



REFERENCE MANUAL | PUBLIC

SAP Adaptive Server Enterprise 16.0 SP03

Document Version: 1.0 – 2019-06-06

Reference Manual: Tables

Content

- 1 System Tables 8**
- 1.1 Locations of System Tables. 8
 - System Tables in master. 8
 - System Tables in sybsecurity. 9
 - System Table in sybssystemdb. 9
 - System Tables in All Databases. 9
 - The sybdiagdb Database. 10
 - The syblicenseslog Table. 10
- 1.2 Using System Tables in the Cluster Edition. 11
 - timestamp Columns. 11
 - Changed Identity Values. 11
 - Controlling Fake-Table Materialization. 11
- 1.3 Rules for Using System Tables. 12
 - Permissions on System Tables. 12
 - Locking Schemes Used for System Tables. 13
 - Reserved Columns. 14
 - Updating System Tables. 14
 - Triggers on System Tables. 14
- 1.4 ch_events. 14
- 1.5 sysdams. 16
- 1.6 syblicenseslog. 16
- 1.7 sysalternates. 17
- 1.8 sysaltusages. 17
- 1.9 sysattributes. 19
- 1.10 sysauditoptions. 21
- 1.11 sysaudits_01 – sysaudits_08. 22
- 1.12 syscacheinfo. 24
- 1.13 syscachepoolinfo. 25
- 1.14 syscharsets. 27
- 1.15 syscolumns. 28
- 1.16 syscomments. 32
- 1.17 sysconfigures. 34
- 1.18 sysconstraints. 36
- 1.19 syscoordinations. 37
- 1.20 syscurconfigs. 38
- 1.21 sysdatabases. 41

1.22	sysdepends.	47
1.23	sysdevices.	47
1.24	sysencryptkeys.	50
1.25	sysengines.	51
1.26	sysgams.	52
1.27	sysimrlogs.	52
1.28	sysindexes.	53
1.29	sysinstances.	56
1.30	sysjars.	57
1.31	syskeys.	58
1.32	syslanguages.	59
1.33	syslisteners.	60
1.34	syslocks.	60
1.35	syslogininfo.	63
1.36	sysloginroles.	63
1.37	syslogins.	65
1.38	syslogs.	68
1.39	syslogshold.	68
1.40	sysmessages.	70
1.41	sysmonitors.	71
1.42	sysobjects.	71
1.43	sysoptions.	78
1.44	syspartitionkeys.	80
1.45	syspartitions.	80
1.46	syspoolinfo.	82
1.47	sysprocedures.	83
1.48	sysprocesses.	84
1.49	sysprotects.	89
1.50	sysquerymetrics.	92
1.51	sysqueryplans.	94
1.52	sysreferences.	95
1.53	sysremotelogins.	97
1.54	sysresourcelimits.	98
1.55	sysroles.	99
1.56	syssecmechs.	100
1.57	syssegments.	100
1.58	sysservers.	101
1.59	syssequences.	103
1.60	syssessions.	103
1.61	syssslices.	104
1.62	sysssrvroles.	105

1.63	sysstatistics.	106
1.64	systabstats.	108
1.65	systhresholds.	111
1.66	systimeranges.	113
1.67	systransactions.	113
1.68	systypes.	116
1.69	sysusages.	119
1.70	sysusermessages.	121
1.71	sysusers.	121
1.72	sysversions.	123
1.73	sysxtypes.	123
2	dbccdb Tables.	125
2.1	dbccdb Workspaces.	125
2.2	dbccdb Log.	126
2.3	dbcc_config.	126
2.4	dbcc_counters.	127
2.5	dbcc_exclusions.	128
2.6	dbcc_fault_params.	129
2.7	dbcc_faults.	130
2.8	dbcc_operation_log.	131
2.9	dbcc_operation_results.	132
2.10	dbcc_types.	133
3	Monitoring Tables.	142
3.1	monBucketPool.	142
3.2	monCachedObject.	144
3.3	monCachePool.	145
3.4	monCachedProcedures.	147
3.5	monCachedStatement.	149
3.6	monCIPC.	154
3.7	monCIPCEndpoints.	155
3.8	monCIPCLinks.	156
3.9	monCIPCMesh.	157
3.10	monCLMObjectActivity.	158
3.11	monClusterCacheManager.	160
3.12	monCMSFailover.	161
3.13	monDataCache.	162
3.14	monDBRecovery.	164
3.15	monDBRecoveryLRTypes.	166
3.16	monDeadLock.	167
3.17	monDeviceIO.	171

3.18	monDeviceSegmentIO	172
3.19	monDeviceSpaceUsage	172
3.20	monDeviceSegmentUsage	173
3.21	monEngine	175
3.22	monErrorLog	177
3.23	monFailoverRecovery	178
3.24	monHADRMembers	179
3.25	monHANANonPushdown	180
3.26	monHCBGCTasks	181
3.27	monHCBPartitionActivity	182
3.28	monHCBTuningActivity	184
3.29	monIMRSCache	186
3.30	monIMRSCacheActivity	186
3.31	monIMRSGCTasks	190
3.32	monImrslogRecovery	192
3.33	monIMRSPartitionActivity	195
3.34	monInmemoryStorage	201
3.35	monIOController	202
3.36	monIOQueue	203
3.37	monLatchFreeIndex	204
3.38	monLicense	205
3.39	monLocks	206
3.40	monLockTimeout	208
3.41	monLogicalCluster	213
3.42	monLogicalClusterAction	215
3.43	monLogicalClusterInstance	217
3.44	monLogicalClusterRoute	218
3.45	monMemoryUsage	218
3.46	monNetworkIO	222
3.47	monNVCache	223
3.48	monOpenDatabases	224
3.49	monOpenObjectActivity	225
3.50	monOpenPartitionActivity	230
3.51	monPCIBridge	235
3.52	monPCIEngine	235
3.53	monPCISlots	236
3.54	monPCM	237
3.55	monProcedureCache	239
3.56	monProcedureCacheMemoryUsage	240
3.57	monProcedureCacheModuleUsage	241
3.58	monProcess	242

3.59	monProcessActivity.	244
3.60	monProcessLookup.	246
3.61	monProcessMigration.	247
3.62	monProcessNetIO.	248
3.63	monProcessObject.	249
3.64	monProcessProcedures.	250
3.65	monProcessSQLText.	252
3.66	monProcessStatement.	253
3.67	monProcessWaits.	254
3.68	monProcessWorkerThread.	255
3.69	monRepCoordinator.	256
3.70	monRepLogActivity.	257
3.71	monRepMemoryStatistics.	260
3.72	monRepScanners.	262
3.73	monRepScannersTotalTime.	263
3.74	monRepSchemaCache.	265
3.75	monRepSenders.	266
3.76	monRepStreamStatistics.	268
3.77	monRepSyncTaskStatistics.	271
3.78	monRepTruncationPoint.	275
3.79	monSpinlockActivity.	277
3.80	monSQLRepActivity.	278
3.81	monSQLRepMisses.	279
3.82	monSSLCertInfo.	280
3.83	monState.	280
3.84	monStatementCache.	282
3.85	monSysExecutionTime.	283
3.86	monSysLoad.	284
3.87	monSysPlanText.	285
3.88	monSysSQLText.	286
3.89	monSysStatement.	287
3.90	monSysWaits.	290
3.91	monSysWorkerThread.	290
3.92	monTableColumns.	291
3.93	monTableCompression.	293
3.94	monTableParameters.	294
3.95	monTables.	295
3.96	monTableTransfer.	296
3.97	monTask.	298
3.98	monTempdbActivity.	298
3.99	monThread.	300

3.100	monThreadPool.	301
3.101	monThresholdEvent.	302
3.102	monWaitClassInfo.	304
3.103	monWaitEventInfo.	304
3.104	monWorkload.	305
3.105	monWorkloadPreview.	306
3.106	monWorkloadProfile.	307
3.107	monWorkloadRaw.	308
3.108	monWorkQueue.	309
4	sybpcidb Tables.	310
4.1	pca_jre_arguments.	310
4.2	pca_jre_directives.	311
4.3	pci_arguments.	312
4.4	pci_directives.	313
4.5	pci_slotinfo.	314
4.6	pci_slot_syscalls.	315
5	Workload Profiler Tables.	316
5.1	wlp_tables.	316
5.2	wlp_table_columns.	317
5.3	wlp_exec_defaults.	318
5.4	wlp_exec_control.	319
5.5	wlp_exec_commands.	321
5.6	wlp_control.	322
5.7	wlp_valid_features.	322
5.8	wlp_exec_cmd_timings.	323
5.9	wlp_exec_cmd_show.	324
5.10	wlp_unique_collist_for_mda_archive.	325
5.11	wlp_exec_cmd_generate_deltas.	326
5.12	wlp_afs_inputs.	327
5.13	wlp_plan_objscores.	327
5.14	WLP_spaceusage_object.	329
5.15	WLP_spaceusage_syslogs.	330
5.16	WLP_spaceusage_sysimrslogs.	331
5.17	WLP_monOpenObjectActivity.	333
5.18	WLP_monDataCache.	334
5.19	WLP_monSysWaits.	335

1 System Tables

System tables are supplied for use with SAP Adaptive Server Enterprise. Most SAP ASE system tables are row-locked tables. Those that are not, are noted in the individual system table descriptions.

1.1 Locations of System Tables

Some system tables are located only in the `master`, `sybsecurity`, or `sybsystemdb` databases, while others may be located in all databases.

Most tables in the `master` database are system tables. Some of these tables also occur in user databases. They are automatically created when the `create database` command is issued.

1.1.1 System Tables in master

Certain system tables occur *only* in the `master` database.

- `syscharsets` – one row for each character set or sort order.
- `sysconfigures` – one row for each configuration parameter that can be set by users.
- `syscurconfigs` – information about configuration parameters currently being used by the SAP ASE server.
- `sysdatabases` – one row for each database on the SAP ASE server.
- `sysdevices` – one row for each tape dump device, disk dump device, disk for databases, and disk partition for databases.
- `sysengines` – one row for each SAP ASE engine currently online.
- `syslanguages` – one row for each language (except U.S. English) known to the server.
- `syslisteners` – one row for each type of network connection used by the current SAP ASE server.
- `syslocks` – information about active locks.
- `sysloginroles` – one row for each server login that possesses a system role.
- `syslogins` – one row for each valid SAP ASE user account.
- `syslogshold` – information about the oldest active transaction and the Replication Server® truncation point for each database.
- `sysmessages` – one row for each system error or warning.
- `sysmonitors` – one row for each monitor counter.
- `sysprocesses` – information about server processes .
- `sysremotelogins` – one row for each remote user .
- `sysresourcelimits` – one row for each resource limit.
- `syssecmechs` – information about the security services available for each security mechanism that is available to the SAP ASE server.

- `syservers` – one row for each remote SAP ASE server.
- `sysessions` – used only when the SAP ASE server is configured for failover in a high availability system. `sysessions` contains one row for each client that connects to the SAP ASE server with the failover property.
- `sysrvroles` – one row for each server-wide role.
- `sysrimeranges` – one row for each named time range.
- `systransactions` – one row for each transaction.
- `sysusages` – one row for each disk piece allocated to a database.

1.1.2 System Tables in sybsecurity

Two audit-related system tables occur *only* in the `sybsecurity` database:

- `sysauditoptions` – one row for each global audit option.
- `sysaudits_01` – `sysaudits_08` – the audit trail. Each audit table contains one row for each audit record.

All auditing-related system tables are all-pages locked.

1.1.3 System Table in sybssystemdb

The `syscoordinations` system table, which consists of one row for each remote participant of a distributed transaction, occurs only in `sybssystemdb`.

1.1.4 System Tables in All Databases

Certain system tables occur in all databases.

- `sysalternates` – one row for each SAP ASE user mapped to a database user.
- `sysattributes` – one row for each object attribute definition.
- `syscolumns` – one row for each column in a table or view, and for each parameter in a procedure.
- `syscomments` – one or more rows for each view, rule, default, trigger, and procedure, giving SQL definition statement.
- `sysconstraints` – one row for each referential and check constraint associated with a table or column.
- `sysdepends` – one row for each procedure, view, or table that is referenced by a procedure, view, or trigger.
- `sysencryptkeys` – one row for each key created in a database, including the default key.
- `sysgams` – allocation bitmaps for an entire database.
- `sysindexes` – one row for each clustered or nonclustered index, one row for each table with no indexes, and an additional row for each table containing text or image data.
- `sysjars` – one row for each Java archive (JAR) file that is retained in the database.
- `syskeys` – one row for each primary, foreign, or common key; set by user (not maintained by the SAP ASE server).

- `syslogs` – transaction log.
- `sysobjects` – one row for each table, view, procedure, rule, trigger default, log, and (in `tempdb` only) temporary object.
- `syspartitionkeys` – one row for each partition key.
- `syspartitions` – one row for each partition of a partitioned table or index.
- `sysprocedures` – one row for each view, rule, default, trigger, and procedure, giving internal definition.
- `sysprotects` – user permissions information.
- `sysquerymetrics` – gathers aggregated historical query information in a persistent catalog. `sysquerymetrics` is a view, not a table.
- `sysqueryplans` – abstract query plans and SQL text.
- `sysreferences` – one row for each referential integrity constraint declared on a table or column.
- `sysroles` – maps server-wide roles to local database groups.
- `syssegments` – one row for each segment (named collection of disk pieces).
- `syslices` – obsolete, used only during upgrade. Formerly called `syspartitions` before SAP ASE version 15.0.
- `sysstatistics` – one or more rows for each indexed column on a user table. May also contain rows for unindexed column.
- `systabstats` – one row for each table, plus one row for each nonclustered index.
- `systhresholds` – one row for each threshold defined for the database.
- `systypes` – one row for each system-supplied and user-defined datatype.
- `sysusermessages` – one row for each user-defined message.
- `sysusers` – one row for each user allowed in the database.
- `sysxtypes` – one row for each extended, Java-SQL datatype. Uses row-level locking.

1.1.5 The `sybdiagdb` Database

SAP Product Support may create the `sybdiagdb` database on your system for debugging purposes. This database holds diagnostic configuration data for use by Product Support representatives.

1.1.6 The `syblicenseslog` Table

The `syblicenseslog` table is described in `syblicenseslog`. It is not technically a system table, but you may need to consult it for license information related to shutting down the SAP ASE server.

Related Information

[syblicenseslog \[page 16\]](#)

1.2 Using System Tables in the Cluster Edition

There are some differences in system tables if you are using a Cluster Edition of SAP ASE.

1.2.1 timestamp Columns

In the SAP ASE server, if a table includes a `timestamp` column, its value is updated when a row is changed. Client applications can use this functionality to detect changes to rows using an access method called “optimistic locking.”

The values in the `timestamp` column are unique in a database. However, in the Cluster Edition, `timestamp` column values are not guaranteed to be in increasing order in a database across tables, but they are guaranteed to be in increasing order for a particular table.

1.2.2 Changed Identity Values

Identity columns in the Cluster Edition behave differently from those in non-clustered editions of SAP ASE. Although the Cluster Edition guarantees that identity values are unique, for performance reasons the values may not monotonically increase.

In a non-clustered SAP ASE server, a set of identity values are burned into memory to reduce disk I/Os as `inserts` access the next value from memory. In the Cluster Edition, the same size `set` is burned into memory, but the `set` is split among the cluster instances. In a two-instance cluster with an identity `set` size of 250000, the first instance inserts values {1,2,3, and so on}, and the second instance inserts values {125000,125001,125002, and so on}.

When you restart an instance after a shutdown or a crash, a new block of identity values may be allocated for that instance when it rejoins the cluster. As a result, you may see a jump in the identity value for the instance.

The `next-identity` function reports the next identity value for a table from the instance in which `next-identity` is executed. For example, `next-identity` returns 4 for instance 1 and 125003 for instance 2.

The behavior of the `identity-burn-max` remains the same as for a non-clustered SAP ASE server because the burn size and burn behavior is unchanged in the Cluster Edition.

1.2.3 Controlling Fake-Table Materialization

Certain stored procedures, such as `sp_who` and `sp_lock`, read from fake tables such as `sysprocesses` and `syslocks`. Because their rows are not stored on disk, fake tables present an exception to the shared-data nature of a shared-disk cluster, and special features apply.

You can control whether a fake-table query returns rows from the local instance or all instances in the cluster by using the `set system_view` command. `set system_view` is a session-level command. In addition, `set system_view` also controls monitoring table materialization.

By default, the SAP ASE server retrieves rows only from the local instance.

- To specify that fake-table queries materialize rows for all instances, use the `cluster` option. For example:

```
set system_view cluster
```

- To specify that fake-table queries materialize rows for the local instance, use the `instance` option. For example:

```
set system_view instance
```

To retrieve the current `system_view` setting, select the `<@@system_view>` global variable.

The SAP ASE server supports cluster-wide materialization for these fake tables:

- `sysprocesses`
- `syslocks`
- `sysengines`
- `syslisteners`
- `sysmonitors`
- `syssecmechs`
- `syscurconfigs`

i Note

`sysinstances` is always set for cluster-wide materialization, regardless of the `system_view` setting.

1.3 Rules for Using System Tables

This section describes rules, restrictions, and usage information for system tables.

i Note

By default, a column is defined as NOT NULL. Nullable columns are described using the “null” keyword, and are listed in the column descriptions for the tables in this book.

1.3.1 Permissions on System Tables

Permissions for use of the system tables can be controlled by the Database Owner, just like permissions on any other tables.

By default, when SAP ASE is installed, the `installmodel` script grants `select` access to “public” (all users) for most system tables and for most fields in the tables. Instead, the default permissions on the system tables are assigned when the SAP ASE server builds a new database. However, no access is granted to some system tables, such as `sysserveroles`, and no access is granted to certain fields in other system tables. For example, all users, by default, can select all columns of `sysobjects` except `audflags`.

See the *Security Administration Guide* for more information.

```
sp_helprotect <system_table_name>
```

For example, to check the permissions of `sysrvroles` in `master`, execute:

```
use master
go
sp_helprotect sysrvroles
go
```

1.3.2 Locking Schemes Used for System Tables

In the allpages locking scheme in SAP ASE, locks are acquired on data and index pages.

See the *Performance and Tuning Guide: Locking* for more information on locking schemes.

All system tables use datarow locking except for the following, which use allpages locking:

- `sysusermessages`
- `syslices`
- `sysmessages`

In addition, the following system tables are “fake”—or non-row-oriented—catalogs that give the appearance of using allpages locking:

- `syslogs`
- `sysgams`
- `sysprocesses`
- `syslocks`
- `syscurconfigs`
- `syssecmechs`
- `sysmonitors`
- `sysengines`
- `systestlog`
- `syslisteners`
- `syslogshold`

1.3.3 Reserved Columns

The word “reserved” in the column description means that the column is not currently used by the SAP ASE server.

1.3.4 Updating System Tables

Direct updates on system tables are not allowed – even for the Database Owner. Instead, SAP ASE includes system procedures that you should use to make any normally needed updates and additions to system tables.

You can allow direct updates to the system tables if it you must modify them in a way that cannot be accomplished with a system procedure. To allow direct updates, a system security officer must use `sp_configure` to reset the configuration parameter called `allow updates to system tables`. For more information, see the *Security Administration Guide*.

1.3.5 Triggers on System Tables

You cannot create triggers on system tables. If you try to create a trigger on a system table, the SAP ASE server returns an error message and cancels the trigger.

1.4 ch_events

Contains one row for each configuration change event. `ch_events` is located in the `sybmgmtdb` database. `ch_events` is a view based on the `extrainfo` columns. You must have the `mon_role` to view `ch_events`.

Columns

The columns for `ch_events` are:

Name	Datatype	Description
<code>area</code>	<code>varchar(10) not null</code>	Area in which the event occurs. One of: <ul style="list-style-type: none"><code>server</code> – server-level events.<code>database</code> – database-level events.<code>cache</code> – cache-level events.<code>traceflag</code> – <code>dbcc traceflag</code> and <code>set switch</code> events.<code>SUSD</code> – for startup/shutdown.<code>audit</code> – auditing state changes.

Name	Datatype	Description
type	varchar(30) not null	Type of auditable event. One of: <ul style="list-style-type: none"> • sp_configure • sp_serveroption • sp_dboption • sp_cacheconfig • sp_poolconfig • create thread pool • alter thread pool • drop thread pool • dbcc traceflag • set switch • configuration file change • startup • shutdown • shutdown with wait • shutdown with nowait • abrupt shutdown • global auditing • config history auditing
target	varchar(30) null	Name of the objects to which the change applies.
element	varchar(255) null	Configuration parameter or other option name.
oldvalue	varchar(255) null	Value of event prior to change.
newvalue	varchar(255) null	Value of event after change.
mode	varchar(10) null	Status for configuration parameters: static or dynamic.
timestamp	datetime not null	Date and time the event takes place. For changes to the configuration file and abrupt shutdowns, <code>timestamp</code> indicates the time the event was detected, not when the event took place.
username	varchar(30) null	Name of the user who made the change. Set to null for: <ul style="list-style-type: none"> • Startup • Configuration file change • Abrupt shutdown
instanceid	tinyint null	(Cluster Edition only) ID of the instance.

1.5 sysdams

`sysdams` stores the dump allocation map (DAM) for the database. The DAM stores the list of allocation units that have been modified since the last full database dump. It is a bitmap with one bit per allocation unit in the database.

A value of:

- 0 – indicates that no page in the allocation unit has changed since the last full database dump.
- 1 – indicates that at least one page in the allocation unit has changed since the last database dump.

`sysdams` is automatically increased in size by an `alter database` operation. You cannot select from or view `sysdams`.

1.6 syblicenseslog

Applies to `master` database only. `syblicenseslog` contains one row for each update of the maximum number of licenses used in the SAP ASE server per 24-hour period. `syblicenseslog` is updated every 24 hours. If the SAP ASE server is shut down at any time, License Use Manager logs the number of licenses currently being used in `syblicenseslog` before the shutdown is complete. The 24-hour period restarts when you start the SAP ASE server.

i Note

`syblicenseslog` is not a system table. Its type is “U” and its object ID is greater than 255.

Columns

The columns for `syblicenseslogs` are:

Name	Datatype	Description
<code>status</code>	<code>smallint</code>	Status of the maximum number of licenses used; one of the following: <ul style="list-style-type: none">• 0 = number of licenses not exceeded• 1 = number of licenses is exceeded• -1 = housekeeper is unable to monitor number of licenses
<code>logdate</code>	<code>datetime</code>	Date and time the log was written
<code>maxlicenses</code>	<code>int</code>	Maximum number of licenses used during the 24-hour period

1.7 sysalternates

Applies to all databases. `sysalternates` contains one row for each SAP ASE user that is mapped or aliased to a user of the current database. When a user tries to access a database, the SAP ASE server looks for a valid `uid` entry in `sysusers`. If none is found, it looks in `sysalternates.suid`. If the user's `suid` is found there, he or she is treated as the database user whose `suid` is listed in `sysalternates.altsuid`.

Columns

The columns for `sysalternates` are:

Name	Datatype	Description
<code>suid</code>	<code>int</code>	Server user ID of user being mapped
<code>altsuid</code>	<code>int</code>	Server user ID of user to whom another user is mapped

Indexes

Unique clustered index on `suid`.

1.8 sysaltusages

Applies to the scratch database. The `sysaltusages` system table maps page numbers in an archive database to the actual page within either the database dump and its stripes, or the modified pages section.

Unlike the `sysusages` table in a traditional database, however, the `sysaltusages` table does not map every logical page in the database. `sysaltusages` maps pages that have been:

- Stored in a database dump
- Modified, and therefore, relocated to the modified pages section

See *Archive Database Access* in the *System Administration Guide, Volume 2*.

Columns

The columns for `sysaltusages` are:

Name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	The database ID of the archive database
<code>location</code>	<code>int</code>	The location of the archive database segment where the physically contiguous block of pages resides. In the <code>location</code> column, a value of 5 and 6 means the location is in the database dump, transaction log dump, or their stripes, and a value of 7 or 8 means that the location is in the modified pages section. A value of 4 is used to fill the gaps for pages that are not physically available.
<code>lstart</code>	<code>unsigned int</code>	The logical page number of the start of the block of physically contiguous pages.
<code>size</code>	<code>unsigned int</code>	The number of logical pages in the block of physically contiguous pages.
<code>vstart</code>	<code>int</code>	The offset of the start of the contiguous block of pages on the device given by <code>vdevno</code> .
<code>vdevno</code>	<code>int</code>	The device number on which the contiguous block of pages resides.
<code>segmap</code>	<code>int</code>	A map of the segments to which this block of pages is allocated.

Note

Because `sysaltusages` is a row-locked catalog, you may need to periodically use `reorg` to reclaim logically deleted space.

The scratch database stores the new `sysaltusages` table. The scratch database is used to provide flexibility as to where the `sysaltusages` table is located.

The scratch database can be any database (with some exceptions like `master` and `temporary` databases). You should dedicate a database that is used only as a scratch database, because:

- The size of `sysaltusages` may vary depending on the number of archive databases it supports. You cannot decrease the size of a database, but if it is too large, you can drop it and re-create a smaller database when required.
- It allows you to turn on the `"trunc log on checkpoint"` option so that the database log be automatically truncated.

Apart from hosting the `sysaltusages` table, this database is like any other. You can use threshold procedures and other space management mechanisms to manage space within the database.

You must specify a database that is to be used as a scratch database, by entering:

```
sp_dboption <db name>, "scratch database", "true"
```

Each archive database can be assigned to only one scratch database at a time, but multiple archive databases can use the same scratch database. If you have a large number of archive databases, you may want to define multiple scratch databases.

`sysaltusages` includes a unique clustered index named `csysaltusages` on `dbid`, `location`, and `lstart`.

1.9 sysattributes

Applies to all databases. System attributes define properties of objects such as databases, tables, indexes, users, logins, and procedures. `sysattributes` contains one row for each of an object's attribute definitions (configured by various system procedures). `master..sysattributes` defines the complete set of valid attribute values and classes for the SAP ASE server as a whole. It also stores attribute definitions for server-wide objects, such as databases and logins.

Use only system procedures to access `sysattributes`. The permissions required for modifying `sysattributes` depend on the system procedure you use.

Columns

The columns for `sysattributes` are:

Name	Datatype	Description
<code>class</code>	<code>smallint</code>	The attribute class ID. This describes the category of the attribute. In <code>master..sysattributes</code> , the special class 0 identifies all valid <i>classes</i> of attributes for the SAP ASE server.
<code>attribute</code>	<code>smallint</code>	The attribute ID, this column specifies a default decrypt on an encrypted column with a value of 1 (<code>DECRYPT-DEFAULT_ID</code>) for objects with a type of EC and a class of 25.
<code>object_type</code>	<code>char(2)</code>	A one- or two-letter character ID that defines the type of object to associate with the attribute.
<code>object_cinfo</code>	<code>varchar(255) null</code>	
<code>object_cinfo</code> <code>2</code>	<code>varchar(255) null</code>	A string identifier for the object (for example, the name of an application) in a SDC environment. This field is not used by all attributes.
<code>object</code>	<code>int null</code>	The object identifier. This may be an object ID, user ID, decrypt default ID, or database ID, depending on the type of object. If the object is a part of a table (for example, an index), this column contains the object ID of the associated table.

Name	Datatype	Description
object_info1 , object_info2 , object_info3	int null	Defines additional information required to identify the object. This field is not used by all attributes. The contents of this field depend on the attribute that is defined. <ul style="list-style-type: none"> object_info_1 – includes the table ID for a table whose encrypted column defines the decrypt default. object_info2 – specifies the <colid> of the encrypted column that includes the decrypt default.
int_value	int null	An integer value for the attribute (for example, the display level of a user).
char_value	varchar(768) null	A character value for the attribute (for example, a cache name).
text_value	text null	A text value for the attribute.
image_value	image null	An image value for the attribute.
comments	varchar(255) null	A string identifier for the object (forComments or additional information about the attribute definition).
object_datetime	null	datetime value for the attribute. Its use depends on the module using the attribute, but it typically refers to the date and time the attribute was created.

The relevant values most frequently used in `object_type` are:

- D – Database
- I – Index
- L – Login
- LR – Login Profile
- P – Proc
- T – Table
- U – User
- AP – Application
- DC – Dump Condition
- EL – External Login (OMNI)
- OD – Object Definition (OMNI)
- TC – Transaction Coordination (ASTC)
- TG – Temporary Database Group (multi tempdb)
- TP – Text Page (OMNI)
- QP – Query Plans (abstract plans)
- UR – User Role
- GR – Group Role
- LG – Login (for MTDB binding)
- EG – Engine Group

- `PS` – Password Security
- `SP` – Keypair Regeneration Period

These values provide additional information for `sysattributes`, and are not for use as standalone values. For this reason, use these values only in conjunction with the class ID.

Indexes

- Unique clustered index on `class`, `attribute`, `object_type`, `object`, `object_info1`, `object_info2`, `object_info3`, `object_cinfo`.
- Nonclustered index on `object_type`, `object`, `object_info1`, `object_info2`, `object_info3`, `object_cinfo`.

1.10 sysauditoptions

Applies to `sybsecurity` database. `sysauditoptions` contains one row for each server-wide audit option and indicates the current setting for that option.

Other types of auditing option settings are stored in other tables. For example, database-specific option settings are stored in `sysdatabases`, and object-specific option settings are stored in `sysobjects`. The default value for each option is 0, or “off.” Only system security officers can access `sysauditoptions`.

Columns

The columns for `sysauditoptions` are:

Name	Datatype	Description
<code>num</code>	<code>smallint</code>	Number of the server-wide option.
<code>val</code>	<code>smallint</code>	Current value; one of the following: <ul style="list-style-type: none"> • 0 = off • 1 = pass • 2 = fail • 3 = on
<code>minval</code>	<code>smallint</code>	Minimum valid value for this option.
<code>maxval</code>	<code>smallint</code>	Maximum valid value for this option.
<code>name</code>	<code>varchar(30)</code>	Name of option.
<code>sval</code>	<code>varchar(30)</code>	String equivalent of the current value: for example, “on”, “off”, “nonfatal”.

Name	Datatype	Description
comment	varchar(255)	Description of option.
id	int	Server user ID (suid) of the login in syslogins.
status	int	Internal status information.

1.11 sysaudits_01 – sysaudits_08

Applies to sybsecurity database. These system tables contain the audit trail. Only one table at a time is active. The active table is determined by the value of the `current_audit_table` configuration parameter. An installation can have as many as eight audit tables. For example, if your installation has three audit tables, the tables are named `sysaudits_01`, `sysaudits_02`, and `sysaudits_03`. An audit table contains one row for each audit record.

Columns

The columns for `sysaudits_01` – `sysaudits_08` are:

Name	Datatype	Description
event	smallint	Type of event being audited.
eventmod	smallint	Further information about the event. Possible values are: <ul style="list-style-type: none"> • 0 = no modifier for this event. • 1 = the event passed permission checking. • 2 = the event failed permission checking.
spid	smallint int for the Cluster Edition	Server process ID of the process that caused the audit record to be written.
eventtime	datetime	Date and time of the audited event.
sequence	smallint	Sequence number of the record within a single event; some events require more than one audit record.
suid	smallint	Server login ID of the user who performed the audited event.
dbid	int null	Database ID in which the audited event occurred or the object/stored procedure/trigger resides, depending on the type of event.

Name	Datatype	Description
objid	int null	ID of the accessed object or stored procedure/trigger.
xactid	binary(6) null	ID of the transaction containing the audited event. For a multidatabase transaction, this is the transaction ID from the database where the transaction originated.
loginname	varchar(30) null	Login name corresponding to the <code>suid</code> .
dbname	varchar(30) null	Database name corresponding to the <code>dbid</code> .
objname	varchar(255) null	Object name corresponding to the <code>objid</code> .
objowner	varchar(30) null	Name of the owner of <code>objid</code> .
extrainfo	varchar(255) null	Additional information about the audited event. The <code>extrainfo</code> column contains a sequence of items separated by semicolons: <ul style="list-style-type: none"> • Roles – lists the roles that are active. The roles are separated by blanks. • For commands supported by full text auditing, full text of an audited command. For other commands, the name of the keyword or command option that was used for the event. Multiple keywords or options are separated by commas. • Previous value – the value prior to the update if the event resulted in the update of a value. • Current value – the new value if the event resulted in the update of a value. • Other information – additional security-relevant information that is recorded for the event. For example, parameter names and values can be recorded here. • Proxy information – the original login name, if the event occurred while a <code>set proxy</code> was in effect. • Principal information – the principal name from the underlying security mechanism, if the user's login is the secure default login, and the user logged in to the SAP ASE server using unified login. The value of this field is NULL, if the secure default login is not being used.
nodeid	tinyint null	Reserved for future use (not available for cluster environments)
instanceid	tinyint	ID of the instance (available only for cluster environments)

An example of an `extrainfo` column for the security-relevant event of changing an auditing configuration parameter might be:

```
sso_role;suspend auditing when full; 1; 0; ; ; ;
```

This example indicates that a system security officer changed the configuration parameter `suspend auditing when full` from 1 (suspend all processes that involve an auditing event) to 0 (truncate the next audit table and make it the current audit table).

1.12 syscacheinfo

Applies to master Database. Provides information about data caches.

`syscacheinfo` is a view of the `master` database that provides information about the configuration of data caches and pools.

Access to the views is restricted to users with the `sa_role` role.

Columns

The columns for `syscacheinfo` are:

Name	Datatype	Description
<code>cache_name</code>	<code>varchar (30)</code>	Name of the cache in which this pool is allocated.
<code>cache_status</code>	<code>varchar (8)</code>	Status of the cache. One of: <ul style="list-style-type: none">• Active• Pend/Act• Act/Del
<code>cache_type</code>	<code>varchar (16)</code>	Type of cache. One of: <ul style="list-style-type: none">• Mixed, HK Ignore• Mixed• Log Only• In-Memory Storage• Default• Row Storage
<code>config_size</code>	<code>float</code>	The currently configured size of the cache, in megabytes. May be different from the actual size of the cache, reported in the <code>run_size</code> column.
<code>run_size</code>	<code>float</code>	The current amount of memory, in megabytes, allocated to the cache. May be different from the configured size reported by the <code>config_size</code> column.

Name	Datatype	Description
config_replacement	varchar(11)	Currently configured buffer replacement strategy. None, or one of: <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
run_replacement	varchar(11)	Current buffer replacement strategy for the cache. None, or one of: <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
config_partitions	int	Configured number of partitions in the data cache.
run_partitions	int	The current number of partitions in the data cache.
overhead	numeric	Amount of memory overhead for the data cache.
cacheid	int	ID of the data cache. A value of -1 for caches of type Row Storage.
instanceid	int	ID of the instance (zero for non-Cluster Edition servers).
scope	varchar(6)	Indicates whether the data cache is local or global for Cluster Edition. The value is always Global for nonclustered servers.

1.13 syscachepoolinfo

Provides a row for each data cache pool that includes configuration information for the data cache. This view is a join between the `syscacheinfo` and `syspoolinfo` views.

Columns

Access to the views is restricted to users with the `sa_role` role.

The columns for `syscacheinfo` are:

Name	Datatype	Description
cache_name	varchar(30)	Name of the cache in which this pool is allocated.

Name	Datatype	Description
cache_status	varchar(8)	Status of the cache. One of: <ul style="list-style-type: none"> • Active • Pend/Act • Act/Del
cache_type	varchar(16)	Type of cache. One of: <ul style="list-style-type: none"> • Mixed, HK Ignore • Mixed • Log Only • In-Memory Storage • Default
cache_config_size	float	The currently configured size of the cache, in megabytes. May be different from the actual size of the cache, reported in the run_size column.
cache_run_size	float	The current amount of memory allocated to the cache, in megabytes. May be different from the configured size reported by the config_size column.
cache_config_replacement	varchar(11)	Currently configured buffer replacement strategy. None, or one of: <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
cache_run_replacement	varchar(11)	Current buffer replacement strategy for the cache. None, or one of: <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
cache_config_partitions	int	Configured number of partitions in the data cache.
cache_run_partitions	int	The current number of partitions in the data cache.
cache_overhead	numeric	Amount of memory overhead for the data cache.
pool_io_size	varchar(3)	The size of the buffers, in kilobytes, used to perform I/O for this pool.
pool_config_size	float	Configured amount of memory, in megabytes, allocated to the pool. May be different from the amount reported in the run_size column.
pool_run_size	float	The current amount of memory, in megabytes, allocated to the pool.
pool_apf_percent	int	The percentage of buffers in the pool that can be used to hold buffers that have been read into cache by asynchronous prefetch.

Name	Datatype	Description
pool_wash_size	varchar (10)	The size of the wash area, in megabytes, in the pool.
cacheid	int	ID of the data cache.
instanceid	int	ID of the instance (zero for non-Cluster Edition servers).
scope	varchar (6)	Indicates whether the data cache is local or global for Cluster Edition. The value is always Global for nonclustered servers.

1.14 syscharsets

Applies to master database only. `syscharsets` contains one row for each character set and sort order defined for use by the SAP ASE server. One of the sort orders is marked in `master..sysconfigures` as the default sort order, which is the only one actually in use.

Columns

The columns for `syscharsets` are:

Name	Datatype	Description
type	smallint	The type of entity this row represents. Numbers from 1001 to 1999 represent character sets. Numbers from 2000 to 2999 represent sort orders.
id	tinyint	The ID for a character set or sort order. A sort order is defined by the combination of the sort order ID and the character set ID (<code>csid</code>). The character set is defined by <code>id</code> , which must be unique. SAP reserves ID numbers 0 – 200.
csid	tinyint	If the row represents a character set, this field is unused. If the row represents a sort order, this is the ID of the character set that sort order is built on. A character set row with this ID must exist in this table.
status	smallint	Internal system status information bits.
name	varchar (30)	A unique name for the character set or sort order. Can use only the 7-bit ASCII letters A – Z or a – z, digits 0 – 9, and underscores (<code>_</code>), and must begin with a letter.
description	varchar (255)	An optional description of the features of the character set or sort order.

Name	Datatype	Description
definition	image	The internal definition of the character set or sort order. The structure of the data in this field depends on the <code>type</code> .
sortfile	varchar(30) null	The name of the sort order file.

Indexes

- Unique clustered index on `id, csid`
- Unique nonclustered index on `name`

1.15 syscolumns

Applies to all databases. `syscolumns` contains one row for every column in every table and view, and a row for each parameter in a procedure.

Contains one row for each computed column and function-based index key associated with a table.

Columns

The columns for `syscolumns` are:

Name	Datatype	Description
id	int	ID of table to which this column belongs, or of procedure with which this parameter is associated.
number	smallint	Sub-procedure number when the procedure is grouped (0 for nonprocedure entries).
colid	smallint	Column ID.

Name	Datatype	Description
status	tinyint	<ul style="list-style-type: none"> Bits 0–2 (values 1, 2, and 4) – indicate bit positioning if the column uses the <code>bit</code> datatype. If the column uses the <code>text</code>/<code>image</code> datatype, bits 0 and 1 indicate replication status as follows: <ul style="list-style-type: none"> 01 = always replicate 10 = replicate only if changed 00 = never replicate Bit 3 (value 8) – indicates whether NULL values are legal in this column. Bit 4 (value 16) – indicates whether more than one check constraint exists for the column. Bits 5 and 6 – are used internally. Bit 7 (value 128) – indicates an identity column.
type	tinyint	Physical storage type; copied from <code>systypes</code> .
length	int	Physical length of data; copied from <code>systypes</code> or supplied by user.
offset	smallint	Offset into the row where this column appears; if negative, this is a variable-length column.
usertype	smallint	<p>User type ID; copied from <code>systypes</code>.</p> <p>If the value of <code>type</code> indicates a column is of nullable datatype, the value of <code>usertype</code> is copied from the <code>usertype</code> of the corresponding base type that can be specified in DDL statements. For example:</p> <pre> create table t1 (dt datetime, dt_null datetime null) go select left(name,6), type, usertype from syscolumns where id = object_id("t1") go type usertype ----- dt 61 12 dt_nul 111 12 </pre>
cdefault	int	ID of the procedure that generates default value for this column.
domain	int	Constraint ID of the first rule or check constraint for this column.
name	varchar(255) not null	Column name
printfmt	varchar(255) null	Reserved
prec	tinyint null	Number of significant digits, if the column uses the <code>numeric</code> datatype.
scale	tinyint null	Number of digits to the right of the decimal point, if the column uses the <code>numeric</code> datatype.

Name	Datatype	Description
remote_type	int null	Maps local names to remote names. Required by the access methods of Component Integration Services to allow the software to pass native column datatype information in parameters to servers of class <code>access_server</code> .
remote_name	varchar(255) null	Maps local names to remote names. Required by the access methods of Component Integration Services to construct a query using the proper column names for a remote table.
xstatus	int null	The status of a column with extended datatypes. The values are: <ul style="list-style-type: none"> • 0 = in row • 1 = off row xstatus is NULL for columns that do not have an extended datatype.
xtype	int null	ID of the class. Used if a column in a table or a parameter in a procedure has a Java class as its datatype. When used, fields are not NULL, and the value of <code>type</code> is 0x39. See <i>Java in Adaptive Server Enterprise</i> for more information.
xdbid	int null	The database ID of the class. For system classes, the value is -1. Otherwise, the value is the current database ID. Used if a column in a table or a parameter in a procedure has a Java class as its datatype. Fields are not NULL, and the value of <code>type</code> is 0x39. See <i>Java in Adaptive Server Enterprise</i> for more information.
accessrule	int null	The object ID of the access rule in <code>sysprocedures</code> . See "Row-level access control" in Chapter 11, "Managing User Permissions" of the <i>Security Administration Guide</i> for more information.

Name	Datatype	Description
status2	int null	<p>Indicates the parameter mode of a SQLJ stored procedure, and the return type of a SQLJ function:</p> <ul style="list-style-type: none"> • 0x00000001, value 1 – parameter mode “in” • 0x00000002, value 2 – parameter mode “out” <p>These internal bits support computed columns:</p> <ul style="list-style-type: none"> • 0x00000010, value 16 – the column is a computed column. • 0x00000020, value 32 – the column is a materialized computed column. • 0x00000040, value 64 – the column is a computed column in a view. • 0x00001000, value 4096 – the encrypted column has a decrypt default. <p>The status2 field from <code>syscolumns</code> uses this encoding to indicate a column's encryption properties:</p> <ul style="list-style-type: none"> • 0x80, value 128 – the column is encrypted. • 0x100, value 256 – the column is encrypted with initialization vector. • 0x200, value 512 – the column is encrypted with random padding. • 0x400, value 1024 – the proxy table is encrypted. • 0x1000, value 4096 – the encrypted column has a decrypt default. • 0x20000, value 131072 – the column is explicitly defined as not compressed. • 0x00040000, value 262144 – the user-specified, or derived in-row length for LOB columns created as in-row.
status3	smallint	0x0001, value 1 – Indicates a hidden computed column for a function-based index key.
computedcolumn1	int	Stores the object ID of the computed column definition.
enctype	int null	Type of data in encrypted form.
lobcomp_level	tinyint	Compression level of the columns defined for large objects.
enclen	int null	Length of encrypted data.
enckeyid	int null	Object ID of key.
enckeydb	varchar(30) null	Name of the database where the encryption key was created; NULL if it is in the same database as the encrypted column.
enckdate	datetime null	Creation date of encryption key; copied from <code>sysobjects.crdate</code> .
inrowlen	smallint	Stores the user-specified, or derived in-row length for LOB columns created as in-row.

Indexes

Unique clustered index on `id`, `number`, `colid`

1.16 syscomments

Applies to all databases. `syscomments` contains entries for each view, rule, default, trigger, table constraint, partition, procedure, computed column, function-based index key, and other forms of compiled objects. The `text` column contains the original definition statements. If the `text` column is longer than 255 bytes, the entries span rows. Each object can occupy as many as 65,025 rows.

It also stores the text of a computed column, function-based index, or partition definition—for example, “`values <= <value_list>`” for a range partition.

The `create service` command stores text in `syscomments`, as it uses the `create procedure` infrastructure.

Columns

The columns for `syscomments` are:

Name	Datatype	Description
<code>id</code>	<code>int</code>	Object ID to which this text applies.
<code>number</code>	<code>smallint</code>	Sub-procedure number when the procedure is grouped (0 for nonprocedure entries).
<code>colid</code>	<code>smallint</code>	The low portion of a column counter for this procedure's comments. Can vary from 0 to 32767. If a procedure has more text than fits in that many rows, this counter works together with <code>colid2</code> .
<code>texttype</code>	<code>smallint</code>	Indicates the comment type. Values are: <ul style="list-style-type: none">• 0 – system-supplied comment, for views, rules, defaults, triggers, and procedures• 1 – user-supplied comment for adding entries that describe an object or column
<code>language</code>	<code>smallint</code>	Reserved.
<code>text</code>	<code>varchar(255) null</code>	Actual text of SQL definition statement.
<code>colid2</code>	<code>smallint</code>	The high portion of a column counter for this procedure's comments. Can vary from 0 to 32767. Is only greater than 0 for procedures containing more than 32,768 rows of procedure text.

Name	Datatype	Description
status	smallint null	Bits indicating the status of the objects: <ul style="list-style-type: none"> • 0x1 – SYSCOM_TEXT_HIDDEN indicates that the text is hidden • 0x2 – Reserved for internal use • 0x4 – SYSCOM_QUOTED_ID_ON indicates that quoted identifiers were on when the object was created • 0x8 – SYSCOM_SHARED_INLINE_DEF indicates the text is for a sharable inline default
version	smallint null	The version of encryption that encodes the algorithm used to encrypt the hidden text for this row. One of: <ul style="list-style-type: none"> • Null – no encryption for hidden text • 1 – (the default) the SAP ASE server obfuscation algorithm used in versions of SAP ASE 15.0 and earlier • 2 – (optional) Advanced Encryption Standard (“AES”) strong encryption
partitionid	int null	Partition ID. Otherwise, null.
encrkeyid	int null	The encryption key ID from the key object in <code>sysencryptkeys</code> in the current database that the SAP ASE server used to encrypt the hidden text of this object when <code>version</code> has a value of 2. Otherwise, the SAP ASE server uses a value of null for <code>encrkeyid</code> .

Note

Do not delete the definition statements from the text column of `syscomments`. These statements are required for the SAP ASE upgrade process. To encrypt a definition statement, execute the system procedure `sp_hidetext`. To see if a statement created in version 11.5 or later was deleted, execute `sp_checksourc`. If the statement was deleted, you must either re-create the object that created the statement or reinstall the application that created the object, which re-creates the statement.

You can protect the text of a database object against unauthorized access by restricting `select` permission on the `text` column of the `syscomments` table to the owner of the object and the system administrator. This restriction, which applies to direct access through `select` statements as well as access through stored procedures, is required to run SAP ASE in the evaluated configuration. To enact this restriction, a system security officer must reset the parameter called `select on syscomments.text` using the system procedure `sp_configure`. For information about the evaluated configuration, see the *Security Administration Guide: Volume 1*.

Indexes

Unique clustered index on `id, number, colid2, colid, texttype`

1.17 sysconfigures

Applies to master database only. `sysconfigures` contains one row for each configuration parameter that can be set by the user.

Columns

The columns for `sysconfigures` are:

Name	Datatype	Description
<code>config</code>	<code>smallint</code>	Configuration parameter number.
<code>value</code>	<code>int</code>	The user-modifiable value for the parameter with <code>integer</code> datatype.
<code>comment</code>	<code>varchar (255)</code>	Name of the configuration parameter.
<code>status</code>	<code>int</code>	Value that represents the type of configuration parameter.
<code>name</code>	<code>varchar (255) null</code>	Name of the configuration parameter (the same value as <code>comment</code>).
<code>parent</code>	<code>smallint null</code>	Configuration parameter number of the parent; if more than one parent, the additional parent numbers are stored in <code>sysattributes</code> .
<code>value2</code>	<code>varchar (255) null</code>	The user-modified value for the parameter with the character datatype. Its value is NULL for parameters with <code>integer</code> datatype. <code>value2</code> is also used to store the pool size of a buffer pool and the size of the IMRS cache for entries corresponding to IMRS caches.
<code>value3</code>	<code>int null</code>	Stores the wash size of a buffer pool.
<code>value4</code>	<code>int null</code>	Stores the number of the cache partition for an IMRS cache. The value for an IMRS cache is always 1 since IMRS caches are not partitioned.
<code>instanceid</code>	<code>tinyint</code>	ID of the instance. Available only for cluster environments.

The `value2` column of the `sysconfigures` table stores the size in terms of string such as 5G, or 20M.

In versions earlier than 16.0 SP02 PL06 the size of cache and pool was stored as an absolute value in terms of Kilobytes. In versions 16.0 SP02 PL06 and later, the `value` column of `sysconfigures` stores the absolute value in terms of 2K.

For example, if the size of cache is 10MB, then the following will be stored from 16.0 SP02 PL06 and later:

```
sysconfigures.value = (10 * 1024KB)/2 = 5120 [absolute value in terms of 2K]
sysconfigures.value2 = "10MB" -> String
```

The following table provides information about the `status` column:

Status type	Decimal	Hex	Description
CFG_NO_OPTIONS	0	0x0	Parameter has no options.
CFG_SYSTEM_OPTION	1	0x01	Parameter is a system option.
CFG_SYSTEM_GROUP	2	0x02	Parameter is a system group.
CFG_STATIC	4	0x04	Parameter is static.
CFG_DYNAMIC	8	0x08	Parameter is dynamic.
CFG_CALCULATED	16	0x10	Parameter is calculated.
CFG_READONLY	32	0x20	Parameter is read-only.
CFG_MEMORY_USED	64	0x40	Parameter consumes memory.
CFG_CONFIG_FILE	128	0x80	Parameter is externally visible.
CFG_SYSTEM_TAB	256	0x100	Parameter is externally visible only in system table.
CFG_EXTRAS_OPTION	512	0x200	Parameter is for CFG_EXTRAS not DS_CONFIG.
CFG_CFGBLK	1024	0x400	Parameter is stored in the configuration block.
CFG_CACHE_GROUP	2048	0x800	Parameter is a cache group.
CFG_CACHE_OPTION	4096	0x1000	Parameter is a cache option.
CFG_BUFFER_POOL_GROUP	8192	0x2000	Parameter is a buffer pool group.
CFG_BUFFER_POOL_OPTION	16384	0x4000	Parameter is a buffer pool option.
CFG_INTERNAL	32768	0x8000	Parameter is for internal use only.
CFG_FNOF_LPAGESIZE	65536	0x10000	Parameter entry depends on logical pagesize.

Indexes

- Unique clustered index on `name, parent, config`
- Nonclustered index on `config`
- Nonclustered index on `parent, config`

1.18 sysconstraints

Applies to all databases. Whenever a user declares a new check constraint or referential constraint using `create table` or `alter table`, the SAP ASE server inserts a row into the `sysconstraints` table. The row remains until a user executes `alter table` to drop the constraint. Dropping a table by executing `drop table` removes all rows associated with that table from the `sysconstraints` table.

This table also contains one row for each check constraint, referential constraint, rules, computed column, multiple triggers, and function-based index key associated with a specific table.

Columns

The columns for `sysconstraints` are:

Name	Datatype	Description
<code>colid</code>	<code>smallint</code>	Column number in the table
<code>constrid</code>	<code>int</code>	Object ID of the constraint
<code>tableid</code>	<code>int</code>	ID of the table on which the constraint is declared
<code>error</code>	<code>int</code>	Constraint-specific error message
<code>status</code>	<code>int</code>	The type of constraint: 0x0040 = a referential constraint 0x0080 = a check constraint 0x0100 = a computed column object constraint The status of triggers: 0x0080 = a delete trigger 0x0100 = an insert trigger 0x0200 = an update trigger 0x0400 = trigger is disabled
<code>spare2</code>	<code>int</code>	Unused

Indexes

- Unique clustered index on `tableid`, `colid`
- Nonclustered index on `constrid`

1.19 syscoordinations

Applies to sybsystemdb Database Only. `syscoordinations` contains information about remote SAP ASE servers participating in distributed transactions (remote participants) and their coordination states.

Columns

The columns for `syscoordinations` are:

Name	Datatype	Description
participant	smallint	Participant ID
starttime	datetime	Date the transaction started
coordtype	tinyint	Value indicating the coordination method or protocol in the <code>systransactions</code> table definition
owner	tinyint	Row owner (for internal use)
protocol	smallint	Reserved for internal use
state	int	Value indicating the current state of the remote participant: <ul style="list-style-type: none">• 1 – Begun• 4 – Prepared• 7 – Committed• 9 – In AbortTrans
bootcount	int	Reserved for internal use
dbid	smallint	Database ID at the start of the transaction.
logvers	tinyint	Reserved for internal use
spare	tinyint	Reserved for internal use
status	int	Reserved for internal use
xactkey	binary(14)	Unique SAP ASE transaction key
gtrid	varchar(255) null	Global transaction ID for distributed transactions coordinated by the SAP ASE server (reserved for internal use)

Name	Datatype	Description
partdata	varbinary(255) null	Reserved for internal use
srvname	varchar(30) null	Name of local server (null for remote servers)
nodeid	tinyint null	Not available for non-cluster environments – reserved for future use
instance id	tinyint	<i>Cluster environments only</i> – ID of the instance

Indexes

Unique clustered index on `xactkey`, `participant`, `owner`

1.20 syscurconfigs

Applies to master database only. `syscurconfigs` is built dynamically when queried. It contains an entry for each of the configuration parameters, as does `sysconfigures`, but with the current values rather than the default values. In addition, it contains four rows that describe the configuration structure.

Columns

The columns for `syscurconfigs` are:

Name	Datatype	Description
config	smallint	Configuration parameter number.
value	int	The current run value for the parameter with <code>integer</code> datatype. Its value is 0 for the parameters with <code>character</code> datatype.
comment	varchar(255)	Comments about the configuration parameter. For internal use.
status	int	Value that represents the type of configuration parameter.
value2	varchar(255) null	The current run value for the parameter with the <code>character</code> datatype. Its value is NULL for parameters with the <code>integer</code> datatype.

Name	Datatype	Description
defvalue	varchar(255) null	Default value of the configuration parameter.
minimum_value	int null	Minimum value of the configuration parameter.
maximum_value	int null	Maximum value of the configuration parameter.
memory_used	bigint null	Integer value for the amount of memory used by each configuration parameter. Negative values indicate memory shared.
display_level	int null	Display level of the configuration parameter. The values are 1, 5, and 10.
datatype	int null	Datatype of the configuration parameter.
message_num	int null	Message number of the sp_helpconfig message for this parameter.
apf_percent	int null	The current run value for the asynchronous prefetch percent for a buffer pool. Valid only for rows that represent buffer pools.
nodeid	tinyint null	Reserved for future use (not available in cluster environments)
instanceid	tinyint	ID of the instance (available only for cluster environments)

Name	Datatype	Description
unit	varchar(20)	Unit of the parameter. Values are: <ul style="list-style-type: none"> • Not applicable – parameter has no units • Number – number of items • Clock ticks – number of clock ticks • Microseconds • Milliseconds • Seconds • Minutes • Hours • Days • Bytes • Kilobytes • Megabytes • Memory pages (2K) • Virtual pages (2K) • Logical pages • Percent • Ratio • Switch – a Boolean value • ID – ID number • Name • Rows
type	varchar(10) null	Specifies whether a configuration parameter is declared dynamic or static in its structure definition. Values are: <ul style="list-style-type: none"> • Dynamic – takes effect immediately. • Static – takes effect after restarting the SAP ASE server.

This table describes status types:

Status type	Decimal	Hex	Description
CFG_NO_OPTIONS	0	0x0	Parameter has no options.
CFG_SYSTEM_OPTION	1	0x01	Parameter is a system option.
CFG_SYSTEM_GROUP	2	0x02	Parameter is a system group.
CFG_STATIC	4	0x04	Parameter is static.
CFG_DYNAMIC	8	0x08	Parameter is dynamic.
CFG_CALCULATED	16	0x10	Parameter is calculated.

Status type	Decimal	Hex	Description
CFG_READONLY	32	0x20	Parameter is read-only.
CFG_MEMORY_USED	64	0x40	Parameter consumes memory.
CFG_CONFIG_FILE	128	0x80	Parameter is externally visible.
CFG_SYSTEM_TAB	256	0x100	Parameter is only externally visible in system table.
CFG_EXTRAS_OPTION	512	0x200	Parameter is for CFG_EXTRAS not DS_CONFIG.
CFG_CFGBLK	1024	0x400	Parameter is stored in the configuration block.
CFG_CACHE_GROUP	2048	0x800	Parameter is a cache group.
CFG_CACHE_OPTION	4096	0x1000	Parameter is a cache option.
CFG_BUFFER_POOL_GROUP	8192	0x2000	Parameter is a buffer pool group.
CFG_BUFFER_POOL_OPTION	16384	0x4000	Parameter is a buffer pool option.
CFG_INTERNAL	32768	0x8000	Parameter is for internal use only.
CFG_FNOF_LPAGESIZE	65536	0x10000	Parameter entry depends on logical pagesize.

1.21 sysdatabases

Applies to master database only. `sysdatabases` contains one row for each database in the SAP ASE server. When the SAP ASE server is installed, `sysdatabases` contains entries for the following databases:

- `master`
- `model`
- `sybserverprocs`
- `tempdb` database.
- `sybsecurity`

Columns

The columns for `sysdatabases` are:

Name	Datatype	Description
<code>name</code>	<code>sysname</code>	Name of the database.

Name	Datatype	Description
dbid	smallint	Database ID.
suid	int	Server user ID of Database Owner.
status	smallint	Control bits.
status5	int	Indicates the status of a database: <ul style="list-style-type: none"> • 0x00000001 Indicates whether the database is encrypted or not. • 0x00000002 – The database is being encrypted, and the encryption is still in progress. • 0x00000004 – The database is being decrypted, and the decryption is still in progress. • 0x00000008 – The database is only partially encrypted, either due to an error or because the process was suspended by the user. • 0x00000010 – The database is only partially decrypted, either due to an error or because the process was suspended by the user. • 0x00000100 – The <code>imrscache</code> column stores the name of the default IMRS cache for a database. • 0x00000200 – Database-wide snapshot isolation and MVCC is enabled. • 0x00000400 – Database wide DRC is enabled. • 0x00001000 – Database wide On-disk versioning is enabled. • 0x00004000 – Version storage is enabled and a valid temporary database name is specified. • 0x00010000 – ILM partition tuning is enabled.
version	smallint	Unused.
logptr	int	Pointer to transaction log.
crdate	datetime	Creation date.
dumptrdate	datetime	Date of the last dump transaction.
status2	smallint	Additional control bit.
audflags	int	Audit settings for database.
deftabaud	int	Bit-mask that defines default audit settings for tables.
defvwaud	int	Bit-mask that defines default audit settings for views.
defpraud	int	Bit-mask that defines default audit settings for stored procedures.
def_remote_ty pe	smallint	Identifies the default object type to be used for remote tables if no storage location is provided via the stored procedure <code>sp_addobjectdef</code> .

Name	Datatype	Description
def_remote_lo c	varchar (349)	Identifies the default storage location to be used for remote tables if no storage location is provided via the stored procedure <code>sp_addobjectdef</code> .
status3	int	Additional control bits.
status4	int	Additional control bits.
audflags2	varbinary (16)	Reserved for future use.
instanceid spare	tinyint	(Cluster Edition only) Instance ID of owning instance of a local temporary databases. For other databases, it remains NULL. In nonclustered editions of SAP ASE, this is the <code>spare</code> column, and is reserved for future use.
inrowlen	smalldint	Database-wide in-row LOB column length.
dcompdefaultl evel	tinyint	The level that <code>create table</code> , <code>alter table</code> , or <code>reorg rebuild</code> uses to set the level of compression for on a table (or partition).
durability	int	Durability level of the database. Values are: <ul style="list-style-type: none"> • 1 - full • 5 - at_shutdown • 6 - no_recovery
lobcomp_lvl	tinyint	LOB compression level.
enckeyid	int	Database encryption key ID. A null indicates the corresponding database is not encrypted.
imrscache	varchar (255)	Name of the row storage cache assigned to the database.
imrslogptr	int	Pointer to the start of the <code>imrslog</code> .
imrsloglastpt r	int	Pointer to the end of the <code>imrslog</code> .
versiondbid	smallint	Database ID of the temporary database used for version storage for on-disk MVCC-enabled databases.

This table lists the bit representations for the `status` column:

Decimal	Hex	Status
2	0x02	Database is using signed pages.

Decimal	Hex	Status
4	0x04	<code>select into/bulkcopy</code> ; can be set by user.
8	0x08	<code>trunc log on chkpt</code> ; can be set by user.
16	0x10	<code>no chkpt on recovery</code> ; can be set by user.
32	0x20	Database created with <code>for load</code> option, or crashed while loading database, instructs recovery not to proceed.
64	0x04	Recovery started for all databases to be recovered.
256	0x100	<ul style="list-style-type: none"> • Database suspect • Not recovered • Cannot be opened or used • Can be dropped only with <code>dbcc dbrepair</code>
512	0x200	<code>ddl in tran</code> ; can be set by user.
1024	0x400	<code>read only</code> ; can be set by user.
2048	0x800	<code>dbo use only</code> ; can be set by user.
4096	0x1000	<code>single user</code> ; can be set by user.
8192	0x2000	<code>allow nulls by default</code> ; can be set by user.

This table lists the bit representations for the `status2` column:

Decimal	Hex	Status
1	0x0001	<code>abort tran on log full</code> ; can be set by user.
2	0x0002	<code>no free space acctg</code> ; can be set by user.
4	0x0004	<code>auto identity</code> ; can be set by user.
8	0x0008	<code>identity in nonunique index</code> ; can be set by user.
16	0x0010	Database is offline.
32	0x0020	Database is offline until recovery complete.s
64	0x0040	The table has an auto identity feature, and a unique constraint on the <code>identity</code> column.
128	0x0080	Database has suspect pages.
256	0x0100	Table structure written to disk. If this bit appears after recovery completes, server may be under-configured for open databases. Use <code>sp_configure</code> to increase this parameter.

Decimal	Hex	Status
512	0x0200	Database is in the process of being upgraded.
1024	0x0400	Database brought online for standby access.
2048	0x0800	When set by the user, prevents cross-database access via an alias mechanism.
-32768	0xFFFF8000	Database has some portion of the log which is not on a log-only device.

This table lists the bit representations for the `status3` column:

Decimal	Hex	Status
0	0x0000	A normal or standard database, or a database without a proxy update in the <code>create</code> statement.
1	0x0001	You specified the <code>proxy_update</code> option, and the database is a user-created proxy database.
2	0x0002	Database is a proxy database created by high availability.
4	0x0004	Database has a proxy database created by high availability.
8	0x0008	Disallow access to the database, since database is being shut down.
16	0x0010	Database is a failed-over database.
32	0x0020	Database is a mounted database of the type <code>master</code> .
64	0x0040	Database is a mounted database.
128	0x0080	Writes to the database are blocked by the <code>quiesce database</code> command.
256	0x0100	User-created <code>tempdb</code> .
512	0x0200	Disallow external access to database in the server in failed-over state.
1024	0x0400	User-provided option to enable or disable asynchronous logging service threads. Enable through <code>sp_dboption enable async logging service</code> option set to true on a particular database.
4096	0x1000	Database has been shut down successfully.
8192	0x2000	A <code>drop database</code> is in progress.

This table lists the bit representations for the `status4` column:

Decimal	Hex	Status
512	0x0200	The in-memory database has a template database with it.
4096	0x1000	Database is an in-memory databases.

Decimal	Hex	Status
16384	0x4000	64-bit atomic operations have been enabled on this database.
32768	0x00008000	Enforce <code>dump tran</code> sequence. Disallows operations that will fail a subsequent <code>dump tran</code> .
16777216	0x01000000	All tables in the database are created as page compressed.
33554432	0x02000000	All tables in the database are created as row compressed.

This table lists the bit representations for the `status5` column:

Decimal	Hex	Status
128	0x80	The name of the IMDB template database is stored in the <code>def_remote_loc</code> column.
256	0x100	A row storage cache is currently assigned to this database.
512	0x200	Timestamp-based snapshot isolation (MVCC) is automatically enabled for all newly created user tables in this database.
1024	0x400	Data row caching is automatically enabled for all newly created user tables in this database.

Indexes

- Unique clustered index on `name`
- Nonclustered index on `dbid`

1.22 sysdepends

Applies to all databases. `sysdepends` contains one row for each procedure, view, or table that is referenced by a procedure, view, or trigger.

Columns

The columns for `sysdepends` are:

Name	Datatype	Description
<code>id</code>	<code>int</code>	Object ID.
<code>number</code>	<code>smallint</code>	Procedure number.
<code>depid</code>	<code>int</code>	Dependent object ID.
<code>depnumber</code>	<code>smallint</code>	Dependent procedure number.
<code>status</code>	<code>smallint</code>	Internal status information.
<code>selall</code>	<code>bit</code>	On if object is used in <code>select *</code> statement.
<code>resultobj</code>	<code>bit</code>	On if object is being updated.
<code>readobj</code>	<code>bit</code>	On if object is being read.
<code>columns</code>	<code>varbinary</code>	Stores a bitmap of column IDs of columns that are referenced in the body of a stored procedure. This bitmap gives column-level dependency tracking information, including predicated privileges, for compiled objects, and is decoded by <code>sp_depends</code> to report on column-level dependencies for stored procedures, triggers, and views.

Indexes

Unique clustered index on `id, number, depid, depnumber`

1.23 sysdevices

Applies to master database only. `sysdevices` contains one row for each tape dump device, disk dump device, disk for databases, and disk partition for databases. There are four entries in `sysdevices` in the SAP ASE

distribution media: one for the master device (for databases), one for a disk dump device, and two for tape dump devices.

i Note

The device identification number is stored in the `vdevno` column and is not as part of the `high` or `low` column. As a consequence, you may need to modify scripts and stored procedures that determine the device identification number based on the earlier schema.

Columns

The columns for `sysdevices` are:

Name	Datatype	Description
<code>low</code>	<code>int</code>	(Not used for dump devices) Block offset of virtual page in 2K bytes.
<code>high</code>	<code>int</code>	Block offset of last virtual page in 2K bytes.
<code>status</code>	<code>smallint</code>	Bitmap indicating type of device, default, and mirror status.
<code>cntrltype</code>	<code>smallint</code>	Controller type: <ul style="list-style-type: none"> • 0 = Database device • 2 = Disk dump device or streaming tape • 3–8 = Tape dump device
<code>name</code>	<code>sysname</code>	Logical name of dump device, database device, or in-memory storage cache.
<code>phyname</code>	<code>varchar(127)</code>	Name of physical device or in-memory storage cache.
<code>mirrorname</code>	<code>varchar(127)</code> <code>null</code>	Name of mirror device.
<code>vdevno</code>	<code>int</code>	Device identification number.
<code>crdate</code>	<code>datetime</code> <code>null</code>	Date on which the device was added.
<code>resizedate</code>	<code>datetime</code> <code>null</code>	Date on which <code>disk resize</code> was most recently run for this device.
<code>status2</code>	<code>int</code>	Additional status bits for this device.
<code>class</code>	<code>varchar(2)</code>	
<code>instanceid</code>	<code>tinyint</code>	ID of the instance (available only for cluster environments).

Name	Datatype	Description
uuid	varbinary(16)	Reserved for future use (available only for cluster environments).

The bit representations for the `status` column, shown below, are additive. For example, 3 indicates a physical disk that is also a default.

Decimal	Hex	Status
1	0x01	Default disk
2	0x02	Physical disk
4	0x04	(Not used) – logical disk
8	0x08	Skip header
16	0x10	Dump device
32	0x20	Serial writes
64	0x40	Device mirrored
128	0x80	Reads mirrored
256	0x100	Secondary mirror side only
512	0x200	Mirror enabled
1024	0x400	Master device is mirrored
2048	0x800	(Used internally) mirror disabled
4096	0x1000	(Used internally) primary device must be unmirrored
8192	0x2000	(Used internally) secondary device must be unmirrored
16384	0x4000	UNIX file device uses <code>dsync</code> setting (writes flushed to physical media)

The bit representation for the `status2` column is:

Decimal	Hex	Status
1	0x01	Direct I/O is enabled for this device
16	0x0010	Identifies <code>imrslog</code> devices

Indexes

Unique clustered index on name

1.24 sysencryptkeys

Applies to all databases. Each key created in a database, including the default key, has an entry in the database-specific system catalog `sysencryptkeys`.

Columns

The columns for `sysencryptkeys` are:

Field	Type	Description
id	int	Encryption key ID.
ekalgori thm	int	Encryption algorithm.
type	smallint	Identifies the key type. The values are: <ul style="list-style-type: none">• 0x1 (decimal 1) – Symmetric key• 0x4 (decimal 4) – Default key• 0x10 (decimal 16) – Key copy• 0x40 (decimal 64) – Recovery key copy
status	int	Internal status information. The bit representations are: <ul style="list-style-type: none">• 0x1 (decimal 1) – Key uses initialization vector• 0x2 (decimal 2) – Key uses random pad• 0x4 (decimal 4) – Key is encrypted for lost password protection• 0x8 (decimal 8) – Key copy encrypted for login access• 0x10 (decimal 16) – Key copy encrypted with login password• 0x20 (decimal 32) – Key copy encrypted with system encryption password• 0x100 (decimal 256) – Key encrypted with user password
eklen	smallint	User-specified length of key.
value	varbinary(1282)	Encrypted value of a key. Contains a symmetric encryption of the key. To encrypt keys, the SAP ASE server uses AES with a 128-bit key from the system encryption, user-specified, or login password.

Field	Type	Description
uid	int null	User ID of key copy assignee.
eksalt	varbinary (20)	Random values used to validate decryption of the encryption key.
ekpairid	int null	Not used.
pwdate	datetime null	Date the password was last changed.
expdate	int null	Not used.
ekpwdwar n	int null	Not used.

The status bits for `sysencryptkeys`

Decimal	Hex	Status
	0x00000004	EK_KEYRECOVERY() – keys encrypted for lost password protection.
	0x00000008	EK_LOGINACCESS() – key encrypted for login access
	0x00000010	EK_LOGINPASS () – key encrypted with login password
	0x00000100	EK_USERPWD() – keys encrypted with user-encryption passwords

1.25 sysengines

Applies to master database only. `sysengines` contains one row for each SAP ASE engine currently online.

Columns

The columns for `sysengines` are:

Name	Datatype	Description
engine	smallint	Engine number.
osprocid	int	<ul style="list-style-type: none"> Process mode – operating system process ID Threaded mode – operating system thread (LWP) ID

Name	Datatype	Description
osprocname	char (32)	Operating system process name (may be NULL).
status	char (12)	One of: online, in offline, in create, in destroy, debug, bad status.
affinitied	int	Number of SAP ASE processes with affinity to this engine..
cur_kpid	int	Kernel process ID of process currently running on this engine, if any
last_kpid	int	Kernel process ID of process that previously ran on this engine.
idle_1	tinyint	Reserved.
idle_2	tinyint	Reserved.
idle_3	tinyint	Reserved.
idle_4	tinyint	Reserved.
starttime	datetime	Date and time engine came online.
nodeid	tinyint null	Reserved for future use (not available for cluster environments).
instanceid	tinyint	ID of the instance (available only for cluster environments).

1.26 sysgams

Applies to all databases. *sysgams* stores the global allocation map (GAM) for the database. The GAM stores a bitmap for all allocation units of a database, with one bit per allocation unit. You cannot select from or view *sysgams*.

1.27 sysimrlogs

Applies to all in-memory databases. *sysimrlogs* contains the in-memory transaction log. It is used by the SAP ASE server for recovery and roll forward.

You can run select queries against the *sysimrlogs* system table, but you cannot delete from, insert into, or update *sysimrlogs*. Every data modification operation is logged, so before you can change *sysimrlogs*, the change must be logged. This means that a change operation on *sysimrlogs* adds a row to *sysimrlogs*, which then must be logged, adding another row to *sysimrlogs*, and so on, producing an infinite loop. The loop continues until the database becomes full.

Columns

The columns for `sysimrlogs` are:

Name	Datatype	Description
<code>xactid</code>	<code>binary(6)</code>	Transaction ID
<code>op</code>	<code>tinyint</code>	Number of update operation

1.28 sysindexes

Applies to all databases. `sysindexes` contains one row for each clustered index, one row for each nonclustered index, one row for each table that has no clustered index, and one row for each table that contains `text` or `image` columns. This table also contains one row for each function-based index or index created on a computed column.

Columns

The columns for `sysindexes` are:

Name	Datatype	Description
<code>name</code>	<code>varchar(255)) null</code>	Index or table name.
<code>id</code>	<code>int</code>	ID of an index, or ID of table to which index belongs.
<code>indid</code>	<code>smallint</code>	Valid values are: <ul style="list-style-type: none">• 0 = if a table.• 1 = if a clustered index on an allpages-locked table.• >1 = if a nonclustered index or a clustered index on a data-only-locked table.• 255 = if <code>text</code>, <code>image</code>, text chain, or Java off-row structure (large object—or LOB—structure).
<code>doampg</code>	<code>int</code>	Obsolete.
<code>ioampg</code>	<code>int</code>	Obsolete.
<code>oampgtrips</code>	<code>int</code>	Number of times OAM pages cycle in the cache without being reused, before being flushed.

Name	Datatype	Description
status3	smallint	Internal system status information.
status2	smallint	Internal system status information.
ipgtrips	int	Number of times index pages cycle in the cache, without being reused, before being flushed
first	int	Obsolete.
root	int	Obsolete.
distribution	int	Unused. Formerly used to store the page number of the distribution page for an index.
usagecnt	smallint	Reserved.
segment	smallint	Number of segment in which object resides.
status	smallint	Internal system status information.
maxrowsperpage	smallint	Maximum number of rows per page.
minlen	smallint	Minimum size of a row.
maxlen	smallint	Maximum size of a row.
maxirow	smallint	Maximum size of a non-leaf index row.
keycnt	smallint	Number of keys for a clustered index on an allpages-locked table; number of keys, plus 1 for all other indexes.
keys1	varbinary (255) null	Description of key columns if entry is an index.
keys2	varbinary (255) null	Description of key columns if entry is an index.
soid	tinyint	Sort order ID with which the index was created; 0 if there is no character data in the keys.
csid	tinyint	Character set ID with which the index was created; 0 if there is no character data in the keys.
base_partition	int null	Obsolete.

Name	Datatype	Description
fill_factor	smallint null	Value for the fillfactor of a table set with <code>sp_chgattribute</code> .
res_page_gap	smallint null	Value for the reservepagegap on a table.
exp_rowsize	smallint null	Expected size of data rows.
keys3	varbinary (255) null	Description of key columns if entry is an index.
identitygap	int null	Identity gap for a table.
crdate	datetime null	Creation date.
partitiontype	smallint null	Values are: <ul style="list-style-type: none"> • 1 = range • 2 = hash • 3 or NULL = [default] round robin • 4 = list
conditionid	int null	ID of the partition condition. Null if <code>partitiontype</code> is round-robin or hash.

This table lists the bit representations for the `status` column:

Decimal	Hex	Status
1	0x1	Abort current command or trigger if attempt to insert duplicate key.
2	0x2	Unique index.
4	0x4	Abort current command or trigger if attempt to insert duplicate row; always 0 for data-only-locked tables.
16	0x10	Table is an all-pages-locked table with a clustered index.
64	0x40	Index allows duplicate rows, if an allpages-locked table; always 0 for data-only-locked tables.
128	0x80	Sorted object toggle that is being used internally. Can be set by <code>create clustered index</code> , <code>reorg rebuild</code> , or <code>alter table locking scheme</code> commands.
512	0x200	<code>sorted data</code> option used in <code>create index</code> statement.
2048	0x800	Index on primary key.

Decimal	Hex	Status
32768	0x8000	Suspect index; index was created under another sort order.

This table lists the bit representations for the `status2` column:

Decimal	Hex	Status
1	0x1	Index supports foreign-key constraint.
2	0x2	Index supports primary key/unique declarative constraint.
4	0x4	Index includes an IDENTITY column.
8	0x8	Constraint name not specified.
16	0x10	Large I/Os (prefetch) not enabled for table, index, or text chain.
32	0x20	Most recently used (MRU) cache strategy not enabled for table, index, or text chain.
64	0x40	Ascending inserts turned on for the table.
256	0x0100	Index is presorted and does not need to be copied to new extents.
512	0x0200	Index is a DOL clustered index.
8192	0x2000	Index on a data-only-locked table is suspect.
32768	0x8000	The index is function-based.

Indexes

Unique clustered index on `id`, `indid`

1.29 sysinstances

Applies to the Cluster Edition only. A fake table that reports on the state of the instances. `sysinstances` includes a row for each instance defined in the cluster configuration.

Although `sysinstances` is a fake table, it is not impacted by the setting of `set system_view`, and always returns a row for each instance, regardless of the `system_view` setting.

Columns

The columns for `sysinstances` are:

Column name	Datatype	Description
<code>id</code>	<code>tiny int</code>	ID of the instance
<code>name</code>	<code>varchar(30)</code>	Name of the instance
<code>state</code>	<code>char(17)</code>	State of the instance (one of <code>online</code> , <code>offline</code> , <code>joining</code> , <code>leaving</code> , and <code>initiating</code>)
<code>hostname</code>	<code>varchar(255)</code>	Name of the operating system host running this instance
<code>starttime</code>	<code>datetime</code>	Date and time the instance started
<code>connections_active</code>	<code>int</code>	Number of active connections on the instance
<code>engines_online</code>	<code>smallint</code>	Number of online engines for this instance

1.30 sysjars

Applies to all databases. `sysjars` contains one row for each Java archive (JAR) file that is retained in the database.

For more information about JAR files, Java classes, and Java datatypes, see *Java in Adaptive Server Enterprise*.

Columns

The columns for `sysjars` are:

Name	Datatype	Description
<code>jid</code>	<code>int</code>	The ID of the JAR.
<code>jstatus</code>	<code>smallint</code>	Internal status information. Unused.
<code>jname</code>	<code>varchar(255) null</code>	The JAR name.
<code>jbinary</code>	<code>image null</code>	The contents of the JAR: the Java classes.

Indexes

- Unique clustered index on `jid`
- Unique nonclustered index on `jname`

1.31 syskeys

Applies to all databases. `syskeys` contains one row for each primary, foreign, or common key.

Columns

The columns for `syskeys` are:

Name	Datatype	Description
<code>id</code>	<code>int</code>	Object ID.
<code>type</code>	<code>smallint</code>	Record type. Valid values are: <ul style="list-style-type: none">• 1 = primary key• 2 = foreign key• 3 = common key
<code>depid</code>	<code>int null</code>	Dependent object ID.
<code>keycnt</code>	<code>int null</code>	Number of non-null keys.
<code>size</code>	<code>int null</code>	Reserved.
<code>key1 ... key8</code>	<code>smallint null</code>	Column ID.
<code>depkey1 ... depkey8</code>	<code>smallint null</code>	Column ID.
<code>spare1</code>	<code>smallint</code>	Reserved.

Indexes

Clustered index on `id`

1.32 syslanguages

Applies to master database only. `syslanguages` contains one row for each language known to SAP ASE. `us_english` is not in `syslanguages`, but it is always available to the SAP ASE server.

Columns

The columns for `syslanguages` are:

Name	Datatype	Description
<code>langid</code>	<code>smallint</code>	Unique language ID.
<code>dateformat</code>	<code>char(3)</code>	Date order; for example, "dmy".
<code>datefirst</code>	<code>tinyint</code>	First day of the week—1 for Monday, 2 for Tuesday, and so on, up to 7 for Sunday.
<code>upgrade</code>	<code>int</code>	SAP ASE version of last upgrade for this language.
<code>name</code>	<code>varchar(30)</code>	Official language name, for example, "french".
<code>alias</code>	<code>varchar(30)</code> <code>null</code>	Alternate language name, for example, "français".
<code>months</code>	<code>varchar(251)</code> <code>)</code>	Comma-separated list of full-length month names, in order from January to December—each name is at most 20 characters long.
<code>shortmonths</code>	<code>varchar(119)</code> <code>)</code>	Comma-separated list of shortened month names, in order from January to December—each name is at most 9 characters long.
<code>days</code>	<code>varchar(216)</code> <code>)</code>	Comma-separated list of day names, in order from Monday to Sunday—each name is at most 30 characters long.

Indexes

- Unique clustered index on `langid`
- Unique nonclustered index on `name`
- Unique nonclustered index on `alias`

1.33 syslisteners

Applies to master database only. `syslisteners` contains a row for each network protocol available for connecting with the current SAP ASE server. The SAP ASE server builds `syslisteners` dynamically when a user or client application queries the table.

Columns

The columns for `syslisteners` are:

Name	Datatype	Description
<code>net_type</code>	<code>char(32)</code>	Network protocol
<code>address_info</code>	<code>char(255)</code>	Information that uniquely identifies this SAP ASE server on the network; usually the name of the current SAP ASE server and an identifying number, such as the server's port number for the protocol
<code>spare</code>	<code>tinyint</code>	Unused
<code>nodeid</code>	<code>tinyint</code> <code>null</code>	Reserved for future use (not available for cluster environments)
<code>instanceid</code>	<code>tinyint</code>	ID of the instance (available only for cluster environments)

1.34 syslocks

Applies to master database only. `syslocks` contains information about active locks, and built dynamically when queried by a user. No updates to `syslocks` are allowed.

Columns

The columns for `syslocks` are:

Name	Datatype	Description
<code>id</code>	<code>int</code>	Table ID.

Name	Datatype	Description
dbid	smallint	Database ID.
page	unsigned int	Page number.
type	smallint	Type of lock.
spid	smallint int for the Cluster Edition	ID of process that holds the lock.
class	varchar (30)	Name of the cursor this lock is associated with, if any.
fid	smallint int for the Cluster Edition	The family (coordinating process and its worker processes) to which the lock belongs. <i>fid</i> values are: <ul style="list-style-type: none"> • 0 – the task represented by the <i>spid</i> is a single task executing a statement in serial • Nonzero – the task (<i>spid</i>) holding the lock is a member of a family executing a statement in parallel. If the value is equal to the <i>spid</i>, it indicates that the task is the coordinating process in a family executing a query in parallel.
context	tinyint	Context type of lock request.
row	smallint	Row number.
loid	int	Unique lock owner ID.
partitionid	int null	Partition ID.
nodeid	tinyint null	Reserved for future use (not available for cluster environments)
instanceid	tinyint	ID of the instance (available only for cluster environments)

The bit representations for the *type* column are:

Decimal	Hex	Status
1	0x1	Exclusive table lock

Decimal	Hex	Status
2	0x2	Shared table lock
3	0x3	Exclusive intent lock
4	0x4	Shared intent lock
5	0x5	Exclusive page lock
6	0x6	Shared page lock
7	0x7	Update page lock
8	0x8	Exclusive row lock
9	0x9	Shared row lock
10	0xA	Update row lock
11	0xB	Shared next key lock
256	0x100	Lock is blocking another process
512	0x200	Demand lock

The values for the `context` column are:

Value	Interpretation
null	The task holding this lock is either executing a query in serial, or it is a query being executed in parallel in transaction isolation level 1.
0x1	The task holding the lock will hold the lock until the query is complete. A lock's context may be FAM_DUR (0x1H) when the lock is: <ul style="list-style-type: none"> • A table lock held as part of a parallel query. • Held by a worker process at transaction isolation level 3. • Held by a worker process in a parallel query and must be held for the duration of the transaction.
0x2	Range lock held by serializable read task.
0x4	Infinity key lock.
0x8	Lock acquired on an index pages of an allpages-locked table.
0x10	Lock on a page or row acquired to delete a row.
0x20	Address lock acquired on an index page during a shrink or split operation.
0x40	Intent lock held by a transaction performing repeatable reads. Valid only for shared intent and exclusive intent locks on data-only-locked tables.

1.35 syslogininfo

Returns information for all logins. `syslogininfo` is a view of the master database that provides information about all logins. You must have the `sa_role` or `sso_role` to select from the `syslogininfo` view.

Columns

The columns for `syslogininfo` are:

Name	Datatype	Description
<code>suid</code>	<code>int</code>	Server user ID or login profile ID.
<code>login_name</code>	<code>sysname</code>	Login name.
<code>pwd_expired</code>	<code>varchar(3)</code>	Indicates if the password is expired.
<code>locked</code>	<code>varchar(3)</code>	Indicates if the login or account are locked.
<code>date_of_last_password_change</code>	<code>datetime</code> <code>null</code>	Date when the password was last changed.
<code>pwd_expiry_interval</code>	<code>int</code>	Interval, in number of days, for the password expiration.
<code>pwd_expiry_remaining</code>	<code>int</code>	Number of days remaining before the password expires.

1.36 sysloginroles

Applies to master database only. `sysloginroles` contains a row for each instance of a server login or login profile possessing a role. One row is added for each role granted to each login. For example, if a single server user is granted `sa_role`, `sso_role`, and `oper_role`, three rows are added to `sysloginroles` associated with that user's system user ID (`suid`).

Note

When you change the status bit using `alter login`, you must log out and relog for the change to take effect. To see immediate results, use `set role role_name off`.

Columns

The columns for `sysloginroles` are:

Name	Datatype	Description
<code>suid</code>	<code>int</code>	Server user ID or login profile ID
<code>srid</code>	<code>int</code>	Server role ID. See the list below.
<code>status</code>	<code>smallint</code>	Status bit that indicates whether the various server roles are set to their defaults at login: <ul style="list-style-type: none">• 0 – default off• 1 – default on
<code>predid</code>	<code>int</code>	The object ID for the predicate of a <code>grant role</code> command. See <i>Security Administration Guide > Predicated role activation</i> .

The values for `srid` are:

- 0 – `sa_role`
- 1 – `sso_role`
- 2 – `oper_role`
- 4 – `navigator_role`
- 5 – `replication_role`
- 6 – Currently unused
- 7 – `dtm_tm_role`
- 8 – `ha_role`
- 9 – Used internally
- 10 – `mon_role`
- 11 – `js_admin_role`
- 12 – `messaging_role`
- 13 – `js_client_role`
- 14 – `js_user_role`
- 15 – `webservices_role`

Indexes

Clustered index on `suid`

1.37 syslogins

Applies to master database only. `syslogins` contains one row for each valid SAP ASE user account or login profile.

Columns

The columns for `syslogins` are:

Name	Datatype	Description
<code>suid</code>	<code>int</code>	Server user ID or login profile ID.
<code>status</code>	<code>smallint</code>	Status of the account.
<code>accdate</code>	<code>datetime</code>	Date <code>totcpu</code> and <code>totio</code> were last cleared.
<code>totcpu</code>	<code>int</code>	CPU time accumulated by login.
<code>totio</code>	<code>int</code>	I/O accumulated by login.
<code>spacelimit</code>	<code>int</code>	Reserved.
<code>timelimit</code>	<code>int</code>	Reserved.
<code>resultlimit</code>	<code>int</code>	Reserved.
<code>dbname</code>	<code>sysname</code> <code>null</code>	Name of database in which to put user when connection established. Column is not applicable for a login row if a login profile is associated with the login account.
<code>name</code>	<code>sysname</code>	Login name of user.
<code>password</code>	<code>varbinary(128)</code> <code>null</code>	One-way hash of user password. The contents of <code>syslogins.password</code> depend on the value for <code>sp_passwordpolicy allow password downgrade</code> .
<code>language</code>	<code>varchar(30)</code> <code>null</code>	User's default language. If a login profile is associated with the login account, this column is not applicable for a login row.
<code>pwdate</code>	<code>datetime</code> <code>null</code>	Date the password was last changed.

Name	Datatype	Description
audflags	int null	User's audit settings. One of: <ul style="list-style-type: none"> • 0x00000001 – successful reference to a user-created table • 0x00000002 – failure • 0x00000004 – successful reference to a user-created view • 0x00000008 – failure • 0x00000010 – user cmdtext auditing • 0x00000020 – required padding • 0x00000040 – all successful user action auditing • 0x00000080 – all failed user action auditing
fullname	varchar(30) null	Full name of the user.
srvname	varchar(30) null	Name of server to which a passthrough connection must be established if the AUTOCONNECT flag is turned on.
logincount	smallint null	Number of failed login attempts; reset to 0 by a successful login.
procid	int null	Stores the login trigger registered with the <code>login script</code> . If a login profile is associated with the login account, this column is not applicable for a login row.
lastlogin date	datetime	Timestamp for the user's last login.
crdate	datetime	Timestamp when the login or login profile was created.
locksuid	int	The server user ID (suid) responsible for locking the login.
lockreason	int	Reasons for lock; one of: <ul style="list-style-type: none"> • NULL – account has not been locked • 0 – locked by <code>locksuid</code> by executing <code>sp_locklogin</code> • 1 – inactive account locked by executing <code>sp_locklogin 'all', 'lock', 'ndays'</code> • 2 – the SAP ASE server locked the account because number of failed login attempts reached <code>max failed logins</code>. • 3 – locked by <code>locksuid</code> because the password downgrade period has ended and a login or role was not transitioned to SHA-256 • 4 – automatically locked by <code>locksuid</code> due to inactivity.

Name	Datatype	Description
lockdate	datetime	If: <ul style="list-style-type: none"> The login account is locked – <code>syslogins.lockdate</code> specifies the timestamp when the login was locked. The login account is not locked, and: <ul style="list-style-type: none"> <code>syslogins.lockdate</code> is non-NULL – specifies the timestamp when the login was unlocked. <code>syslogins.lockdate</code> is NULL – specifies that the login was never locked.
lpid	int	Login profile ID. One of: <ul style="list-style-type: none"> <code>null</code> – the login account is associated with the default login profile, if any <code>-1</code> – the login profile is ignored for the login account. <code>suid</code> – the login profile ID.
crsuid	int	Server user ID of the creator of login or login profile.

On the SAP ASE distribution media, `syslogins` contains an entry in which the name is “sa”, the `suid` is 1, and the password is null. It also contains the entry “probe” with an unpublished password. The login “probe” and the user “probe” exist for the two-phase commit probe process, which uses a challenge and response mechanism to access the SAP ASE server.

The status control bits in `syslogins` are:

Decimal	Hex	Status
2	0x2	Account is locked.
4	0x4	Password has expired. This is set on the user's first login attempt after expiration.
8	0x8	Indicates that the value of <code>exempt_inactive_lock</code> is set to TRUE. It is not applicable for login profile rows.
16	0x10	OMNI:autoconnect mode is enabled.
32	0x20	May use SAP ASE internal authentication mechanism – <code>syslogins</code> .
64	0x40	May use LDAP external authentication.
128	0x80	May use PAM external authentication.
256	0x100	May use Kerberos external authentication.
512	0x200	Indicates a login profile.
1536	0x200 0x400	Indicates a default login profile.
2048	0x800	Indicates an authentication mechanism specified in a login profile.

Indexes

- Unique clustered index on `suid`
- Unique nonclustered index on `name`

1.38 syslogs

Applies to all databases. `syslogs` contains the transaction log. It is used by the SAP ASE server for recovery and roll forward. It is not useful to users.

You cannot delete from, insert into, or update `syslogs`. Every data modification operation is logged, so before you can change `syslogs`, the change must be logged. This means that a change operation on `syslogs` adds a row to `syslogs`, which then must be logged, adding another row to `syslogs`, and so on, producing an infinite loop. The loop continues until the database becomes full.

Columns

The columns for `syslogs` are:

Name	Datatype	Description
<code>xactid</code>	<code>binary(6)</code>	Transaction ID
<code>op</code>	<code>tinyint</code>	Number of update operation

1.39 syslogshold

Applies to master database only. `syslogshold` contains information about each database's oldest active transaction (if any) and the Replication Server truncation point (if any) for the transaction log, but it is not a normal table. Rather, it is built dynamically when queried by a user. No updates to `syslogshold` are allowed.

i Note

Because of this change in the datatypes for the Cluster Edition, you should archive and truncate audit tables before you upgrade. This reduces the likelihood of a failed upgrade because of insufficient space in the `sybsecurity` database.

Columns

The columns for `syslogshold` are:

Name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	Database ID.
<code>reserved</code>	<code>int</code>	Unused.
<code>spid</code>	<code>smallint</code> <code>int</code> for cluster environments	Server process ID of the user that owns the oldest active transaction (always 0 for Replication Server).
<code>page</code>	<code>unsigned int</code>	Starting page number of active portion in <code>syslogs</code> defined by oldest transaction (or the truncation page in <code>syslogs</code> for Replication Server).
<code>xactid</code>	<code>binary(6)</code>	ID of the oldest active transaction (always 0x000000 for Replication Server).
<code>masterxac tid</code>	<code>binary(6)</code>	ID of the transaction's master transaction (if any) for multidatabase transactions; otherwise 0x000000 (always 0x000000 for Replication Server).
<code>starttime</code>	<code>datetime</code>	Date and time the transaction started (or when the truncation point was set for Replication Server).
<code>name</code>	<code>char(67)</code>	Name of the oldest active transaction. It is the name defined with <code>begin transaction</code> , "\$user_transaction" if no value is specified with <code>begin transaction</code> , or "\$chained_transaction" for implicit transactions started by the ANSI chained mode. Internal transactions started by the SAP ASE server have names that begin with the dollar sign (\$) and are named for the operation, or are named "\$replication_truncation_point" for Replication Server.
<code>xloid</code>	<code>int null</code>	Lock ownership ID based on <code>spid</code> if the owner is a task, or on <code>xdes</code> if the owner is a transaction.

1.40 sysmessages

Applies to master database only. `sysmessages` contains one row for each system error or warning that can be returned by the SAP ASE server. The SAP ASE server displays the error description on the user's screen.

Columns

The columns for `sysmessages` are:

Name	Datatype	Description
<code>error</code>	<code>int</code>	Unique error number
<code>severity</code>	<code>smallint</code>	Severity level of error
<code>dlevel</code>	<code>smallint</code>	Reserved
<code>description</code>	<code>varchar(1024)</code>	Explanation of error with placeholders for parameters
<code>langid</code>	<code>smallint null</code>	Language; null for <code>us_english</code>
<code>sqlstate</code>	<code>varchar(5) null</code>	SQLSTATE value for the error

Indexes

- Clustered index on `error, dlevel`
- Nonclustered index on `error, dlevel, langid`

1.41 sysmonitors

Applies to master database only. `sysmonitors` contains one row for each monitor counter.

Columns

The columns for `sysmonitors` are:

Name	Datatype	Description
<code>field_name</code>	<code>char(79)</code>	Name of the counter
<code>group_name</code>	<code>char(25)</code>	Group to which this counter belongs
<code>field_id</code>	<code>smallint</code>	Unique identifier for the row
<code>value</code>	<code>int</code>	Current value of the counter
<code>description</code>	<code>varchar(255)</code> <code>null</code>	Description of the counter; not used
<code>nodeid</code>	<code>tinyint null</code>	Reserved for future use (not available for cluster environments)
<code>instanceid</code>	<code>tinyint</code>	ID of the instance (available only for cluster environments)

1.42 sysobjects

Applies to all databases. `sysobjects` contains one row for each table, view, stored procedure, extended stored procedure, log, rule, default, trigger, check constraint, referential constraint, computed column, function-based index key, encryption key, predicated privilege, and (in `tempdb` only) temporary object, and other forms of compiled objects. It also contains one row for each partition condition ID when object `type` is N.

For cross-database key references, `syscolumns.encrdate` matches `sysobjects.cdate`. `encrkeyid` in `sysencryptkeys` matches the `id` column in `sysobjects`.

Columns

The columns for `sysobjects` are:

Name	Datatype	Description
<code>name</code>	<code>varchar(255)</code> <code>not null</code>	Object name.
<code>id</code>	<code>int</code>	Object ID.
<code>uid</code>	<code>int</code>	User ID of object owner.
<code>type</code>	<code>char(2)</code>	One of the following object types: <ul style="list-style-type: none">• C – computed column• D – default• DD – decrypt default• EK – encryption key• F – SQLJ function• N – partition condition• P – Transact-SQL or SQLJ procedure• PP – the predicate of a privilege• PR – prepare objects (created by dynamic SQL)• R – rule• RI – referential constraint• RS – precomputed result set• S – system table• SF – scalar or user-defined functions• SQ – sequence object• TR – trigger• U – user table• V – view• XP – extended stored procedure
<code>userstat</code>	<code>smallint</code>	Application-dependent type information (32768 decimal [0x8000 hex] indicates to Data Workbench that a procedure is a report).
<code>sysstat</code>	<code>smallint</code>	Internal status information (256 decimal [0x100 hex] indicates that table is read-only)
<code>indexdel</code>	<code>smallint</code>	Recounts the changes in the schema of an object and updates <code>schemacnt</code> .
<code>schemacnt</code>	<code>smallint</code>	Count of changes in the schema of an object (incremented if a rule or default is added)
<code>sysstat2</code>	<code>int</code>	Additional internal status information

Name	Datatype	Description
sysstat3	unsigned smallint	Additional internal status information
crdate	datetime	Date the object was created
expdate	datetime	Reserved
deltrig	int	Stored procedure ID of a delete trigger if the entry is a table. Table ID if the entry is a trigger.
instrig	int	Stored procedure ID of a table's insert trigger if the entry is a table
updtrig	int	Stored procedure ID of a table's update trigger if the entry is a table
seltrig	int	Reserved
ckfirst	int	ID of first check constraint on the table
cache	smallint	Reserved
audflags	int null	Object's audit settings
objspare	smallint	Spare
versionts	binary(6) null	The version timestamp of the last schema change for this object (used by Replication Server)
loginame	varchar(30) null	Login name of the user who created the object
identburnmax	numeric(17) null	Maximum burned value for identity column if any in this object
<div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>i Note</p> <p>The <code>identburnmax</code> column is stored in an internal format. Use the <code>identity_burn_max()</code> function if you need a value.</p> </div>		
spacestate	smallint null	For internal use only.
erlchgts	binary(8) null	For internal use only.
lobcomp_lv 1	tinyint	LOB compression level.
status5	int	Additional internal status information.

The valid values for `t_type` are:

Object Type	Description
C	Computed column
D	Default
DD	Decrypt default
EK	Encryption key
F	SQLJ function
N	Partition condition
P	Transact-SQL or SQLJ procedure
PP	The predicate of a privilege
PR	Prepare objects (created by dynamic SQL)
R	Rule
RI	Referential constraint
RS	Precomputed result set
S	System table
SF	Scalar or user-defined functions
TR	Trigger
U	User table
V	View
XP	Extended stored procedure

The bit representations for the `sysstat` column are:

Decimal	Hex	Description
0	0x0	Any illegal object
1	0x1	System object
2	0x2	View
3	0x3	User object

Decimal	Hex	Description
4	0x4	Stored procedure
5	0x5	Predicated privilege
6	0x6	Default value spec
7	0x7	Domain rule
8	0x8	Trigger procedure
9	0x9	Referential integrity constraint
10	0xA	SQL Function
11	0xB	Extended type
12	0xC	Stored function
13	0xD	Computed column
14	0xE	Partition condition
15	0xF	Encryption key
16	0x10	Has clustered index
32	0x20	Has nonclustered index
64	0x40	If the object is a table, changes to the object are logged. If the object is a procedure, indicates that replication can subscribe to executions of the procedure.
128	0x80	The object is being created
256	0x100	The object contains suspect indexes and can only be used for read-only purposes until you have run <code>dbcc reindex</code> .
512	0x200	The object flagged by recovery as possibly damaged; run <code>dbcc</code> . Checked by <code>opentable</code> .
1024	0x400	The object is "fake"; that is, it resides in <code>tempdb</code> and is redefined for every query step that uses it
2048	0x800	The object is a definition time object created for query compilation.
4096	0x1000	Tags a system table that will have its index(es) re-created.
8192	0x2000	The object contains <code>text/image</code> fields
16384	0x4000	Unused
32768	0x8000	The table or procedure is replicated

The bit representations for the `sysstat2` column are:

Decimal	Hex	Status
0	0x00	Unchained transaction mode.
1	0x1	Table has a referential constraint.
2	0x2	Table has a foreign-key constraint.
4	0x4	Table has more than one check constraint.
8	0x8	Table has a primary-key constraint.
16	0x10	Stored procedure can execute only in chained transaction mode.
32	0x20	Stored procedure can execute in any transaction mode.
64	0x40	Table has an IDENTITY field.
128	0x80	Object is a virtually hashed table.
256	0x100	Allow implicit grant in execute immediate calls inside the stored procedure (dynamic ownership chain).
512	0x200	Table does not contain variable-length columns.
1024	0x400	Table is remote.
2048	0x800	Table is a proxy table created with the <code>existing</code> keyword.
4096	0x1000	Object should be replicated with owner name.
8192	0x2000	Table uses allpages-locking scheme.
16384	0x4000	Table uses datapages-locking scheme.
32768	0x8000	Table uses datarows-locking scheme.
65536	0x10000	Table was created in a version 11.9 or later server.
131072	0x20000	Table has a clustered index.
262144	0x40000	Object represents an Embedded SQL procedure.
524288	0x80000	Hybrid table.
16777216	0x1000000	An access rule.
33554432	0x2000000	Object represents a SQLJ stored procedure.
67108864	0x4000000	Object represents an OR access rule.
1073741824	0x40000000	Table contains one or more function-based indexes.

Decimal	Hex	Status
2147483648	0x80000000	Object has an extended index

The bit representations for the `sysstat3` column are:

Table 1: `sysstat3` Control Bits in the `sysobjects` Table

Decimal	Hex	Status
128	0x80	Indicates deferred table status. The table is deferred until SAP ASE allocates their pages.
256	0x0100	Stored procedure created with execute as owner clause.
512	0x0200	Stored procedure created with execute as caller clause.
2048	0x0800	Table contains LOB compressed data.
4096	0x1000	Table uses row-level compression.
8192	0x2000	Table uses page-level compression.
16384	0x4000	Table contains compressed data.
32768	0x8000	Table participates in incremental transfer.

The bit representations for the `sysstat4` column are:

Decimal	Hex	Status
1	0x1	Automatically compress indexes on this table.
2	0x2	Erase Residual Data is on for this table.
4	0x4	Erase Residual Data is off for this table.
8	0x8	Internal use.
16	0x10	Table has had a compressed index at some point in its life.
32	0x20	Unused.
64	0x40	DOL index root page access is optimized with CAS latching.

The bit representations for the `sysstat5` column are:

Decimal	Hex	Status
1	0x1	Multiversion concurrency control-enabled table. Supports snapshot isolation.

Decimal	Hex	Status
2	0x2	Datarow cache-enabled table. Supports row caching.
4	0x4	Multiversion concurrency control-enabled table, defined by <code>create</code> or <code>alter table</code> , but snapshot isolation is temporarily suspended.
8	0x8	Datarow cache-enabled table, defined by <code>create</code> or <code>alter table</code> , but row caching is temporarily suspended.
16	0x10	For internal use only.

Indexes

- Unique clustered index on `id`
- Nonclustered index on `name, uid`

1.43 sysoptions

Applies to all databases. `sysoptions` is a fake table queried by `sp_options`. When you are querying `sysoptions`, the names of the rows are case sensitive.

Columns

Name	Datatype	Attributes	Description
<code>spid</code>	<code>smallint</code>		Contains the process ID.
<code>name</code>	<code>varchar(100)</code>		Contains the name of the option.
<code>category</code>	<code>varchar(100)</code>		Contains the name of the category to which the option belongs.
<code>currentsetting</code>	<code>varchar(100)</code>	NULL	Contains the current setting of the option.
<code>defaultsetting</code>	<code>varchar(100)</code>	NULL	Contains the default setting of the option.

Name	Datatype	Attributes	Description
scope	int		Contains the bitmap used to capture information about options. The bits are ordered as follows: <ul style="list-style-type: none"> • Bit 1 – compiled time options • Bit 2 – stored procedure specific options • Bit 3 – binary options
number	int		The switch ID as an integer.

sysoptions shows:

- Trace flags set in the runserver file with the `-T` options
- Trace flags set with `dbcc traceon(<flag_number>)` or `set switch serverwide on`
- Trace flags and switches set by a specific system process ID (SPID) using `set switch on`

sysoptions displays only the switches that are visible to the user querying the sysoptions table. That is, the user cannot see switches set privately by other SPIDs with `set switch on`. However, traceflags enabled using the runserver file `-T` option, `dbcc traceon`, or `set switch serverwide on` are visible to all users.

Query sysoptions using `sp_options`. The datatype for the current and default value is `varchar` so settings with `varchar` values can be used directly. Settings with `integer` values can be used after typecasting.

You do not need special privileges to query sysoptions. For example:

```
select * from sysoptions
where spid = 13
go
```

You can also use string manipulation or typecasting. For example, if an option is numeric, you can query sysoptions by entering:

```
if (isnumeric(currentsetting))
    select@int_val = convert(int, currentsetting)
    ...
else
    select@char_val = currentsetting
    ...
```

1.44 syspartitionkeys

Applies to all databases. `syspartitionkeys` contains one row for each partition key for hash, range, and list partitioning of a table. All columns are not null.

Columns

The columns for `syspartitionkeys` are:

Name	Datatype	Description
<code>indid</code>	<code>smallint</code>	Type of index. Values are: <ul style="list-style-type: none">• 0 = table• 1 = clustered index• >1 = nonclustered index
<code>id</code>	<code>int</code>	Object ID of the partitioned table
<code>colid</code>	<code>smallint</code>	Column ID of the partition key of the partitioned table
<code>position</code>	<code>smallint</code>	Position of key among key positions

Indexes

Unique clustered index on `id`, `indid`, `colid`

1.45 syspartitions

Applies to all databases. `syspartitions` contains one row for each data partition and one row for each index partition.

For each database, `syspartitions` contains one row for:

- Each table partition. `indid` is 0.
- Each clustered index partition. `indid` is 1.
- Each nonclustered index partition. `indid` is >1.
- Each single-partitioned (unpartitioned) table.
- Each single-partitioned (unpartitioned) clustered or nonclustered index.

If an index is local, the value for `partitionid` (data partition row) and `data_partitionid` (associated index row) are the same. If the index is not local, the value for `data_partitionid` (index row) is zero (0), and it does not equal that for `partitionid` (data partition row).

i Note

The `syspartitions` table in versions of SAP ASE earlier than 15.0 has been renamed `syslices` and made obsolete. With SAP ASE version 15.0, `syspartitions` is completely redefined, and now supports data and index partitioning.

Columns

The columns for `syspartitions` are:

Name	Datatype	Description
<code>name</code>	<code>varchar (255)</code>	Partition name.
<code>indid</code>	<code>smallint</code>	on an allpages-locked table Index ID. Values are: <ul style="list-style-type: none"> • 0 – data pages (table) • 1 – clustered index on an allpages-locked table • >1 and <255 – nonclustered index or a clustered index on a data-only-locked table • 255 – text chain
<code>id</code>	<code>int</code>	Table ID.
<code>partitionid</code>	<code>int</code>	ID of data or index partition.
<code>segment</code>	<code>smallint</code>	ID of segment on which partition resides.
<code>status</code>	<code>int</code>	Internal status information.
<code>datoampage</code>	<code>unsigned int</code>	Page number for the object allocation map of a data partition.
<code>indoampage</code>	<code>unsigned int</code>	Page number of the object allocation map of an index partition.
<code>firstpage</code>	<code>unsigned int</code>	Page number of the first data or leaf page.
<code>rootpage</code>	<code>unsigned int</code>	Page number of: <ul style="list-style-type: none"> • Root page if entry is an index partition • Last page if entry is a data partition

Name	Datatype	Description
data_partitionid	int	ID of data partition this index spans. Values are: <ul style="list-style-type: none"> • 0 – for global indexes spanning the entire table • Non-zero – partition ID of the data partition that a local index's partition spans.
crdate	datetime	Date the partition created.
cdataptname	varchar(255) null	Name of data partition.
lobcomp_lvl	tinyint	LOB compression level
ptndcompver	tinyint	Version of datacompression algorithm used

Indexes

- Unique clustered index on `id, indid, partitionid`
- Unique nonclustered index on `id, indid, name`
- Unique nonclustered index on `partitionid, indid`

1.46 syspoolinfo

Applies to master database. Provides information about data caches and pools.

Access to the views is restricted to users with the `sa_role` role.

Columns

The columns for `syspoolinfo` are:

Name	Datatype	Description
cache_name	varchar(30)	Name of the cache in which this pool is allocated.
io_size	varchar(30)	The size of the buffers, in kilobytes, used to perform I/O for this pool.

Name	Datatype	Description
config_size	float	Configured amount of memory, in megabytes, allocated to the pool. May be different from the amount reported in the run_size column.
run_size	float	The current amount of memory, in megabytes, allocated to the pool.
apf_percent	int	The percentage of buffers in the pool that can be used to hold buffers that have been read into cache by asynchronous prefetch.
wash_size	varchar(10)	The size of the wash area, in megabytes, in the pool.
cacheid	int	ID of the data cache.
instanceid	int	ID of the instance (zero for non-Cluster Edition servers).
scope	varchar(6)	Indicates whether the data cache is local or global for Cluster Edition. The value is always Global for nonclustered servers.

1.47 sysprocedures

Applies to all databases. `sysprocedures` contains entries for each view, default, rule, trigger, procedure, declarative default, partition condition, check constraint, computed column, function-based index key, and other forms of compiled objects.

The sequence tree for each object, including computed columns or function-based index definition, is stored in binary form. If the sequence tree does not fit into one entry, it is broken into more than one row. The `sequence` column identifies the sub-rows.

Columns

The columns for `sysprocedures` are:

Name	Datatype	Description
type	smallint	Object type
qp_setting	varbinary(6) null	For future use only
id	int	Object ID

Name	Datatype	Description
sequence	int	Sequence number if more than one row is used to describe this object
status	smallint	Internal system status
number	smallint	Sub-procedure number when the procedure is grouped (0 for nonprocedure entries)
version	int null	The version of SAP ASE that created the sequence tree stored in this catalog for a given object

The bit representations for the `type` column are:

Decimal	Hex	Status
1	0x1	Entry describes a plan (reserved)
2	0x2	Entry describes a tree

Indexes

Unique clustered index on `id`, `number`, `type`, `sequence`

1.48 sysprocesses

Applies to `master` database only. `sysprocesses` contains information about SAP ASE processes, but it is not a normal table. It is built dynamically when queried by a user. No updates to `sysprocesses` are allowed. Use the `kill` statement to kill a process.

Columns

The columns for `sysprocesses` are:

Name	Datatype	Description
spid	smallint	Process ID.
	int for the Cluster Edition	
kpid	int	Kernel process ID.

Name	Datatype	Description
enginenum	int	Number of engine on which process is being executed.
status	char(12)	Process ID status.
suid	int	Server user ID of user who issued command.
hostname	varchar(30) null	Name of host computer.
program_name	varchar(30) null	Name of front-end module.
hostprocess	varchar(30) null	Host process ID number..
cmd	varchar(30) null	Command or process currently being executed. Evaluation of a conditional statement, such as an if or while loop, returns cond.
cpu	int	Cumulative CPU time for process in ticks
physical_io	int	Number of disk reads and writes for current command.
memusage	int	Amount of memory allocated to process.
blocked	smallint int for the Cluster Edition	Process ID of blocking process, if any.
dbid	smallint	Database ID.
uid	int	ID of user who executed command.
gid	int	Group ID of user who executed command.
tran_name	varchar(64) null	Name of the active transaction.
time_blocked	int null	Time blocked in seconds.
network_pktsz	int null	Current connection's network packet size.
fid	smallint int for the Cluster Edition	Process ID of the worker process' parent.

Name	Datatype	Description
<code>execlass</code>	<code>varchar(30) null</code>	Execution class that the process is bound to.
<code>priority</code>	<code>varchar(10) null</code>	Base priority associated with the process.
<code>affinity</code>	<code>varchar(30) null</code>	Name of the engine to which the process has affinity.
<code>id</code>	<code>int null</code>	Object ID of the currently running procedure (or 0 if no procedure is running).
<code>stmtnum</code>	<code>int null</code>	The current statement number within the running procedure (or the SQL batch statement number if no procedure is running).
<code>linenum</code>	<code>int null</code>	The line number of the current statement within the running stored procedure (or the line number of the current SQL batch statement if no procedure is running).
<code>origsuid</code>	<code>int null</code>	Original server user ID. If this value is not NULL, a user with an <code>suid</code> of <code>origsuid</code> executed <code>set proxy</code> or <code>set session authorization</code> to impersonate the user who executed the command.
<code>block_xloid</code>	<code>int null</code>	Unique lock owner ID of a lock that is blocking a transaction.
<code>clientname</code>	<code>varchar(30) null</code>	(Optional) Name by which the user is known for the current session.

i Note

The SAP ASE server automatically stores one or more spaces in `clientname`, `clienthostname`, and `clientapplname` columns. For this reason, a query using any of these three columns that includes "is null" does not return an expected result set.

Name	Datatype	Description
clienthostname	varchar(30) null	(Optional) Name by which the host is known for the current session.
clientapplname	varchar(30) null	(Optional) Name by which the application is known for the current session.
sys_id	smallint null	Unique identity of companion node.
ses_id	int null	Unique identity of each client session.
loggedindatetime	datetime null	Shows the time and date when the client connected to the SAP ASE server. See "Row-level access control" in Chapter 11, "Managing User Permissions" of the <i>Security Administration Guide</i> for more information.
ipaddr	varchar(64) null	IP address of the client where the login is made. See "Row-level access control" in Chapter 11, "Managing User Permissions" of the <i>Security Administration Guide</i> for more information.
nodeid	tinyint null	Reserved for future use (not available for cluster environments).
instanceid	tinyint	ID of the instance (available only for cluster environments).
pad	smallint	(Cluster Edition) Column added for alignment purposes.
spare2	int	Reserved for future use
lcid	int	(Cluster Edition) ID of the cluster.
execution_time	int	Execution time (including compilation time) is the time that the process has been running and the precision is milliseconds

i Note

Because of this change in the datatypes for the Cluster Edition, you should archive and truncate audit tables before you upgrade. This reduces the likelihood of a failed upgrade because of insufficient space in the `sybsecurity` database.

The values for the `status` column are:

Status	Meaning
alarm sleep	Waiting for alarm to wake process up (user executed a <code>waitfor delay</code> command)
background	A process, such as a threshold procedure, run by the SAP ASE server rather than by a user process
infected	Server has detected a serious error condition; extremely rare
latch sleep	Waiting on a latch acquisition
lock sleep	Waiting on a lock acquisition
PLC sleep	Waiting to access a user log cache
recv sleep	Waiting on a network read
remote i/o	Performing I/O with a remote server
runnable	In the queue of runnable processes
running	Actively running on one of the server engines
send sleep	Waiting on a network send
sleeping	Waiting on a disk I/O, or some other resource (often indicates a process that is running, but doing extensive disk I/O)
stopped	Stopped process
sync sleep	Waiting on a synchronization message from another process in the family

1.49 sysprotects

Applies to all databases. `sysprotects` contains information on permissions that have been granted to, or revoked from, users, groups, and roles.

Columns

The columns for `sysprotects` are:

Name	Datatype	Description
<code>id</code>	<code>int</code>	ID of the object to which this permission applies. Has an ID of 0 when the permission granted is <code>create table</code> , <code>create default</code> , and so on.
<code>uid</code>	<code>int</code>	ID of the user, group, or role to which this permission applies.
<code>action</code>	<code>smallint</code>	See the following list for permissions.
<code>protectt ype</code>	<code>tinyint</code>	One of the following values: <ul style="list-style-type: none">• 0=grant with grant• 1=grant• 2=revoke
<code>columns</code>	<code>varbinary(1 33)</code>	Bitmap of columns to which this <code>select</code> , <code>update</code> , <code>decrypt</code> , or references permission applies. <code>columns</code> is also a bitmap of permitted roles for <code>set session authorization</code> .
<code>grantor</code>	<code>int</code>	User ID of the grantor. If the grantor is a system administrator, the user ID of the object owner is used.
<code>predid</code>	<code>int</code>	Object ID of predicated privilege
<code>status</code>	<code>smallint</code>	0x0001 – indicates that the privilege (or denial) is predicated

Use this query to look up names for action values in the `spt_values` table:

```
select number, type, name from master..spt_values
where type in ("T", "GP")
```

The `action` column values are:

- 1=alter any object owner *
- 2=alter any table *
- 3=change password *
- 99=manage any statistics *
- 100=manage any user *
- 101=manage auditing *

- 4=checkpoint any database*
- 5=select builtin
- 6=checkpoint*
- 7=create any default*
- 8=create any function*
- 9=create any index*
- 10=create any object*
- 11=create any procedure*
- 12=create any rule*
- 13=create any table*
- 14=create any trigger*
- 15=create any view *
- 16=allow exceptional login*
- 17=dbcc checkalloc
- 18=dbcc checkalloc any database
- 19=map external file*
- 20=manage dump configuration*
- 21=dbcc checkcatalog
- 22=dbcc checkcatalog any database
- 24=Manage HADR
- 25=dbcc checkdb
- 26=dbcc checkdb any database
- 29=dbcc checkindex
- 30=dbcc checkindex any database
- 33=dbcc checkstorage
- 34=dbcc checkstorage any database
- 37=dbcc checktable
- 38=dbcc checktable any database
- 41=dbcc checkverify
- 42=dbcc checkverify any database
- 45=dbcc fix_text
- 46=dbcc fix_text any database
- 49=dbcc indexalloc
- 50=dbcc indexalloc any database
- 53=dbcc reindex
- 54=dbcc reindex any database
- 57=dbcc tablealloc
- 58=dbcc tablealloc any database
- 61=dbcc textalloc
- 62=dbcc textalloc any database
- 102=manage checkstorage*
- 103=manage cluster*
- 104=manage data cache*
- 105=manage database*
- 106=manage database permissions*
- 107=manage disk*
- 108=manage lock promotion threshold*
- 109=manage master key*
- 110=manage replication*
- 111=manage resource limit*
- 112=manage roles*
- 113=manage security configuration*
- 114=manage security permissions*
- 115=manage server*
- 116=manage server configuration*
- 117=manage server permissions*
- 118=monitor qp performance*
- 119=monitor server replication*
- 120=mount any database*
- 121=online any database*
- 122=online database*
- 123=own any database*
- 125=own database *
- 126=quiesce any database*
- 129=references any table *
- 130=report checkstorage*
- 131=reorg any table *
- 132=select any audit table*
- 133=select any system catalog*
- 134=select any table *
- 135=set tracing any process *
- 136=setuser
- 137=shutdown *
- 138=transfer any table*
- 139=manage any thread pool*
- 140=truncate any table*
- 141=unmount any database*
- 144=update any security catalog*
- 145=update any table*
- 146=use any database*

- 65=dbcc tune
- 66=delete any table*
- 67=drop any default*
- 68=drop any function*
- 70=drop any object*
- 71=drop any procedure*
- 72=drop any rule*
- 73=drop any table*
- 74=drop any trigger*
- 75=drop any view*
- 76=dump database*
- 77=dump any database*
- 79=execute any function*
- 80=Execute Any Procedure*
- 81=Identity_insert Any Table*
- 82=Identity_update Any Table*
- 83=Identity_insert
- 84=Identity_update
- 85=insert any table*
- 86=kill*
- 87=kill any process*
- 88=load database*
- 89=load any database*
- 90=manage service key*
- 91=manage abstract plans*
- 92=manage any encryption key*
- 93=manage any esp*
- 94=manage any execution class*
- 95=manage any login*
- 96=manage any login profile*
- 97=manage any object permission*
- 98=manage any remote login*
- 148=use database*
- 149=set switch*
- 150=show switch*
- 151=references
- 152=truncate any audit table*
- 153=decrypt any table*
- 155=manage column encryption key*
- 156=manage any database*
- 167=set proxy
- 193=select
- 195=insert
- 196=delete
- 197=update
- 198=create table
- 203=create database
- 207=create view
- 221=create trigger
- 222=create procedure
- 224=execute
- 233=create default
- 235=dump transaction
- 236=create rule
- 253=connect
- 280=create function
- 282=delete statistics
- 320=truncate table
- 326=update statistics
- 347=set tracing
- 353=decrypt
- 354=create encryption key
- 368=transfer table

The description of bits in column are:

Bit	Decimal Value	Description
0	1	Permission on all columns
1	2	Permission on column 1
2	4	Permission on column 2

Bit	Decimal Value	Description
[...]		
<n>	2 ⁿ	Permission on column <n>

Values that are not an exact power of 2 indicate a combination of columns.

i Note

Permissions for the `action` column marked with an asterisk (*) take effect only when granular permissions is enabled.

Indexes

Unique clustered index on `id`, `action`, `grantor`, `uid`, `protecttype`, `predid`

1.50 sysquerymetrics

Applies to all databases. Presents aggregated historical query processing metrics for individual queries from persistent data. In addition to monitoring tables, use performance metrics information from this catalog.

Columns

The columns for `sysquerymetrics` are:

Name	Datatype	Description
<code>uid</code>	<code>int</code>	User ID
<code>gid</code>	<code>int</code>	Group ID
<code>hashkey</code>	<code>int</code>	Hashkey over the SQL query text
<code>id</code>	<code>int</code>	Unique ID
<code>sequence</code>	<code>smallint null</code>	Sequence number for a row when multiple rows are required for the text of the SQL
<code>exec_min</code>	<code>unsigned bigint</code>	Minimum execution time

Name	Datatype	Description
exec_max	unsigned bigint	Maximum execution time
exec_avg	unsigned bigint	Average execution time
elap_min	unsigned bigint	Minimum elapsed time
elap_max	unsigned bigint	Maximum elapsed time
elap_avg	unsigned bigint	Average elapsed time
lio_min	unsigned bigint	Minimum logical IO
lio_max	unsigned bigint	Maximum logical IO
lio_avg	unsigned bigint	Average logical IO
pio_min	unsigned bigint	Minimum physical IO
pio_max	unsigned bigint	Maximum physical IO
pio_avg	unsigned bigint	Average physical IO
cnt	unsigned bigint	Number of times the query has been executed.
abort_cnt	int null	Number of times a query is aborted by the Resource Governor when a resource limit is exceeded
qtext	varchar(255) null	Query text

The number of metrics shared among user IDs increased for SAP ASE release 15.0.2 and later, reducing the number of entries in `sysquerymetrics` (a view of `sysqueryplans`), and automatically aggregates the metrics for identical queries across different user IDs.

The user ID (`uid`) of `sysquerymetrics` is 0 when all table names in a query that are not qualified by user name are owned by the DBO.

For example, if table `t1` is owned only by the DBO and shared by different users:

```
select * from t1 where c1 = 1
```

The SAP ASE server uses 0 as the `uid` for the `sysquerymetrics` entry for all users executing this query who do not have a private table named `t1`.

In this example, if table `t2` is owned and qualified by "user1," the SAP ASE server also uses an UID of 0:

```
select * from user1.t2 where c1 = 1
```

However, if table `t3` is owned only by “user1,” but is unqualified and not owned by the DBO, the UID of “user1” is used in the `sysquerymetrics` entry:

```
select * from t3 where c1 = 1
```

1.51 sysqueryplans

Applies to all databases. `sysqueryplans` contains two or more rows for each abstract query plan. Uses datarow locking.

Columns

The columns for `sysqueryplans` are:

Name	Datatype	Description
uid	int	User ID of user who captured the abstract plan.
dbid	int null	For future use only
qupdate	datetime null	For future use only
sprocid	int null	For future use only
hashkey2	int null	For future use only
key1	int null	For future use only
key2	int null	For future use only
key3	int null	For future use only
key4	int null	For future use only
gid	int	The abstract plan group ID under which the abstract plan was saved.
hashkey	int	The hash key over the SQL query text.
id	int	The unique ID of the abstract plan.
type	smallint	10 if the text column contains query text, or 100 if the text column contains abstract plan text.

Name	Datatype	Description
sequence	smallint	Sequence number if multiple rows are required for the text of the SQL query or abstract plan.
status	int null	Reserved.
text	varchar (255) null	The SQL text, if <code>type</code> is 10, or the abstract query plan text, if the <code>type</code> is 100.

Indexes

- Unique clustered index on `uid, gid, hashkey, id, type, sequence`
- Nonclustered index on `id, type, sequence`

1.52 sysreferences

Applies to all databases. `sysreferences` contains one row for each referential integrity constraint declared on a table or column.

Columns

The columns for `sysreferences` are:

Name	Datatype	Description
indexid	smallint	ID of the unique index on referenced columns
constrid	int	Object ID of the constraint from <code>sysobjects</code>
tableid	int	Object ID of the referencing table
reftabid	int	Object ID of the referenced table
keycnt	smallint	Number of columns in the foreign key
status	smallint	Options and indicators
frgndbid	smallint null	Database ID of the database that includes the referencing table.

Name	Datatype	Description
pmrydbid	smallint	Database ID of the database that includes the referenced table (the table with the primary key)
spare2	int	Reserved
fokey1 ... fokey16	smallint	Column ID of the first to the 16th referencing column
refkey1 ... refkey16	smallint	Column ID of the first to the 16th referenced column
frgndbname	varchar(30) null	Name of the database that includes the referencing table (the table with the foreign key); NULL if the referencing table is in the current database
pmrydbname	varchar(30) null	Name of the database that includes the referenced table (the table with the primary key); NULL if the referenced table is in the current database

The status bit in `sysreferences` is:

Decimal	Hex	Status
2	0x2	The referential constraint has a <code>match full</code> option

Indexes

- Clustered index on `tableid, frgndbname`
- Nonclustered index on `constrid, frgndbname`
- Nonclustered index on `reftabid, indexid, pmrydbname`

1.53 sysremotelogins

Applies to master database only. `sysremotelogins` contains one row for each remote user that is allowed to execute remote procedure calls on this SAP ASE server.

Columns

The columns for `sysremotelogins` are:

Name	Datatype	Description
<code>remoteserverid</code>	<code>smallint</code>	Identifies the remote server
<code>remoteusername</code>	<code>varchar(30) null</code>	User's login name on remote server
<code>suid</code>	<code>int</code>	Local server user ID
<code>status</code>	<code>smallint</code>	Bitmap of options

Indexes

Unique clustered index on `remoteserverid`, `remoteusername`

1.54 sysresourcelimits

Applies to master database only. `sysresourcelimits` contains a row for each resource limit defined by the SAP ASE server. Resource limits specify the maximum amount of server resources that can be used by an SAP ASE login or an application to execute a query, query batch, or transaction.

Columns

The columns for `sysresourcelimits` are:

Name	Datatype	Description
name	varchar(30) null	Login name
appname	varchar(30) null	Application name
rangeid	smallint	id column from <code>systimeranges</code>
limitid	smallint	id column from <code>spt_limit_types</code>
enforced	tinyint	Subset of the <code>enforced</code> column from <code>spt_limit_types</code> : <ul style="list-style-type: none">• 1 = prior to execution• 2 = during execution• 3 = both
action	tinyint	Action to take on a violation: <ul style="list-style-type: none">• 1 = issue warning• 2 = abort query batch• 3 = abort transaction• 4 = kill session
limitvalue	int	Value of limit
scope	tinyint	Scope of user limit (a bitmap indicating one or more of the following): <ul style="list-style-type: none">• 1 = query• 2 = query batch• 4 = transaction
spare	tinyint	Reserved

Indexes

Clustered index on `name`, `appname`

1.55 sysroles

Applies to all databases. `sysroles` maps server role IDs to local role IDs.

When a database permission is granted to a role, if an entry for the role does not exist in `sysroles`, the SAP ASE server adds an entry to `sysroles` to map the local role ID (`lrid`) to the server-wide role ID (`srid`) in `sysserverroles`.

Columns

The columns for `sysroles` are:

Name	Datatype	Description
<code>id</code>	<code>int</code>	Server role ID (<code>srid</code>)
<code>lrid</code>	<code>int</code>	Local role ID
<code>type</code>	<code>smallint</code>	Unused
<code>status</code>	<code>int</code>	Unused

Indexes

Unique clustered index on `lrid`

1.56 syssecmechs

Applies to master database only. `syssecmechs` contains information about the security services supported by each security mechanism that is available to the SAP ASE server. `syssecmechs` is not created during installation, rather, it is built dynamically when queried by a user.

Columns

The columns for `syssecmechs` are:

Name	Datatype	Description
<code>sec_mech_name</code>	<code>varchar(30)</code>	Name of the security mechanism; for example, "NT LANMANAGER"
<code>available_service</code>	<code>varchar(30)</code>	Name of the security service supported by the security mechanism; for example, "unified login"

1.57 syssegments

Applies to all databases. `syssegments` contains one row for each segment (named collection of disk pieces). In a newly created database, the entries are: segment 0 (`system`) for system tables; segment 2 (`logsegment`) for the transaction log; and segment 1 (`default`) for other objects. Segment 3 includes information about `imrslogsegment`.

Columns

The columns for `syssegments` are:

Name	Datatype	Description
<code>segment</code>	<code>smallint</code>	Segment number
<code>name</code>	<code>sysname</code>	Segment name
<code>status</code>	<code>smallint null</code>	Indicates which segment is the default segment

1.58 sys.servers

Applies to master database only. `sys.servers` contains one row for each remote SAP ASE server, Backup Server, or Open Server on which this SAP ASE server can execute remote procedure calls.

Columns

The columns for `sys.servers` are:

Name	Datatype	Description
<code>srvid</code>	<code>smallint</code>	ID number (for local use only) of the remote server.
<code>srvstatus</code>	<code>smallint</code>	Bitmap of options.
<code>srvstatus</code> 2	<code>unsigned int</code>	Bitmap of options.
<code>srvstat2</code>	<code>unsigned int</code>	Bitmap of server options.
<code>srvname</code>	<code>varchar(30)</code>	Server name.
<code>srvnetnam</code> e	<code>varchar (255)</code>	Interfaces file name for the server.
<code>srvclass</code>	<code>smallint null</code>	Server category defined by the class parameter of <code>sp_addserver</code> .
<code>srvsecmeh</code>	<code>varchar(30)</code> <code>null</code>	Security mechanism.
<code>svrcost</code>	<code>smallint null</code>	Provides the network cost in milliseconds for accessing a server over a network. Used only by the SAP ASE query optimizer for evaluating the cost of a query when accessing a proxy table, the default is set to 1,000 ms.
<code>srvprincipal</code>	<code>varchar(255)</code> <code>null</code>	Specifies the Kerberos principal name for the server. Default value is NULL.

The bit representations for the `srvstatus` column are:

Decimal	Hex	Status
0	0x0	Timeouts are enabled
1	0x1	Timeouts are disabled
2	0x2	Network password encryption is enabled

Decimal	Hex	Status
4	0x4	Remote server is read-only
64	0x40	Use message confidentiality
128	0x80	Use message integrity
256	0x100	Mutual authentication

The bit representations for the `srvstatus2` column are:

Decimal	Hex	Status
1	0x01	Supports fully qualified table names
2	0x02	Reserved for future use

The server categories for the `srvclass` column are:

<code>srvclass</code>	Server category
0	Local server
1	sql_server class server
3	direct_connect class server
4	DB2 class server
6	sds class server
7	SAP ASE class server
8	Adaptive Server Anywhere class server
9	ASIQ class server

Indexes

- Unique clustered index on `srvid`
- Nonclustered index on `srvname`

1.59 syssequences

(SQLScript only) Stores the parameter settings for a sequence.

Columns

The columns for `syssequences` are:

Name	Datatype	Description
<code>id</code>	<code>int</code>	Object ID of the sequence
<code>startwith</code>	<code>bigint</code>	Initial value in the sequence
<code>minval</code>	<code>bigint</code>	Minimum value to which the sequence can be set
<code>maxval</code>	<code>bigint</code>	Maximum value to which the sequence can be set
<code>increment</code>	<code>bigint</code>	Value by which the sequence is incremented
<code>cache</code>	<code>smallint</code>	The cache size setting if the cache flag is set
<code>status</code>	<code>int</code>	Internal system status information

1.60 syssessions

Applies to master database only. `syssessions` is used only when SAP ASE is configured for failover in a high availability system. `syssessions` contains one row for each client that connects to the SAP ASE server with the failover property. Clients that have an entry in `syssessions` during failover are moved to the secondary companion. Clients that do not have an entry in `syssessions` are dropped during failover. Clients that have an entry in `syssessions` during failback are moved to the primary companion. Clients that do not have an entry in `syssessions` during failback are dropped.

Columns

The columns for `syssessions` are:

Name	Datatype	Description
<code>sys_id</code>	<code>smallint</code>	Unique identity of companion node
<code>ses_id</code>	<code>int</code>	Unique identity of each client session
<code>state</code>	<code>tinyint</code>	Describes whether the session is active or inactive
<code>spare</code>	<code>tinyint</code>	Reserved for future use
<code>status</code>	<code>smallint</code>	Reserved for future use
<code>dbid</code>	<code>smallint</code>	Reserved for future use
<code>name</code>	<code>varchar(30)</code> <code>null</code>	Same as client's login name as specified in <code>syslogins</code>
<code>nodeid</code>	<code>tinyint</code> <code>null</code>	Reserved for future use (not available for cluster environments)
<code>instanceid</code>	<code>tinyint</code>	ID of the instance (available only for cluster environments)
<code>ses_data</code>	<code>image null</code>	Reserved for future use

1.61 `syslices`

Applies to all databases. `syslices` contains one row for each slice (page chain) of a sliced table. `syslices` is used only during the SAP ASE upgrade process. After the upgrade is complete, all the data is removed.

i Note

In versions of SAP ASE earlier than 15.0 `syspartitions` stored partition-related *information*. This has been renamed to `syslices` for SAP ASE 15.0, and later; `syspartitions` now refers to the catalog that tracks all partition-related *data* in the SAP ASE server.

Columns

The columns for `syslices` are:

Name	Datatype	Description
<code>state</code>	<code>smallint</code>	Internal information about the state of the partition
<code>id</code>	<code>int</code>	Object ID of the partitioned table
<code>partitionid</code>	<code>int</code>	Partition ID number
<code>firstpage</code>	<code>int</code>	Page number of the partition's first page
<code>controlpage</code>	<code>int</code>	Page number of the partition's control page
<code>spare</code>	<code>binary(32)</code>	Reserved

Indexes

Unique clustered index on `id`, `partitionid`

1.62 sysrvroles

Applies to master database only. `sysrvroles` contains a row for each system or user-defined role.

Columns

The columns for `sysrvroles` are:

Name	Datatype	Description
<code>srid</code>	<code>int</code>	Server role ID
<code>name</code>	<code>varchar(30)</code>	Name of the role
<code>password</code>	<code>varbinary(128)</code> <code>null</code>	Password for the role (encrypted) and readable only by a user with <code>sso_role</code>
<code>pwdate</code>	<code>datetime</code> <code>null</code>	Date the password was last changed

Name	Datatype	Description
status	smallint null	Bitmap for role status.
logincount	smallint null	Number of failed login attempts; reset to 0 by a successful login
locksuid	int null	The user who locked the role.
lockreason	int null	The reason why a role was locked.
lockdate	datetime null	The date and time a role was locked.

The bit representations for the `status` column are:

Decimal	Hex	Status
2	0x2	Role is locked
4	0x4	Role is expired
8	0x8	Role has exclusive activation role with another role
16	0x10	Role is granted child roles

Indexes

Unique clustered index on `srid`

1.63 sysstatistics

Applies to all databases. `sysstatistics` contains one or more rows for each indexed column on a user table and for each partition. May also contain rows for unindexed column.

Columns

The columns for `sysstatistics` are:

Name	Datatype	Description
statid	smallint	Reserved

Name	Datatype	Description
id	int	Object ID of table
sequence	int	Sequence number if multiple rows are required for this set of statistics
moddate	datetime	Date this row was last modified
formatid	tinyint	Type of statistics represented by this row
usedcount	tinyint	Number of fields c0 to c79 used in this row
colidarray	varbinary(100)	An ordered list of column IDs
c0...c79	varbinary(255)	Statistical data
indid	smallint	Index ID of partition
ststatus	smallint	Status bits for this statistics row; possible values vary according to the type of row.
partitionid	int	Partition ID
spare2	int	For future use
spare3	int	For future use

Indexes

Unique clustered index `csystatistics` on `id`, `indid`, `partitionid`, `statid`, `colidarray`, `formatid`, `sequence`

1.64 systabstats

Applies to all databases. `systabstats` contains one row for each clustered index, one row for each nonclustered index, one row for each table that has no clustered index, and one row for each partition.

Columns

The columns for `systabstats` are:

Name	Datatype	Description
<code>indid</code>	<code>smallint</code>	<ul style="list-style-type: none">• 0 = if a table• 1 = if a clustered index on an allpages-locked table• >1 = if a nonclustered index or a clustered index on a data-only-locked table <p><code>systabstats</code> does not maintain statistics on text or image objects (255)</p>
<code>id</code>	<code>int</code>	ID of table to which index belongs
<code>activesta</code> <code>tid</code>	<code>smallint</code>	Reserved
<code>indexheig</code> <code>ht</code>	<code>smallint</code>	Height of the index; maintained if <code>indid</code> is greater than 1
<code>leafcnt</code>	<code>unsigned</code> <code>int</code>	Number of leaf pages in the index; maintained if <code>indid</code> is greater than 1
<code>pagecnt</code>	<code>unsigned</code> <code>int</code>	Number of pages in the table or index
<code>rowcnt</code>	<code>float</code>	Number of rows in the table; maintained for <code>indid</code> of 0 or 1
<code>forrowcn</code> <code>t</code>	<code>float</code>	Number of forwarded rows; maintained for of 0 or 1
<code>delrowcnt</code>	<code>float</code>	Number of deleted rows
<code>dpagecrn</code> <code>t</code>	<code>float</code>	Number of extent I/Os that need to be performed to read the entire table
<code>ipagecrn</code> <code>t</code>	<code>float</code>	Number of extent I/Os that need to be performed to read the entire leaf level of a nonclustered index
<code>drowrcnt</code>	<code>float</code>	Number of page I/Os that need to be performed to read an entire table

Name	Datatype	Description
oamapgcnt	int	Number of OAM pages for the table, plus the number of allocation pages that store information about the table
extent0pgcnt	int	Count of pages that are on the same extent as the allocation page
datarowsize	float	Average size of the data row
leafrowsize	float	Average size of a leaf row for nonclustered indexes and clustered indexes data-only-locked tables
status	int	Internal system status information.
plljoindegree	int	The degree of parallelism used for a nested loop join operation, <code>plljoindegree</code> is the parallel scan degree of the table (whose <code>sysabstats</code> has this field) that is the inner table in a nested loop join.
rslastoam	int	Last OAM page visited by a <code>reorg reclaim_space</code> or <code>reorg compact</code> command
rslastpage	int	Last data or leaf page visited by a <code>reorg reclaim_space</code> or <code>reorg compact</code> command
frlastoam	int	Last OAM page visited by the <code>reorg forwarded_rows</code> command
frlastpage	int	Last data page visited by the <code>reorg forwarded_rows</code> command
conopt_threshold	smallint	Concurrency optimization threshold
plldegree	smallint	Maximum degree of parallelism possible on table or index for data manipulation languages (DMLs). A value of 0 (zero) indicates a nonexistent maximum; the query processor configures maximum degree of parallelism.
emptypgcnt	unsigned int	Number of empty pages in extents allocated to the table or index
partitionid	int	Partition ID
warmcachepgcnt	unsigned int	
statmoddate	datetime	Last time the row was flushed to disk

Name	Datatype	Description
unusedpgcnt	unsigned int	Number of unused pages
oampagecnt	unsigned int	Number of allocation pages listed in the object allocation map
pioclmdata	real	
pioclmindex	real	
piocsmdata	real	
piocsmindex	real	
spare2	float	Reserved Reserved
spare4	float	Reserved Reserved
spare5	int	Spare field for alignment

The status bit for `systabstats` is:

Decimal	Hex	Status
1	0x1	Statistics are the result of upgrade (not update statistics)

Indexes

Unique clustered index on `id, indid, partitionid`

1.65 systhresholds

Applies to all databases. `systhresholds` contains one row for each threshold defined for the database.

Columns

The columns for `systhresholds` are:

Name	Datatype	Description
<code>segment</code>	<code>smallint</code>	Segment number for which free space is being monitored.
<code>free_space</code>	<code>unsigned int</code>	Size of threshold, in logical pages.
<code>status</code>	<code>smallint</code>	Bit 1 equals 1 for the log segment's last-chance threshold, 0 for all other thresholds.
<code>proc_name</code>	<code>varchar(255)</code>	Name of the procedure that is executed when the number of unused pages on <code>segment</code> falls below <code>free_space</code> .
<code>suid</code>	<code>int null</code>	The server user ID of the user who added the threshold or modified it most recently.
<code>currauth</code>	<code>varbinary(255) null</code>	A bitmask that indicates which roles were active for <code>suid</code> at the time the threshold was added or most recently modified. When the threshold is crossed, <code>proc_name</code> executes with this set of roles, less any that have been deactivated since the threshold was added or last modified.

The possible bitmasks you might see, individually or in combination, in the `currauth` column.

Decimal	Hex	Description
1	0x1	<code>sa_role</code>
2	0x2	<code>sso_role</code>
4	0x4	<code>oper_role</code>
8	0x8	<code>sybase_ts_role</code>
16	0x10	<code>sybase_ts_role</code>
32	0x20	<code>navigator_ole</code>
128	0x80	<code>replication_role</code>
256	0x100	<code>dtm_tm_role</code>

Decimal	Hex	Description
1024	0x400	ha_role
2048	0x800	mon_role
4096	0x1000	js_admin_role
16384	0x4000	messaging_role
32768	0x8000	web_services

To find out what role ID is associated with the bitmask output in `currauth` in your SAP ASE server, perform the following `select` statement:

```

1> select (c.number - 1) as role_id,role_name(c.number - 1) as role_name
2> from systhresholds ,master.dbo.spt_values c
3> where convert(tinyint,substring(isnull(currauth,0x1), c.low,1)) &
4> c.high != 0
5> and c.type = "P"
6> and c.number <= 1024
7> and c.number >0
8> and role_name(c.number - 1) is not null
9> go

```

The SAP ASE server returns something similar to the following:

```

role_id   role_name
-----
0    sa_role
1    sso_role
2    oper_role
3    sybase_ts_role
4    navigator_role
7    dtm_tm_role
10   mon_role
11   js_admin_role
12   messaging_role
13   js_client_role

```


1.66 systimeranges

Applies to master database only. `systimeranges` stores named time ranges, which are used by the SAP ASE server to control when a resource limit is active.

Columns

The columns for `systimeranges` are:

Name	Datatype	Description
name	<code>varchar(255)</code>	Unique name of the time range.
id	<code>smallint</code>	Unique identifier for the time range. 1 represents the "at all times" limit.
startday	<code>tinyint</code>	Day of week (1 – 7) for the beginning of the range. Monday = 1, Sunday = 7.
endday	<code>tinyint</code>	Day of week (1 – 7) for the end of the range. Monday = 1, Sunday = 7.
starttime	<code>varchar(10)</code>	Time of day for the beginning of the range.
endtime	<code>varchar(10)</code>	Time of day for the end of the range.

Indexes

Clustered index on `id`

1.67 systransactions

Applies to master database only. `systransactions` contains information about SAP ASE transactions, but it is not a normal table. Portions of the table are built dynamically when queried by a user, while other portions

are stored in the master database. Updates to the dynamically built columns of `systransactions` are not allowed.

i Note

Because of this change in the datatypes for the Cluster Edition, SAP strongly recommends that you archive and truncate audit tables before you upgrade. This reduces the likelihood of a failed upgrade because of insufficient space in the `sybsecurity` database.

Columns

The columns for `systransactions` are:

Name	Datatype	Description
<code>xactkey</code>	<code>binary(14)</code>	Unique SAP ASE transaction key
<code>starttime</code>	<code>datetime</code>	Date the transaction started
<code>failover</code>	<code>int</code>	Value indicating the transaction failover state. Valid values are: <ul style="list-style-type: none">• 0 – Resident Tx• 1 – Failed-over Tx• 2 – Tx by Failover-Conn
<code>type</code>	<code>int</code>	Value indicating the type of transaction. Valid values are: <ul style="list-style-type: none">• 1 – Local• 3 – External• 98 – Remote• 99 – Dtx_State
<code>coordinator</code>	<code>int</code>	Value indicating the coordination method or protocol. Valid values are: <ul style="list-style-type: none">• 0 – None• 1 – Syb2PC• 2 – ASTC• 3 – XA• 4 – DTC
<code>state</code>	<code>int</code>	Value indicating the current state of the transaction.

Name	Datatype	Description
connection	int	Value indicating the connection state. The connection values and states are: <ul style="list-style-type: none"> • 1 – Attached • 2 – Detached
status	int	Internal transaction status flag
status2	int	Additional internal transaction status flags
spid	smallint int for the Cluster Edition	Server process ID, or 0 if the process is detached
masterdbid	smallint	Starting database of the transaction
loid	int	Lock owner ID
namelen	smallint	Length of xactname
xactname	varchar(255) null	Transaction name or <XID>
srvname	varchar(30) null	Name of the remote server (null for local servers)
nodeid	tinyint null	Reserved for future use (not available for cluster environments)
instanceid	tinyint	ID of the instance (available only for cluster environments)

The values for the `state` column are:

state Value	Transaction State
1	Begun
2	Done Command
3	Done
4	Prepared
5	In Command
6	In Abort Cmd
7	Committed

state Value	Transaction State
8	In Post Commit
9	In Abort Tran
10	In Abort Savept
65537	Begun-Detached
65538	Done Cmd-Detached
65539	Done-Detached
65540	Prepared-Detached
65548	Heur Committed
65549	Heur Rolledback

1.68 systypes

Applies to all databases. `systypes` contains one row for each system-supplied and user-defined datatype. Domains (defined by rules) and defaults are given, if they exist.

You cannot alter the rows that describe system-supplied datatypes.

Columns

The columns for `systypes` are:

Name	Datatype	Description
<code>uid</code>	<code>int</code>	User ID of datatype creator
<code>usertype</code>	<code>smallint</code>	User type ID
<code>variable</code>	<code>bit</code>	1 if datatype is of variable length; 0 otherwise
<code>allownulls</code>	<code>bit</code>	Indicates whether nulls are allowed for this datatype
<code>type</code>	<code>tinyint</code>	Physical storage datatype
<code>length</code>	<code>int</code>	Physical length of datatype

Name	Datatype	Description
tdefault	int	ID of system procedure that generates default for this datatype
domain	int	ID of system procedure that contains integrity checks for this datatype
name	varchar(255)	Datatype name
printfmt	varchar(255) null	Reserved
prec	tinyint null	Number of significant digits
scale	tinyint null	Number of digits to the right of the decimal point
ident	tinyint null	1 if column has the IDENTITY property; 0 if it does not
hierarchy	tinyint null	Precedence of the datatype in mixed-mode arithmetic
xtypeid	int null	The internal class ID
xdbid	int null	The dbid where a class is installed: <ul style="list-style-type: none"> -1 = the system database -2 = the current database
accessrule	int null	The object ID of the access rule in sysprocedures

This table lists each system-supplied datatype's name, hierarchy, type (not necessarily unique), and usertype (unique). The datatypes are ordered by hierarchy. In mixed-mode arithmetic, the datatype with the lowest hierarchy takes precedence:

Name	Hierarchy	Type	Usertype
floatn	1	109	14
float	2	62	8
datetimn	3	111	15
datetime	4	61	12
real	5	59	23
numericn	6	108	28
numeric	7	63	10
decimaln	8	106	27

Name	Hierarchy	Type	Usertype
decimal	9	55	26
moneyn	10	110	17
money	11	60	11
smallmoney	12	122	21
smalldatetime	13	58	22
intn	14	38	13
uintn	15	68	47
bigint	16	191	43
ubigint	17	67	46
int	18	56	7
uint	19	66	45
smallint	20	52	6
usmallint	21	65	44
tinyint	22	48	5
bit	23	50	16
univarchar	24	155	35
unichar	25	135	34
unitext	26	174	36
varchar	27	39	2
sysname	27	39	18
nvarchar	27	39	25
longsysname	27	39	42
char	28	47	1
nchar	28	47	24
varbinary	29	37	4

Name	Hierarchy	Type	Usertype
timestamp	29	37	80
binary	30	45	3
text	31	35	19
image	32	34	20
date	33	49	37
time	34	51	38
daten	35	123	39
timen	36	147	40
extended type	99	36	-1

Datatypes with names ending with 'n' are internal, nullable datatypes. When you define a column with a nullable datatype (for example, `datetime null`), SAP ASE converts it to the internal nullable form of the datatype (for example, `datetimn`). You cannot define a column with these internal nullable datatypes when creating a table. For example:

```
create table t1 (c1 intn)
go
Msg 2715, Level 16, State 2:
Server 'ASE', Line 1:
Can't find type 'intn'
```

Indexes

- Unique clustered index on `name`
- Unique nonclustered index on `usertype`

1.69 sysusages

Applies to master database only. `sysusages` contains one row for each disk allocation piece assigned to a database. Each database contains a specified number of database (logical) page numbers.

The `create database` command checks `sysdevices` and `sysusages` to find available disk allocation pieces. One or more contiguous disk allocation pieces are assigned to the database, and the mapping is recorded in `sysusages`.

See *System Tables That Manage Space Allocation* in the *System Administration Guide: Volume 2*.

i Note

In SAP ASE version 15.0 and later, the device identification number is stored in the `vdevno` column and not as part of the `vstart` column. As a consequence, you may need to modify scripts and stored procedures that determine the device identification number based on the earlier schema.

Columns

The columns for `sysusages` are:

Name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	Database ID
<code>segmap</code>	<code>int</code>	Bitmap of possible segment assignments
<code>lstart</code>	<code>unsigned int</code>	First database (logical) page number
<code>size</code>	<code>unsigned int</code>	Number of contiguous database (logical) pages
<code>vstart</code>	<code>int</code>	Starting virtual page number
<code>location</code>	<code>smallint</code>	<p>The location of the archive database segment where the physically contiguous block of pages resides.</p> <p>In the <code>location</code> column, a value of 5 and 6 means the location is in the database dump, transaction log dump, or their stripes, and a value of 7 or 8 means that the location is in the modified pages section. A value of 4 is used to fill the gaps for pages that are not physically available.</p>
<code>unreservedpgs</code>	<code>unsigned int</code>	Free space not part of an allocated extent
<code>crdate</code>	<code>datetime null</code>	Creation date
<code>vdevno</code>	<code>int</code>	Device identification number

Indexes

- Unique clustered index on `dbid`, `lstart`
- Unique nonclustered index on `vdevno`, `vstart`

1.70 sysusermessages

Applies to all databases. `sysusermessages` contains one row for each user-defined message that can be returned by the SAP ASE server.

Columns

The columns for `sysusermessages` are:

Name	Datatype	Description
<code>error</code>	<code>int</code>	Unique error number. Must be 20,000 or higher.
<code>uid</code>	<code>int</code>	Server user ID (<code>suser_id</code>) of the message creator.
<code>description</code>	<code>varchar(1024)</code>	User-defined message with optional placeholders for parameters.
<code>langid</code>	<code>smallint</code> <code>null</code>	Language ID for this message; null for <code>us_english</code> .
<code>dlevel</code>	<code>smallint</code> <code>null</code>	Stores the <code>with_log</code> bit, which is used to call the appropriate routine to log a message.

Indexes

- Clustered index on `error`
- Unique nonclustered index on `error`, `langid`

1.71 sysusers

Applies to all databases. `sysusers` contains one row for each user allowed in the database, and one row for each group or role.

On the SAP ASE distribution media, `master..sysusers` contains some initial users:

- `dbo` – with an `suid` of 1 and `uid` of 1.
- `guest` – with an `suid` of -1 and `uid` of 2.

The user `guest` provides a mechanism for giving users not explicitly listed in `sysusers` access to the database with a restricted set of permissions. The `guest` entry in `master` means any user with an account on the SAP ASE server (that is, with an entry in `syslogins`) can access `master`.

- `public` – with an `suid` of -2 and `uid` of 0.
The user `public` refers to all users. The keyword `public` is used with the `grant` and `revoke` commands to signify that permission is being given to or taken away from all users.

In addition, both system-defined and user-defined roles are listed in `sysusers`.

Columns

The columns for `sysusers` are:

Name	Datatype	Description
<code>suid</code>	<code>int</code>	Server user ID, copied from <code>syslogins</code> .
<code>uid</code>	<code>int</code>	User ID, unique in this database, is used for granting and revoking permissions. User ID 1 is “ <code>dbo</code> ”.
<code>gid</code>	<code>int</code>	Group ID to which this user belongs. If <code>uid = gid</code> , this entry defines a group. Negative values may be used for user IDs (<code>uid</code>). Every <code>suid</code> associated with a group or a role in <code>sysusers</code> is set to -2 (<code>INVALID_SUID</code>).
<code>name</code>	<code>sysname</code>	User or group name, unique in this database.
<code>enviro</code> <code>n</code>	<code>varchar(255)</code> <code>null</code>	Reserved.
<code>user_s</code> <code>tatus</code>	<code>smallint</code>	Indicates the user is resolved as another user. A value of: <ul style="list-style-type: none"> • 1 – indicates that the system administrator issued <code>sp_modifyuser <user_name>, 'resolve as', <other_user_name></code> against this user. • 0 – indicates the user is not resolved as another user.

Indexes

- Nonunique clustered index with “`allow duplicate rows`” on `suid`
- Unique nonclustered index on `name`
- Unique nonclustered index on `uid`

1.72 sysversions

(In-memory databases only) Stores old row versions created for version storage. `sysversions` exists only in on-disk MVCC temporary databases.

Columns

The columns for `sysversions` are:

Name	Datatype	Description
<code>vdbid</code>	<code>smallint</code>	Database ID of the database containing the table
<code>vobjid</code>	<code>int</code>	Object ID of the table
<code>vptnid</code>	<code>int</code>	Partition ID of the table
<code>vrid</code>	<code>bigint</code>	Row ID of the row

Indexes

- Unique clustered index on `<jid>`
- Unique nonclustered index on `<jname>`

1.73 sysxtypes

Applies to all databases. `sysxtypes` contains one row for each extended, Java-SQL datatype.

See *Java in Adaptive Server Enterprise* for more information about Java-SQL classes and datatypes.

Columns

The columns for `sysxtypes` are:

Name	Datatype	Description
<code>xtid</code>	<code>int</code>	System-generated ID for the extended type.

Name	Datatype	Description
<code>xtutid</code>	<code>smallint</code>	Unused.
<code>xtstatus</code>	<code>int</code>	Internal status information. Unused.
<code>xtmetatype</code>	<code>int</code>	Unused.
<code>xtcontainer</code>	<code>int</code>	The ID of the JAR file containing the class. Can be NULL.
<code>xtname</code>	<code>varchar (255)</code> <code>null</code>	The name of the extended type.
<code>xtsource</code>	<code>text null</code>	Source code for the extended type. Unused.
<code>xtbinaryinrow</code>	<code>varbinary (255)</code> <code>null</code>	Object code for the extended type. For Java classes, it contains the class file. Data is stored in-row up to a length of 255 bytes.
<code>xtbinaryoffrow</code>	<code>image</code>	Object code for the extended type. For Java classes, it contains the class file. Data is stored off-row as an image column.

Indexes

- Unique clustered index on `xtid`
- Unique nonclustered index on `xtname`

2 dbccdb Tables

In addition to the standard system tables included in all databases, the `dbcc` management database, `dbccdb`, contains seven tables that define inputs to and outputs from `dbcc checkstorage`. It also contains at least two workspaces.

2.1 dbccdb Workspaces

Workspaces are special tables in `dbccdb` that store intermediate results of the `dbcc checkstorage` operation.

Workspaces differ from worktables in that they:

- Are preallocated contiguously to improve I/O performance
- Are persistent
- Do not reside in the `tempdb` database

When you create `dbccdb`, two workspaces are created automatically. They are preallocated as follows:

- *Scan workspace* – contains a row for each page of the target database. The allocation is approximately 1 percent of the database size. Each row consists of a single `binary (18)` column.
- *Text workspace* – contains a row for each table in the target database that contains `text` or `image` columns. The size of this table depends on the design of the target database, but it is usually significantly smaller than the scan workspace. Each row consists of a single `binary (22)` column.

If either allocation is larger than needed by `dbcc checkstorage`, the operation uses only what is required. The allocation does not change. If the text workspace allocation is too small, `dbcc checkstorage` reports this, recommends a new size, and continues checking; however, not all text chains are checked. If the scan workspace allocation is too small, the `dbcc checkstorage` operation fails immediately.

You must have at least one scan and one text workspace, but you may create as many as you need. While in use, the workspaces are locked so that only one `dbcc checkstorage` operation can use them at any given time. You can execute concurrent `dbcc checkstorage` operations by supplying each one with a separate scan and text workspace.

For more information on creating workspaces, see the *System Administration Guide* and the *Reference Manual*.

Ideally, you should access workspaces only through `dbcc checkstorage`, but this is not a requirement. `dbcc checkstorage` exclusively locks the workspaces it uses, and the content of the workspaces is regenerated with each execution of `dbcc checkstorage`. The workspaces do not contain any secure data.

i Note

While the contents of the workspaces are accessible through SQL, no interpretation of the binary values is available. Access through SQL might return data from different `dbcc` checks mixed together. The presence of a row in these tables does not ensure that it contains valid data. `dbcc` tracks valid rows only during execution. That information is lost when the operation completes.

Most of the update activity in `dbccdb` is performed in the text and scan workspaces. The workspaces are preallocated, and only one `dbcc checkstorage` operation can use the workspaces at any given time, so the workspaces are less susceptible to corruption than most user tables. Corruption in a workspace can cause the `dbcc checkstorage` operation to fail or behave erratically. If this happens, drop and re-create the corrupt workspace.

Checks of databases using different workspaces can proceed simultaneously, but the performance of each operation might be degraded as it competes for disk throughput.

To delete a workspace, in `dbccdb`, enter:

```
drop table <workspace_name>
```

2.2 dbccdb Log

The results of each `dbcc checkstorage` operation are recorded in the `dbccdb` log. Updates to the text and scan workspaces are not recorded there.

You must size the `dbccdb` log to handle updates to the tables. The log requirement is related to the number of tables and indexes in the target database. It is not related to the target database size.

To minimize the log requirement and the recovery time, use the `truncate log on checkpoint` option with `dbccdb`.

2.3 dbcc_config

The `dbcc_config` table describes the currently executing or last completed `dbcc checkstorage` operation.

`dbcc_config` defines:

- The location of resources dedicated to the `dbcc checkstorage` operation
- Resource usage limits for the `dbcc checkstorage` operation

The primary key is the combination of `dbid` and `type_code`

Columns

The columns for `dbcc_config` are:

Column name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	Matches the <code>dbid</code> from a row in <code>sysdatabases</code> .

Column name	Datatype	Description
type_code	int	Matches the type_code from a row in dbcc_types. Valid values are 1 – 9.
value	int null	Specifies the value of the item identified by type_code. Can be null only if the value of stringvalue is not null.
stringvalue	varchar(255) null	Specifies the value of the item identified by type_code. Can be null only if the value of value is not null.

Related Information

[dbcc_types \[page 133\]](#)

2.4 dbcc_counters

The dbcc_counters table stores the results of the analysis performed by dbcc checkstorage. Counters are maintained for each database, table, index, partition, device, and invocation of dbcc.

The primary key is the combination of dbid, id, indid, partitionid, devid, opid, and type_code

Columns

The columns for dbcc_counters are:

Column name	Datatype	Description
dbid	smallint	Identifies the target database.
id	int	Identifies the table. The value is derived from sysindexes and sysobjects.
indid	smallint	Identifies the index. The value is derived from sysindexes.
partitionid	int	Identifies the defined object-page affinity. The value is derived from sysindexes and syspartitions.
devid	int	Identifies the disk device. The value is derived from sysdevices.
opid	smallint	Identifies the dbcc operation that was performed.

Column name	Datatype	Description
type_code	int	Matches the type_code column of a row in dbcc_types. Valid values are 5000 through 5024.
value	real null	Matches the appropriate type_name for the given type_code as described in dbcc_types.

Related Information

[dbcc_types \[page 133\]](#)

2.5 dbcc_exclusions

The dbcc_exclusions table stores the faults, tables or a combination of them that should be excluded from processing by checkverify and fault reporting via sp_dbcc_faultreport.

The primary key is the combination of dbid, fault_type, and table_name

Columns

The columns for dbcc_exclusions are:

Column name	Datatype	Description
dbid	smallint	Identifies the target database.
type	tinyint	Exclusion type code. The valid values are: <ul style="list-style-type: none"> • 1 – faults • 2 – tables • 3 – combo
fault_type	int null	The fault type to be excluded when type is 1 (faults) or 3 (combo). See \dbcc_types for more information.
table_name	varchar (30) null	The table name to be excluded when type is 2 (faults) or 3 (combo). See dbcc_types for more information.

Related Information

[dbcc_types \[page 133\]](#)

2.6 dbcc_fault_params

The `dbcc_fault_params` table provides additional descriptive information for a fault entered in the `dbcc_faults` table.

Each “value” column (`intvalue`, `realvalue`, `binaryvalue`, `stringvalue`, and `datevalue`) can contain a null value. At least one must be not null. If more than one of these columns contains a value other than null, the columns provide different representations of the same value.

The primary key is the combination of `dbid`, `opid`, `faultid`, and `type_code`

Columns

The columns for `dbcc_fault_params` are:

Column name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	Identifies the target database.
<code>opid</code>	<code>smallint</code>	Identifies the <code>dbcc</code> operation that was performed.
<code>faultid</code>	<code>int</code>	Identifies the fault ID.
<code>type_code</code>	<code>int</code>	Defines the interpretation of the value, which is provided by the “value” columns. Valid values are 1000 – 1009. They are described in <code>dbcc_types</code> .
<code>intvalue</code>	<code>int null</code>	Specifies the integer value.
<code>realvalue</code>	<code>real null</code>	Specifies the real value.
<code>binaryvalue</code>	<code>varbinary (255) null</code>	Specifies the binary value.
<code>stringvalue</code>	<code>varchar (255) null</code>	Specifies the string value.
<code>datevalue</code>	<code>datetime null</code>	Specifies the date value.

Related Information

[dbcc_types \[page 133\]](#)

2.7 dbcc_faults

The `dbcc_faults` table provides a description of each fault detected by `dbcc checkstorage`.

The primary key is the combination of `dbid`, `id`, `indid`, `partitionid`, `devid`, `opid`, `faultid`, and `type_code`

Columns

The columns for `dbcc_faults` are:

Column name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	Identifies the target database.
<code>id</code>	<code>smallint</code>	Identifies the table. The value is derived from <code>sysindexes</code> and <code>sysobjects</code> .
<code>indid</code>	<code>smallint</code>	Identifies the index. The value is derived from <code>sysindexes</code> .
<code>partitionid</code>	<code>int</code>	Identifies the partition. The value is derived from <code>sysindexes</code> and <code>syspartitions</code> . Counters are maintained for page ranges, so "partition" refers to the defined object-page affinity, rather than the actual object page chain.
<code>devid</code>	<code>int</code>	Identifies the disk device. The value is derived from <code>sysdevices</code> .
<code>opid</code>	<code>smallint</code>	Identifies the <code>dbcc</code> operation that was performed.
<code>faultid</code>	<code>int</code>	Provides a unique sequence number assigned to each fault recorded for the operation.
<code>type_code</code>	<code>int</code>	Identifies the type of fault. Valid values are 100000 – 100032. They are described in <code>dbcc_types</code> .
<code>status</code>	<code>int</code>	Classifies the fault. For more information, see the <i>System Administration Guide</i> .

The values for the `status` column are:

Value	Description
0	Soft fault, possibly transient.

Value	Description
1	Hard fault.
2	Soft fault that proved to be transient.
3	Soft fault upgraded to a hard fault.
5	Repaired hard fault.
7	Repaired upgraded hard fault.
9	Hard fault not repairable.
11	Soft fault upgraded to a hard fault and not repairable.
16	Soft fault, object dropped (inaccessible).
17	Hard fault, object dropped (inaccessible).
18	Transient soft fault, object dropped (inaccessible).
19	Soft fault upgraded to a hard fault and object dropped (inaccessible).

Related Information

[dbcc_types \[page 133\]](#)

2.8 dbcc_operation_log

The `dbcc_operation_log` table records the use of the `dbcc checkstorage` operations.

Summary results are recorded in the `dbcc_operation_results` table.

The primary key is the combination of `dbid`, `opid`, and `optype`

Columns

The columns for `dbcc_operation_log` are:

Column Name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	Identifies the target database.
<code>opid</code>	<code>smallint</code>	Identifies the sequence number of the <code>dbcc checkstorage</code> operation. <code>opid</code> is an automatically incrementing number, unique for each <code>dbid</code> and <code>finish</code> pair.
<code>optype</code>	<code>smallint</code>	The valid value is valid for <code>optype</code> is 2 = <code>checkstorage</code> .
<code>suid</code>	<code>int</code>	Identifies the user executing the command.
<code>start</code>	<code>datetime</code>	Identifies when the operation started.
<code>finish</code>	<code>datetime</code> <code>null</code>	Identifies when the operation ended.
<code>seq</code>	<code>smallint</code> <code>null</code>	The sequence number for a <code>checkverify</code> operation.
<code>id</code>	<code>int null</code>	The object ID, if used, for a <code>checkverify</code> operation.
<code>maxseq</code>	<code>smallint</code> <code>null</code>	The maximum sequence used by <code>checkverify</code> for a <code>checkstorage</code> operation.

2.9 dbcc_operation_results

The `dbcc_operation_results` table provides additional descriptive information for an operation recorded in the `dbcc_operation_log` table.

Each "value" column (`intvalue`, `realvalue`, `binaryvalue`, `stringvalue`, and `datevalue`) may contain a null value. At least one must be not null. If more than one of these columns contains a value other than null, the columns provide different representations of the same value.

Results of the `dbcc checkstorage` operations include the number of:

- Hard faults found
- Soft faults found
- Operations stopped due to a hard error

The primary key is the combination of `dbid`, `opid`, `optype`, and `type_code`

Columns

The columns for `dbcc_operation_results` are:

Column Name	Datatype	Description
<code>dbid</code>	<code>smallint</code>	Identifies the target database.
<code>opid</code>	<code>smallint</code>	Identifies the <code>dbcc</code> operation ID.
<code>optype</code>	<code>smallint</code>	Identifies the <code>dbcc</code> operation type.
<code>type_code</code>	<code>int</code>	Defines the <code>dbcc</code> operation type. Valid values are 1000 – 1007. They are described in <code>dbcc_types</code> .
<code>intvalue</code>	<code>int null</code>	Specifies the integer value.
<code>realvalue</code>	<code>real null</code>	Specifies the real value.
<code>binaryvalue</code>	<code>varbinary (255) null</code>	Specifies the binary value.
<code>stringvalue</code>	<code>varchar (255) null</code>	Specifies the string value.
<code>datevalue</code>	<code>datetime null</code>	Specifies the date value.
<code>seq</code>	<code>smallint null</code>	The sequence number for a <code>checkverify</code> operation.

Related Information

[dbcc_types \[page 133\]](#)

2.10 dbcc_types

Provides the definitions of the datatypes used by `dbcc checkstorage`.

This table is not actually used by the `dbcc` stored procedures. It is provided to facilitate the use of the other tables in `dbccdb`, and to document the semantics of the datatypes. Type codes for operation configuration, analysis data reported, fault classification, and fault report parameters are included. If you create your own stored procedures for generating reports, you can use the values listed in the `type_name` column as report headings.

Columns

The columns for `dbcc_types` are:

<code>type_code</code>	<code>type_name</code>	Description
1	<code>max worker processes</code>	(Optional) Specifies the maximum number of worker processes that can be employed. This is also the maximum level of concurrent processing used. Minimum value is 1.
2	<code>dbcc named cache</code>	Specifies the size (in kilobytes) of the cache used by <code>dbcc checkstorage</code> and the name of that cache.
3	<code>scan workspace</code>	Specifies the ID and name of the workspace to be used by the database scan.
4	<code>text workspace</code>	Specifies the ID and name of the workspace to be used for text columns.
5	<code>operation sequence number</code>	Specifies the number that identifies the <code>dbcc</code> operation that was started most recently.
6	<code>database name</code>	Specifies the name of the database in <code>sysdatabases</code> .
7	<code>OAM count threshold</code>	Specifies the percentage by which the OAM counts must vary before they can be considered to be an error.
8	<code>IO error abort</code>	Specifies the number of I/O errors allowed on a disk before <code>dbcc</code> stops checking the pages on that disk.
9	<code>linkage error abort</code>	Specifies the number of linkage errors allowed before <code>dbcc</code> stops checking the page chains of an object. Some kinds of page chain corruptions might require a check to be stopped with fewer linkage errors than other kinds of page chain corruptions.
10	<code>enable automatic workspace expansion</code>	The flag that enables or disables automatic expansion of workspaces when estimated size exceeds the actual workspace size.
1000	<code>hard fault count</code>	Specifies the number of persistent inconsistencies (hard faults) found during the consistency check.
1001	<code>soft fault count</code>	Specifies the number of suspect conditions (soft faults) found during the consistency check.
1002	<code>checks aborted count</code>	Specifies the number of linkage checks that were stopped during the consistency check.
1007	<code>text column count</code>	Specifies the number of non-null <code>text/image</code> column values found during the consistency check.

<i>type_code</i>	<i>type_name</i>	Description
5000	<code>bytes data</code>	Specifies (in bytes) the amount of user data stored in the partition being checked.
5001	<code>bytes used</code>	Specifies (in bytes) the amount of storage used to record the data in the partition being checked. The difference between <code>bytes used</code> and <code>bytes data</code> shows the amount of overhead needed to store or index the data.
5002	<code>pages used</code>	Specifies the number of pages linked to the object being checked that are actually used to hold the object.
5003	<code>pages reserved</code>	Specifies the number of pages that are reserved for the object being checked, but that are not allocated for use by that object. The difference between $(8 * \text{extents used})$ and $(\text{pages used} + \text{pages reserved})$ shows the total uncommitted deallocations and pages incorrectly allocated.
5004	<code>pages overhead</code>	Specifies the number of pages used for the overhead functions such as OAM pages or index statistics.
5005	<code>extents used</code>	Specifies the number of extents allocated to the object in the partition being checked. For object 99 (allocation pages), this value is the number of extents that are not allocated to a valid object. Object 99 contains the storage that is not allocated to other objects.
5006	<code>count</code>	Specifies the number of component items (rows or keys) found on any page in the part of the object being checked.
5007	<code>max count</code>	Specifies the maximum number of component items found on any page in the part of the object being checked.
5008	<code>max size</code>	Specifies the maximum size of any component item found on any page in the part of the object being checked.
5009	<code>max level</code>	Specifies the maximum number of levels in an index. This datatype is not applicable to tables.
5010	<code>pages misallocated</code>	Specifies the number of pages that are allocated to the object, but are not initialized correctly. This is a fault counter.
5011	<code>io errors</code>	Specifies the number of I/O errors encountered. This datatype is a fault counter.
5012	<code>page format errors</code>	Specifies the number of page format errors reported. This datatype is a fault counter.
5013	<code>pages not allocated</code>	Specifies the number of pages linked to the object through its chain, but not allocated. This datatype is a fault counter.

<i>type_code</i>	<i>type_name</i>	Description
5014	pages not referenced	Specifies the number of pages allocated to the object, but not reached through its chains. This datatype is a fault counter.
5015	overflow pages	Specifies the number of overflow pages encountered. This datatype is applicable only to clustered indexes.
5016	page gaps	Specifies the number of pages not linked to the next page in ascending sequence. This number indicates the amount of table fragmentation.
5017	page extent crosses	Specifies the number of pages that are linked to pages outside of their own extent. As the number of <code>page extent crosses</code> increases relative to <code>pages used</code> or <code>extents used</code> , the effectiveness of large I/O buffers decreases.
5018	page extent gaps	Specifies the number of page extent crosses where the subsequent extent is not the next extent in ascending sequence. Maximal I/O performance on a full scan is achieved when the number of <code>page extent gaps</code> is minimized. A seek or full disk rotation is likely for each gap.
5019	ws buffer crosses	Specifies the number of pages that are linked outside of their workspace buffer cache during the <code>dbcc checkstorage</code> operation. This information can be used to size the cache, which provides high performance without wasting resources.
5020	deleted rows	Number of deleted rows in the object.
5021	forwarded rows	Number of forwarded rows in the object.
5022	empty pages	Number of pages allocated but not containing data.
5023	pages with garbage	Number of pages that could benefit from garbage collection.
5024	non-contiguous free space	Number of bytes of noncontiguous free space.
10000	page id	Specifies the location in the database of the page that was being checked when the fault was detected. All localized faults include this parameter.
10001	page header	Specifies the hexadecimal representation of the header of the page that was being checked when the fault was detected. This information is useful for evaluating soft faults and for determining if the page has been updated since it was checked. The server truncates trailing zeros.
10002	text column id	Specifies an 8-byte hexadecimal value that gives the page, row, and column of the reference to a text chain that had a fault. The server truncates trailing zeros.

<i>type_code</i>	<i>type_name</i>	Description
10003	<code>object id</code>	<p>Specifies a 9-byte hexadecimal value that provides the <code>object id</code> (table), the <code>partition id</code> (partition of the table) if applicable, and the <code>index id</code> (index) of the page or allocation being checked.</p> <p>For example, if a page is expected to belong to table T1 because it is reached from T1's chain, but is actually allocated to table T2, the <code>object id</code> for T1 is recorded, and the <code>object id expected</code> for T2 is recorded. The server truncates trailing zeros.</p>
10007	<code>page id expected</code>	<p>Specifies the page ID that is expected for the linked page when there is a discrepancy between the page ID that is expected and the page ID that is actually encountered.</p> <p>For example, if you follow the chain from P1 to P2 when going forward, then, when going backward, P1 is expected to come after P2. The value of <code>page id expected</code> is P1, and the value of <code>page id</code> is P2. When the actual value of P3 is encountered, it is recorded as <code>page id actual</code>.</p>
10008	<code>page id actual</code>	<p>When there is a discrepancy between the page ID that is encountered and the expected page ID, this value specifies the actual page ID that is encountered. (See also, <code>type_code</code> 10007.)</p> <p>For example, if you follow the chain from P1 to P2 when going forward, then, when going backward, P1 is expected to come after P2. The value of <code>page id expected</code> is P1, and the value of <code>page id</code> is P2. When the actual value of P3 is encountered, it is recorded as <code>page id actual</code>.</p>
10009	<code>object id expected</code>	<p>Specifies a 9-byte hexadecimal value that provides the expected <code>object id</code> (table), the <code>partition id</code> (partition of the table) if applicable, and the <code>index id</code> (index) of the page or allocation being checked.</p> <p>For example, if a page is expected to belong to table T1 because it is reached from T1's chain, but is actually allocated to table T2, the <code>object id</code> for T1 is recorded, and the <code>object id expected</code> for T2 is recorded. The server truncates trailing zeros.</p>
10010	<code>data-only locked data page header</code>	Indicates the 44-byte page header for the page where the fault is located.
10011	<code>data-only locked b-tree leaf page header</code>	Indicates the 44-byte page header for the page where the fault is located.
10012	<code>data-only locked b-tree header</code>	Indicates the 44-byte page header for the page where the fault is located.
20001	<code>rerun checkstorage reco</code>	Reruns <code>checkstorage</code> .

<i>type_code</i>	<i>type_name</i>	Description
20002	indexalloc reco	Runs dbcc indexalloc with the fix option.
20003	tablealloc reco	Runs dbcc tablealloc with the fix option.
20004	checktable fix_spacebits reco	Runs dbcc tablealloc with the fix_spacebits option.
20005	checktable reco	Runs dbcc checktable.
20006	reorg reco	Runs the reorg command
20007	no action reco	This fault is harmless; no action is required.
30000	drop object reco	Drops the object and re-creates it.
30001	bulk copy reco	Bulk copies the data out and back in.
40000	check logs for hardware failure reco	Checks your operating system logs and corrects all reported hardware problems on disks containing an SAP device.
40001	checkalloc reco	Runs dbcc checkalloc with the fix option.
40002	reload db reco	Reloads the database from a clean backup.
100000	IO error	Indicates that part of the identified page could not be fetched from the device. This is usually caused by a failure of the operating system or the hardware.
100001	page id error	Indicates that the identifying ID (page number) recorded on the page is not valid. This might be the result of a page being written to or read from the wrong disk location, corruption of a page either before or as it is being written, or allocation of a page without subsequent initialization of that page.
100002	page free offset error	Indicates that the end of data on a page is not valid. This event affects insertions and updates on this page. It might affect some access to the data on this page.
100003	page object id error	Indicates that the page appears to be allocated to some other table than the one expected. If this is a persistent fault, it might be the consequence of either: <ul style="list-style-type: none"> • An incorrect page allocation, which might only result in the effective loss of this page to subsequent allocation, or • A corrupted page chain, which might prevent access to the data in the corrupted chain.
100004	timestamp error	Indicates that the page has a timestamp that is later than the database timestamp. This error can result in failure to recover when changes are made to this page.

<i>type_code</i>	<i>type_name</i>	Description
100005	wrong dbid error	Indicates that the database ID <code>dbid</code> is stored on the database allocation pages. When this ID is incorrect, the allocation page is corrupt and all the indicated allocations are suspect.
100006	wrong object error	Indicates that the page allocation is inconsistent. The page appears to belong to one table or index, but it is recorded as being allocated to some other table or index in the allocation page. This error differs from <code>page object id error</code> in that the allocation is inconsistent, but the consequences are similar.
100007	extent id error	Indicates that an allocation was found for a table or index that is unknown to <code>dbcc checkstorage</code> . Typically, this results in the inability to use the allocated storage.
100008	fixed format error	Indicates that the page incorrectly indicates that it contains only rows of a single fixed length. <code>dbcc checkstorage</code> reports this error. <code>dbcc checktable</code> does not report it, but does repair it.
100009	row format error	Indicates that at least one row on the page is incorrectly formatted. This error might cause loss of access to some or all the data on this page.
100010	row offset error	Indicates that at least one row on the page is not located at the expected page offset. This error might cause loss of access to some or all of the data on this page.
100011	text pointer error	Indicates that the location of the table row that points to the corrupted text or image data. This information might be useful for correcting the problem.
100012	wrong type error	Indicates that the page has the wrong format. For example, a data page was found in an index or a <code>text/image</code> column.
100013	non-OAM error	This error is a special case of <code>wrong type error</code> . It is not reported as a separate condition in the current release.
100014	reused page error	Indicates that a page is reached by more than one chain and that the chains belong to different objects. This error indicates illegal sharing of a page through corrupt page chain linkages. Access to data in either or both tables might be affected.
100015	page loop error	Indicates that a page is reached a second time while following the page chain for an object, which indicates a loop in the page chain. A loop can result in a session hanging indefinitely while accessing data in that object.
100016	OAM ring error	Indicates that a page is allocated but not reached by the page chains for the object. Typically, this results in the inability to use the allocated storage.
100017	OAM ring error	Indicates that the OAM page ring linkages are corrupted. This might not affect access to the data for this object, but it might affect insertions, deletions, and updates to that data.

<i>type_code</i>	<i>type_name</i>	Description
100018	missing OAM error	Indicates that <code>dbcc checkstorage</code> found an allocation for the object that was not recorded in the OAM. This error indicates a corruption that might affect future allocations of storage, but probably does not affect access to the presently stored data.
100019	extra OAM error	Indicates that an allocation for this object was recorded in the OAM, but it was not verified in the allocation page. This error indicates a corruption that might affect future allocations of storage, but probably does not affect access to the presently stored data.
100020	check aborted error	Indicates that <code>dbcc checkstorage</code> stopped checking the table or index. To prevent multiple fault reports, the check operation on a single chain might be stopped without reporting this error. When an object contains several page chains, failure of the check operation for one chain does not prevent the continuation of the check operation on the other chains unless a fault threshold is exceeded.
100021	chain end error	Indicates that the end of the chain is corrupted. As a soft fault, it might indicate only that the chain was extended or truncated by more than a few pages during the <code>dbcc checkstorage</code> operation.
100022	chain start error	Indicates that the start of a chain is corrupted or is not at the expected location. If this is a persistent fault, access to data stored in the object is probably affected.
100023	used count error	Indicates an inconsistency between the count of the pages used that is recorded in the OAM page and the count of the pages used that is determined by examining the allocation pages.
100024	unused count error	Indicates an inconsistency between the count of the pages reserved but unused that is recorded in the OAM page and the count of the pages reserved but unused that is determined by examining the allocation pages.
100025	row count error	Indicates an inconsistency between the row count recorded in the OAM page and the row count determined by <code>dbcc checkstorage</code> .
100026	serialalloc error	Indicates a violation of the serial allocation rules applied to log allocations.
100027	text root error	Indicates a violation of the format of the root page of a <code>text</code> or <code>image</code> index. This check is similar to the root page checks performed by <code>dbcc textalloc</code> .
100028	page misplaced	Indicates that pages of this object were not found where they were expected to be from examination of the system tables. This usually indicates that <code>sp_placeobject</code> was used sometime in the past. In the <code>dbcc_counters</code> table, all misplaced pages are counted together, rather than being reported by device and partition.

<i>type_code</i>	<i>type_name</i>	Description
100029	page header error	Indicates an internal inconsistency in the page's header other than the kind described by the other type codes. The severity of this error depends on the type of page and the inconsistency found.
100030	page format error	Indicates an internal inconsistency in the page's body other than the kind described by the other type codes. The severity of this error depends on the type of page and the inconsistency found.
100031	page not allocated	Indicates that <code>dbcc checkstorage</code> reached an unallocated page by following a page chain. This condition might affect access to data stored in this object.
100032	page linkage error	Indicates that <code>dbcc checkstorage</code> detected a fault with either the next or previous linkage of an interior page of a chain. If this is a persistent fault, access to data stored in the object is probably affected.
100033	non-contiguous free-space error	Indicates an invalid or inconsistent value for the noncontiguous free space on the page.
100034	insert free space error	Indicates an invalid or inconsistent value for the contiguous free space on the page.
100035	spacebits mismatch	Indicates an inconsistency in the page fullness indicator.
100036	deleted row count error	Indicates an invalid or inconsistent value for the deleted row count on the page.
100037	forwarded rows error	Indicates an inconsistency between the forwarded rows indicator and the number of forwarded rows on the page.
100038	page header type error	Indicates that a page header format indicator set incorrectly.
100039	incorrect extent oampage	Extent OAM page reference is set incorrectly
100040	OAM page format error	Non-first OAM page has non-zero first OAM page-specific data.

i Note

To allow for future additions to `dbcc_types`, some `type_code` numbers are not used at this time.

3 Monitoring Tables

The Attributes column in monitoring and diagnostic (MDA) tables provides information about how the SAP ASE server manages the column.

An Attribute value of:

- “Counter” indicates value in this column may wrap, or become zero and start incrementing again, because the value exceeds the maximum possible value of 2^{31} . The SAP ASE server resets the monitor counters when you run `sp_sysmon` without the `noclear` option. In SAP ASE version 15.0.1 and later, the `noclear` option is, by default, included as a `sp_sysmon` parameter. In versions earlier than 15.0.1, you must specify `noclear` to prevent the SAP ASE server from resetting the monitor counters. Resetting monitor counters may skew your results if you run `sp_sysmon` on the same SAP ASE server on which you are using the monitoring tables.
- “Null” indicates the column value may be null.
- “Reset” indicates the column is reset when you run `sp_sysmon` in a manner that causes it to clear the monitoring counters (see *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon*).

3.1 monBucketPool

Collects information about allocation of memory for memory pools managed by the bucket pool manager.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	The Server Instance Identifier (cluster only).
BucketPoolName	varchar(30)	None	Name of the bucket pool.
NumBuckets	int	None	Number of buckets in a bucket pool instance
NumInstances	int	None	Number of instances in each bucket pool
Flags	int		For internal use.
NumSets	int	None	Number of sets in each bucket pool instance. Additional sets are added when the pool grows. Sets can be removed by shrinking the pool.

Name	Datatype	Attributes	Description
BucketPoolSize	int	None	Size of the bucket pool (in bytes).
BucketPoolUsed	int	None	Size of the used portion of the bucket pool (in bytes).
BucketPoolUsedMax	int	None	Largest value of <BucketPoolUsed> since the last time the counters were reset.
BucketPoolOverhead	int	None	Overhead from fragment headers.
BucketPoolAllocs	int	None	Total number of allocation requests made to this bucket pool (whether successful or not).
BucketPoolRetries	int	None	Number of times the allocation was retried due to engine contention, or bucket or instance being empty.
BucketPoolFailures	int	None	Number of times the allocation failed and returned no memory to the caller.
BucketPoolStats	int	None	For internal use.
BucketPoolAllocStats	int	None	For internal use.
BucketPoolOversize	int	None	For internal use.
BucketPoolAutotune	int		Autotune on and off status bit.
BucketPoolMinFragSize	int		Size of the smallest fragment available in the global bucket.
BucketPoolMaxFragSize	int		Size of the largest fragment available in the global bucket.
BucketSize	int	None	Size of the bucket (in bytes).
BucketEmpty	int	None	Number of times the bucket was empty and the allocation was made from the global bucket instead.
BucketAllocs	int	None	Number of allocations made from the bucket since the last time the counters were reset.
BucketFrees	int	None	Number of fragments freed or added to the bucket since the last time the server reset the counters.
BucketAllocsMissed	int	None	Number of times the allocation had to be retried because another engine allocated the intended fragment first.
BucketSeed	int	None	Number of fragments initially added to each newly-created bucket.

Name	Datatype	Attributes	Description
BucketNonEmptyInstances	int		Number of instances of a bucket having at least one free fragment to allocate.
InstanceEmpty	int	None	Number of times the allocation could not be made by the preferred instance for the current engine, and the engine had to try the next instance.
InstanceNumFragments	int	None	Current number of free fragments in the instance.
InstanceHWM	int	None	Highest value recorded for <Numfrags> since the counters were last reset.
InstanceLWM	int	None	Lowest value recorded for <Numfrags> since the counters were last reset.
InstanceAllocs	int	None	Number of allocations made from the instance since the server last reset the counters.
InstanceFrees	int	None	Number of fragments freed or added to the instance since the last time the server reset the counters.
InstanceFragSkipped	int		Number of fragments skipped in an instance of the global bucket before finding the right fragment for allocation.

3.2 monCachedObject

Stores statistics for all tables, partitions, and indexes with pages currently in a data cache.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
CacheID	int	None	Unique identifier for the cache.
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
DBID	int	None	Unique identifier for the database.

Name	Datatype	Attributes	Description
IndexID	int	None	Unique identifier for the index.
PartitionID	int	None	Unique identifier for the partition. This is the same value as ObjectID for nonpartitioned objects.
CachedKB	int	None	Number of kilobytes of the cache the object is occupying.
CacheName	varchar (30)	None	Name of the cache.
ObjectID	int	None	Unique identifier for the object. Null if the descriptor for the object has been removed from the server's metadata cache. In this situation, you can determine the object identifier by querying <code>syspartitions</code> in the specified database for the value of <code>PartitionID</code> .
DBName	varchar (30)	None	Name of the database (NULL if the descriptor for the object was removed from the server's metadata cache).
OwnerUserID	int	None	Unique identifier for the object owner.
OwnerName	varchar (30)	None	Name of the object owner (null if the descriptor for the object was removed from the server's metadata cache).
ObjectName	varchar (30)	None	Name of the object (null if the descriptor for the object was removed from the server's metadata cache).
PartitionName	varchar (30)	None	Name of the object partition (null if the descriptor for the object was removed from the server's metadata cache).
ObjectType	varchar (30)	None	Object type (null if the object is no longer open).
TotalSizeKB	int	Counter	Partition size, in kilobytes.
ProcessesAccessing	int	Counter	Number of processes currently accessing pages for this object in the data cache.

3.3 monCachePool

Stores statistics for all pools allocated for all data caches.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
CacheID	int	None	Unique identifier for the cache
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
IOBufferSize	int	None	Size (in bytes) of the I/O buffer for the pool
AllocatedKB	int	None	Number of bytes allocated for the pool
PhysicalReads	int	Counter	Number of buffers read from disk into the pool
Stalls	int	Counter, reset	Number of times I/O operations were delayed because no clean buffers were available in the wash area for this data cache
PagesTouched	int	Counter	Number of pages that are currently being used within the pool
PagesRead	int	Counter	Number of pages read into the pool
BuffersToMRU	int	Counter	Number of buffers fetched and replaced in the most recently used portion of the pool
BuffersToLRU	int	Counter	Number of buffers fetched and replaced in the least recently used portion of the pool: fetch and discard
CacheName	varchar(30)	None	Name of the cache
LogicalReads	int	Counter	Number of buffers read from the pool
PhysicalWrites	int	Counter	Number of write operations performed for data in this pool (one write operation may include multiple pages)
APFReads	int	Counter	Number of asynchronous prefetch (APF) read operations that loaded pages into this pool
APFPercentage	int	None	The configured asynchronous prefetch limit for this pool
WashSize	int	None	The wash size, in kilobytes, for a memory pool

3.4 monCachedProcedures

Stores statistics for all stored procedures, triggers, and compiled plans currently stored in the procedure cache.

Enable the `enable_monitoring` and `statement_statistics_active` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
ObjectID	int	None	Unique identifier for the procedure
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
OwnerUID	int	None	Unique identifier for the object's owner
DBID	int	None	Unique identifier for the database in which the object exists
PlanID	int	None	Unique identifier for the query plan for the object in the procedure cache
MemUsageKB	int	None	Number of kilobytes of memory used by the procedure
CompileDate	datetime	None	Date that the procedure was compiled
ObjectName	varchar (30)	None	Name of the procedure
ObjectType	varchar (32)	None	Type of procedure (for example, stored procedure or trigger)
OwnerName	varchar (30)	None	Name of the object owner
DBName	varchar (30)	None	Name of the database
RequestCnt	int	Counter	Number of times this procedure was requested from cache
TempdbRemapCnt	int	Counter	Number of times this procedure was remapped for the temporary database's ID.

Name	Datatype	Attributes	Description
AvgTempdbRemapTime	int	None	Average time (in milliseconds) spent remapping the temporary databases's ID.
ExecutionCount	int	Counter	Number of times the SAP ASE server executed the stored procedure plan or tree since it was cached
CPUTime	int	Counter	Total number of milliseconds of CPU time used
ExecutionTime	int	Counter	Total amount of elapsed time, in milliseconds, the SAP ASE server spent executing the stored procedure plan or tree
PhysicalReads	int	Counter	Number of physical reads performed
LogicalReads	int	Counter	Number of pages read
PhysicalWrites	int	Counter	Number of physical writes performed
PagesWritten	int	Counter	Number of pages written
SnapCodegenTime	int	Counter	Total number of microseconds of CPU time used by the stored procedure plan's compiled queries code generation for simplified native access plans.
SnapJITTime	int	Counter	Total number of microseconds of CPU time used by the stored procedure plan's just-in-time compilation for simplified native access plans.
SnapExecutionTime	int	Counter	Total accumulated amount of elapsed time, in microseconds, that the stored procedure plan's for compiled queries (simplified native access plans) has executed across multiple procedure executions.
SnapExecutionCount	int	Counter	Number of times the stored procedure plan's compiled queries has been executed since it was compiled.
Active	varchar	None	Indicates if the plan for this procedure is up-to-date with the schema of tables that it accesses.
DMLStatementCount	int	None	Number of SQL DML statements executed against HANA.
NonpushdownCount	int	None	Number of SQL DML statements not fully pushed down to HANA.

3.5 monCachedStatement

Stores detailed monitoring information about the statement cache.

The `monCachedStatement` table includes information about resources used during the previous executions of a statement, how frequently a statement is executed, the settings in effect for a particular plan, the number of concurrent uses of a statement, and so on. This information can be helpful when troubleshooting, and when deciding which statements to retain in the cache.

i Note

Machines that use multiple CPUs with different clock frequencies may report inaccurate elapsed time.

- The columns in `monCachedStatement` allow two attributes: “counter” if the column has a counter value, and “reset” if you can reset the column using `sp_sysmon`.
- Enable the `enable monitoring, statement cache size, and enable stmt cache monitoring` configuration parameters for this monitoring table to collect data.
- Versions of SAP ASE earlier than 16.0 updated metrics in `monCachedStatement` when the statement finished. However, when SAP ASE 16.0 and later executes a statement cache, it periodically updates these values while it executes a query:

- `TotalLIO`
- `MaxLIO`
- `TotalPIO`
- `MaxPIO`
- `TotalCPUtime`
- `MaxCPUtime`
- `TotalElapsedTime`
- `MaxElapsedTime`

- Increments the `UseCount` column when statement begin execution. The value for `UseCount` is:

```
(number of completed queries) + (number of ongoing queries)
```

The `CurrentUsageCount` column includes the number of active queries for a statement. The number of completed executions for a statement is:

```
(Value of UseCount) - (value of CurrentUsageCount)
```

- Increments the value for columns that describe maximums (for example, `MaxCPUtime`) for currently executing statements if the metric described by the column (in this case, `CpuTime`) exceeds the maximum value used during an intermediate update. Maximum columns reflect up-to-date metrics (including metrics for active queries), which helps determine if a currently executing query is consuming resources that exceed previous or normal usage.

Other metrics (for example, `MinLIO` and `AvgLIO`) are updated after query executions are finished.

Columns

Names	Datatypes	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
SSQLID	int	None	Unique identifier for each cached statement. This value is treated as a primary key for <code>monCachedStatement</code> , and is used in functions. <code>show_cached_text</code> uses <code>SSQLID</code> to refer to individual statements in the cache.
Hashkey	int	None	Hash value of the SQL text of the cached statement. A hash key is generated based on a statement's text, and can be used as an approximate key for searching other monitoring tables.
StmtType	tinyint	None	
UserID	int	None	User ID of the user who initiated the statement that has been cached.
SUserID	int	None	Server ID of the user who initiated the cached statement.
DBID	smallint	None	Database ID of the database from which the statement was cached.
UseCount	int	None	Number of times the statement was accessed after it was cached.
StatementSize	int	None	Size of the cached statement, in bytes.
MinPlanSizeKB	int	None	Size of the plan when it is not in use, in kilobytes.
MaxPlanSizeKB	int	None	Size of the plan when it is in use, in kilobytes.
CurrentUsageCount	int	None	Number of concurrent users of the cached statement. Attribute is counter.
MaxUsageCount	int	None	Maximum number of times the cached statement's text was simultaneously accessed. Attribute is counter.

Names	Datatypes	Attributes	Description
NumRecompilesSchemaChanges	int	None	Number of times the statement was recompiled due to schema changes. Running <code>update statistics</code> on a table may result in changes to the best plan. This change is treated as a minor schema change. Recompiling a statement many times indicates that it is not effective to cache this particular statement, and that you may want to delete the statement from the statement cache to make space for some other, more stable, statement. Attribute is counter.
NumRecompilesPlanFlushes	int	None	Number of times the cached statement was recompiled because a plan was not found in the cache. Attribute is counter.
HasAutoParams	tinyint	None	"true" if the statement has any parameterized literals, "false" if it does not.
ParallelDegree	tinyint	None	Degree of parallelism used by the query that is stored for this statement
QuotedIdentifier	tinyint	None	Specifies whether the plan compiled with <code>set quoted_identifier</code> is enabled.
TransactionIsolationLevel	tinyint	None	Transaction isolation level for which the statement was compiled.
TransactionMode	tinyint	None	Specifies whether "chained transaction mode" is enabled for the statement.
SAAuthorization	tinyint	None	Specifies whether the plan was compiled with <code>sa_role</code> authorization.
SystemCatalogUpdate	tinyint	None	Specifies whether <code>allow catalog updates</code> was enabled when the plan was compiled.
MetricsCount	int	None	Number of times metrics were aggregated for this statement.
MinPIO	int	None	Maximum physical I/Os that occurred during any execution of this statement.
MaxPIO	int	None	Maximum physical I/Os that occurred during any execution of this statement.
AvgPIO	int	None	Average number of physical I/Os that occurred during execution of this statement.
MinLIO	int	None	Minimum logical I/Os that occurred during any execution of this statement.

Names	Datatypes	Attributes	Description
MaxLIO	int	None	Maximum logical I/Os that occurred during any one execution of this statement.
AvgLIO	int	None	Average number of logical I/Os that occurred during execution of this statement.
MinCpuTime	int	None	The minimum amount of CPU time, in milliseconds, consumed by any execution of this statement.
MaxCpuTime	int	None	The maximum amount of CPU time, in milliseconds, consumed by any execution of this statement.
AvgCpuTime	int	None	The average amount of CPU time, in milliseconds, consumed by this statement.
MinElapsedTime	int	None	Minimum elapsed execution time for this statement.
MaxElapsedTime	int	None	Maximum elapsed execution time for this statement.
AvgElapsedTime	int	None	Average elapsed execution time for this statement.
AvgScanRows	int	None	Average number of scanned rows read per execution
MaxScanRows	int	None	Maximum number of scanned rows read per execution
AvgQualifyingReadRows	int	None	Average number of qualifying data rows per read command execution
MaxQualifyingReadRows	int	None	Maximum number of qualifying data rows per query execution
AvgQualifyingWriteRows	int	None	Average number of qualifying data rows per query execution
MaxQualifyingWriteRows	int	None	Maximum number of qualifying data rows per query execution
LockWaits	int	None	Total number of lock waits
LockWaitTime	int	None	Total amount of time, in milliseconds, spent waiting for locks
SortCount	int	None	Total number of sort operations
SortSpilledCount	int	None	Total number of sort operations spilled to disk
TotalSortTime	int	None	Total amount of time, in milliseconds, spent in sorts
MaxSortTime	int	None	Maximum amount of time, in milliseconds, spent in a sort

Names	Datatypes	Attributes	Description
DBName	varchar (30)	None	Name of database from which the statement was cached. Attribute is null.
CachedDate	datetime	None	Timestamp of the date and time when the statement was first cached.
LastUsedDate	datetime	None	Timestamp of the date and time when the cached statement was last used. Use this information with CachedDate to determine how frequently this statement is used, and whether it is helpful to have it cached.
LastRecompiledDate	datetime	None	Date when the statement was last recompiled, because of schema changes or because the statement was not found in the statement cache.
OptimizationGoal	varchar (30)	None	The optimization goal used to optimize this statement.
OptimizerLevel	varchar (30)	None	The optimizer level used to optimize this statement.
ParallelDegreeReduced	int	None	Indicates if an insufficient number of worker threads were available to execute the query with the full degree of parallelism the query plan calls for, but the query did execute with some parallelism.
ParallelPlanRanSerial	int	None	Indicates if an insufficient number of worker threads were available to execute the query in parallel so the query was executed serially.
WorkerThreadDeficient	int	None	Indicates that the cumulative total number of worker threads were unavailable to execute this query since it was added to the statement cache.
TotalLIO	bigint	None	Cumulative logical I/O
TotalPIO	bigint	None	Cumulative physical I/O
TotalCpuTime	bigint	None	Cumulative elapsed time, in seconds, this statement spent using CPU
TotalElapsedTime	bigint	None	Cumulative amount of time, in seconds spent executing this statement

3.6 monCIPC

Applies to cluster environments only. Provides summary figures for total messaging within the cluster, as viewed from the current instance or all instances.

One row is returned in the `monCIPC` table for each instance in the cluster, if the system view is set to `cluster`; otherwise, a single row is returned for the instance on which the query is executed.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster
ReceiveCount	int	Counter, reset	Number of messages received by this instance
TransmitCount	int	Counter, reset	Number of messages sent by this instance
Multicast	int	Counter, reset	Number of messages sent that were addressed to all other instances in the cluster
Synchronous	int	Counter, reset	Number of those messages sent synchronously
ReceiveSoftError	int	Counter, reset	Number of recoverable errors received on this instance
ReceiveHardError	int	Counter, reset	Number of unrecoverable errors received on this instance
TransmitSoftError	int	Counter, reset	Number of recoverable transmit errors on this instance
TransmitHardError	int	Counter, reset	Number of unrecoverable transmit errors on this instance
Retransmits	int	Counter, reset	Number of retransmissions performed by this instance
Switches	int	Counter, reset	Number of switches between the primary interconnect network and the secondary interconnect network
FailedSwitches	int	Counter, reset	Number of attempts to switch between primary and secondary interconnect networks that failed
RegularBuffersInUse	int	None	Number of buffers from the CIPC regular buffer pool currently allocated.
FreeRegularBuffers	int	None	Number of buffers available in the CIPC regular buffer pool.

Name	Datatype	Attributes	Description
MaxRegularBuffersInUse	int	None	Maximum number of buffers from the CIPC regular buffer pool allocated at any time since the server was started.
LargeBuffersInUse	int	None	Number of buffers from the CIPC large buffer pool currently allocated.
FreeLargeBuffers	int	None	Number of buffers available in the CIPC large buffer pool.
MaxLargeBuffersInUse	int	None	Maximum number of buffers from the CIPC large buffer pool allocated at any time since the server was started.

3.7 monCIPCEndpoints

Applies to cluster environments only. Provides a detailed summary, giving traffic data for each subsystem within the cluster instance.

One row is returned for each logical endpoint in the instance. If the system view is set to `cluster`, a set of rows is returned for each node in the cluster.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster
ReceiveCount	int	Counter, reset	Number of messages received by this logical endpoint within the cluster
TransmitCount	int	Counter, reset	Number of messages sent by this logical endpoint within the instance
ReceiveBytes	int	Counter, reset	Number of bytes received by this logical endpoint within the instance
TransmitBytes	int	Counter, reset	Number of bytes sent by this logical endpoint within the instance
ReceiveQ	int	Counter	Current number of messages queued for this logical endpoint

Name	Datatype	Attributes	Description
MaxReceiveQ	int	Counter	Maximum number of messages ever observed queued for this logical endpoint
DoneQ	int	None	Current number of messages for this logical endpoint that were processed and await further action
MaxDoneQ	int	None	Maximum number of messages ever observed for this logical endpoint, which have been processed and await further action
MaxRecvQTime	real	None	Maximum time (in milliseconds) a message spends in the queues of the current logical endpoint.
AvgRecvQTime	real	None	Average time (in milliseconds) a message spends in the queues of the current logical endpoint.
EndPoint	varchar (30)	None	Name of CIPC endpoint

3.8 monCIPCLinks

Applies to cluster environments only. Monitors the state of the links between instances in the cluster.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster.
LocalInterface	varchar (30)	None	Name of the link's local network endpoint. Same name that appears in the <code>hosts</code> file for a server name.
RemoteInterface	varchar (30)	None	Name of the link's remote end point. Same name that appears in the <code>hosts</code> file for a server name.
PassiveState	varchar (10)	None	Latest state listed in the traffic on the link.
PassiveStateAge	int	None	Time since the <code>PassiveState</code> column was updated, in milliseconds.

Name	Datatype	Attributes	Description
ActiveState	varchar(10)	None	Latest state used, as determined by active monitoring (when no traffic was present on the link).
ActiveStateAge	int	None	Time since the ActiveState column was updated, in milliseconds.

3.9 monCIPCMesh

Applies to cluster environments only. Gives summary figures for the mesh of connections, from the current instance to all other instances in the cluster, on a per-instance basis.

One row is returned for each of the four connections to each of the other nodes in the cluster, up to the maximum configured. If the system view is `cluster`, a set of rows for each instance active in the cluster is returned.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster.
FarInstanceID	tinyint	None	Instance number of the far-end instance in the cluster.
Received	int	Counter	Number of messages received by this instance from the FarInstanceID instance.
Dropped	int	Counter	Number of messages from the FarInstanceID instance that were dropped, due to a lack of resources.
Transmitted	int	Counter	Number of messages transmitted to the FarInstanceID instance.
Resent	int	Counter	Number of messages re-sent to the FarInstanceID instance.
Retry	int	Counter	Number of packets retried to the FarInstanceID instance.

Name	Datatype	Attributes	Description
ControlRx	int	Counter	Number of control messages received by the InstanceID instance.
ControlTx	int	Counter	Number of control messages sent by the InstanceID instance for this mesh.
SendQ	int	Counter	Current number of messages waiting to be sent to the FarInstanceID instance for this mesh.
MaxSendQ	int	Counter	Maximum number of packets in the send queue for this mesh since the InstanceID instance was started.
SentQ	int	Counter	Number of packets sent by the InstanceID instance to the FarInstanceID instance that have not yet been acknowledged by the FarInstanceID instance.
MaxSentQ	int	Counter	Maximum number of messages sent, but notification of sending is not yet processed.
MaxSendQTime	real	None	Maximum time that has been required to process a message in the send queue for this mesh. In milliseconds.
AvgSendQTime	real	None	Average amount of time required to process a message in the send queue for this mesh. In milliseconds.
Mesh	varchar(30)	None	The channel name for the connection. One of: <ul style="list-style-type: none"> • Out of Band • Message • Large Message • Direct memory access (DMA)
MinRTT	int	None	Minimum round-trip delay observed for messages (applies only to user datagram protocol (UDP) transport).
MaxRTT	int	None	Maximum round trip delay observed for messages (applies only to UDP transport).
AverageRTT	int	None	Average round trip delay observed for messages (applies only to UDP transport).

3.10 monCLMObjectActivity

Applies to cluster environments only. Collects cluster lock information.

monCLMObjectActivity tracks:

- Activity for objects only in the master and user databases.
- Physical lock activity at the partition level.

Cluster object locks for a database have an Object-PartitionID of 0.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
InstanceID	tinyint	None	Instance ID.
DBID	int	None	Database ID.
Object_PartitionID	int	None	Identity of the object making the lock request.
LockRequests	int	Counter	Number of cluster lock requests.
LocalMaster	int	Counter	Number of times a lock request finds the current instance to be the lock master. One instance in the cluster becomes the “lock master.” When an instance needs a cluster lock, it contacts the lock master for the lock.
Waited	int	Counter	Number of lock requests granted with contention at the remote instance.
Granted	int	Counter	Number of lock requests granted without contention at the remote instance.
RWConflictWaited	int	Counter	Number of lock requests that waited because of a read-write conflict lock that was granted to a task at a remote instance.
AvgRWConflictWaitTime	real	None	Average amount of time spent performing the wait described by RWConflictWaited.
MaxRWConflictWaitTime	real	None	Maximum amount of time spent performing the wait described by RWConflictWaited.
WWConflictWaited	int	None	Number of lock requests that waited because of a write-write conflict lock that was granted to a task at a remote instance.
AvgWWConflictWaitTime	real	None	Average amount of time spent performing the wait described by WWConflictWaited.
MaxWWConflictWaitTime	real	None	Maximum amount of time spent performing the wait described in WWConflictWaited.

Column name	Type	Attributes	Description
ClusterMsgWaits	int	Counter	Number of waits due to cluster messaging.
AvgClusterMsgWaitTime	real	None	Average wait time due to cluster messaging.
MaxClusterMsgWaitTime	real	None	Maximum wait time due to cluster messaging.
DowngradeReqRecv	int	Counter	Number of downgrade service requests received at the cluster lock owner.
DowngradeReqRecvWithNoBlocker	int	Counter	Number of the downgrade service requests received without any blocking task ownership at cluster lock owner.
ClusterDeadlock	int	Counter	Number of deadlocks caused by multiple instances attempting to acquire the same cluster lock simultaneously.
LockType	varchar(30)	None	Type of lock.

3.11 monClusterCacheManager

Applies to cluster environments only. Stores diagnostic information about the cluster cache manager daemon running on each instance. `monClusterCacheManager` reports cluster-wide information on a per-instance basis.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster
RequestsQueued	int	Counter, reset	Number of requests queued to the cluster cache manager daemon
RequestsRequeued	int	Counter, reset	Number of requests requeued to the cluster cache manager daemon

Name	Datatype	Attributes	Description
RequestsServed	int	Counter, reset	Number of requests serviced by the cluster cache manager daemon
DiskWrites	int	Counter, reset	Number of disk writes initiated by the cluster cache manager daemon
SleepCount	int	Counter, reset	Number of times the cluster cache manager daemon went to sleep
DaemonName	varchar	None	Name of the cluster cache manager daemon
TransfersInitiated	int	Counter, reset	Number of transfers initiated by the cluster cache manager daemon
Downgrades	int	Counter, reset	Number of downgrades performed by the cluster cache manager daemon
Releases	int	Counter, reset	Number of releases performed by the cluster cache manager daemon
AvgServiceTime	real	None	Average time (in milliseconds) spent servicing a request
MaxQSize	int	None	Maximum number of requests queued to the cluster cache manager daemon at any time since the instance started

3.12 monCMSFailover

Applies to cluster environments only. Tracks the time at which the cluster membership service (CMS) detects the failure, gets a new cluster view, resynchronizes the heartbeat, posts the failure event, and completes the failure event. There is a row for each instance.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
InstanceID	tinyint	None	Instance performing the failover.
FailedInstanceID	varchar (96)	None	List of failed instance IDs, separated by commas.

Column name	Type	Attributes	Description
FailDetectTime	datetime	None	Time when cluster failure is detected.
InitViewTime	datetime	None	Time when initial cluster view is obtained.
FinalViewTime	datetime	None	Time when final cluster view is obtained.
ResynchHBTime	datetime	None	Time when cluster-wide heartbeat is resynchronized.
NotifyFailTime	datetime	None	Time when failure event is posted.
EventdoneTime	datetime	None	Time when last failure event is finished.

3.13 monDataCache

Stores statistics relating to SAP ASE data caches.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
CacheID	int	None	Unique identifier for the cache
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
RelaxedReplacement	int	None	Specifies whether the cache is using relaxed cache replacement strategy
BufferPools	int	None	Number of buffer pools within the cache
CacheSearches	int	Counter, reset	Cache searches directed to the cache
PhysicalReads	int	Counter, reset	Number of buffers read into the cache from disk
LogicalReads	int	Counter, reset	Number of buffers retrieved from the cache
PhysicalWrites	int	Counter, reset	Number of buffers written from the cache to disk

Name	Datatype	Attributes	Description
Stalls	int	Counter, reset	Number of times I/O operations were delayed because no clean buffers were available in the wash area
CachePartitions	smallint	None	Number of partitions currently configured for the cache
CacheName	varchar (30)	None	Name of cache
Status	varchar (30)	None	Status of cache. One of: <ul style="list-style-type: none"> • Active • Pending/Active • Pending/Delete • Update Cache • Cache Create • Cache Delete • (Cluster Edition only) Cache Skip
Type	varchar (30)	None	Type of cache. One of: <ul style="list-style-type: none"> • Default • Mixed • Mixed, HK Ignore • Log Only • In-Memory Storage
CacheSize	int	None	Total size of cache, in kilobytes
ReplacementStrategy	varchar (30)	None	Cache replacement strategy
APFReads	int	Counter	Number of asynchronous prefetch (APF) reads for this data cache
Overhead	int	None	Cache overhead
CASGrabs	bigint	None	Number of times the cache spinlock was acquired
CASSpins	bigint	None	Number of times the process spun, waiting for the cache spinlock
CASWaits	bigint	None	Number of times process spun, waiting for the cache spinlock

3.14 monDBRecovery

Applies to cluster environments only. Contains rows from all instances in the cluster and contains rows for every database that contributes to recovery.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
DBID	int	None	Unique identifier for the database
InstanceID	int	None	Instance that performed the recovery (applicable only to the Cluster Edition)
MaxOpenXacts	int	None	Maximum number of open transactions seen during recovery
MaxPFTSEntries	int	None	Maximum number of PFTS entries seen during recovery
Buckets	int	None	Number of buckets
LogBTotPages	int	None	Number of <code>log scan getpage</code> requests during the log boundary determination pass.
LogBTotAPFWaited	int	None	Number of <code>log scan getpage</code> requests that found the I/O in progress during the log boundary determination pass
LogBTotIO	int	None	Number of <code>log scan getpage</code> requests with physical I/O during the log boundary determination pass
AnlTotRec	int	None	Total number of log records to be scanned by the recovery process
AnlPhase1Recs	int	None	Number of log records in phase 1 recovery process
AnlPhase1RedoRecs	int	None	Number of log records to redo in phase 1 recovery
AnlPhase2Recs	int	None	Number of log records in phase 2 recovery process
AnlPhase2RedoRecs	int	None	Number of log records to redo in phase 2 recovery
AnlTotPages	int	None	Number of <code>log scan getpage</code> requests during the analysis process
AnlTotAPFWaited	int	None	Number of <code>log scan getpage</code> requests that found the I/O in progress during the analysis pass

Column name	Type	Attributes	Description
AnlTotIO	int	None	Number of <code>log scan getpage</code> requests with physical I/O during the analysis pass
RedoOps	int	None	Total operations considered for redo
RedoOpsNotRedonePFTS	int	None	Operations that did not need redo (PFTS check)
RedoOpsRedonePFTS	int	None	Operations that might need redo (PFTS check)
RedoOpsRedoneTS	int	None	Operations that needed redo (timestamp check)
RedoOpsNotRedoneTS	int	None	Operations that did not need redo (timestamp check)
RedoLogTotPages	int	None	Number of <code>log scan getpage</code> requests during the redo pass
RedoLogTotAPFWaited	int	None	Number of <code>log scan getpage</code> requests that found the I/O in progress during the redo pass
RedoLogTotIO	int	None	Number of <code>log scan getpage</code> requests with physical I/O during the redo pass
RedoRecTotPage	int	None	Number of <code>recovery pages getpage</code> requests during the redo pass
RedoRecTotAPFWaited	int	None	Number of <code>recovery pages getpage</code> requests that found the I/O in progress during the redo pass
RedoRecTotIO	int	None	Number of <code>recovery pages getpage</code> requests with physical I/O in progress during the redo pass
UndoRecsUndone	int	None	Number of log records undone
UndoLogTotPages	int	None	Number of <code>log scan getpage</code> requests during the undo pass
UndoLogTotAPFWaited	int	None	Number of <code>log scan getpage</code> requests that found the I/O in progress during the undo pass
UndoLogTotIO	int	None	Number of <code>log scan getpage</code> requests with physical I/O during the undo pass
UndoRecTotPages	int	None	Number of <code>recovery pages getpage</code> requests during the undo pass
UndoRecTotAPFWaited	int	None	Number of <code>recovery pages getpage</code> requests that found the I/O in progress during the undo pass

Column name	Type	Attributes	Description
UndoRedTotIO	int	None	Number of <code>recovery pages</code> <code>getpage</code> requests with physical I/O during the undo pass
DBName	varchar(30)	None	Name of the database
FailedInstances	int	None	ID of the failed instances (applicable only to the Cluster Edition)
Command	varchar(30)	None	One of <code>load database</code> , <code>load transaction</code> , <code>online database</code> , <code>mount database</code> , and <code>start</code> or <code>failover</code> commands executed by the process that is running recovery
RecType	varchar(30)	None	Type of recovery – one of <code>server start</code> , <code>load database</code> , <code>load transaction</code> , or <code>node failover</code>
LogBStartTime	datetime	None	Start time for the log boundaries determination pass
LogBEndTime	datetime	None	End time for the log boundaries determination pass
AnlStartTime	datetime	None	Start time of analysis pass
AnlEndTime	datetime	None	End time of the analysis pass
RedoStartTime	datetime	None	Start time of the redo pass
RedoEndTime	datetime	None	End time of the redo pass
UndoStartTime	datetime	None	Start time of the undo pass
UndoEndTime	datetime	None	End time of the undo pass

3.15 monDBRecoveryLRTypes

Applies to cluster environments only. Tracks log records seen during recovery. Contains a row for each log record type for which at least one log record was seen by recovery.

You need not enable any configuration parameters for this monitoring table to collect data.

`monDBRecoveryLRTypes` requires no parameters.

Columns

Column name	Type	Attributes	Description
DBID	int4	None	Unique identifier for the database
InstanceID	int1	None	(Cluster environments only) Instance that performed the recovery
NumRecs	int4	None	Number of records seen during recovery, by type
LogRecType	varchar(30)	None	Log record type

3.16 monDeadLock

Provides information about deadlocks. Use `deadlock pipe max messages` to tune the maximum number of messages returned.

`monDeadLock` is an historical monitoring table. See *Performance and Tuning: Monitoring Tables*.

Use `sp_monitor 'deadlock'` to check current deadlock options. The `deadlock` parameter provides a number of reports based on `monDeadLock`, which are useful for analyzing the history of server deadlocks.

Enable the `enable monitoring`, `deadlock pipe max messages`, and `deadlock pipe active` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
DeadlockID	int	None	Unique identifier for the deadlock
VictimKPID	int	None	Kernel process ID (kpid) of the victim process for the deadlock
InstanceID	tinyint	None	ID of an instance in a shared-disk cluster.
HeldInstanceID	tinyint	None	Instance ID of process holding the lock.
ResolveTime	datetime	None	Time when the deadlock was resolved
ObjectDBID	int	None	Unique database identifier for database where the object resides

Name	Datatype	Attributes	Description
PageNumber	int	None	Page number requested for the lock, if applicable
RowNumber	int	None	Row number requested for the lock, if applicable
HeldFamilyID	tinyint	None	spid of the parent process holding the lock
HeldSPID	int	None	spid of process holding the lock
HeldKPID	int	None	kpid of process holding the lock
HeldProcDBID	int	None	Unique identifier for the database where the stored procedure that caused the lock to be held resides, if applicable
HeldProcedureID	int	None	Unique object identifier for the stored procedure that caused the lock to be held, if applicable
HeldBatchID	int	None	Identifier of the SQL batch executed by the process holding the lock when the deadlock occurred
HeldContextID	int	None	Unique context identifier for the process holding the lock when it was blocked by another process (not when it acquired the lock)
HeldLineNumber	int	None	Line number within the batch of the statement being executed by the process holding the lock when it was blocked by another process (not when it acquired the lock)
WaitFamilyID	int	None	spid of the parent process waiting for the lock
WaitSPID	int	None	spid of the process waiting for the lock
WaitKPID	int	None	kpid of the process waiting for the lock
WaitTime	int	None	Amount of time, in milliseconds, that the waiting process was blocked before the deadlock was resolved
ObjectName	varchar (30)	None	Name of the object
HeldUserName	varchar (30)	None	Name of the user for whom the lock is being held
HeldApplName	varchar (30)	None	Name of the application holding the lock
HeldTranName	varchar (255)	None	Name of the transaction in which the lock was acquired

Name	Datatype	Attributes	Description
HeldLockType	varchar (20)	None	Type of lock being held
HeldCommand	varchar (30)	None	Category of process or command that the process was executing when it was blocked
WaitUserName	varchar (30)	None	Name of the user for whom the lock is being requested
WaitLockType	varchar (20)	None	Type of lock requested
HeldSourceCodeID	varchar (30)	None	For internal use only.
WaitSourceCodeID	varchar (30)	None	For internal use only.
HeldClientAppName	varchar (30)	None	Value for the <clientapplname> property set by the application holding the lock
HeldClientName	varchar (30)	None	Value of the <clientname> property set by the application holding the lock
HeldClientHostName	varchar (30)	None	Value for the <clienthostname> property set by the application holding the lock
HeldHostName	varchar (30)	None	Name of the host machine on which the application that executed the query holding the lock is running
HeldNumLocks	int	None	Number of locks currently held by holding spid
HeldProcDBName	varchar (30)	None	Name of the database in which the stored procedure was executing the blocking process at the time the deadlock occurred, if applicable
HeldProcedureName	varchar (30)	None	Name of the stored procedure the blocking process was executing at the time the deadlock occurred, if applicable
HeldStmtNumber	int	None	Statement number in the SQL batch of the SQL statement holding the lock
ObjectDBName	varchar (30)	None	Name of the database
ObjectID	int	None	Unique identifier for the object

Name	Datatype	Attributes	Description
WaitApplName	varchar(30)	None	Name of the application waiting for the lock
WaitBatchID	int	None	Identifier of the SQL batch executed by the process waiting for the lock when the lock timeout occurred
WaitClientApplName	varchar(30)	None	Value of the <clientapplname> property set by the application waiting for the lock
WaitClientHostName	varchar(30)	None	Value of the <clienthostname> property set by the application waiting for the lock
WaitClientName	varchar(30)	None	Value of the <clientname> property set by the application waiting for the lock
WaitCommand	varchar(30)	None	Category of process or command that the process was executing when it was blocked and then timed out
WaitContextID	int	None	Unique context identifier for the process waiting for the lock when it was blocked by another process
WaitHostName	varchar(30)	None	Name of the host running the process waiting for the lock.
WaitLineNumber	int	None	Line number of the SQL statement in the SQL batch or stored procedure waiting for the lock
WaitProcDBID	int	None	Unique identifier for the database in which the stored procedure waiting for the lock resides, if applicable
WaitProcDBName	varchar(30)	None	Name for the database where the stored procedure that is waiting for the lock resides, if applicable
WaitProcedureID	int	None	ID of the stored procedure waiting for the lock, if applicable
WaitProcedureName	varchar(30)	None	Name for the stored procedure waiting for the lock, if applicable
WaitStmtNumber	int	None	Line number in SQL batch waiting for the lock
WaitTranName	varchar(255)	None	Name of the transaction in which the lock was requested
PartitionID	int	None	Unique identifier for the partition

3.17 monDeviceIO

Returns statistical information relating to activity on database devices.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
Reads	int	Counter, reset	Number of reads from the device
APFReads	int	Counter, reset	Number of asynchronous prefetch (APF) reads from the device
Writes	int	Counter, reset	Number of writes to the device
DevSemaphoreReques ts	int	Counter, reset	Number of I/O requests to a mirrored device (if mirrored)
DevSemaphoreWaits	int	Counter, reset	Number of tasks forced to wait for synchronization of an I/O request to a mirrored device (if mirrored)
IOTime	int	Counter	Total amount of time (in milliseconds) spent waiting for I/O requests to be satisfied
ReadTime	int	Counter	Cumulative amount of time spent performing reads on this device
WriteTime	int	Counter	Cumulative amount of time spent performing writes on this device
LogicalName	varchar (3 0)	None	Logical name of the device
PhysicalName	varchar (1 28)	None	Full hierarchic file name of the device

3.18 monDeviceSegmentIO

The `monDeviceSegmentIO` monitoring table displays the DSAM collection information by device and segment.

Columns

Name	Datatype	Attributes	Description
DBID	int	None	The database ID associated with this record. Unique identifier for the database.
DeviceNumber	int	None	The device ID from <code>sysdevices</code> . Unique identifier for the device.
SegmentNumber	int	None	The segment ID from the local database's segments. Unique identifier for the segment.
PhysicalReads	unsigned bigint	None	The number of physical reads recorded for this DBID, device, or segment.
LogicalReads	unsigned bigint	None	The number of logical reads recorded for this DBID, device, or segment.
PhysicalWrites	unsigned bigint	None	The number of writes recorded for this DBID, device, or segment.

3.19 monDeviceSpaceUsage

Provides information about the file systems on which database devices are allocated. Space information is available only for file system devices. File system size and free space values are NULL for database devices allocated on raw devices.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of the instance.
VDevNo	int	None	Virtual number of the device.

Name	Datatype	Attributes	Description
LogicalName	varchar (30)	None	Logical name of the device.
PhysicalName	varchar (128)	None	Physical name of the device.
DeviceSizeMB	int	None	Size of the device, in megabytes.
FileSystemName	varchar (128)	None	Name of the file system.
FileSystemSizeMB	int	None	Size of the file system, in megabytes.
FileSystemFreeMB	int	None	Amount of available free space, in megabytes, on the file system.
DeviceType	varchar (20)	None	Type of device. One of: <ul style="list-style-type: none"> • Raw device • Block device • File system device • Unknown

3.20 monDeviceSegmentUsage

The `monDeviceSegmentUsage` monitoring tables displays pages used in a database by device and segment.

Queries on `monDeviceSegmentUsage` can take a very long time to run, because the table's data is generated by reading system catalogs and scanning disks at the time the query is issued. To minimize the time spent getting results, specify as many of the table's keys (`DBID`, `DeviceNumber`, `SegmentNumber`) as possible, to limit the amount of work SAP ASE performs to obtain results. The bigger the database or the more databases that you scan, the longer the query takes to produce results.

Columns

Name	Datatype	Attributes	Description
DBID	int	None	The database ID from <code>sysdatabases</code> . Unique identifier for the database.
DeviceNumber	int	None	The device ID from <code>sysdevices</code> . Unique identifier for the device.

Name	Datatype	Attributes	Description
SegmentNumber	int	None	The segment ID from the local database's segments. Unique identifier for the segment.
PagesUsed	bigint	None	How many logical pages are in use on this DBID, device, or segment. Pages are allocated in groups of 8, even though some of the allocated pages might not be used. Thus, subtracting this number from the total size of the device for that database in <code>sysusages</code> will say how much space is available for new objects. It does not say whether or not space is available for rows added to objects already stored in that place.
Stranded	int	None	How many logical pages should not be in this segment. Not required by DSAM. It is an indication of whether or not objects should be where they are and the column is usually zero. When it is not zero, the device's segment map, as stored in <code>sysusages</code> does not permit storage of that object on this device. This situation is rare, but can occur after the customer does <code>sp_placeobject</code> to assign a partition to a different segment: the existing data for that partition does not automatically move, so if the new segment is not permitted where that data currently resides then the data is 'stranded'.

This example displays space used on the master device.

```

1> select * from monDeviceSegmentUsage
2> where DeviceNumber = 0
3> order by DBID, SegmentNumber
4> compute sum(PagesUsed) by DBID
5>

```

DBID	DeviceNumber	SegmentNumber	PagesUsed	Stranded
1	0	0	2336	0
1	0	1	552	0
1	0	2	16	0

Compute Result:

DBID	DeviceNumber	SegmentNumber	PagesUsed	Stranded
2904				
2	0	0	744	0
2	0	1	104	0
2	0	2	8	0

Compute Result:

```
-----
                        856
DBID      DeviceNumber  SegmentNumber  PagesUsed      Stranded
-----
          3              0              0              744            0
          3              0              1              104            0
          3              0              2               8             0
```

Compute Result:

```
-----
                        856
DBID      DeviceNumber  SegmentNumber  PagesUsed      Stranded
-----
    31513              0              0              768            0
    31513              0              1              104            0
    31513              0              2               16             0
```

Compute Result:

```
-----
                        888
```

This table only shows databases where the information is cached in memory. If a display of a database not currently cached is requested, no rows will return. Specify `use <database>`, then query the table again. The act of using a database caches its information.

3.21 monEngine

Provides statistics regarding SAP ASE engines.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
EngineNumber	smallint	None	Number of the engine.
ThreadID	int	None	ID of the thread associated with the engine.
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
CurrentKPID	int	None	Kernel process identifier (kpid) for the currently executing process.
PreviousKPID	int	None	kpid for the previously executing process.
CPUTime	int	Counter, reset	Total time, in seconds, the engine has been running.

Name	Datatype	Attributes	Description
SystemCPUTime	int	Counter, reset	Time, in seconds, the engine has been executing system database services.
UserCPUTime	int	Counter, reset	Time, in seconds, the engine has been executing user commands.
IOCPUTime	int	Counter, reset	The amount of time, in seconds, the engine has been waiting for issued IOs to complete.
IdleCPUTime	int	Counter, reset	Time, in seconds, the engine has been in idle spin mode.
Yields	int	Counter, reset	Number of times this engine yielded to the operating system. If you are running the SAP ASE server in process mode, modify the rate of yielding during idle periods using <code>runnable process search count</code> . If you are running the SAP ASE server in threaded mode, modify the rate of yielding during idle periods with <code>alter thread pool .. idle timeout</code> .
Connections	int	Counter	Number of connections this engine handles.
DiskIOChecks	int	Counter, reset	Number of times the engine or disk controller (for process or threaded mode, respectively) checks for asynchronous disk I/O. In process mode, use <code>i/o polling process count</code> to modify the frequency of these checks.
DiskIOPolled	int	Counter, reset	Number of times the engine or disk controller (for process or threaded mode, respectively) polls for completion of outstanding asynchronous disk I/O, which occurs when disk I/O checks indicate that asynchronous I/O has been posted, but is not yet complete.
DiskIOCompleted	int	Counter, reset	Number of asynchronous disk I/Os completed when the engine or disk controller (for process or threaded mode, respectively) polls for outstanding asynchronous disk I/O.
MaxOutstandingIOs	int	None	Current number of I/O requests initiated by this engine that are not completed.
ProcessesAffinitied	int	None	Number of processes associated with this engine.
ContextSwitches	int	Counter, reset	Number of context switches.
HkqcMaxQSize	int	None	Maximum number of items the SAP ASE server can queue for housekeeper garbage collection in this engine.

Name	Datatype	Attributes	Description
HkgcPendingItems	int	None	Number of items yet to be collected by housekeeper garbage collector on this engine.
HkgcHWMItems	int	None	Maximum number of pending items queued for housekeeper garbage collector at any instant since server started.
HkgcOverflows	int	None	Number of items that could not be queued to housekeeper garbage collector due to queue overflows.
HkgcPendingItemsDc omp	int	None	Number of items on this engine waiting for the housekeeper
HkgcOverflowsDcomp	int	None	Number of items on this engine that could not be queued to the housekeeper
Status	varchar (20)	None	Status of the engine (online, offline, and so on).
Starttime	datetime	None	Date that the engine came online.
StopTime	datetime	None	Date that the engine went offline.
AffinitiedToCPU	int	None	Number of the CPU to which the engine is affinitied.
OSPID	int	None	Identifier for the operating system process executing the engine.

3.22 monErrorLog

Returns the most recent error messages from the SAP ASE error log.

Use `errorlog pipe max messages` to tune the maximum number of messages returned. See *Performance and Tuning: Monitoring Tables*.

Enable the `enable monitoring`, `errorlog pipe max messages`, and `errorlog pipe active` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier (spid)

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier (kpid)
FamilyID	int	None	spid of the parent process
EngineNumber	smallint	None	Engine on which the process was running
ErrorNumber	int	None	Error message number
Severity	int	None	Severity of error. SAP ASE versions 15.7 and later use a value of 99 to indicate stack traces; versions earlier than 15.7 use a value of 0.
State	int	None	State of error
Time	datetime	None	Timestamp when error occurred
ErrorMessage	varchar (512)	None	Text of the error message. Attribute is null.

3.23 monFailoverRecovery

Applies to cluster environments only. Contains aggregated failover recovery diagnostic information for the cluster lock manager (CLM), database recovery, and cluster membership service (CMS) modules.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
InstanceID	tinyint	None	Instance performing the recovery.
ModuleName	varchar (30)	None	Name of the module. One of CML, CMS, or Database
FailedInstances	varchar (30)	None	ID of the failed instance.

Column name	Type	Attributes	Description
StartTime	datetime	None	Start time for the module's recovery.
EndTime	datetime	None	End time for the module's recovery.

3.24 monHADRMembers

Provides information about the members in an HADR system.

Columns

Column name	Type	Attributes	Description
GroupName	varchar (30)	None	Name of the group to which the member belongs.
ServerName	varchar (30)	None	Name of the HADR member.
Mode	varchar (30)	None	Current mode in which the HADR member is running.
State	varchar (30)	None	Current state of the HADR member.
ServerMap	varchar (932)	None	Server map containing the host name and port number.

3.25 monHANANonPushdown

(Support for SAP HANA accelerator for SAP ASE) A message log of queries that are not pushed down to SAP HANA. Contains information such as which procedure the statement is executed from, the line number, SQL text, reason of non-pushdown and the execution timestamp.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
DBID	smallint	None	Unique identifier for the database in which the object exists.
DBName	varchar (30)	None	Name of the database.
ProcedureID	int	None	Unique identifier for the procedure.
ProcedureName	varchar (255)	None	Name of the procedure.
LineNumber	int	None	Line number of the statement within the SQL batch.
SQLText	varchar (255)	None	SQL text of the executed statement.
Reason	varchar (255)	None	Reason for non-pushdown.

3.26 monHCBGCTasks

Provides information about garbage collection tasks for indexes that have index hash caching enabled, and the statistics they collect (for example, total hash nodes freed, total memory freed, and so on).

Columns

The columns for `monHCBGCTasks` are:

Name	Datatype	Description
InstanceID	int	The server instance identifier (cluster only)
DBID	int	Unique identifier for the database
DBName	varchar(30)	Name of the database
HashTablesFreed	bigint	Total number of hash tables freed by the GC Task
SPID	int	System process identifier of the garbage collector
Status	varchar(30)	Status of the GC Task
WaitStatus	varchar(30)	Wait status of the GC Task
ExitStatus	varchar(30)	Exit status of the GC Task
LastWakeup	datetime	The date and time of the last wakeup of this task
NumWakeUps	int	Number of times the GC Task woke up
MemoryFreed	bigint	Total amount of memory (in bytes) freed by the GC Task
NodesFreed	bigint	Total number of nodes freed by the GC Task

3.27 monHCBPartitionActivity

Collects monitoring data of indexes when hash caching is enabled.

Columns

i Note

You must enable the `enable_monitoring` configuration parameter to collect HCB usage statistics. If you do not enable this parameter, the `HashCacheHits` and other statistics columns will display 0 in the result set, regardless of the activity.

The columns for `monHCBPartitionActivity` are:

Name	Datatype	Attributes	Description
DBID	int	None	Database ID.
ObjectID	int	None	Unique identifier for the object.
IndexID	int	None	Unique identifier for the index.
PartitionID	int	None	Unique identifier for the partition.
TotalBuckets	bigint	None	Total number of buckets for hash-cached BTree indexes in the partition.
UsedBuckets	bigint	None	Total number of buckets used for hash-cached BTree indexes in the partition.
MaxChainLength	bigint	None	Maximum length of hash nodes in the collision chain.
AvgChainLengt	bigint	None	Average length of hash nodes in the collision chain.
MaxScanLength	bigint	None	Maximum length of hash nodes in the collision chain perceived by DMLs.

Name	Datatype	Attributes	Description
AvgScanLength	bigint	None	Average length of hash nodes in the collision chain perceived by DMLs.
HashCacheScans	bigint	Counter	Total number of scans that attempt to find data rows from the hash table.
HashCacheHits	bigint	Counter	Total number of scans that find and return data rows from hash tables (that is, all search arguments match).
HashCacheSkips	bigint	Counter	Total number of scans that use a BTree index instead of hash caching to retrieve data rows.
MemoryUsed	bigint	None	Total amount of memory used by hash tables in this partition.
NScanFailedNoHashTable	bigint	Counter	Total number of scan failures due to the absence of hash tables.
NScanFailedNoHashNode	bigint	Counter	Total number of scan failures due to the absence of hash nodes.
NScanFailedNotQualified	bigint	Counter	Total number of scan failures due to the absence of qualified data rows.
NScanRestarts	bigint	Counter	Total number of scan restarts.
NScanFailedMaxScanLength	bigint	Counter	Number of scan failures due to exceeding the maximum scan length.
NHashNodesInserted	bigint	Counter	Total number of hash nodes inserted.
NHashNodesDeleted	bigint	Counter	Total number of hash nodes deleted.
NHashNodesEvicted	bigint	Counter	Number of hash nodes evicted.

Name	Datatype	Attributes	Description
NHashNodesAllocOOM	bigint	Counter	Number of times the server failed to insert hash nodes due to unavailable memory.
NHashBucketsAllocOOM	bigint	Counter	Number of times the server failed to insert hash buckets due to unavailable memory.
NHashTablesCreatesOOM	bigint	Counter	Number of times the server failed to create hash buckets due to unavailable memory.
DBName	varchar(30) NULL	None	Database name.
ObjectName	varchar(30) NULL	None	Object name.
PartitionName	varchar(30) NULL	None	Partition name.
Status	varchar(12) NULL	None	Status of the index partition when hash caching is enabled.

3.28 monHCBTuningActivity

Columns

i Note

Monitor tables automatically track statistics of indexes that have hash caching enabled using the auto tuning functionality of SAP ASE.

You must enable the `HCB index auto tuning` configuration parameter to enable HCB index auto tuning and collect tuning statistics. If you do not enable this configuration parameter, a query against this monitor table will display 0 in the result set.

The columns for `monHCBTuningActivity` are:

Name	Datatype	Description
DBID	int	Database ID
ObjectID	int	Unique identifier for the object
IndexID	int	Unique identifier for the index
CurPQueryRate	smallint	Current point query rate for this index
LastPQueryRate	smallint	Historic point query rate for this index
AvgPQueryRate	smallint	Average point query rate for this index
PQueryRateThreshold	smallint	Threshold for the point query rate at which this index is turned off
CurHashScanHitRate	smallint	Current hash table hit rate for this index
LastHashScanHitRate	smallint	Historical hash table hit rate for this index
AvgHashScanHitRate	smallint	Average hash table hit rate for this index
HashScanHitRateThreshold	smallint	Threshold for the hash table hit rate at which this index is turned off
NumHashNodes	bigint	Total number of hash nodes in hash tables for all index partitions
MemoryUsed	bigint	Amount of memory used, in bytes, for hash tables
NumDisabled	tinyint	Number of times this index was disabled by automatic tuning
DBName	varchar (255)	Database name
ObjectName	varchar (255)	Name of the object
IndexName	varchar (255)	Name of the index
Status	varchar (12)	The HCB automatic tuning status for this index
LastTuneTime	datetime	Time of the last automatic tuning decision made for this index
AvgTuneInterval	int	Length, in seconds, of the average automatic tuning decision interval

3.29 monIMRSCache

Provides information about the IMRS cache.

Columns

The columns for `monIMRSCache` are:

Name	Datatype	Description
CacheSizeKB	bigint	Size (in KB) of the cache.
UsedSizeKB	bigint	Amount of the cache (in KB) that is used.
CacheName	varchar(30)	Name of the IMRS cache.
DBName	varchar(30)	Name of the database to which the cache is bound.
Status	varchar(30)	Status of the cache.

Indexes

`monIMRSCache` does not have an index.

3.30 monIMRSCacheActivity

Stores statistics relating to in-memory data caches.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
CacheName	varchar(30)	None	Name of the IMRS cache.

Column name	Type	Attributes	Description
DBName	varchar (30)	None	Name of the database to which the cache is assigned.
Status	varchar (30)	None	Status of the cache.
CacheSizeKB	bigint	None	Size, in kilobytes, of the cache.
NumCacheShrinks	int	None	Number of times the cache size was reduced.
NumCacheGrows	int	None	Number of times the cache size was increased.
NumRows	bigint	None	Number of rows in the cache (across all tables using the cache).
NumRowsHWM	bigint	None	High water mark for the number of rows in the cache. That is, the greatest number of rows in the cache at one time.
MemForLatestVersions	bigint	None	Amount of memory allocated for the latest row versions, including overheads.
MemForLatestVersionsHWM	bigint	None	High water mark for memory in the latest row versions. That is, the greatest amount of memory allocated at one time for latest row versions.
NumVersions	bigint	None	Number of older row versions in the IMRS, excluding the latest row version.
NumVersionsHWM	bigint	None	High water mark for the number of versions. That is, the greatest number of older row versions, excluding the latest row version, at one time.
MemForOlderVersions	bigint	None	Amount of memory allocated for the older row versions, including all overheads.
MemForOlderVersionsHWM	bigint	None	High water mark for memory allocated for older row versions. That is, the greatest amount of memory, at one time, allocated for the older row versions, including all overheads.
NumRowsPendingGC	bigint	None	Number of rows removed from the IMRS that are waiting for garbage collection.
MemForRowVersionsOverhead	bigint	None	Extra memory overheads incurred for rounding off memory requests for row versions
NumTransactions	int	None	Number of transactions, completed or active, executed in this database.

Column name	Type	Attributes	Description
NumTransactionsHWM	int	None	High water mark for the number of transactions, completed or active, executed in this database. That is, the greatest number of transactions, completed or active, executed in this database.
MemForTransactions	bigint	None	Amount of memory allocated for transactions.
NumStatements	int	None	Number of statements active in this database.
NumStatementsHWM	int	None	High water mark for the number of statements active in this database. That is, the greatest number of statements active in this database at one time.
MemForStatements	bigint	None	Amount of memory allocated for tracking active statements.
MemForLookupTable	bigint	None	Amount of memory used by lookup tables (metadata structures used for IMRS-enabled databases).
MemForLookupTableHWM	bigint	None	High water mark for memory used by look up tables. That is, the greatest amount of memory used for look up tables at one time.
MemForQPFRowFmt	bigint		Memory used for a query processor-useable row format.
NumStmtsCachingRows	bigint		Number of statement descriptors allocated for caching rows in the IMRS.
NumStmtsCachingRowsHWM	bigint		High water mark for the number of statement descriptors allocated for caching rows in the IMRS.
MemForStmtsCachingRows	bigint		Memory allocated for statement descriptors, specifically for caching rows in the IMRS.
MemForStmtsCachingRowsHWM	bigint		High water mark for memory allocated for statement descriptors, specifically for caching rows in the IMRS.
OtherMem	bigint	None	Memory used for allocation of other metadata structures.
OldestSPID	int		SPID of the oldest process that registered with IMRS garbage collector.
NumRowsPacked	bigint		Number of rows packed by the pack operation.
MemPacked	bigint		Amount of memory occupied by rows that have been packed by the pack operation.
NumRowsSkippedbyPack	bigint		Number of rows skipped by the pack operation.

Column name	Type	Attributes	Description
NRowsSkippedPackNoLock	bigint		Number of rows skipped by the pack operation because it could not acquire a lock on the row.
NRowsSkippedPackGT1Versions	bigint		Number of rows skipped by the pack operation because the row had more than one version.
NRowsSkippedPackMisc	bigint		Number of rows skipped by the pack operation due to miscellaneous reasons.
NRowsSkippedPackHotRow	bigint		Number of rows skipped by the pack operation because the rows were found to be active.
NumCommittedPackTransactions	bigint		Number of transactions committed by the pack operation.
NumRolledbackPackTransactions	bigint		Number of transactions rolled back by the pack operation.
NumPackWakeups	bigint		Number of times the pack operation was awakened.
NumDelRowsPackWakeups	bigint		Number of times the pack operation was awakened early to pack deleted rows.
NumPageStoreInsertsOOM	bigint		Number of times a row was inserted in the page store due to lack of memory.
NumBytesLoggedInsertPack	bigint		Number of bytes logged by the pack operation for inserted rows.
NumBytesLoggedMigratePack	bigint		Number of bytes logged by the pack operation for migrated rows
NumSysversPgAlloc	bigint		Number of pages allocated for <code>sysversions</code> .
NumSysversPgDeAlloc	bigint		Number of pages deallocated for <code>sysversions</code> .
NumSysversActivePages	bigint		Number of active pages in <code>sysversions</code> .
NumSysversActivePagesHWM	bigint		The high water mark for active pages in <code>sysversions</code> .
NumPageStoreUpdatesOOM	bigint		Number of times a row was updated in the page store due to lack of memory.
NumSysversPgGCProcessed	bigint		Number of <code>sysversions</code> pages processed by the garbage collector.

Column name	Type	Attributes	Description
NumSysversPgReques ts	bigint		Number of <code>sysversions</code> page requests.
LastPackWakeup	datetime		Date and time at which the pack operation was last woken up.
LastPacked	datetime		Date and time at which the pack operation last packed rows.

3.31 monIMRSGCTasks

Provides information about garbage collection tasks that are running and the statistics they collect, such as total memory freed large object garbage collection tasks, total disk space freed by deleting the obsolete LOB versions, and so on.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
InstanceID	tinyint	None	(Cluster only) server instance identifier.
DBID	int	None	Unique identifier for the database currently being used by the process.
SPID	int	None	Server process ID.
NumWakeUps	int	None	Number of times the garbage collector tasks woke up.
MemoryFreed	bigint		Amount of memory freed (in bytes) by the garbage collector tasks.
DiskSpaceFreed	bigint		Amount of disk space freed by large object (LOB) garbage collection tasks.
VersionsFreed	bigint		Row versions freed by garbage collection tasks.
TransactionsFreed	bigint		Transaction structure related to datarow cache, multiversion concurrency control, or on-disk multiversion concurrency control freed by garbage collection tasks.

Column name	Type	Attributes	Description
NumStmtsCachingRowsFreed	bigint		Number of IMRS statement descriptors freed for the statement that cached rows in the IMRS.
NumSysversPgDeallocated	bigint		Number of <code>sysversion</code> pages that were deallocated.
NumSysversDeallocGCCommits	bigint		Number of internal transactions that were opened by the garbage collector while deallocating <code>sysversions</code> pages and are committed.
NumTranPending	bigint		Number of finished transactions that are not yet processed by the garbage collector.
NumTranBlocked	bigint		Number of transactions with blocked garbage collection processing due to concurrent open transactions.
BlockingSPID	int		SPID of the oldest process that can block the IMRS garbage collector
BlockingTime	int		Amount of time, in seconds, that garbage collection process can be blocked due to the current BlockingSPID
Type	varchar (30)	None	Garbage collector task type. One of <code>imrsgc</code> or <code>lobgc</code> .
Status	varchar (30)	None	Garbage collector task status. One of <code>running</code> or <code>sleeping</code> .
WaitStatus	varchar (30)	None	Garbage collector task wait status. One of <code>needed</code> , <code>notneeded</code> , or <code>inprogress</code> .
LastWakeup	datetime		Time when the the garbage collection task was most recently woken up.
DBName	varchar (30)	None	Name of the database (NULL if the descriptor for the object was removed from the server's metadata cache.
ExitStatus	varchar (30)		Garbage collector task exit status. One of <code>exiting</code> , <code>exiting (after cleanup)</code> , or <code>alive</code> .
BlockingSPIDType	varchar (30)		Type of the oldest process that can block the IMRS garbage collector.

3.32 monImrslogRecovery

Stores IMRS-recovery statistics. Each IMRS-enabled database contains one row in this table, which contains only previously completed recovery diagnostics information.

Columns

The columns for `monImrslogRecovery` are:

Name	Datatype	Attributes	Description
DBID	int		Database ID.
LogBCommitAfterFullyC	int		Number of transactions committed after existing transactions are fully committed.
LogBPRTItems	int		Number of purged row table items in the table.
LogBPRTBucket	int		Number of purged row table buckets.
LogBLogRScannedForPRT	int		Number of log rows scanned for purged row tables.
LogBLogPScannedForPRT	int		Number of log pages scanned for purged row tables.
LogBTotPages	int		Total number of <code>imrslog</code> pages scanned during the log boundary calculation phase.
LogBTotAPFWaited	int		Total number of asynchronous prefetches to the <code>imrslog</code> that waited during the <code>imrslog</code> boundary calculation phase.
LogBTotIO	int		Total amount of I/O to the <code>imrslog</code> during the <code>imrslog</code> boundary calculation phase..
FixCommitTransactionLinkedlogs	int		Number of fixed committed transactions that linked to <code>syslogs</code> .
FixPageCountAfterLastknownpg	int		Number of fixed pages after last known page.

Name	Datatype	Attributes	Description
FixValidPagesAfterFixing	int		Number of pages after fixing hole.
FixInvalidTableFragments	int		Number of invalid table fragments.
FixInvalidTableBlocks	int		Number of invalid table blocks.
FixImrslogTotalPages	int		Total number of imrslog pages scanned during the imrslog fix phase.
FixImrslogTotalAPFWaited	int		Total number of asynchronous prefetches to the imrslog that waited during the imrslog fix phase.
FixImrslogTotalIO	int		Total amount of I/O to the imrslog during the fix phase.
FixAuxScanTotalPages	int		Total number of pages scanned during the auxiliary scan of the imrslog fix phase.
FixAuxScanTotalAPFWaited	int		Total number of asynchronous prefetches that waited for the imrslog scan during the auxiliary scan of the imrslog fix phase.
FixAuxScanTotalIO	int		Total amount of I/O spent on the imrslog during the auxiliary scan of the imrslog fix phase.
RedoOps	int		Total number of redo operations.
RedoPRTConsulted	int		Number of purged rows tables consulted during the redo phase.
RedoPRTOpsNotRedone	int		Number of insert records skipped after consulting the purged rows table during the redo phase.
RedoPRTPurgeNotRedone	int		Number of purges skipped after consulting the purged rows table during the redo phase.
RedoPRTPurgeRecordNotRedone	int		Number of purge record skipped after consulting the purged rows table during the redo phase.
RedoLogTotalPages	int		Total number of imrslog pages at the beginning of the redo phase.

Name	Datatype	Attributes	Description
RedoLogTotAPFWaited	int		Total number of asynchronous prefetches that waited at the beginning of the redo phase.
RedoLogTotIO	int		Total amount of I/O in the <code>imrsllog</code> at the beginning of the redo phase.
RedoRecTotPages	int		Total number of pages used at end of the redo phase.
RedoRecTotAPFWaited	int		Total number of asynchronous prefetches that waited at end of the redo phase.
RedoRecTotIO	int		Total amount of I/O used at end of the redo phase.
ReconOps	int		Total number of reconstruction operations.
ReconPRTConsulted	int		Purged row tables consulted during the reconciliation phase.
ReconPRTOpsNotRedone	int		Number of <code>insert</code> records skipped after consulting the purged row tables during the reconciliation phase.
ReconPRTPurgeNotRedone	int		Number of purges skipped after consulting the purged row tables during the reconciliation phase.
ReconPRTPurgeRecordNotRedone	int		Number of purge records skipped after consulting the purged row tables during the reconciliation phase.
ReconBeginTotPages	int		Total number of pages used during the beginning of the reconciliation phase.
ReconBeginTotAPFWaited	int		Total number of asynchronous prefetches that waited during the beginning of the reconciliation phase.
ReconBeginTotIO	int		Total amount of I/O used during the beginning of the reconciliation phase.
ReconEndTotPages	int		Total number of page used during the end of the reconciliation phase.
ReconEndTotAPFWaited	int		Total number of asynchronous prefetches that waited during the end of the reconciliation phase.
ReconEndTotIO	int		Total amount of I/O used during the end of the reconciliation phase.

Name	Datatype	Attributes	Description
DBName	varchar(30)		Database name.
Command	varchar(30)		Command being issued.
RecType	varchar(30)		Type of recovery.
LogBStartTime	datetime		Start time of the imrslog boundary calculation phase.
LogEndTime	datetime		End time of the imrslog boundary calculation phase.
FixStartTime	datetime		Start time of the fix phase.
FixEndTime	datetime		End time of the fix phase.
RedoStartTime	datetime		Start time of the redo phase.
RedoEndTime	datetime		End time of the redo phase.
ReconStartTime	datetime		Start time of the reconciliation phase.
ReconEndTime	datetime		End time of the reconciliation phase.

3.33 monIMRSPartitionActivity

Provides information about the activity for in-memory row storage caches across individual partitions or objects.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
CacheName	varchar(30)		Name of the IMRS cache.
DBID	int	None	Unique identifier of the database to which the IMRS cache is bound.

Column name	Type	Attributes	Description
ObjectID	int	None	Unique identifier for the object. Null if the descriptor for the object has been removed from the server's metadata cache. In this situation, you can determine the object identifier by querying syspartitions in the specified database for the value of PartitionID.
PartitionID	int	None	Unique identifier for the partition. This is the same value as ObjectID for non-partitioned objects.
DBName	varchar (30)		Name of the database to which the cache is assigned (NULL if the descriptor for the object was removed from the server's metadata cache).
ObjectName	varchar (30)		Name of the object (null if the descriptor for the object was removed from the server's metadata cache).
PartitionName	varchar (30)		Name of the object partition (null if the descriptor for the object was removed from the server's metadata cache).
NumRows	bigint	None	Number of rows across all the tables in the partition.
NumRowsHWM	bigint	None	High water mark for the number of rows across all the tables in the partition. That is, the greatest number of rows across all the tables in the partition at one time.
MemForLatestVersions	bigint	None	Memory allocated for the latest versions of the rows, including overheads.
MemForLatestVersionsHWM	bigint	None	High water mark for memory in latest versions of the rows, including overheads. That is, the greatest amount of memory allocated for the latest versions of the row, including overhead.
MemReqdForAllRows	bigint		Memory requested from the IMRS cache to store all rows across this partition.
TotalMemForAllRows	bigint	None	Total memory used from the IMRS Cache to store all the rows across this partition.

Column name	Type	Attributes	Description
NumVersions	bigint	None	Number of older versions of rows in the partition, excluding the latest version.
NumVersionsHWM	bigint	None	High water mark for number of versions.
MemForOlderVersions	bigint	None	Memory allocated for the older versions of the rows, including all over heads.
MemForOlderVersionsHWM	bigint	None	High water mark for memory allocated for older versions of the rows, including all over heads. That is, the greatest amount of memory allocated for older versions of the rows, including all over heads.
MemForRowVersionsOverhead	bigint	None	Extra memory overheads incurred for rounding off memory requests for row versions.
NumInsertedRows	bigint	None	Total number of rows that were inserted to the cache and are still found in the cache (that is, not deleted).
NumMigratedRows	bigint	None	Total number of rows that were migrated to the cache and are still found in the cache.
MemForInsertedRows	bigint	None	Amount of memory used for the latest versions of inserted rows.
MemForMigratedRows	bigint	None	Amount of memory used for the latest versions of migrated rows.
NumInsertedVersions	bigint	None	Number of versions of rows inserted into the cache.
NumMigratedVersions	bigint	None	Number of versions of rows migrated to the cache.
MemForInsertedVersions	bigint	None	Memory allocated for the older versions of inserted versions of rows, including all overheads.
MemForMigratedVersions	bigint	None	Memory allocated for the older versions of migrated versions of rows, including all overheads.
MemForQPFRowFmt	bigint		Memory used to allocate a query processor-useable row format.

Column name	Type	Attributes	Description
RowsInserted	bigint	None	Number of rows inserted.
RowsDeleted	bigint	None	Number of rows deleted.
RowsUpdated	bigint	None	Number of rows updated.
RowsMigrated	bigint	None	Number of rows migrated.
NumRowsPendingGC	bigint	None	Number of rows removed from the IMRS that are waiting for garbage collection.
NumSelCachedRows	bigint		Number of rows of the partition that are currently cached due to a <code>select</code> .
NumRidmPgPackTrans	bigint		Number of pack transactions started for <code>ridmap page pack</code> mode.
NumRidmPgPackRows	bigint		Number of rows packed in <code>ridmap page pack</code> mode.
NumRidmPgPackTranMissed	bigint		Number of times a page could not be packed for <code>ridmap page pack</code> mode.
NumSelCachedRowsHWM	bigint		High water mark for the number of rows cached for selects in the IMRS.
MemForSelCachedRows	bigint		Total amount of memory in cached or migrated rows, and the amount of memory in rows cached for selects.
MemForSelCachedRowsHWM	bigint		High water mark for memory in cached rows.
NumUpdToInsRows	bigint		Number of updates to rows inserted in the IMRS.
NumSelToInsRows	bigint		Number of selects of rows inserted in the IMRS.
NumDelToInsRows	bigint		Number of deletes of rows inserted in IMRS.
NumUpdToMigRows	bigint		Number of updates to rows that DMLs migrated to the IMRS.
NumSelToMigRows	bigint		Number of selects to rows that DMLs migrated to the IMRS.

Column name	Type	Attributes	Description
NumDelToMigRows	bigint		Number of deletes to rows that DMLs migrated to the IMRS.
NumUpdToSelCachedRows	bigint		Number of updates to rows that DMLs migrated to the IMRS.
NumSelToSelCachedRows	bigint		Number of selects to rows that <code>select</code> commands migrated to the IMRS.
NumDelToSelCachedRows	bigint		Number of deletes to rows that <code>select</code> commands migrated to the IMRS.
RowsCached	bigint		Number of rows migrated.
NumRowsPacked	bigint		Number of rows packed by the pack operation.
NumInsertedRowsPacked	bigint		Number of inserted rows packed by the pack operation.
NumMigratedRowsPacked	bigint		Number of migrated rows packed by the pack operation.
NumCachedRowsPacked	bigint		Number of cached rows packed by the pack operation.
MemPacked	bigint		Amount of memory occupied by rows packed by the pack operation.
MemPackedForInsertedRows	bigint		Amount of memory occupied by inserted rows packed by the pack operation.
MemPackedForMigratedRows	bigint		Amount of memory occupied by migrated rows packed by the pack operation.
MemPackedForCachedRows	bigint		Amount of memory occupied by cached rows packed by the pack operation.
NumRowsSkippedbyPack	bigint		Number of rows skipped by the pack operation.
NRowsSkippedPackNoLock	bigint		Number of rows skipped by the pack operation because it could not acquire row locks.
NRowsSkippedPackGT1Versions	bigint		Number of rows skipped by the pack operation because the row had more than one version.

Column name	Type	Attributes	Description
NRowsSkippedPackMisc	bigint		Number of rows skipped by the pack operation due to miscellaneous reasons.
NRowsSkippedPackHotRow	bigint		Number of rows skipped by the pack operation because the row was active.
NumCommittedPackTrans	bigint		Number of transactions committed by the pack operation.
NumRolledbackPackTrans	bigint		Number of transactions rolled back by the pack operation.
NumDelPackTrans	bigint		Number of pack transactions started for an early deleted-rows pack operation.
NumSteadyPackTrans	bigint		Number of pack transactions started in steady pack mode.
NumAggrPackTrans	bigint		Number of pack transactions started in aggressive pack mode.
NumDelPackRows	bigint		Number of deleted rows packed by early deleted-rows pack mode.
NumSteadyPackRows	bigint		Number of rows packed in steady pack mode.
NumAggrPackRows	bigint		Number of rows packed in aggressive pack mode.
NumDelRowsOtherPack	bigint		Number of deleted rows packed by a mechanism other than the early deleted-rows pack mode.
NumUnsuccessfulDelrowsTran	bigint		Number of early deleted-rows pack transactions that could not pack a row.
MemForPackableDelRows	bigint		Lower bound on memory that could be freed if deleted rows in the partition are packed.
NumBytesLoggedInsPack	bigint		Number of bytes logged by the pack operation for inserted rows.
NumBytesLoggedMigPack	bigint		Number of bytes logged for migrated rows by the pack operation.
NumWriteConflicts	bigint		Number of times write conflicts occurred.

Column name	Type	Attributes	Description
NumNoWriteConflicts	bigint		Number of times <code>write</code> conflicts did not occur.
NumLatestVersionScanned	bigint		Number of times the latest version of the row was scanned.
NumOlderVersionScanned	bigint		Number of times an older version of the row was scanned.
NumOldLOBVersions	bigint		Number of old LOB versions in this partition.
NumOldLOBVersionsHWM	bigint		High water mark for the number of old LOB versions in this partition.
OldLOBVersionsSpace	bigint		Amount of disk space (in KB) used by all active old LOB versions in this partition.
OldLOBVersionsSpaceHWM	bigint		High water mark for the disk space (in KB) used by all active old LOB versions in this partition.
AllocatedOldLOBVersionSpace	bigint		Accumulated total disk space (in KB) allocated to old LOB versions in this partition.
ReclaimedOldLOBVersionSpace	bigint		Accumulated total disk space (in KB) reclaimed by garbage collector threads in this partition.
LastPackVisited	bigint		The date and time the pack operation last visited this partition.
LastPacked	bigint		The date and time the pack operation last packed rows in this partition.
DisabledRowTypes	bigint		Row types for which IMRS usage of the partition is disabled by internal ILM tuning.

3.34 monInmemoryStorage

Provides information about inmemory devices configured to store the contents of inmemory databases.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
InstanceID	tinyint	None	ID of an instance
ID	int	None	ID of the data cache to which this device is bound.
DeviceNum	int	None	Device number. Always -1 for inmemory devices.
StartPage	int	None	Page ID for the first page in this device.
NumPages	int	None	Number of pages in this device.
SizeKB	int	None	Device size, in kilobytes.
Name	varchar (30)	None	Name of the data cache for this device.
DeviceName	varchar (30)	None	Name of the inmemory storage device.
Type	varchar (30)	None	The type of storage. Always set to "cache".
Status	varchar (30)	None	Status of the device.

3.35 monIOController

Provides information about I/O controllers.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environment) ID of an instance in a shared-disk cluster
InstanceID	tinyint	None	(Non-cluster environment) ID of an instance
ControllerID	int	None	ID of the I/O controller

Name	Datatype	Attributes	Description
KTID	int	None	ID of the kernel task
EngineNumber	int	None	Engine that owns this controller
BlockingPolls	bigint	Counter	Number of blocking polls
NonBlockingPolls	bigint	Counter	Number of nonblocking polls
EventPolls	bigint	Counter	Number of polls returning an event
NonBlockingEventPolls	bigint	Counter	Number of nonblocking polls returning an event
FullPolls	bigint	Counter	Number of polls returning the maximum number of events
Events	bigint	Counter	Number of events polled
EventHWM	bigint	Counter	Highest number of events returned in a single poll
Pending	int	Counter	Number of pending I/O operations
Completed	bigint	Counter	Number of completed I/O operations
Reads	bigint	Counter	Number of read or receive operations
Writes	bigint	Counter	Number of write or send operations
Deferred	bigint	Counter	Number of I/O operations deferred or delayed
Type	varchar(30)	None	I/O controller type

3.36 monIOQueue

Provides device I/O statistics displayed as data and log I/O for normal and temporary databases on each device.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environment) ID of an instance in a shared-disk cluster
InstanceID	tinyint	None	(Non-cluster environment) ID of an instance.
IOs	int	Counter	Total number of I/O operations
IOTime	int	Counter	Amount of time (in milliseconds) spent waiting for I/O requests to be satisfied
LogicalName	varchar(30)	None	Logical name of the device
IOType	varchar(12)	None	Category for grouping I/O. One of UserData, UserLog, TempdbData, TempdbLog, or System.

3.37 monLatchFreeIndex

Collects information related to latch free indexes.

Columns

Name	Datatype	Attributes	Description
DBID	int	None	Database ID for the latch free index
ObjectID	int	None	Object ID of the latch free index
IndexID	int	None	Name of the latch free index
LFBKeyDeleted	bigint	Counter	Number of LFB index keys deleted
LFBKeyInserted	bigint	Counter	Number of LFB index keys inserted
LFBRIDDeleted	bigint	Counter	Number of LFB index RIDs deleted
DMLConsolidation	bigint	Counter	Number of consolidations via DML

Name	Datatype	Attributes	Description
BUFFlushConsolidation	bigint	Counter	Number of consolidations via buffer flush
ConsolidationAbort	bigint	Counter	Number of aborted consolidations
RestartScanByConso	bigint	Counter	Number of restart of scan due to consolidation
GetPageFromELC	bigint	Counter	Number of get LFB page from ELC
GetPageFromMPTBL	bigint	Counter	Number of get LFB page from mapping table
GetPageFromBCM	bigint	Counter	Number of get LFB page from BCM
TableName	varchar (255) NULL	None	Name of the table of latch free index
IndexName	varchar (255) NULL	None	Name of the latch free index

3.38 monLicense

Provides a list of all licenses currently checked out by the SAP ASE server.

You need not enable any configuration parameters for this monitoring table to collect data.

i Note

monLicense does not require mon_role permission; any user can use it.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
Quantity	int	None	Quantity of licenses used for this feature.
Name	varchar (30)	None	Name of the feature license.

Name	Datatype	Attributes	Description
Edition	varchar (30)	None	Edition of SAP ASE for which this feature is licensed.
Type	varchar (64)	None	License type.
Version	varchar (16)	None	Version of the feature license in use
Status	varchar (30)	None	Status of this feature license (that is, whether the license is within a grace period or expired).
LicenseExpiry	datetime	None	Date that the license expires, if this is an expiring license.
GraceExpiry	datetime	None	Date this license expires, if this license was awarded on grace. Refer to the Status column to determine whether this license was awarded a grace period.
LicenseID	varchar (150)	None	License identifier. This may not be available if the license has been awarded a grace period.
Filter	varchar (14)	None	Filter used when selecting this feature license. Use <code>sp_lmconfig</code> to change the filter.
Attributes	varchar (64)	None	License attributes. These attributes are " <code><name>=<value></code> " pairs which, if specified, limit certain characteristics of SAP ASE. Possible limiters are: <ul style="list-style-type: none"> • ME = maximum number of engines • MC = maximum number of connections • MS = maximum number of disk space • MM = maximum number of memory • CP = maximum number of CPUs

3.39 monLocks

Returns a list of granted locks and pending lock requests.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier of process holding or requesting the lock.
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
DBID	int	None	Unique identifier for this database object.
ParentSPID	int	None	Parent process ID.
LockID	int	None	Lock object ID.
Context	int	None	Lock context (bit field). These values are the same as for those of the context column in syslocks. See the <i>Reference Manual: Tables</i> for information about syslocks.
DBName	varchar(30)	None	Name of the database for the locked object. This column is NULL if the database is not open when monLocks is queried.
ObjectID	int	None	Unique identifier for the object
LockState	varchar(20)	None	Indicates if the lock is granted. Values are: <ul style="list-style-type: none"> Granted Requested
LockType	varchar(20)	None	Type of lock. Values are: <ul style="list-style-type: none"> Exclusive Shared Update
LockLevel	varchar(30)	None	The type of object for which the lock was requested. Values are: <ul style="list-style-type: none"> Row Page Table Address
WaitTime	int	None	The time (in seconds) for which the lock request was not granted.
PageNumber	int	None	Page that is locked when LockLevel = 'PAGE'
RowNumber	int	None	Row that is locked when LockLevel = 'ROW'

Name	Datatype	Attributes	Description
BlockedBy	int	None	If the lock request is blocked, the <code>BlockedBy</code> column is the lock object ID for the process holding the lock that is blocking this lock request. Null if request is not blocked.
BlockedState	varchar(64)	None	Lock state if the lock being held is blocking other lock requests or if the lock request is blocked. Values are: <ul style="list-style-type: none"> Blocked Blocking Demand Detached Null (if there is no blocking condition)
SourceCodeID	varchar(30)	None	For internal use only.
PartitionID	int	None	Unique identifier for the partition

3.40 monLockTimeout

Provides information about lock timeouts. Each row identifies the object on which a blocked lock request occurred, and identities of the blocked and blocking processes.

The `monLockTimeout` table records lock timeout events (called “timeouts”), that occur when:

- Two server processes are in contention for the same object lock, and,
- The `lock wait period` has expired

By default, the `lock wait period` on the server is infinite, so lock timeouts occur only if the user has changed the lock timeout configuration at the:

- Server level – by changing the `lock wait period` configuration parameter
- Session level – by executing the `set lock wait <n>`, where `<n>` is the time, in seconds, for the `lock wait period`

When the `lock wait period` expires – at the server or session level – the SAP ASE server writes a row to `monLockTimeout`, recording the lock timeout event and describing the objects and processes involved in the lock contention.

You must enable the `enable monitoring`, `lock timeout pipe active`, and `lock timeout pipe max messages` configuration parameters for `monLockTimeout` monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a cluster.
LockWaitPeriod	int	None	Configured amount of time processes wait before a timeout occurs.
LockTimeOutLevel	varchar (20)	None	Timeout level. One of: <ul style="list-style-type: none"> • DTM_SERVER • SERVER • SESSION • COMMAND • INVALID
ObjectDBID	int	None	Unique database identifier for database in which the object resides.
ObjectDBName	varchar (30)	None	Name of database in which the object resides.
ObjectID	int	None	Unique identifier for the object.
ObjectName	varchar (255)	None	Name of the object.
PageNumber	int	None	Page number requested for the lock, if applicable.
RowNumber	int	None	Row number requested for the lock, if applicable.
ExpiredAtTime	datetime	None	Time when lock expires.
HeldSPID	int	None	Server process ID (spid) of process holding the lock.
HeldKPID	int	None	Kernel process ID (kpid) of process holding the lock.
HeldInstanceID	int	None	Instance ID for the instance on which the process holding the lock was executing
HeldUserName	varchar (30)	None	Name of the user for whom the lock is held.
HeldApplName	varchar (30)	None	Name of the application holding the lock.
HeldHostName	varchar (30)	None	Name of the host machine on which the application that executed the query holding the lock is running.

Name	Datatype	Attributes	Description
HeldClientName	varchar (30)	None	Value of the <code>clientname</code> property set by the application holding the lock.
HeldClientApplName	varchar (30)	None	Value for the <code><clientapplname></code> property set by the application holding the lock.
HeldClientHostName	varchar (30)	None	Value for the <code><clienthostname></code> property set by the application holding the lock.
HeldTranName	varchar (255)	None	Name of the transaction that acquired the lock.
HeldCommand	varchar (30)	None	Category of process or command the process was executing when the process was blocked.
HeldFamilyID	int	None	spid of the parent process holding the lock.
HeldProcDBID	int	None	Unique identifier for the database where the stored procedure that caused the lock to be held resides, if applicable.
HeldProcDBName	varchar (30)	None	Name for the database where the stored procedure that caused the lock to be held resides, if applicable.
HeldProcedureName	varchar (255)	None	Name for the stored procedure that caused the lock to be held, if applicable.
HeldBatchID	int	None	Identifier of the SQL batch executed by the process holding the lock when the lock timeout occurred.
HeldContextID	int	None	Unique context identifier for the process holding the lock when it was blocked by another process (not when it acquired the lock).
HeldLineNumber	int	None	Line number in the SQL batch of the SQL statement holding the lock.
HeldStmtNumber	int	None	Statement number in the SQL batch of the SQL statement holding the lock.

Name	Datatype	Attributes	Description
HeldLockType	varchar (20)	None	Type of lock. One of: <ul style="list-style-type: none"> • Exclusive table • Shared table • Exclusive intent • Shared intent • Exclusive page • Shared page • Update page • Exclusive row • Shared row • Update row • Next key • Exclusive address • Shared address • Semaphore
HeldNumLocks	int	None	Number of locks currently held by holding spid.
HeldNumTimeoutsCausedByTransaction	int	None	Number of timeouts caused by this holding transaction.
HeldNumTimeoutsCausedByLock	int	None	Number of timeouts caused by this lock resource.
HeldSourceCodeID	varchar (30)	None	Location of the source code where the lock being held was acquired (internal use only).
WaitSPID	int	None	spid of the process waiting for the lock.
WaitKPID	int	None	kpid of the process waiting for the lock.
WaitUserName	varchar (30)	None	Name of the user for whom the lock is being requested.
WaitApplName	varchar (30)	None	Name of the application waiting for the lock.
WaitHostName	varchar (30)	None	Name of the host running the process waiting for the lock.
WaitClientName	varchar (30)	None	Value of the <clientname> property set by the application waiting for the lock.
WaitClientApplName	varchar (30)	None	Value of the <clientapplname> property set by the application waiting for the lock.

Name	Datatype	Attributes	Description
WaitClientHostName	varchar (30)	None	Value of the <clienthostname> property set by the application waiting for the lock.
WaitTranName	varchar (255)	None	Name of the transaction in which the lock was requested.
WaitCommand	varchar (30)	None	Category of process or command that the process was executing when it was blocked and then timed out.
WaitFamilyID	int	None	spid of the parent process waiting for the lock.
WaitProcDBID	int	None	Unique identifier for the database in which the stored procedure waiting for the lock resides, if applicable.
WaitProcDBName	varchar (255)	None	Name for the database where the stored procedure that is waiting for the lock resides, if applicable.
WaitProcedureName	varchar (255)	None	Name for the stored procedure waiting for the lock, if applicable.
WaitBatchID	int	None	Identifier of the SQL batch executed by the process waiting for the lock when the lock timeout occurred.
WaitContextID	int	None	Unique context identifier for the process waiting for the lock when it was blocked by another process.
WaitLineNumber	int	None	Line number of the SQL statement in the SQL batch waiting for the lock.
WaitStmtNumber	int	None	Line number in SQL batch waiting for the lock.

Name	Datatype	Attributes	Description
WaitLockType	varchar(30)	None	Type of lock. One of: <ul style="list-style-type: none"> • Exclusive table • Shared table • Exclusive intent • Shared intent • Exclusive page • Shared page • Update page • Exclusive row • Shared row • Update row • Next key • Exclusive address • Shared address • Semaphore
WaitNumTimeoutsCausedByTransaction	int	None	Number of timeouts caused by a waiting transaction.
PartitionID	int	None	ID of the partition.
WaitSourceCodeID	int	None	Location in the source code when the timeout occurred and the waiting lock request was made (for internal use only).
HeldProcedureID	int	None	Unique object identifier for the stored procedure that the blocking process was executing when the timeout occurred
WaitProcedureID	int	None	Unique object identifier for the stored procedure that is waiting for the lock, if applicable

3.41 monLogicalCluster

Applies to cluster environments only. Displays information about the logical clusters currently configured on the system.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
LCID	int	None	Logical cluster ID.
Attributes	int	None	Bitmask of logical cluster attributes.
ActiveConnections	int	None	Number of active connections using this logical cluster.
BaseInstances	tinyint	None	Number of instances configured as base instances for this logical cluster.
ActiveBaseInstances	tinyint	None	Number of base instances on which this logical cluster is currently active.
FailoverInstances	tinyint	None	Number of instances configured as failover instances for this logical cluster.
ActiveFailoverInstances	tinyint	None	Number of failover instances on which this logical cluster is currently active.
LCType	int	None	Type of logical cluster: application, alias, or login.
Name	varchar(30)	None	Logical cluster name.
State	varchar(20)	None	Current state. One of: <ul style="list-style-type: none"> • Online • Offline • Failed • Inactive • Time_wait
DownRoutingMode	varchar(20)	None	Down routing-mode setting. One of: <ul style="list-style-type: none"> • System • Open • Disconnect
FailoverMode	varchar(20)	None	Failover mode setting, instance or cluster.
FailoverRecovery	varchar(30)	None	Failover recovery diagnostic information for cluster lock manager, database recovery, and cluster membership service modules.
StartupMode	varchar(20)	None	Start-up mode setting, automatic or manual.

Name	Datatype	Attributes	Description
SystemView	varchar(20)	None	System view setting, instance or cluster.
Roles	varchar(20)	None	Comma-delimited list of special roles for this logical cluster. The "system" logical cluster always has the system role. The open logical cluster has the "open" role. If the system logical cluster also has the open role, the value for this column is <code>system, open</code> . Logical clusters without any special roles return a null value.
LoadProfile	varchar(30)	None	Load profile associated with this logical cluster.
ActionRelease	varchar(20)	None	The current action release mode for this logical cluster. Values are: <ul style="list-style-type: none"> Manual Automatic Manual indicates that the user must execute the action release command to release the actions for this cluster.
Gather	varchar(30)	None	Indicates whether this logical cluster is configured to automatically gather routable connections to this logical cluster. Values are: <ul style="list-style-type: none"> Manual Automatic

3.42 monLogicalClusterAction

Applies to cluster environments only. Shows all administrative actions against logical clusters from start-up until these actions are released.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
Handle	int	None	Unique handle used to cancel this action.

Name	Datatype	Attributes	Description
State	varchar (20)	None	State of the action: active, complete, releasing, or canceled.
LCID	int	None	Logical cluster ID to which this action applies.
LogicalClusterName	varchar (30)	None	Logical cluster name of this logical cluster (denormalized to reduce joins).
Action	varchar (15)	None	Action being performed. A combination of the command running and its scope. For example, <code>offline instance</code> or <code>failover cluster</code> .
FromInstances	varchar (96)	None	A comma-separated list of <code>from instances</code> for this command and action (instance being brought offline).
ToInstances	varchar (96)	None	A comma-separated list of <code>to instances</code> for this command and action (instances being brought online).
InstancesWaiting	int	None	Number of instances waiting to go offline (this is a count of <code>FromInstances</code> that are in the <code>time_wait</code> state).
WaitType	varchar (20)	None	Current wait state for this action. One of: <code>wait</code> , <code>until</code> , or <code>nowait</code> .
StartTime	datetime	None	Date and time the command was issued.
Deadline	datetime	None	Date and time the command must be finished (based on the time value supplied to the <code>wait</code> or <code>until</code> options).
CompleteTime	datetime	None	Date and time the command and action completed (when <code>InstancesWaiting</code> is zero and the action went from active to the complete state). Returns NULL for incomplete actions.
ConnectionsRemaining	int	None	Number of connections remaining to move as a result of this command.
NonMigConnections	int	None	Number of connections to be terminated because they do not support the migration protocol.
NonHAConnections	int	None	Number of connections that do not support the high availability failover protocol. These connections are disconnected and cannot fail over when the command finishes.

3.43 monLogicalClusterInstance

Applies to cluster environments only. Displays information about the many-to-many relationship between instances and logical clusters.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
LCID	int	None	Logical cluster ID
LogicalClusterName	varchar (30)	None	Logical cluster name
InstanceID	tinyint	None	ID of the instance within the cluster
InstanceName	varchar (30)	None	Instance name
Type	varchar (20)	None	Instance type
FailoverGroup	tinyint	None	Failover group to which this instance is a member (failover instances only)
State	varchar (20)	None	State of this instance with respect to the logical cluster
ActiveConnections	int	None	Number of active connections for this logical cluster on this instance
NonMigConnections	int	None	Number of active connections that do not support the connection migration protocol
NonHAConnections	int	None	Number of active connections that do not support the high availability failover protocol
LoadScore	real	None	Workload score for this instance using the load profile associated with its logical cluster

3.44 monLogicalClusterRoute

Applies to cluster environments only. Displays information about the configured routes (application, login, and alias bindings). You need not have the `mon_role` role to query this monitor table.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
LCID	int	None	Logical cluster ID
LogicalClusterName	varchar (30)	None	Logical cluster name
RouteType	varchar (20)	None	Route type. One of: application, login, or alias
RouteKey	varchar (30)	None	Application, login, or alias name associated with this route.

3.45 monMemoryUsage

The `monMemoryUsage` monitoring table provides information about server and kernel memory pools, including metrics about their sizes, usage patterns, and availability.

Columns

Name	Datatype	Attribute	Description
Flags	int	None	Status flags that describe the memory pool.
ConfigNum	int	None	Configuration number for the primary configuration option controlling the size of the memory pool.
TotalSize	bigint	None	Total size, in bytes, of the memory pool.
UsedSize	bigint	None	Currently used size, in bytes, of the memory pool.

Name	Datatype	Attribute	Description
FreeSize	bigint	None	Amount of free memory, in bytes, in the pool.
NumAllocs	bigint	Counter	Total number of allocations requested.
NumFrees	bigint	Counter	Total number of frees performed. "Frees" are the opposite of allocations. Adaptive Server allocates memory fragments, and the number of allocations is tracked by NumAllocs. When the task is finished, the memory fragment is freed (that is, returned to the memory pool). NumFrees tracks the total number of these free operations
NumSleeps	bigint	Counter	Total number of sleeps encountered while allocating memory fragments.
PoolOwnerKPID	int	None	Kernel process ID (KPID) of task that owns this fragment of the memory pool.
MemoryPoolName	varchar (32)	None	Name of the memory pool.
PoolType	varchar (30)	None	Type of memory pool. One of: <ul style="list-style-type: none"> • Block • Bucket • Object • Fragment • Stack
ConfigOption	varchar (255)	None	Name of the primary configuration option controlling the size of the memory pool.
NumSearches	bigint	Counter	Total number of free fragments examined before satisfying memory allocation requests from this memory pool.
NumRetries	bigint	Counter	Number of retries performed for all free fragments.
ItemSize	int	None	Size of an individual item (applies to object pool).
MinNumItems	int	None	Minimum number of items in this pool.
MaxNumItems	int	None	Maximum number of items in this pool.
NumUsedItems	int	None	Number of used items in this pool.
NumItemsUsedHWM	int	None	High-water mark for the number of items used in this pool.
MinUsedItemSize	bigint	None	Size, in bytes, of smallest used item.
AvgUsedItemSize	bigint	None	Average size, in bytes, of used items.
MaxUsedItemSize	bigint	None	Size, in bytes, of largest used item.
NumUsedItemsMinSize	int	None	Number of minimum-sized used items in this pool.

Name	Datatype	Attribute	Description
NumUsedItemsMaxSize	int	None	Number of maximum-sized used items in this pool.
NumFreeItems	int	None	Number of free items in this pool.
MinFreeItemSize	bigint	None	Size, in bytes, of smallest item that is free.
AvgFreeItemSize	bigint	None	Average size, in bytes, of free items.
MaxFreeItemSize	bigint	None	Size, in bytes, of largest item that is free.
NumFreeItemsMinSize	int	None	Number of minimum-sized free items in this pool.
NumFreeItemsMaxSize	int	None	Number of maximum-sized free items in this pool.
NumBlocks	int	None	Number of blocks of memory used for this pool.
MemSize1	int	None	Memory pool specific request size 1, in bytes.
NumUsedItemsSize1	int	None	Number of used items in this pool of size MemSize1.
NumFreeItemsSize1	int	None	Number of free items in this pool of size MemSize1.
MemSize2	int	None	Memory pool specific request size 2, in bytes.
NumUsedItemsSize2	int	None	Number of used items in this pool of size MemSize2.
NumFreeItemsSize2	int	None	Number of free items in this pool of size MemSize2.

Not all output from all `monMemoryUsage` columns applies, or is relevant to, all memory pools, and depending on the type of memory pool, you may need to select the relevant columns. Typically, columns return a value of NULL if they do not apply to a specific memory pool.

These columns report the metrics for used versus free fragments for memory pools of type `Fragment`:

- NumUsedItems
- NumItemsUsedHWM
- MinUsedItemSize
- AvgUsedItemSize
- MaxUsedItemSize
- NumUsedItemsMinSize
- NumUsedItemsMaxSize
- NumFreeItems
- MinFreeItemSize
- AvgFreeItemSize
- MaxFreeItemSize
- NumFreeItemsMinSize
- NumFreeItemsMaxSize

The following columns are set to NULL for all fragment memory pools:

- NumBlocks
- NumUsedItems
- NumFreeItems
- MinUsedItemSize
- MaxUsedItemSize
- NumUsedItemsMinSize
- NumUsedItemsMaxSize
- AvgUsedItemSize
- NumUsedItemsSize1
- NumUsedItemsSize2
- MinFreeItemSize
- MaxFreeItemSize
- NumFreeItemsMinSize
- NumFreeItemsMaxSize
- AvgFreeItemSize
- NumFreeItemsSize1
- NumFreeItemsSize2

This example lists memory pools in the server, along with the primary configuration option affecting the size of the memory pool:

```
select PoolType = convert(varchar(10), PoolType),
MemoryPoolName = convert(varchar(30), MemoryPoolName),
ConfigOption = convert(varchar(30), ConfigOption)
from monMemoryUsage order by 1, 2
PoolType  MemoryPoolName  ConfigOption
-----  -
Block     Compression           compression memory size
Block     Global Block Pool     NULL
Block     Kernel Resource Memory kernel resource memory
Block     Proc Cache Header     procedure cache size
Block     Pss Heap Memory       heap memory per user
Block     RTMS Block Heap       messaging memory
Fragment  CPINFO memory pool    compression info pool size
Fragment  Column Default Pool   column default cache size
Fragment  Data Cache Frag       NULL
Fragment  Data Change Frag      NULL
Fragment  Data Transfer Utility transfer utility memory size
[...]
```

This example lists the common metrics that are applicable to most memory pools:

```
select MemoryPoolName = convert(varchar(30), MemoryPoolName),
TotalSize, UsedSize, FreeSize, NumAllocs, NumFrees
from monMemoryUsage order by 1
```

This example lists the metrics that apply to fragment memory pools:

```
select MemoryPoolName = convert(varchar(30), MemoryPoolName),
NumUsedItems, NumItemsUsedHWM, MinUsedItemSize,
AvgUsedItemSize, MaxUsedItemSize
from monMemoryUsage
where PoolType = "Fragment"
MemoryPoolName  NumUsedItems  NumItemsUsedHWM  MinUsedItemSize
AvgUsedItemSize  MaxUsedItemSize
```

Pss Frag Pool	0	983048	0	0	16960
Pss Frag Pool	0	1376267	0	0	17040
Pss Frag Pool	0	2293778	0	0	16960

3.46 monNetworkIO

Returns network I/O statistics for all communication between the SAP ASE server and client connections.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster
PacketsSent	int	Counter, reset	Number of packets sent
PacketsReceived	int	Counter, reset	Number of packets received
BytesSent	int	Counter, reset	Number of bytes sent
BytesReceived	int	Counter, reset	Number of bytes received
PacketsSentMln	int	Counter, reset	Number of packets, in millions, sent by the server
PacketsReceivedMln	int	Counter, reset	Number of packets, in millions, received by the server
BytesSentMB	int	Counter, reset	Number of bytes, in megabytes, sent by the server
BytesReceivedMB	int	Counter, reset	Number of bytes, in megabytes, received by the server

3.47 monNVCache

Stores statistics relating to SAP ASE named non-volatile caches, including cache definition, current status, and usage of NV cache(s):

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
CacheID	int	None	Unique identifier for the NV cache
CacheSize	int	None	Total size of NV cache, in kilobytes
Overhead	int	None	NV cache overhead
CachePartitions	smallint	None	Number of partitions currently configured for the cache
CacheSearches	int	None	Cache searches directed to the cache
CacheWrites	int	Counter, reset	Number of buffers written from the NV cache to disk
CacheReads	int	Counter, reset	Number of buffers read into the cache from disk
HDDWrites	int	Counter, reset	Number of buffers written from the device disk to the NV cache.
CacheName	varchar (30)	None	Name of cache
Status	varchar (30)	None	Status of NV cache. One of: <ul style="list-style-type: none">• Active• Pending/Active• Pending/Delete• Update Cache• Cache Create• Cache Delete

3.48 monOpenDatabases

Provides state and statistical information pertaining to databases that are currently in the server's metadata cache.

If the value of `number of open databases` is too low, the SAP ASE server may flush database descriptors from the metadata cache. If this occurs, the SAP ASE server loses the database statistics, but the statistics are reinitialized the next time the database descriptor is installed in the metadata cache.

Enable the `enable monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
BackupInProgress	int	None	Specifies whether a backup is currently in progress for the database
LastBackupFailed	int	None	Specifies whether the last backup of the database failed
TransactionLogFull	int	None	Specifies whether the database transaction log is full
AppendLogRequests	int	Counter	Number of semaphore requests when attempting to append to the database transaction log
AppendLogWaits	int	Counter	Number of times a task had to wait for the append log semaphore to be granted
DBName	varchar(30)	None	Name of the database
BackupStartTime	datetime	None	Date the last full database backup started
SuspendedProcesses	int	None	Number of processes currently suspended due to the database transaction log being full
QuiesceTag	varchar(30)	None	Tag used in the <code>quiesce database</code> command for this database if the database is in a quiesced state

Name	Datatype	Attributes	Description
LastCheckpointTime	datetime	None	Date and time checkpoint last ran for this database
LastTranLogDumpTime	datetime	None	Date and time of this database's most recently successful transaction log dump. The time is not updated if the transaction is dumped using the <code>truncate_only</code> or <code>no_log</code> .
PRUpdateCount	int	Counter	Number of updates to precomputed result sets caused by <code>insert</code> , <code>update</code> , or <code>delete</code> s to the base table.
PRSelectCount	int	Counter	The number of times the optimizer selected precomputed result sets in this database when generating a query plan.
PRRewriteCount	int	Counter	The number of times the optimizer determined the precomputed result sets in this database were valid when generating the query plan.

3.49 monOpenObjectActivity

Provides statistics for all open tables and indexes.

Enable the `enable monitoring`, `per object statistics active`, and `object lockwait timing` configuration parameters for this monitoring table to collect data.

Note

The value of `OptSelectCount` may be less than that of `UsedCount` since you can use the plan for a stored procedure or trigger multiple times. Also, because the SAP ASE server may decide not to execute certain portions of a query plan during execution, `UsedCount` may be less than `OptSelectCount`.

Columns

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database.
ObjectID	int	None	Unique identifier for the object.
IndexID	int	None	Unique identifier for the index..

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) Unique identifier for an instance.
DBName	varchar (30)	None	Name of the database in which the object resides
ObjectName	varchar (30)	None	Name of the object.
LogicalReads	int	Counter	Total number of times a buffer for this object has been retrieved from a buffer cache without requiring a read from disk.
PhysicalReads	int	Counter	Number of buffers read from disk.
APFReads	int	Counter	Number of APF buffers read from disk.
PagesRead	int	Counter	Total number of pages read.
PhysicalWrites	int	Counter	Total number of buffers written to disk.
PagesWritten	int	Counter	Total number of pages written to disk.
RowsInserted	int	Counter	Number of rows inserted.
RowsDeleted	int	Counter	Number of rows deleted.
RowsUpdated	int	Counter	Number of updates.
Operations	int	Counter	Number of times the object was accessed.
LockRequests	int	Counter	Number of requests for a lock on the object.
LockWaits	int	Counter	Number of times a task waited for an object lock.
OptSelectCount	int	Counter	Number of times the optimizer selected this index to be used in a query plan.
LastOptSelectDate	datetime	None	Last date the index was selected for a plan during compilation.
UsedCount	int	Counter	Number of times the object was used in a plan during execution.
LastUsedDate	datetime	None	Last date the index was used in a plan during execution.
HkgcRequests	int	Counter	Total number of events queued for an object. A large value implies the system is generating large amounts of garbage for the specified object.

Name	Datatype	Attributes	Description
HkgcPending	int	Counter	The number of pending events for an object. A large value implies that a lot of garbage is yet to be collected, although the housekeeper will clean it up. If you restart the SAP ASE server, all entries in the housekeeper queue are lost, and the garbage from those pages is not collected when you restart the SAP ASE server.
HkgcOverflows	int	Counter	The number of overflow object events. A large value implies the housekeeper queues are filling up. Generated garbage will not then be cleaned up because the housekeeper cannot schedule the job.
PhysicalLocks	int	Counter	(Cluster environments only) Number of physical locks requested per object.
PhysicalLocksRetained	int	Counter	(Cluster environments only) Number of physical locks retained. Use to identify the lock hit ratio for each object. Good hit ratios imply balanced partitioning for this object.
PhysicalLocksRetainedWaited	int4	Counter	(Cluster environments only) Number of physical lock requests waiting before a lock is retained.
PhysicalLocksDeadlocks	int	Counter	(Cluster environments only) Number of times a requested physical lock returned a deadlock. The <code>Cluster Physical Locks</code> subsection of <code>sp_sysmon</code> uses this counter to report deadlocks while acquiring physical locks for each object.
PhysicalLocksWaited	int	Counter	(Cluster environments only) Number of times an instance waited for a physical lock request.
PhysicalLocksPageTransfer	int	Counter	(Cluster environments only) Number of page transfers that occurred when an instance requested a physical lock. The <code>Cluster Physical Locks</code> subsection of <code>sp_sysmon</code> uses this counter to report the node-to-node transfer and physical-lock acquisition as a node affinity ratio for this object.
TransferReqWaited	int4	Counter	(Cluster environments only) Number of times physical lock requests waiting before receiving page transfers.
AvgPhysicalLockWaitTime	int4	Counter	(Cluster environments only) Average amount of time clients spend before the physical lock is granted.
MaxPhysicalLockWaitTime	real	Counter	(Cluster environments only) Maximum amount of time this object waited for before a physical lock was granted.

Name	Datatype	Attributes	Description
AvgTransferReqWaitTime	int4	Counter	(Cluster environments only) Average amount of time physical lock requests wait before receiving page transfers.
MaxTransferReqWaitTime	real	Counter	(Cluster environments only) Maximum amount of time physical lock requests waited to receive page transfers.
TotalServiceRequests	int4	Counter	(Cluster environments only) Number of physical lock requests serviced by the cluster cache manager of an instance.
PhysicalLocksDowngraded	int4	Counter	(Cluster environments only) Number of physical lock downgrade requests serviced by the cluster cache manager of an instance.
PagesTransferred	int4	Counter	(Cluster environments only) Number of pages transferred at an instance by the cluster cache manager.
ClusterPageWrites	int4	Counter	(Cluster environments only) Number of pages written to disk by the cluster cache manager of an instance.
AvgServiceTime	int4	None	(Cluster environments only) Average amount of service time spent by the cluster cache manager of an instance.
MaxServiceTime	real	None	(Cluster environments only) Maximum amount of service time spent by the cluster cache manager of an instance.
AvgQueueWaitTime	real	None	(Cluster environment only) Average amount of time, in milliseconds, spent waiting for the SAP ASE server to complete buffer transfers for this object.
MaxQueueWaitTime	real	None	(Cluster environment only) Maximum amount of time, in milliseconds, spent waiting for the SAP ASE server to complete a buffer transfer for this object .
AvgTimeWaitedOnLocalUsers	int4	None	(Cluster environments only) Average amount of time, in milliseconds, an instance's cluster cache manager waited because of page use by users on this instance.
MaxTimeWaitedOnLocalUsers	real	None	(Cluster environments only) Maximum amount of time, in milliseconds, an instance's cluster cache manager waited because of page use by users on this instance.
AvgTransferSendWaitTime	int4	None	(Cluster environments only) Average amount of time an instance's cluster cache manager spends for page transfer.
MaxTransferSendWaitTime	real	None	(Cluster environments only) Maximum amount of time an instance's cluster cache manager waited for a page transfer to complete.

Name	Datatype	Attributes	Description
AvgIOServiceTime	int4	None	(Cluster environments only) Average amount of time used by an instance's cluster cache manager for page transfer.
MaxIOServiceTime	real	None	(Cluster environments only) Maximum amount of time the Cluster Cache Manager took to write pages to disk.
AvgDowngradeServiceTime	int4	None	(Cluster environments only) Average amount of time the cluster cache manager uses to downgrade physical locks.
MaxDowngradeServiceTime	real	None	(Cluster environments only) Maximum time a task spent waiting for the physical lock to be downgraded on a page.
SharedLockWaitTime	int	Counter, reset	The total amount of time, in milliseconds, that all tasks spent waiting for a shared lock.
ExclusiveLockWaitTime	int	Counter, reset	The total amount of time, in milliseconds, that all tasks spent waiting for an exclusive lock.
UpdateLockWaitTime	int	Counter, reset	The total amount of time, in milliseconds, that all tasks spent waiting for an update lock.
ObjectCacheDate	datetime	None	Indicates the date and time when the object was added to the cache.
HkgcRequestsDcomp	int	Counter	Total number of data pages of the partition that were queued for page compression.
HkgcPendingDcomp	int	Counter	Number of data pages of the partition that are still pending for page compression.
HkgcOverflowsDcomp	int	Counter	Total number of pages that could not be compressed because the housekeeper queue was full.
IOSize1Page	int	Counter	Number of IO operations performed for each IO that is one page in size.
IOSize2Pages	int	Counter	Number of IO operations performed for each IO that is 2 pages in size.
IOSize4Pages	int	Counter	Number of IO operations performed for each IO that is 4 pages in size.
IOSize8Pages	int	Counter	Number of IO operations performed for each IO that is 8 pages in size.
PRSSelectCount	int	Counter	The number of times the precomputed result set was used in a query.

Name	Datatype	Attributes	Description
LastPRSSelectDate	datetime	None	Date for the last time the precomputed result set was used in a query.
PRSRewriteCount	int	Counter	Number of times the optimizer determined that the precomputed result set was valid for use in a query. the optimizer may not have used the precomputed result set because it found a better choice.
LastPRSRewriteDate	datetime	None	Date for the last time the optimizer determined that the precomputed result set was valid for use in a query.
Scans	int	Counter	Number of scans performed on this object.
LastScanDate	datetime	None	Date of the last scan on this object
Updates	int	Counter	Number of updates performed on this object.
LastUpdateDate	datetime	None	Date of the last update on this object
Inserts	int	None	Number of inserts performed on this object.
LastInsertDate	datetime	None	Date of the last insert on this object
Deletes	int	Counter	Number of deletes performed on this object.
LastDeleteDate	datetime	None	Date of the last delete on this object
NumLevel0Waiters	int	Counter	Number of times a Level0 Scan start waited because of a utility's wait request.
AvgLevel0WaitTime	real	Counter	Average time, in milliseconds, Adaptive Server waited for Level0 access.

3.50 monOpenPartitionActivity

Provides information about the use of each open partition on the server.

Enable the `enable monitoring` and `per object statistics active` configuration parameters for this monitoring table to collect data.

Note

Because you can use the plan for a stored procedure or trigger multiple times, the value of the `OptSelectCount` column may be less than the value of `UsedCount`. In addition, because the SAP ASE server may decide not to execute certain portions of a query plan during execution, the `UsedCount` may be less than the `OptSelectCount`.

Columns

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database.
ObjectID	int	None	Unique identifier for the object.
IndexID	int	None	Unique identifier for the index.
PartitionID	int	None	Unique identifier for the partition.
InstanceID	int	None	ID of an instance in a shared-disk cluster.
DBName	varchar (30)	None	Name of the database in which the object resides.
ObjectName	varchar (30)	None	Name of the object.
PartitionName	varchar (30)	None	Name of the partition.
LogicalReads64	bigint	Counter	Total number of buffers read.
PhysicalReads64	bigint	Counter	Number of buffers read from disk.
APFReads	int	Counter	Number of asynchronous prefetch (APF) buffers read.
PagesRead	int	Counter	Total number of pages read.
PhysicalWrites64	bigint	Counter	Total number of buffers written to disk.
PagesWritten	int	Counter	Total number of pages written to disk.
RowsInserted	int	Counter	Number of rows inserted.
RowsDeleted	int	Counter	Number of rows deleted.
RowsUpdated	int	Counter	Number of updates.
OptSelectCount	int	Counter	Number of times object was selected for plan during compilation.
LastOptSelectDate	datetime	None	Last date the index was selected for plan during compilation.
UsedCount	int	Counter	Number of times the object was used in a plan during execution.

Name	Datatype	Attributes	Description
LastUsedDate	datetime	None	Last date the index was used in a plan during execution.
HkgcRequests	int	Counter	Total number of events queued for a partition. A large value implies the system is generating large amounts of garbage for the specified partition.
HkgcPending	int	Counter	The number of pending events for a partition. A large value implies that a lot of garbage is yet to be collected, although the housekeeper will clean it up. If you restart the SAP ASE server, all entries in the housekeeper queue are lost, and the garbage from those pages is not collected when you restart the SAP ASE server.
HkgcOverflows	int	Counter	The number of overflow partition events. A large value implies the housekeeper queues are filling up. Generated garbage will not then be cleaned up because the housekeeper cannot schedule the job.
PhysicalLocks	int	Counter	(Cluster environments only) Number of physical locks requested per object.
PhysicalLocksRetained	int	Counter	Number of physical locks retained. Use to identify the lock hit ratio for each object. Good hit ratios imply balanced partitioning for this object.
PhysicalLocksRetainWait	int	Counter	(Cluster environments only) Number of physical lock requests waiting before a lock is retained.
PhysicalLocksDeadlocks	int	Counter	(Cluster environments only) Number of times a physical lock requested returned a deadlock. The <code>Cluster Physical Locks</code> subsection of <code>sp_sysmon</code> uses this counter to report deadlocks while acquiring physical locks for each object.
PhysicalLocksWaited	int	Counter	(Cluster environments only) Number of times an instance waited for a physical lock request.
PhysicalLocksPageTransfer	int	Counter	(Cluster environments only) Number of page transfers that occurred when an instance requested a physical lock. The <code>Cluster Physical Locks</code> subsection of <code>sp_sysmon</code> uses this counter to report the node-to-node transfer and physical-lock acquisition as a node affinity ratio for this object.
TransferReqWaited	int	Counter	(Cluster environments only) Number of times physical lock requests waiting before receiving page transfers.

Name	Datatype	Attributes	Description
MaxPhysicalLockWaitTime	real	Counter	(Cluster environments only) Maximum amount of time this object waited for before a physical lock was granted.
AvgPhysicalLockWaitTime	real	Counter	(Cluster environments only) Average amount of time clients spend before the physical lock is granted.
MaxTransferReqWaitTime	real	Counter	(Cluster environments only) Maximum amount of time physical lock requests waited to receive page transfers.
AvgTransferReqWaitTime	real	Counter	(Cluster environments only) Average amount of time physical lock requests wait before receiving page transfers.
TotalServiceRequests	int	Counter	(Cluster environments only) Number of physical lock requests serviced by the cluster cache manager of an instance.
PhysicalLocksDowngraded	int	Counter	(Cluster environments only) Number of physical lock downgrade requests serviced by the cluster cache manager of an instance.
PagesTransferred	int	Counter	(Cluster environments only) Number of pages transferred at an instance by the cluster cache manager.
ClusterPageWrites	int	Counter	(Cluster environments only) Number of pages written to disk by the cluster cache manager of an instance.
AvgServiceTime	real	None	(Cluster environments only) Average amount of time spent by the cluster cache manager of an instance.
MaxServiceTime	real	None	(Cluster environments only) Maximum amount of time spent by the cluster cache manager of an instance.
AvgQueueWaitTime	real	None	(Cluster environment only) Average amount of time, in milliseconds, spent waiting for the SAP ASE server to complete buffer transfers for this object.
MaxQueueWaitTime	real	None	(Cluster environments only) Maximum amount of time, in milliseconds, spent waiting for the SAP ASE server to complete a buffer transfer for this object.
AvgTimeWaitedOnLocalUsers	real	None	(Cluster environments only) Average amount of service time an instance's cluster cache manager waits because of page use by users on this instance.

Name	Datatype	Attributes	Description
MaxTimeWaitedOnLocalUsers	real	None	(Cluster environments only) Maximum amount of time, in milliseconds, an instance's cluster cache manager waited for a physical lock because the object in question was in use by another process.
AvgTransferSendWaitTime	real	None	(Cluster environments only) Average amount of service time an instance's cluster cache manager spends for page transfer.
MaxTransferSendWaitTime	real	None	(Cluster environments only) Maximum amount of time the Cluster Cache Manager for an instance waited for page transfer to complete
AvgIOServiceTime	real	None	(Cluster environments only) Average amount of service time used by an instance's cluster cache manager for page transfer.
MaxIOServiceTime	real	None	(Cluster environments only) Maximum amount of time the Cluster Cache Manager took to write pages to disk.
AvgDowngradeServiceTime	real	None	(Cluster environments only) Average amount of time the cluster cache manager uses to downgrade physical locks.
MaxDowngradeServiceTime	real	None	(Cluster environments only) Maximum time a task spent waiting for the physical lock to be downgraded on a page.
ObjectCacheDate	datetime	None	Indicates the date and time when the object was added to the cache.
HkgcRequestsDcomp	int	Counter	Total number of data pages of the partition that were queued for page compression
HkgcPendingDcomp	int	Counter	Number of data pages of the partition that are still pending for page compression
HkgcOverflowsDcomp	int	Counter	Total number of pages that could not be compressed because the housekeeper queue was full.
IOSize1Page	int	Counter	Number of IO operations performed for each IO that is one page in size
IOSize2Pages	int	Counter	Number of IO operations performed for each IO that is 2 pages in size
IOSize4Pages	int	Counter	Number of IO operations performed for each IO that is 4 pages in size

Name	Datatype	Attributes	Description
IOSize8Pages	int	Counter	Number of IO operations performed for each IO that is 8 pages in size

3.51 monPCIBridge

Contains information about the Java PCI Bridge. This table provides information about the Java environment. You do not need to enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
Status	char(10)	None	Current status of the PCI Bridge. Values are: <ul style="list-style-type: none"> ACTIVE DOWN
ConfiguredSlots	int	None	Number of configured slots. Set using <code>max_pci_slots</code> configuration parameter.
ActiveSlots	int	None	Number of currently active slots.
ConfiguredPCIMemoryKB	int	None	Total memory configured for the PCI Bridge using the <code>pci_memory</code> configuration parameter.
UsedPCIMemoryKB	int	None	Total memory currently used by the PCI bridge and its components.

3.52 monPCIEngine

Displays engine information for the PCI Bridge and its plug-ins. This table provides information about the Java environment.

You do not need to enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
Engine	int	None	Engine number
Status	char(10)	None	Status of the plug-in on the engine. Values are: <ul style="list-style-type: none">ACTIVEINIT
PLBStatus	char(10)	None	Status of the PCI Launcher Boss. Values are: <ul style="list-style-type: none">ACTIVEDOWN
NumberOfActiveThreads	int	None	Number of active threads currently under control of the PCI Launcher Boss.
PLBRequests	int	None	Number of requests for the PCI Launcher Boss to execute a function for a native thread.
PLBwakeUpRequests	int	None	Number of times the PCI Launcher Boss received a wake-up to execute work for a native thread.

3.53 monPCISlots

Contains information about the plug-in bound to each slot in the PCI Bridge. This table provides information about the Java environment.

You do not need to enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
Slot	int	None	Number of active slot: Values are 1 – 31.

Name	Datatype	Attributes	Description
Status	char(10)	None	Status of the slot. Values are: <ul style="list-style-type: none"> INIT IN USE STOPPED
Modulename	varchar(30)	None	Logical module name bound to the current slot.
engine	int	None	Engine associated with the slot.

3.54 monPCM

Applies to cluster environments only. Tracks the peer coordination module (PCM) client activities in the cluster (for example, the number of fragment that were sent and received), and contains a row for each PCM client.

You do not need to enable any configuration parameters for this monitoring table to collect data.

Columns

Column name	Type	Attributes	Description
InstanceId	tinyint	None	Instance ID for which the information is collected
Sent	int	None	Number of messages sent per module
Fragments_sent	int	None	Number of fragments sent per module
Fragments_received	int	None	Number of fragments received per module
Multicast	int	None	Number of multicast requests
Received	int	None	Number of messages received per module
Reply	int	None	Number of replies received per module
Unicast	int	None	Number of unicast messages sent per module
Mulicat	int	None	Number of multicast messages sent per module
Sync	int	None	Number of synchronous messages sent per module

Column name	Type	Attributes	Description
Async	int	None	Number of asynchronous messages sent per module
MinBytes	int	None	Minimum number of bytes transferred per message
AvgBytes	int	None	Average number of bytes transferred per message
MaxBytes	int	None	Maximum number of bytes transferred per message
MinDialog	int	None	Minimum length of the dialogs
AvgDialog	int	None	Average length of the dialogues
MaxDialog	int	None	Maximum length of the dialogues
Dialog	int	None	Number of the dialogues
MinTimeSyncApi	real	None	Minimum time spent in PCM API in synchronous mode in the PCM layer per module
AvgTimeSyncApi	real	None	Average time spent in PCM API in synchronous mode in the PCM layer per module
MaxTimeSyncApi	real	None	Maximum time spent in PCM API in synchronous mode in the PCM layer per module
MinTimeAsyncApi	real	None	Minimum time spent in PCM API in asynchronous mode in the PCM layer per module
AvgTimeAsyncApi	float	None	Average time spent in PCM API in asynchronous mode in the PCM layer per module
MaxTimeAsyncApi	float	None	Maximum time spent in PCM API in asynchronous mode in the PCM layer per module
MinTimeCIPCMsgAlloc	real	None	Minimum time spent in cipcmsg allocations in the PCM layer per module
AvgTimeCIPCMsgAlloc	real	None	Average time spent in cipcmsg allocations in the PCM layer per module
MaxTimeCIPCMsgAlloc	real	None	Maximum time spent in cipcmsg allocations in the PCM layer per module
MinTimeCIPCSCB	real	None	Minimum time spent in cipc_sendcb per module
AvgTimeCIPCSCB	real	None	Average time spent in cipc_sendcb per module
MaxTimeCIPCSCB	real	None	Maximum time spent in cipc_sendcb per module

Column name	Type	Attributes	Description
MinTimeCIPCUnicast smsg	float	None	Minimum time spent in CIPC while sending the unicasts message per module
AvgTimeCIPCUnicast smsg	real	None	Average time spent in CIPC while sending the unicasts message per module
MaxTimeCIPCUnicast smsg	real	None	Maximum time spent in CIPC while sending the unicasts message per module
MinTimeCIPCMultica stsmsg	real	None	Minimum time spent in CIPC while sending the multicasts message per module
AvgTimeCIPCMultica stsmsg	real	None	Average time spent in CIPC while sending the multicasts message per module
MaxTimeCIPCMultica stsmsg	real	None	Maximum time spent in CIPC while sending the multicasts message per module
MinTimeClientRecvC B	real	None	Minimum time spent in client receive callback in the PCM layer per module
AvgTimeClientRecvC B	real	None	Average time spent in client receive callback in the PCM layer per module
MaxTimeClientRecvC B	real	None	Maximum time spent in client receive callback in the PCM layer per module
ModuleName	varchar (3 0)	None	Name of the PCM client

3.55 monProcedureCache

Returns statistics relating to SAP ASE procedure cache.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
Requests	int	Counter, reset	Number of stored procedures requested
Loads	int	Counter, reset	Number of stored procedures loaded into cache
Writes	int	Counter, reset	Number of times a procedure was normalized and the tree written back to sysprocedures
Stalls	int	Counter, reset	Number of times a process had to wait for a free procedure cache buffer when installing a stored procedure into cache
Instance ID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.

3.56 monProcedureCacheMemoryUsage

Includes one row for each procedure cache allocator. An allocator is identified by an allocator ID, which is internal to SAP ASE.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
AllocatorID	int	Counter, reset	Allocator ID
ModuleID	int	Counter, reset	Module ID (internal to SAP ASE)
Active	int	Counter, reset	Number of memory pages (2KB) currently allocated to this allocator
HWM	int	Counter, reset	Maximum number of memory pages allocated since the server was started

Name	Datatype	Attributes	Description
ChunkHWM	int	Counter, re-set	Largest number of contiguous memory pages allocated since the server was started
AllocatorName	varchar(30)	None	Name of the allocator
NumReuseCaused	int	Counter, re-set	Number of times this allocator has caused replacement

3.57 monProcedureCacheModuleUsage

Includes one row for each module that allocates memory from procedure cache. A module, which is identified with a module ID, is a functional area classification internal to SAP ASE procedure cache management.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
ModuleID	int	Counter, re-set	A module ID
Active	int	Counter, re-set	Number of memory pages (2KB) currently allocated to this module
HWM	int	Counter, re-set	The maximum number of memory pages allocated since the server was started
NumPagesReused	int	Counter, re-set	Number of pages allocated to this module
ModuleName	varchar(30)	None	Name of the module

3.58 monProcess

Provides detailed statistics about processes that are currently executing or waiting.

Enable the `enable_monitoring` and `wait_event_timing` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
ServerUserID	int	None	Server user ID (SUID) of the user associated with this process.
OrigServerUserID	int	None	Server user identifier prior to executing <code>set proxy</code>
BatchID	int	None	Unique identifier for the SQL batch containing the executing statement
ContextID	int	None	A unique identifier generated each time an executing query causes a stored procedure, trigger, execute immediate, deferred compilation, or other compiled object execution to occur
LineNumber	int	None	Line number of the current statement within the SQL batch
SecondsConnected	int	None	Number of seconds since this connection was established
DBID	int	None	Unique identifier for the database used by the process
EngineNumber	smallint	None	Unique identifier of the engine on which the process is executing
Priority	int	None	Priority at which the process is executing
FamilyID	int	None	spid of the parent process, if this is a worker process
Login	varchar(30)	None	Login user name

Name	Datatype	Attributes	Description
Application	varchar(30)	None	Application name. May be blank if the application did not set a name in its login structure.
Command	varchar(30)	None	Category of process or command the process is currently executing
NumChildren	int	None	Number of child processes, if executing a parallel query
SecondsWaiting	int	None	Amount of time, in seconds, the process has been waiting, if the process is currently blocked by a lock held by another process.
WaitEventID	smallint	None	Unique identifier for the event for which the process is waiting, if the process is currently in a wait state.
BlockingSPID	int	None	Session process identifier of the process holding the lock this process requested, if waiting for a lock
BlockingXLOID	int	None	Unique lock identifier for the lock that this process has requested, if waiting for a lock
DBName	varchar(30)	None	Name of the database the process is currently using
EngineGroupName	varchar(30)	None	Engine group for the process
ExecutionClass	varchar(30)	None	Execution class for the process
MasterTransactionID	varchar(255)	None	Name of the transaction the process has open
HostName	varchar(30)	None	Name of the host machine on which the application that started the process is running.
ClientName	varchar(30)	None	Value of the <clientname> property set by the application.
ClientHostName	varchar(30)	None	Value of the <clienthostname> property set by the application.
ClientApplName	varchar(30)	None	Value of the <clientapplname> property set by the application.
ClientDriverVersion	varchar16	None	Version of the connectivity driver used by the client program

3.59 monProcessActivity

Provides detailed statistics about process activity.

Enable the `enable_monitoring` and `wait_event_timing` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	smallint	None	Session process identifier.
InstanceID	int	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier.
ServerUserID	int	None	Server user identifier (SUID) of the user running this process. The value in <code>ServerUserID</code> matches the <code>syslogins.suid</code> column. Use the <code>suser_name</code> function to obtain the corresponding name.
OrigServerUserID	int	None	Server user identifier prior to executing <code>set_proxy</code>
CPUTime	int	Counter	CPU time (in milliseconds) used by the process.
WaitTime	int	Counter	Time (in milliseconds) the process spent waiting.
PhysicalReads	bigint	Counter	Number of buffers read from disk.
LogicalReads	bigint	Counter	Number of buffers read from cache.
PagesRead	bigint	Counter	Number of pages read.
PhysicalWrites	bigint	Counter	Number of buffers written to disk.
PagesWritten	bigint	Counter	Number of pages written.
MemUsageKB	int	None	Amount of memory (in bytes) allocated to the process.
LocksHeld	int	None	Number of locks process currently holds.
TableAccesses	int	Counter	Number of pages read that the SAP ASE server retrieved without using an index.

Name	Datatype	Attributes	Description
IndexAccesses	int	Counter	Number of pages read that the SAP ASE server retrieved using an index.
WorkTables	int	Counter	Total number of work tables the process created.
TempDbObjects	int	Counter	Total number of temporary tables the process created.
ULCBytesWritten	int	Counter	Number of bytes written to the user log cache for the process.
ULCFlushes	int	Counter	Total number of times the user log cache was flushed. The value is a sum of regular and tempdb user log cache.
ULCFlushFull	int	Counter	Number of times the user log cache was flushed because it was full. The value is a sum of regular and tempdb user log cache.
ULCMaxUsage	int	None	The maximum usage (in bytes) of the user log cache by the process. The value is a sum of regular and tempdb user log cache.
ULCCurrentUsage	int	None	The current usage (in bytes) of the user log cache by the process. The value is a sum of regular and tempdb user log cache.
Transactions	int	Counter	Number of transactions started by the process.
Commits	int	Counter	Number of transactions committed by the process.
Rollbacks	int	Counter	Number of transactions rolled back by the process.
HostName	varchar(30)	None	Name of the host machine on which the application that executed the query is running.
Application	varchar(30)	None	Name of the application.
ClientName	varchar(30)	None	Value of the <clientname> property set by the application.
ClientHostName	varchar(30)	None	Value of the <clienthostname> property set by the application.
ClientApplName	varchar(30)	None	Value of the <clientapplname> property set by the application.
IOSize1Page	int	Counter	Number of IO operations performed for each IO one page in size.
IOSize2Page	int	Counter	Number of IO operations performed for each IO that is 2 pages in size.

Name	Datatype	Attributes	Description
IOSize4Page	int	Counter	Number of IO operations performed for each IO that is 4 pages in size.
IOSize8Page	int	Counter	Number of IO operations performed for each IO that is 8 pages in size.
HeapMemoryInUseKB	int	None	Current amount, in kilobytes, of heap memory in use.
HeapMemoryUsedHWM_KB	int	None	Maximum amount, in kilobytes, of heap memory used (the high water mark).
HeapMemoryReservedKB	int	None	Amount, in kilobytes, of heap memory reserved.
HeapMemoryAllocs	int	None	Amount, in kilobytes, of heap memory allocated.
QueryOptimizationTime	int	Counter	CPU time (in milliseconds) used for query optimization.

3.60 monProcessLookup

Provides identifying information about each process on the server.

See `monProcessActivity` for statistics about the activity of each process.

You need not enable any configuration parameters for this monitoring table to collect data.

Use the `set` command to configure `<clientname>`, `<clienthostname>`, `<clientapplname>`. See the *Reference Manual: Commands*.

Columns

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
KTID	int	None	ID of the kernel task

Name	Datatype	Attributes	Description
Login	varchar(30)	None	Login user name
Application	varchar(30)	None	Application name
ClientHost	varchar(30)	None	Host name of client
ClientIP	varchar(24)	None	IP address of client
ClientOSPID	varchar(30)	None	Client application's operating system process identifier
ClientName	varchar(30)	None	Value of the <clientname> property set by the application
ClientHostName	varchar(30)	None	Value of the <clienthostname> property set by the application
ClientAppName	varchar(30)	None	Value of the <clientappliance> property set by the application

Related Information

[monProcessActivity \[page 244\]](#)

3.61 monProcessMigration

Applies to cluster environments only. Displays information about the connection currently migrating.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	int4	None	Pending migration session process ID
KPID	int4	None	Kernel process ID
LogicalCluster	varchar(30)	None	Current logical cluster
Instance	varchar(30)	None	Current instance.
MigrationLogicalCluster	varchar(30)	None	Migration logical cluster.
MigrationInstance	varchar(30)	None	Migration instance.
Command	varchar(30)	None	Migration trigger.

3.62 monProcessNetIO

Provides the network I/O activity information for each process.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
NetworkPacketSize	int	None	Network packet size the session is currently using.

Name	Datatype	Attributes	Description
PacketsSent	int	Counter	Number of packets sent
PacketsReceived	int	Counter	Number of packets received
BytesSent	int	Counter	Number of bytes sent
BytesReceived	int	Counter	Number of bytes received
NetworkEngineNumber	smallint	None	Number of the engine that this process is using as its network engine.

3.63 monProcessObject

Provides statistical information regarding objects currently being accessed by processes.

Enable the `enable_monitoring` and `per_object_statistics_active` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
DBID	int	None	Unique identifier for the database in which the object resides
ObjectID	int	None	Unique identifier for the object
PartitionID	int	None	Unique identifier for the partition
IndexID	int	None	Unique identifier for the index
OwnerUserID	int	None	User identifier for the object owner
LogicalReads	int	Counter	Number of buffers read from cache

Name	Datatype	Attributes	Description
PhysicalReads	int	Counter	Number of buffers read from disk
PhysicalAPFReads	int	Counter	Number of asynchronous prefetch buffers read from disk
DBName	varchar(30)	None	Name of database
ObjectName	varchar(30)	None	Name of the object
PartitionName	varchar(30)	None	Name of the partition
ObjectType	varchar(30)	None	Type of object
PartitionSize	int	Counter	Partition size in kilobytes

3.64 monProcessProcedures

Returns a list of all procedures being executed by processes.

Enable the `enable_monitoring` and `statement_statistics_active` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
DBID	int	None	Unique identifier for object's database
OwnerUID	int	None	Unique identifier for the object owner
ObjectID	int	None	Unique identifier for the procedure

Name	Datatype	Attributes	Description
PlanID	int	None	Unique identifier for the query plan
MemUsageKB	int	None	Amount of memory, in KB, used by the procedure
CompileDate	datetime	None	Date that the procedure was compiled
ContextID	int	None	A unique identifier generated each time an executing query causes a stored procedure, trigger, execute immediate, deferred compilation, or other compiled object execution to occur
LineNumber	int	None	The line in the procedure currently being executed
StatementNumber	int	None	The currently executing statement
DBName	varchar(30)	None	Name of the database that contains the procedure
OwnerName	varchar(30)	None	Name of the owner of the object
ObjectName	varchar(30)	None	Name of the procedure
ObjectType	varchar(32)	None	The type of procedure (for example, stored procedure or trigger)
ExecutionCount	int	Counter	Number of times the SAP ASE server executed this instance of the stored procedure held in the procedure cache
CPUTime	int	Counter	Amount of CPU time, in milliseconds, the SAP ASE server spent executing the instance of this stored procedure held in the procedure cache
ExecutionTime	int	Counter	Total amount of time, in milliseconds, the SAP ASE server spent executing the instance of this stored procedure held in the procedure cache
PhysicalReads	int	Counter	Number of physical reads performed by the instance of this stored procedure held in the procedure cache
LogicalReads	int	Counter	Number of logical reads performed by the instance of this stored procedure held in the procedure cache
PhysicalWrites	int	Counter	Number of physical writes performed by the instance of this stored procedure held in the procedure cache
PagesWritten	int	Counter	Number of pages read by the instance of this stored procedure held in the procedure cache

3.65 monProcessSQLText

Provides the SQL text currently being executed by the process. Use `max SQL text monitored` to tune the maximum size of the SQL text.

`monProcessSQLText` returns a row for each row of the SQL text batch a process executes (specified by `SPID`). That is, if a batch contains three rows, `monProcessSQLText` returns three rows in its result set. The value for `LineNumber` indicates the number of the line in the batch. If the length of a single row exceeds 255 bytes, `monProcessSQLText` returns multiple rows and the value for `LineNumber` is the same for all rows, but the value for `SequenceInLine` is different for each row.

Enable the `enable monitoring,max SQL text monitored,SQL batch capture` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
<code>SPID</code>	<code>int</code>	None	Session process identifier.
<code>InstanceID</code>	<code>tinyint</code>	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
<code>KPID</code>	<code>int</code>	None	Kernel process identifier.
<code>ServerUserID</code>	<code>int</code>	None	Server user identifier (SUID) of the user executing this SQL. The <code>ServerUserID</code> matches the value for the <code>syslogins.suid</code> column. Use the <code>suser_name</code> function to obtain the corresponding name.
<code>OrigServerUserID</code>	<code>int</code>	None	Server user identifier prior to executing <code>set proxy</code>
<code>BatchID</code>	<code>int</code>	None	Unique identifier for the SQL batch containing the SQL text.
<code>LineNumber</code>	<code>int</code>	None	SQL batch line number for the row's SQL text.
<code>SequenceInLine</code>	<code>int</code>	None	Each row has a unique, and increasing, <code>SequenceInLine</code> value. If the length of the SQL text exceeds 255 bytes, the text is split over multiple rows.
<code>SQLText</code>	<code>varchar(255)</code>	None	The text being executed.

3.66 monProcessStatement

Provides information about the statement currently executing.

Enable the `enable monitoring, statement statistics active, per object statistics active,` and `wait event timing` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SPID	smallint	None	Session process identifier.
InstanceID	int	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier.
DBID	int	None	Unique identifier for the database currently being used by the process.
ProcedureID	int	None	Unique identifier for the stored procedure.
PlanID	int	None	Unique identifier for the plan the process is executing.
BatchID	int	None	The batch number for the process in which the statement is executed.
ContextID	int	None	The stack frame of the procedure, if a procedure.
LineNumber	int	None	Line number of the statement within the SQL batch.
CPUTime	int	Counter	CPU time, in milliseconds, used by the statement.
WaitTime	int	Counter	Amount of time, in milliseconds, the task has waited while the statement executes.
MemUsageKB	int	None	Number of kilobytes of memory used for execution of the statement.
PhysicalReads	int	Counter	Number of buffers read from disk.
LogicalReads	int	Counter	Number of buffers read from cache.
PagesModified	int	Counter	Number of pages modified by the statement.
PacketsSent	int	Counter	Number of network packets sent by the SAP ASE server.

Name	Datatype	Attributes	Description
PacketsReceived	int	Counter	Number of network packets received by the SAP ASE server.
NetworkPacketSize	int	None	Size, in bytes, of the network packet currently configured for the session.
PlansAltered	int	Counter	Number of plans altered at execution time.
RowsAffected	int	None	Number of rows affected by the current statement. Queries using an inefficient query plan likely show a high number of logical I/Os per returned row.
SnapCodegenTime	int	Counter	Total number of microseconds of CPU time used by this query plan's SNAP code generation.
SnapJITTime	int	Counter	Total number of microseconds of CPU time used by this query plan's SNAP JIT compilation.
SnapExecutionTime	int	Counter	Total amount of elapsed time that this query plan's SNAP has executed (in microseconds).
SnapExecutionCount	int	Counter	Number of times the query plan's SNAP has been executed since it was compiled.
DBName	varchar(30)	None	Name of the database in which this process is executing. If the process is executing a stored procedure or other compiled object, the database name is the name of the database for that object.
StartTime	datetime	None	Date when the statement began executing.
QueryOptimizationTime	int	Counter	CPU time (in milliseconds) used for query optimization.

3.67 monProcessWaits

Provides a list of all wait events for which current processes on the server are waiting. Returns only wait events whose `waits` value is greater than zero.

Enable the `enable monitoring`, `wait event timing`, and `process wait events` configuration parameters for this monitoring table to collect data.

See *Performance and Tuning: Monitoring Tables* for a descriptions of select wait events.

Columns

Name	Datatype	Attribute	Description
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
ServerUserID	int	None	Server user ID (SUID) of the user associated with this process.
OrigServerUserID	int	None	Server user identifier prior to executing <code>set proxy</code>
WaitEventID	smallint	None	Unique identifier for the wait event
Waits	int	Counter	Number of times the process has waited for the event
WaitTime	int	Counter	Amount of time, in milliseconds, that the process has waited for the event

3.68 monProcessWorkerThread

Provides statistics for the activity of each currently configured worker process.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attribute	Description
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier
ThreadsActive	int	None	Number of worker threads currently in use by the process

Name	Datatype	Attribute	Description
MaxParallelDegree	smallint	None	The maximum degree of parallelism this task can use, which is set with the set <code>parallel_degree</code> option for the session, or the current <code>Run Value</code> for <code>max_parallel_degree</code> .
MaxScanParallelDegree	smallint	None	The maximum degree of parallelism for scans this task can use, which is set with set <code>scan_parallel_degree</code> for the session, or if this is not set, the current <code>Run Value</code> for <code>max_scan_parallel_degree</code> .
ParallelQueries	int	Counter	Total number of parallel queries performed by this process
PlansAltered	int	Counter	Number of plans altered from "optimal" for the process. Plans are altered if the SAP ASE server has an insufficient number of worker threads available to execute the query with an optimal degree of parallelism.
FamilyID	int	None	The spid of the parent process, if this is a worker process

3.69 monRepCoordinator

Provides information about the RepAgent coordinator process.

Besides providing general information about the coordinator process, such as its spid, or the database with which RepAgent is associated, it also provides status information. `monRepCoordinator` display rows if the RepAgent process is running `multiple_scanners` or if `stream_replication` is set to true.

Columns

Note

Columns that display information only when `stream_replication` is set to true are indicated with a ¹

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database currently being used by the process
SPID	int	None	Session process identifier
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster

Name	Datatype	Attributes	Description
DBName	varchar (30)	None	Database name for this Rep Agent
Status	varchar (30)	None	Current task status
SleepStatus	varchar (30)	None	Current sleep status, if sleeping
NumStreamSwSynctoAsync	bigint	None	¹ Number of Stream mode switches from (near) synchronous to asynchronous
NumStreamSwAsynctoSync	bigint	None	¹ Number of Stream mode switches from asynchronous to (near) synchronous
ModeSwitchStatus	varchar (30)	None	The status of synchronous and asynchronous switch modes.

Valid values for ModeSwitchStatus are:

Value	Description
AsyncSwitchRequested	An asynchronous switch has been requested; RAT-CI is waiting for an acknowledgement from Fault Manager or Replication Server.
SyncSwitchRequested	A switch to synchronous mode has been requested.
AsyncSwitchAllowed	The previous SyncSwitchRequested is confirmed and RAT-CI is allowed to execute the switch towards SyncSwitchRequested mode.
AsyncSwitchInProgress	A switch to asynchronous mode is in progress
SyncSwitchInProgress	Switch to synchronous mode is in progress
AsyncSwitchCompleted	A switch to asynchronous mode has occurred.
SyncSwitchCompleted	A switch to synchronous mode has occurred.

3.70 monRepLogActivity

Collects statistics information about Replication Agent activity related to scanning the log.

Enable the RepAgent `activate_monitoring` configuration parameter to collect timing related data.

A RepAgent has a syslogs scanner and a sysimrslogs scanner if in-memory row storage (IMRS) is enabled on a database. For a non-IMRS database, this table reports information for the syslogs scanner. For an IMRS database, this table reports information for both scanners.

Use the `ScannerSPID` and `ScannerType` columns from `monRepScanners` to determine which row is the syslogs scanner or sysimrlogs scanner.

Columns

The columns for `monRepLogActivity` Columns are:

i Note

Columns that display information only when the `RepAgent enable monitoring` configuration parameter is set to true are indicated with an asterisk (*).

Name	Datatype	Description
DBID	int	Unique identifier for the database currently being used by the process
SPID	int	Session process identifier
InstanceID	tinyint	(Cluster environments only) ID of an instance in a shared-disk cluster
LogRecordsScanned	bigint	Total number of log records scanned
LogRecordsProcessed	bigint	Total number of log records processed
NumberOfScans	bigint	Total number of scans performed
TotalTimeForLogScans	bigint	Total amount of time the scanner thread used to scan the log
LongestTimeForLogScans	bigint	Longest time spent on a single scan
AvgTimeForLogScans	bigint	Average amount of time spent on the log scan
Updates	bigint	Total number of updates processed
Inserts	bigint	Total number of inserts processed
Deletes	bigint	Total number of deletes processed
StoredProcedures	bigint	Total number of stored procedures processed
SQLStatements	bigint	Total number of SQL statements processed
DDL	bigint	Total number of DDL log records processed
Writetxt	bigint	Total number of Log records processed by writetxt commands

Name	Datatype	Description
LobColumns	bigint	Total number of DML log records processed for a table with off-row, large object columns
CLRs	bigint	Total number of CLRs processed
Checkpoints	bigint	Total number of checkpoints processed
BeginTransaction	bigint	Total number of begin transactions processed
CommitTransaction	bigint	Total number of commit transactions processed
AbortedTransaction	bigint	Total number of aborted transactions processed
PreparedTransaction	bigint	Total number of transactions found in the prepare state
DelayedCommit	bigint	Total number of delayed commits processed
MaintenanceUserTransaction	bigint	Total number of transactions opened by the maintenance user
NumberOfLogExtentions	bigint	Total number of times the RepAgent waited for extensions to transactions
TotalTimeOfLogExtentions	bigint	Total amount of time, in milliseconds, the RepAgent waited for log extensions*
LongestTimeOfLogExtentions	bigint	Longest amount of time, in milliseconds, the RepAgent waited for log extensions*
AvgTimeOfLogExtentions	bigint	Average amount of time, in milliseconds, the RepAgent waited for log extensions*
NumberOfSchemaFwdLookup	bigint	Total number of schema forward lookups
TotalTimeOfSchemaFwdLookup	bigint	Total amount of time, in milliseconds, spent on forward scans*
LongestTimeOfSchemaFwdLookup	bigint	Longest amount of time, in milliseconds, spent on a forward scan*
AvgTimeOfSchemaFwdLookup	bigint	Average amount of time, in milliseconds, spent on forward scans*
NumberOfSchemaBckwLookup	bigint	Total number of schema backward lookups
TotalTimeOfSchemaBckwLookup	bigint	Total amount of time spent on schema backward lookups*
LongestTimeOfSchemaBckwLookup	bigint	The longest amount of time, in milliseconds, spent on a backward scan *

Name	Datatype	Description
AvgTimeOfSchemaBckwLookup	bigint	Average amount of time, in milliseconds, spent on backward scans*
NumberOfMempoolAllocates	bigint	Total number of RepAgent pool allocates
NumberOfMempoolFrees	bigint	Total number of RepAgent memory pool frees
MempoolCurrentSize	bigint	Current size of the RepAgent memory pool
MempoolHighUsage	bigint	RepAgent memory pool high usage
DBName	varchar (30)	Name of the database in which the task scans
SaveTransaction	bigint	Number of save transactions processed
AbortCommand	bigint	Number of abort commands processed
PartialRollback	bigint	Number rollbacks to a savepoint processed
CachedBeforeImageUsed	bigint	Number of times that a RepAgent thread retrieves the before-image row values from its memory

3.71 monRepMemoryStatistics

Displays information about Replication Agent memory usage.

Columns

The information of this table is collected only when RepAgent uses `stream replication` to replicate data.

Name	Datatype	Description
DBID	int	Unique identifier for the database currently being used by the process.
SPID	int	Session process identifier
InstanceID	tinyint	(Cluster Edition only) ID of an instance in a shared-disk cluster

Name	Datatype	Description
AllocatedMemory	bigint	Amount of memory, in bytes, allocated by RepAgent tasks from the RepAgent memory pool.
NumberOfAllocs	bigint	Number of allocations required to assign memory from the RepAgent memory pool
NumberOfFrees	bigint	Number of frees used to release memory previously allocated from the RepAgent memory pool
NumberOfWaitsOnMemory	bigint	Number of waits on memory.
AllocatedSchemaCacheMemory	bigint	Memory currently allocated from the RepAgent memory pool used for the object schema.
GlobalStreamingRepMemory	bigint	Total amount of globally allocated memory, in bytes, requested for stream replication, and allocated by all RepAgent instances.
AllocsGlobalStreamingRep	bigint	Total number of global allocation requests to stream replication
FreesGlobalStreamingRep	bigint	Total number of global free requests to stream replication
LocalStreamingRepMemory	bigint	Total amount of memory, in bytes, requested for stream replication that was allocated by the RepAgent running on a specific database.
AllocsLocalStreamingRep	bigint	Total number of requests to allocate memory for stream replication allocated by the RepAgent running on a specific database.
FreesLocalStreamingRep	bigint	Total number of requests to free memory for stream replication allocated by the RepAgent running on a specific database.
MemoryPoolConfiguredSize	bigint	Current configured amount of memory, in bytes, for the RepAgent memory pool
MemoryPoolUsageSize	bigint	Current amount of memory used, in bytes, for the RepAgent memory pool
MemoryPoolFreeSize	bigint	Current amount of memory available, in bytes, for the RepAgent memory pool
DBName	varchar(30)	Name of the database on which the RepAgent is running

3.72 monRepScanners

Provides information on where the RepAgent Scanner task spends its time.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Description	Datatype	Description
DBID	int	Unique identifier for the database currently being used by the process.
ScannerSPID	int	Session process identifier of the RepAgent scanner task.
InstanceID	tinyint	(Cluster environments only) ID of an instance in a shared-disk cluster.
EngineBinding	int	Number of the engine to which this task is bound (not applicable in threaded mode).
LogRecordsScanned	int	Total number of log records scanned.
LogRecordsProcessed	int	Total number of log records processed.
NumberOfTruncPointRequested	int	Total number of times RepAgent asked Replication Server for a new truncation point. Populated only when RepAgent uses the single task scanning of a log model. When the multiple replication paths (MRP) model is used, NumberOfTruncPointRequested values are 0. The monRepSenders table contains related information for the MRP model.
NumberOfTruncPointMoved	int	Total number of times RepAgent moved the secondary truncation point. Populated only when RepAgent uses the single task scanning of a log model. When the multiple replication paths (MRP) model is used, NumberOfTruncPointMoved values are 0. The monRepSenders table contains related information for the MRP model.
DBName	varchar(30)	Name of the database in which this task scans.
Status	varchar(30)	Current task status.

Description	Datatype	Description
SleepStatus	varchar (30)	<ul style="list-style-type: none"> • Current sleep status, if sleeping • If the log is full, reports: <ul style="list-style-type: none"> sleeping on log full
StartMarker	varchar (30)	Start marker in the log for this scanner.
EndMarker	varchar (30)	End marker in the log for this scanner.
CurrentMarker	varchar (30)	Current marker in the log for this scanner.
OldestTransaction	varchar (30)	Oldest open transaction.
PathName	varchar (30)	Name of the path the scanner is servicing. Not applicable when RepAgent is configured for stream replication.
LogPagesLeft	int	The number of remaining log pages to be scanned for this scanner, from its CurrentMarker to the end of the log.
GetLogPageCount	int	Controls the retrieval of data for the LogPagesLeft column. The LogPagesLeft column is populated only if the value of GetLogPageCount is 1.
ScannerType	varchar (30)	One of the following scanner types: <ul style="list-style-type: none"> • syslogs_scanner • sysimrslogs_scanner

3.73 monRepScannersTotalTime

Provides information about time spent on different RepAgent Scanner modules.

Enable the RepAgent `activate_monitoring` configuration parameter for this table to start collecting timing related data.

A RepAgent has a syslogs scanner and a sysimrslogs scanner if in-memory row storage (IMRS) is enabled on a database. For a non-IMRS database, this table reports information for the syslogs scanner. For an IMRS database, this table reports information for both scanners.

Use the `ScannerSPID` and `ScannerType` columns from `monRepScanners` to determine which row is the syslogs scanner or sysimrslogs scanner.

Columns

i Note

Columns that display information only when the RepAgent is running in multithreaded mode are indicated with a ¹. Columns that display information only when multi-path replication is enable are indicated with a ²

Name	Datatype	Description
DBID	int	Unique identifier for the database currently being used by the process.
SPID	int	Session process identifier
InstanceID	tinyint	(Cluster environments only) ID of an instance in a shared-disk cluster.
LogRecProcessed	bigint	Total number of log records processed by the scanner thread
BytesPacked	bigint	Amount of LTL bytes packed by the scanner thread.
TotalTime	bigint	Total amount of time used by the scanner thread
BootstrapTime	bigint	Total amount of time required, in microseconds, to complete the multipath replication bootstrap cycle
ScanTime	bigint	Total amount of time spent scanning
ProcessTime	bigint	Total amount of time spent processing log records
SchemaLookupsTime	bigint	Total amount of time spent looking for an object's schema in RepAgent cache
PackTime	bigint	Total amount of time spent packing the LTL
QueueingTime	bigint	Total amount of time spent queuing LTL packets ¹
HashBindingSize	bigint	Total number of buckets in the hash binding table holding an object's binding information ²
HashBindingEntries	bigint	Total number of objects bound to a path when RepAgent was boot strapped ²
HashBindingCollisions	bigint	The length of the longest collision chain used in the hash binding table ²
YieldsOnFullQueue	bigint	Total number of scanner yields on a full queue ¹
WaitsOnSenderThread	bigint	Total number of waits on a sender thread ¹
WaitTimeOnSenderThread	bigint	Total amount of time, in milliseconds, spent waiting on the sender thread ¹

Name	Datatype	Description
LongestWaitOnSenderThread	bigint	Longest amount of time, in milliseconds, spent waiting on the sender thread ¹

3.74 monRepSchemaCache

The `monRepSchemaCache` table reports the schema cache information for each running RepAgent scanner.

Enable the RepAgent `activate_monitoring` configuration parameter to collect timing related data.

A RepAgent has a `syslogs` scanner and a `sysimrslogs` scanner if in-memory row storage (IMRS) is enabled on a database. For a non-IMRS database, this table reports information for the `syslogs` scanner. For an IMRS database, this table reports information for both scanners.

Use the `ScannerSPID` and `ScannerType` columns from `monRepScanners` to determine which row is the `syslogs` scanner or `sysimrslogs` scanner.

Columns

The columns for `<monRepSchemaCache>` are:

i Note

Columns that display information only when the RepAgent `enable_monitoring` configuration parameter is set to true are indicated a ¹.

Description	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database running the Replication Agent.
ScannerSPID	int	None	Session process identifier of the scanner task.
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
ConfiguredSize	bigint	None	Size of the schema cache, in bytes, as configured using <code>max schema cache per scanner</code> .
CurrentUsageSize	bigint	None	Current size, in bytes, of the schema cache for this scanner.
MaxReachedSize	bigint	None	Maximum size, in bytes, reached for the schema cache.
ObjectSchemas	bigint	None	Number of schemas in cache for tables/stored procedures.

Description	Datatype	Attributes	Description
TextImageDescriptors	bigint	None	Number of descriptors for text/image column replication.
WideParameters	bigint	None	Number of descriptors for the wide parameter for stored procedure replication.
ObjectSchemasFlushed	bigint	None	Number of table/stored procedure schemas that have been flushed.
TextImageDescriptorsFlushed	bigint	None	Number of descriptors for text/image column replication that have been flushed.
WideParametersFlushed	bigint	None	Number of descriptors for the wide parameter for stored procedure replication that have been flushed.
CacheTooSmallFlushes	bigint	None	Number of objects flushed because the schema cache could not hold all schemas involved. This might be an indication that you need to increase the schema cache size.
TotalAllocTime	bigint	None	Total amount of time spent allocating objects ¹ .
TotalDeallocTime	bigint	None	Total amount of time spent de-allocating objects ¹ .
DBName	varchar (30)	None	Name of the database in which the task scans.

3.75 monRepSenders

Provides processing information about RepAgent Sender tasks.

This table collects data when you enable the RepAgent `activate_monitoring` configuration parameter to collect timing related data and RepAgent is configured to replicate in LTL mode.

Columns

The columns for `monRepSenders` are:

i Note

Columns that display information only when the RepAgent is running in multithreaded mode are indicated with a ¹. Columns that display information only when the RepAgent `activate_monitoring` configuration parameter is set to true (to begin collecting timing-related data) are indicated a ².

Name	Datatype	Description
DBID	int	Unique identifier for the database currently being used by the process.
SenderSPID	int	Process identifier for the Replication Agent sender task.
InstanceID	tinyint	(Cluster environments only) ID of an instance in a shared-disk cluster.
EngineBinding	int	Number of the engine with which this task is bound (not applicable when the SAP ASE kernel mode is set to "threaded").
MessageQueueSize	int	Maximum size of the message queue ¹
MessagesInQueue	int	Total number of messages in the message queue ¹
NumberOfScannerYields	int	Total number of times the scanner yielded on a full queue ¹
NumberOfScannerSleeps	int	Total number of times the scanner slept on a full queue ¹
NumberOfBytesSent	int	Total number of bytes sent
LastRepServerError	int	Last error from Replication Server
NumberOfRetries	int	Total number of connection retries
SleepsOnEmptyQueue	int	Total number of sleeps spent on an empty message queue
NumberOfQueueFlushes	int	Total number of times a sender flushed its queue
SleepTimeOnEmptyQueue	int	Total amount of time, in milliseconds, spent sleeping in an empty queue ²
LongestSleepTimeOnEmptyQueue	int	Longest amount of time, in milliseconds, spent sleeping on an empty queue ²
MaxQueueSize	int	Maximum queue size ever reached ¹
NumberOfCmdsProcessed	int	Total number of commands processed.
AvgBytesPerCmd	int	Average number of bytes per command.
ScannerSPID	int	Process identifier for the Replication Agent scanner task associated with this sender.
DBName	varchar (30)	Name of the database in which the task scans
Dataserver	varchar (30)	Dataserver name used to connect to Replication Server

Name	Datatype	Description
ReplicationServer	varchar (30)	Replication Server name used to connect to Replication Server
Username	varchar (30)	User name used to connect to Replication Server
Status	varchar (30)	Current status of this task
SleepStatus	varchar (30)	Current sleep status, if sleeping
PathName	varchar (30)	Name of the path the scanner is servicing (only applicable for Multi-path replication)

3.76 monRepStreamStatistics

Provides information about RepAgent activity when it is configured to use stream replication.

This table collects data when you enable the RepAgent `activate_monitoring` configuration parameter to collect timing-related data.

i Note

Columns that display information only when the RepAgent `enable_monitoring` configuration parameter is set to true are indicated with an asterisk (*).

A RegAgent has a syslogs scanner and a sysimrslogs scanner if in-memory row storage (IMRS) is enabled on a database. For a non-IMRS database, this table reports information for the syslogs scanner. For an IMRS database, this table reports information for both scanners.

Use the `ScannerSPID` and `ScannerType` columns from `monRepScanners` to determine which row is the syslogs scanner or sysimrslogs scanner.

Columns

Table 2: monRepStreamStatistics Columns

Name	Datatype	Description
DBID	int	Unique identifier for the database currently being used by the process.

Name	Datatype	Description
SPID	int	Process identifier for the Replication Agent coordinator task.
InstanceID	tinyint	(Cluster Edition only) ID of an instance in a shared-disk cluster.
NumberOfScannersSleep sOpenStream	bigint	Number of times the scanner sleeps while opening a stream.
NumberOfScannersSleep sAllocPkg	bigint	Number of times the scanner sleeps while allocating a package.
NumberOfScannersSleep sFlushPkg	bigint	Number of times the scanner sleeps while flushing a package.
NumberOfPackagesAllo cated	bigint	Number of packages this stream allocated.
NumberOfPkgsFlushed	bigint	Number of packages this stream flushed.
NumberOfFullPkgsFlus hed	bigint	Number of full packages this stream flushed.
NumberOfFullPkgsFlus hedAtEOL	bigint	Number of full packages flushed at the end of the log.
NumberOfPkgsFlushedA tCommit	bigint	Number of packages flushed after a commit.
NumberOfAllocatedCmd s	bigint	Number of commands allocated for all packages.
NumberOfCmdHdrsGen	bigint	Number of command headers generated.
NumberOfObjSchemaGen	bigint	Number of object schemas generated.
NumberOfBytesCmdHdrs Gen	bigint	Number of command header bytes generated.
NumberOfBytesObjSche maGen	bigint	Number of object schema bytes generated.
TotalMbytesSentForSt ream	bigint	Total number of megabytes sent for this stream.
TotalTimeDistCmds	bigint	Total amount of time, in milliseconds, spent distributing com- mands.*
TotalTimeAllocPkg	bigint	Total amount of time, in milliseconds, allocating stream packages.*

Name	Datatype	Description
TotalPopulatePkgTime	bigint	Total amount of time, in milliseconds, populating the packages.
LastPopulatePkgTime	bigint	Amount of time, in microseconds, spent populating the last package.
LongestPopulatePkgTime	bigint	Maximum amount of time, in microseconds, spent populating any package.
TotalTimeFlushingPkg	bigint	Total amount of time, in milliseconds, spent flushing stream packages.*
TotalWaitToSendTime	bigint	Total amount of time, in milliseconds, spent by packages waiting in the send queue before the CI thread sends them over the network.
LastWaitToSendTime	bigint	Amount of time, in microseconds, that the last package spent waiting in the send queue.
LongestWaitToSendTime	bigint	Maximum amount of time, in microseconds, spent by any package waiting in the send queue.
TotalSendTime	bigint	Total amount of time, in milliseconds, required to send packages over the network.
LastSendTime	bigint	Amount of time, in microseconds, required to send the last package over the network.
LongestSendTime	bigint	Maximum amount of time, in microseconds, required to send any package over the network.
AllocatedBufferPoolSize	bigint	Allocated size of the buffer pool that holds the packets available for a RepAgent thread to use to send data towards SAP Replication Server.
ConfiguredBufferPoolSize	bigint	Configured size of the packets buffer pool.
MaxBufferPoolSize	bigint	Maximum size of the packets buffer pool.
BufferPoolExhausted	bigint	Number of times the buffer pool was exhausted (no packets left).
BufferPoolGrows	bigint	Number of times the buffer pool grew to the maximum buffer pool size.
BufferPoolShrinks	bigint	Number of times the buffer pool shrank to the configured buffer pool size.
BufferPoolSleeps	bigint	Number of times a task needed to sleep when a package was allocated from the buffer pool, indicating that the buffer was exhausted.

Name	Datatype	Description
BufferPoolSleepTime	bigint	Total amount of time, in milliseconds, a task needed to sleep because a package was allocated from the buffer pool.
CurUsedPoolSize	bigint	Number of packages in current use.
MaxReachedPoolSize	bigint	Maximum size reached by the buffer pool when RepAgent was running.
NumberOfPkgsInSendQ	bigint	Number of packages currently in the send queue waiting to be sent over network by the CI thread.
DBName	varchar (30)	Name of the database in which the task resides.

3.77 monRepSyncTaskStatistics

Provides information about user task activity when the Replication Agent is configured for stream replication.

`monRepSyncTaskStatistics` collects timing related data when you enable the RepAgent `activate monitoring` configuration parameter to collect timing related data.

Note

Columns that display information only when the RepAgent `enable monitoring` configuration parameter is set to true are indicated with an asterisk (*).

Columns

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database currently being used by the process.
SPID	int	None	Process identifier for the Replication Agent coordinator task.
InstanceID	tinyint	None	(Cluster Edition only) ID of an instance in a shared-disk cluster.
NumberOfTaskWakeup	bigint	None	Number of wakeup calls received by user tasks in synchronous mode.

Name	Datatype	Attributes	Description
NumberTimerTaskWakeup	bigint	None	Total number of user task wakeup calls that resulted from an expired timer.
TotalSleepTime	bigint	None	Total amount of sleep time, in milliseconds, user tasks spent on disk and network I/O completion.*
LongestSleepTime	bigint	None	Longest amount of sleep time, in milliseconds, user tasks spend on disk and network I/O completion.*
AverageSleepTime	bigint	None	Average amount of sleep time, in milliseconds, user tasks spent on disk and network I/O completion.*
MaxTasksInSleepQueue	bigint	None	Maximum number of user tasks in the sleep queue.
TotalSyncCommitTime	bigint	None	Total amount of transaction time, in milliseconds, user tasks spent between the start and the end flush.*
TotalFlushTime	bigint	None	Total amount of disk I/O flush time, in milliseconds, user tasks spent between the start and the end flush.*
AverageFlushTime	bigint	None	Average amount of disk I/O flush time, in milliseconds, user tasks spent between the start and the end flush.*
NumberOfCommits	bigint	None	Total number of commits.
NumberOfSleeps	bigint	None	Total number of sleeps.
CommitToRAProcessTime	bigint	None	Total amount of time, in milliseconds, that elapses between a commit and RepAgent processing the commit.*
AverageCommitToRAProcessTime	bigint	None	Average amount of time, in microseconds, that elapses between a commit and RepAgent processing the commit.*
LongestCommitToRAProcessTime	bigint	None	Longest amount of time, in microseconds, that elapses between a commit and RepAgent processing the commit.*
RAProcessToPkgFlushTime	bigint	None	Total amount of time, in milliseconds, Rep Agent requires to process a commit before the package flushes.*

Name	Datatype	Attributes	Description
AverageRAProcessToPkgFlushTime	bigint	None	Average amount of time, in milliseconds, that elapses to process a commit before packages are flushed.*
LongestRAProcessToPkgFlushTime	bigint	None	Longest amount of time, in microseconds, Rep Agent requires to process a commit before the package flushes.*
PkgFlushToAckTime	bigint	None	Total amount of time, in milliseconds, required for a package flush to receive acknowledgement from Replication Server.*
AveragePkgFlushToAckTime	bigint	None	Average amount of time, in microseconds, for the package flush to receive an acknowledgement from Replication Server.*
LongestPkgFlushToAckTime	bigint	None	Longest amount of time, in microseconds, for the package flush to receive an acknowledgement from Replication Server.*
AckToScheduleTime	bigint	None	Total amount of time, in milliseconds, required between the Replication Server acknowledgement and the task being rescheduled.*
AverageAckToScheduleTime	bigint	None	Average amount of time, in microseconds, required between the Replication Server acknowledgement and the task being rescheduled.*
LongestAckToScheduleTime	bigint	None	Longest amount of time, in microseconds, required between the Replication Server acknowledgement and the task being rescheduled.*
DBName	varchar (30)	None	Name of the database in which the task resides
PreviousAverageCommitWaitTime	bigint	None	Previous average commit wait time for all user tasks.
CurrentAverageCommitWaitTime	bigint	None	Current average commit time
CurrentAverageCommitThreshold	bigint	None	Number of tasks that exceeded the average commit time but did not trigger a mode switch.
CommitThresholdResets	bigint	None	Number of resets of the commit wait threshold

Name	Datatype	Attributes	Description
AcquiredTaskWaitMutex	bigint	None	<p>Number of times the task wait mutex was acquired.</p> <p>A mutex is a mutual exclusion object that allows multiple program threads to share the same resource, but not at the same time.</p> <p>This mutex is acquired when a task needs to insert itself into the list of sleeping tasks waiting for a commit they sleep on to be acknowledged. This mutex is used when streaming replication is set to near sync or sync.</p>
RetriesTaskWaitMutex	bigint	None	<p>Number of times a thread tried to acquire the task wait mutex.</p> <p>The number indicates how often a retry was necessary to acquire the mutex. If this number goes up, it means there is higher traffic for this mutex, which could potentially lead to a performance slowdown. For more information on the mutex contention, enable traceflag 9144 to display the information when the RepAgent thread is stopped.</p>
AcquiredPackageInfoMutex	bigint	None	<p>Number of times the package information mutex was acquired.</p> <p>This mutex is acquired when the RepAgent thread allocates a package from the Component Interface (CI) package pool. The package pool is shared between the RepAgent scanner task and the CI native thread that deals with sending and receiving acknowledgments from Replication Server.</p>
RetriesPackageInfoMutex	bigint	None	<p>Number of times a thread tried to acquire the package information mutex.</p> <p>The number indicates how often a retry was necessary to acquire the mutex. If this number goes up, it means there is higher traffic for this mutex, which could potentially lead to a performance slowdown. For more information on the mutex contention, enable traceflag 9144 to display the information when the RepAgent thread is stopped.</p>

Name	Datatype	Attributes	Description
NumberOfAckBeforeIODone	bigint	None	Number of commits acknowledged by SAP Replication Server before SAP ASE finished flushing the physical I/O from log records.
NumberOfWakeupMissed	bigint	None	Number of SAP ASE tasks still sleeping after receiving a commit.
NumberOfSwitchesSkipped	bigint	None	Number of mode switches that were skipped because the average commit wait time for all tasks indicated a mode switch for an individual task was not needed.

3.78 monRepTruncationPoint

Provides information about the RepAgent truncation point management task's activity while the Replication Agent is configured for stream replication.

`monRepTruncationPoint` collects data when you enable the RepAgent `activate_monitoring` configuration parameter to collect timing related data.

Note

Columns that display information only when the RepAgent `enable_monitoring` configuration parameter is set to true are indicated with an asterisk (*).

Columns

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database currently being used by the process.
SPID	int	None	Process identifier for the Replication Agent truncation point manager task.
InstanceID	tinyint	None	(Cluster Edition only) ID of an instance in a shared-disk cluster.

Name	Datatype	Attributes	Description
TruncpointsProcessed	bigint	None	Total number of truncation points processed.
syslogsTruncpointsProcessed	bigint	None	If IMRS is enabled, displays the number of times a truncation point was moved in syslogs. If IMRS is not enabled, has the same value as TruncpointsProcessed.
sysimrslogsTruncpointsProcessed	bigint	None	If IMRS is enabled, displays the number of times a truncation point was moved in sysimrslogs. If IMRS is not enabled, displays 0.
TotalProcessingTime	bigint	None	Total amount of time, in milliseconds, spent processing truncation point requests.*
LongestProcessingTime	bigint	None	Longest amount of time, in milliseconds, spent processing truncation point requests.*
IdleTime	bigint	None	Total amount of idle time, in milliseconds, for the truncation point manager task.*
NumberOfWakeups	bigint	None	Number of times the truncation point manager task wakes up.
TruncpointsRequested	bigint	None	Number of truncation points requested when opening the stream to Replication Server.
TruncpointsConfirmed	bigint	None	Number of truncation points received from Replication Server that have a confirmation status.
DBName	varchar(30)	None	Name of the database in which the task resides.

Name	Datatype	Attributes	Description
Status	varchar(30)	None	Current status of this task.
SleepStatus	varchar(30)	None	Current sleep status, if sleeping.

3.79 monSpinlockActivity

Provides statistics about spinlock activity.

Enable the `enable spinlock monitoring` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
SpinlockName	varchar(255)	None	Name of spinlock
SpinlockSlotID	int	None	ID for this spinlock in the spinlock memory pool
Grabs	bigint	Counter, reset	Number of grabs for this spinlock
Spins	bigint	Counter, reset	Number of spins on this spinlock
Waits	bigint	Counter, reset	Number of waits for this spinlock
OwnerPID	int	None	Current owner Process Identifier
LastOwnerPID	int	None	Previous owner Process Identifier
Contention	real	None	Spinlock contention, as percentage
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.

3.80 monSQLRepActivity

Provides statistics for SQL statements that were successfully replicated on all open objects.

Enable the `enable_monitoring` and `per_object_statistics_active` configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attribute	Description
DBID	int	None	Unique identifier of the database the process is currently using
ObjectID	int	None	ID of the object being monitored
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster
DBName	varchar(30)	None	Name of database containing the object being monitored for activity
ObjectName	varchar(30)	None	Name of the object being monitored for activity
UpdateStmts	int	Counter	Number of <code>update</code> statements replicated as SQL
InsertSelectStmts	int	Counter	Number of <code>insert</code> and <code>select</code> statements replicated as SQL
DeleteStmts	int	Counter	Number of <code>delete</code> statements replicated as SQL
SelectIntoStmts	int	Counter	Number of <code>select into</code> statements replicated as SQL
RowsThreshold	int	None	Low boundary range for the number of rows affected by the statements

3.81 monSQLRepMisses

Provides statistics for SQL statements that were not successfully replicated for all open objects.

Enable the `enable_monitoring` and `per_object_statistics_active` configuration parameter for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier of the database the process is currently using
ObjectID	int	None	ID of the object being monitored
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster
DBName	varchar(30)	None	Name of database containing the object being monitored for activity
ObjectName	varchar(30)	None	Name of the object being monitored for activity
Threshold	int	Counter	Number of statements that could not be replicated as SQL because the number of affected rows was below the defined threshold
QueryLimitation	int	Counter	Number of statements that could not be replicated as SQL because of a query limitation
Configuration	int	Counter	Number of statements that could not be replicated as SQL because of the configuration

3.82 monSSLCertInfo

Provides statistical information about an activate SSL certificate.

Columns

Name	Datatype	Description
CertificatePath	varchar (30)	The path of the SSL certificate.
ValidFrom	datetime	The start date of the SSL certificate.
ValidTo	datetime	The expiration date of the SSL certificate.

3.83 monState

Provides information regarding the overall state of the SAP ASE server.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
LockWaitThreshold	int	None	Time (in seconds) that a process must wait for a lock before it is counted as blocked and reported in the LockWaits column. The default value for LockWaitThreshold is 5 seconds. The default is used if you do not specify a value in the where clause of the query (for example LockWaitThreshold=30).
LockWaits	int	None	Number of process that have waiting for a lock longer than the value of LockWaitThreshold.
DaysRunning	int	None	Number of days the SAP ASE server has been running.

Name	Datatype	Attributes	Description
CheckPoints	int	None	Specifies if any checkpoint is currently running.
NumDeadlocks	int	Counter	Total number of deadlocks that have occurred.
DiagnosticDumps	int	None	Specifies if a shared memory dump is currently in progress for this server.
Connections	int	None	Number of active inbound connections.
MaxRecovery	int	None	The maximum time (in minutes), per database, that the SAP ASE server uses to complete its recovery procedures in case of a system failure; also, the current Run Value for the <code>recovery interval in minutes</code> configuration option.
Transactions	int4	Counter	Number of transactions run, server-wide.
Rollbacks	bigint	Counter	Total number of transactions rolled back
Selects	bigint	Counter	Total number of <code>select</code> operations executed
Updates	bigint	Counter	Total number of <code>update</code> operations executed
Inserts	bigint	Counter	Total number of <code>insert</code> operations executed
Deletes	bigint	Counter	Total number of <code>delete</code> operations executed
Merges	bigint	Counter	Total number of <code>merge</code> operations executed
TableAccesses	bigint	Counter	Number of pages from which data was retrieved without an index
IndexAccesses	bigint	Counter	Number of pages from which data was retrieved using an index
TempDbObjects	bigint	Counter	Total number of temporary tables created
WorkTables	bigint	Counter	Total number of work tables created
ULCFlushes	bigint	Counter	Total number of times the User Log Cache was flushed
ULCFlushFull	bigint	Counter	Number of times the User Log Cache was flushed because it was full
ULCKBWritten	bigint	Counter	Number of kilobytes written to the user log cache
PagesRead	bigint	Counter	Number of pages read server-wide
PagesWritten	bigint	Counter	Number of pages written server-wide
PhysicalReads	bigint	Counter	Number of buffers read from the disk
PhysicalWrites	bigint	Counter	Number of buffers written to the disk
LogicalReads	bigint	Counter	Number of buffers read from cache

Name	Datatype	Attributes	Description
TotalSyncCommitTime	bigint	None	Total amount of time (in milliseconds) spent performing synchronous commits
SnapsGenerated	bigint	Counter	Number of compiled query compilations since the SAP ASE server was last restarted.
SnapsExecuted	bigint	Counter	Number of actual query plan executions using Simplified Native Access Plans since the SAP ASE server was last restarted.
StartDate	datetime	None	Date and time the SAP ASE server was started.
CountersCleared	datetime	None	Date and time the monitor counters were last cleared.

3.84 monStatementCache

Provides statistical information about the statement cache. You must enable the statement cache before `monStatementCache` table can collect data.

Enable the `enable monitoring, enable stmt cache monitoring, and statement cache size` configuration parameters for this monitoring table to collect data.

Columns

Name	Type	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
TotalSizeKB	int	None	Configured size, in KB, of the statement cache.
UsedSizeKB	int	None	Amount of the statement cache, in KB, currently in use.
NumStatements	int	None	Number of statements in the statement cache.
NumSearches	int	None	Number of times the statement cache was searched.
HitCount	int	None	Number of times the statement cache was searched and a match was found.
NumInserts	int	None	Number of statements that were inserted into the statement cache.

Name	Type	Attributes	Description
NumRemovals	int	None	Number of times statements were removed from the statement cache. This value includes statements that were removed with explicit purges or from a replacement strategy.
NumRecompilesSchemaChanges	int	None	Number of recompiles due to schema changes in the tables referred to in the cached statements.
NumRecompilesPlanFlashes	int	None	Number of recompiles due to the plan flushes from the cache.

3.85 monSysExecutionTime

The `monSysExecutionTime` monitoring table includes one row for each operation module executed by Adaptive Server.

Enable the `enable_monitoring` and `execution_time_monitoring` configuration parameters for this monitoring table to collect data.

Columns

The columns for `monSysExecutionTime` are:

Name	Datatype	Attribute	Description
InstanceID	int	None	(Cluster environments only) ID of an instance in a shared-disk cluster
OperationID	int	None	Unique ID of an operation category
OperationName	varchar (30)	None	Name of the operation category
ExecutionTime	bigint	Counter	Execution time, in microseconds, of each operation performed
ExecutionCnt	bigint	Counter	Total number of occurrences of this operation type

3.86 monSysLoad

Provides trended statistics on a per-engine basis. You need not have the `mon_role` role to query this monitor table.

There is one row per engine per statistic, with the exception of `kernel run queue length`, which is reported only for engine number 0.

Averages are computed using an algorithm that eliminates momentary peaks and valleys and provides an indication of overall trends.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monSysLoad` are:

Name	Datatype	Attribute	Description
<code>InstanceID</code>	<code>tinyint</code>	None	ID of the instance within the cluster.
<code>EngineNumber</code>	<code>smallint</code>	None	Engine to which this row belongs.
<code>SteadyState</code>	<code>real</code>	None	Average value for this statistic since the SAP ASE server started.
<code>Avg_1min</code>	<code>real</code>	None	One-minute moving average for this statistic.
<code>Avg_5min</code>	<code>real</code>	None	Five-minute moving average for this statistic.
<code>Avg_15min</code>	<code>real</code>	None	Fifteen-minute moving average for this statistic.
<code>Max_1min</code>	<code>real</code>	None	Maximum 1-minute average since start-up.
<code>Max_5min</code>	<code>real</code>	None	Maximum 5-minute average since start-up.
<code>Max_15min</code>	<code>real</code>	None	Maximum 15-minute average since start-up.
<code>Max_1min_Time</code>	<code>datetime</code>	None	<datetime> at which <code>Max_1min</code> occurred.
<code>Max_5min_Time</code>	<code>datetime</code>	None	<datetime> at which <code>Max_5min</code> occurred.
<code>Max_15min_Time</code>	<code>datetime</code>	None	<datetime> at which <code>Max_15min</code> occurred.

Name	Datatype	Attribute	Description
Statistic	varchar(30)	None	Name of the statistic this row represents: <ul style="list-style-type: none"> • Percent CPU busy • Percent I/O busy • Run queue length • Kernel run queue length • Outstanding disk I/Os • Disk I/Os per second • Network I/Os per second
Sample	real	None	Value of the metric at the last sample interval (that is, the current value of the metric).
Peak	real	None	The highest Sample value since the instance started (that is, the peak Sample value).
Peak_Time	datetime	None	The date and time the Peak value was achieved.
StatisticID	smallint	None	A fixed identifier for this statistic. You may want to write applications to the fixed StatisticID instead of the localized Statistic name.

3.87 monSysPlanText

Provides the history of the query plans for recently executed queries. `monSysPlanText` returns one row of text from each line of the running query plans (similar to what is returned `sp_showplan` or by `set showplan on`).

To make sure `monSysPlanText` reads the query plan text in the correct sequence, order the query result by `SequenceNumber`. For queries returning data for multiple queries or processes, order the query result by `SPID`, `KPID`, `BatchID`, `SequenceNumber`.

Enable the `enable monitoring`, `plan text pipe max messages`, and `plan text pipe active` configuration parameters for this monitoring table to collect data.

Typically, there are multiple rows in this table for each query plan. Arrange the rows by sorting on the `SequenceNumber` column in ascending order.

`monSysPlanText` is a historical monitoring table. See *Stateful Historical Monitoring Table* in the *Performance and Tuning Guide*.

Columns

The columns for `monSysPlanText` are:

Name	Datatype	Attributes	Description
PlanID	int	None	Unique identifier for the plan.
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
SPID	int	None	Session process identifier.
KPID	int	None	Kernel process identifier.
BatchID	int	None	Unique identifier for the SQL batch for which the plan was created.
ContextID	int	None	The stack frame of the procedure, if a procedure.
SequenceNumber	int	None	A monotonically increasing number indicating the position of the <code>PlanText</code> column within the entire plan text.
DBID	int	None	Unique identifier for the database where the procedure is stored, if the plan is for a stored procedure.
ProcedureID	int	None	Unique identifier for the procedure, if the plan is for a stored procedure.
DBName	varchar (30)	None	Name of the database in which the statement represented by this plan is executed. This column is NULL if this database is not open when <code>monSysPlanText</code> is queried. If the process is executing a stored procedure or other compiled object, the database name is the name of the database for that object.
PlanText	varchar (160)	None	Plan text output.

3.88 monSysSQLText

Provides the most recently executed SQL text, or the SQL text currently executing. The maximum number of rows returned can be tuned with `sql text pipe max messages`.

Enable the `enable monitoring`, `SQL batch capture`, `sql text pipe max messages`, `sql text pipe active` configuration parameters for this monitoring table to collect data.

`monSysSQLText` is a historical monitoring table. See *Performance and Tuning: Monitoring Tables*.

i Note

In many cases, the text for a query spans multiple rows in this table. Arrange rows in proper order by sorting on the `SequenceInBatch` column in ascending order.

Columns

The columns for `monSysSQLText` are:

Name	Datatype	Attributes	Description
SPID	int	None	Session process identifier.
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier.
ServerUserID	int	None	Server user identifier (SUID) of the user who executed this SQL text. The <code>ServerUserID</code> matches the value in <code>syslogins.suid</code> . Use the <code>suser_name</code> function to obtain the corresponding name.
BatchID	int	None	Unique identifier for the SQL batch containing the SQL text.
SequenceInBatch	int	None	Indicates the position of this portion of SQL text within a batch (the SQL text for a batch may span multiple rows).
SQLText	varchar (255)	None	SQL text.

3.89 monSysStatement

Provides a history of the most recently executed statements on the server. Use `statement pipe max messages` to tune the maximum number of statement statistics returned.

Enable the `enable monitoring, statement statistics active, per object statistics active, statement pipe max messages, statement statistics active, and statement pipe activestatement statistics` configuration parameters for this monitoring table to collect data.

`monSysStatement` is a historical monitoring table. See *Performance and Tuning: Monitoring Tables*.

Columns

The columns for `monSysStatements` are:

Name	Datatype	Attributes	Description
SPID	smallint	None	Session process identifier.
InstanceID	int	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
KPID	int	None	Kernel process identifier.
DBID	int	None	Unique identifier for the database.
ProcedureID	int	None	Unique identifier for the procedure.
PlanID	int	None	Unique identifier for the stored plan for the procedure.
BatchID	int	None	Unique identifier for the SQL batch containing the statement.
ContextID	int	None	The stack frame of the procedure, if a procedure.
LineNumber	int	None	Line number of the statement within the SQL batch.
CpuTime	int	Counter	Number of milliseconds of CPU used by the statement.
WaitTime	int	Counter	Number of milliseconds the task has waited during execution of the statement.
MemUsageKB	int		Number of kilobytes of memory used for execution of the statement.
PhysicalReads	int	Counter	Number of buffers read from disk.
LogicalReads	int	Counter	Number of buffers read from cache.
PagesModified	int	Counter	Number of pages modified by the statement.
PacketsSent	int	Counter	Number of network packets sent by the SAP ASE server.
PacketsReceived	int	Counter	Number of network packets received by the SAP ASE server.
NetworkPacketSize	int	None	Size (in bytes) of the network packet currently configured for the session.
PlansAltered	int	Counter	The number of plans altered at execution time.
RowsAffected	int	None	Number of rows affected by the current statement. Queries using an inefficient query plan likely show a high number of logical I/Os per returned row.

Name	Datatype	Attributes	Description
ErrorStatus	int	None	The error return status of the statement.
HashKey	int	None	Hash value for the text of the statement; this is not a unique identifier. This column is zero (0) if the statement is not executed from the statement cache.
SsqlId	int	None	ID of the query plan for this statement within the statement cache. This column is zero (0) if the statement is not executed from the statement cache.
ProcNestLevel	int	None	Nesting level of the statement. This column is zero (0) if the statement is an ad hoc query. If the statement is within a stored procedure, this column indicates the nesting level of that stored procedure.
StatementNumber	int	None	Number indicating the order in which this statement was executed within the SQL batch for the process.
DBName	varchar (30)	None	Name of the database in which the statement is executed. This column is NULL if the database is no longer open when monSysStatement is queried. If the process is executing a stored procedure or other compiled object, the database name is the name of the database for that object.
StartTime	datetime	None	Date the statement began execution.
EndTime	datetime	None	Date the statement finished execution.
SnapCodegenTime	int	Counter	Total number of microseconds of CPU time used by this query plan's SNAP code generation.
SnapJITTime	int	Counter	Total number of microseconds of CPU time used by this query plan's SNAP JIT compilation.
SnapExecutionTime	int	Counter	Total amount of elapsed time that this query plan's SNAP has executed (in microseconds).
SnapExecutionCount	int	Counter	Number of times the query plan's SNAP has been executed since it was compiled.
QueryOptimizationTime	int	Counter	CPU time (in milliseconds) used for query optimization.

3.90 monSysWaits

Provides a server-wide view of the statistics for events on which processes have waited.

Enable the `enable_monitoring` and `wait_event_timing` configuration parameters for this monitoring table to collect data.

See *Performance and Tuning: Monitoring Tables* for more information

You can join the `monSysWaits` table with `monWaitEventInfo` using the `WaitEventID` columns as the join column to obtain the wait event descriptions. For example:

```
select w.Waits, w.WaitTime, w.WaitEventID, i.Description
from master..monSysWaits w, master..monWaitEventInfo i
where w.WaitEventID = i.WaitEventID
```

Columns

The columns for `monSysWaits` are:

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
WaitEventID	smallint	None	Unique identifier for the wait event
WaitTime	int	Counter	Amount of time (in seconds) tasks spent waiting for the event
Waits	int	None	Number of times tasks waited for the event

3.91 monSysWorkerThread

Returns server-wide statistics related to worker thread configuration and execution.

Enable the `enable_monitoring` configuration parameter for this monitoring table to collect data.

Columns

The columns for `monSysWorkerThread` are:

Name	Datatype	Attributes	Description
<code>InstanceID</code>	<code>tinyint</code>	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
<code>ThreadsActive</code>	<code>int</code>	None	Number of worker processes currently active
<code>TotalWorkerThreads</code>	<code>int</code>	None	Maximum number of worker processes (configured by setting <code>number of worker processes</code>)
<code>HighWater</code>	<code>int</code>	reset	The maximum number of worker processes that have ever been in use
<code>ParallelQueries</code>	<code>int</code>	Counter, reset	Number of parallel queries attempted
<code>PlansAltered</code>	<code>int</code>	Counter, reset	Number of plans altered due to unavailable worker processes
<code>WorkerMemory</code>	<code>int</code>	None	The amount of memory currently in use by worker processes
<code>TotalWorkerMemory</code>	<code>int</code>	None	The amount of memory configured for use by worker processes
<code>WorkerMemoryHWM</code>	<code>int</code>	reset	The maximum amount of memory ever used by worker processes
<code>MaxParallelDegree</code>	<code>int</code>	None	The maximum degree of parallelism that can be used: the current <code>Run Value</code> for <code>max parallel degree</code> configuration option
<code>MaxScanParallelDegree</code>	<code>int</code>	None	The maximum degree of parallelism that can be used for a scan: the current <code>Run Value</code> for <code>max scan parallel degree</code> configuration option

3.92 monTableColumns

Describes all the columns for each monitoring table. `monTableColumns` helps determine what columns are in the monitoring tables. You can join `monTableColumns` with `monTables` to report columns and column attributes for the monitoring tables.

The metadata view for this table is identical for all instances in a shared-disk cluster.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Table 3: monTableColumns Columns

Name	Datatype	Attributes	Description
TableID	int	None	Unique identifier for the view
ColumnID	int	None	Position of the column
TypeID	int	None	Identifier for the datatype of the column
Precision	tinyint	None	Precision of the column, if numeric
Scale	tinyint	None	Scale of the column, if numeric
Length	smallint	None	Maximum length of the column (in bytes)
Indicators	int	None	<p>Indicators for specific column properties (for example, if the column is prone to wrapping and should be sampled)</p> <p>The <code>Indicators</code> column is a bitmap. Use a bit mask to determine which bits are turned on. Possible values are:</p> <ul style="list-style-type: none"> 1 – the value for <code>Indicators</code> may increase rapidly and lead to counter wrapping if values reach 2^{32}, which can occur in columns that have the number 1 bit in the <code>Indicators</code> column value turned on. To determine whether the 1 bit is turned on, use: <pre>select TableName, ColumnName from Master..monTableColumns where Indicators & 1 != 0</pre> 2 – the counter is shared with <code>sp_sysmon</code> and is reset if you execute <code>sp_sysmon . . .clear</code>. To display all columns <code>sp_sysmon</code> clears with the <code>clear</code> parameter, use: <pre>Select TableName, ColumnName from master..monTableColumns where Indicators & 2 != 0</pre>
TableName	varchar (30)	None	Name of the table.
ColumnName	varchar (30)	None	Name of the column.
TypeName	varchar (20)	None	Name of the datatype of the column.

Name	Datatype	Attributes	Description
Description	varchar (5 12)	None	Description of the column (includes the column's unit of measurement).
Language	varchar (3 0)	None	Allows you to specify the language in which the SAP ASE server returns the values of the Description column and the Label column. By default, the SAP ASE server returns US English. Queries must use the ISO-639 and ISO-3166 naming conventions.
Label	varchar (1 50)	None	Description of the data presented in the column. You can use these values in application user interfaces instead of the actual column names.

3.93 monTableCompression

Contains the table's compression history. Enable the `enable monitoring, capture compression statistics, and per object statistics active` configuration parameters for this monitoring table to collect data.

Columns

The columns for `monTableCompression` are:

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster Edition only) Server instance ID
DBID	int	None	ID of the database to which this table was transferred
ObjectID	int	None	ID of the compressed object
PartitionID	int	None	ID of the compressed partition
CompRowInserted	bigint	Counter	Number of compressed rows inserted
CompRowUpdated	bigint	Counter	Number of updated compressed rows
CompRowForward	bigint	Counter	Number of compressed rows forwarded from the update
CompRowScan	bigint	Counter	Number of compressed rows accessed

Name	Datatype	Attributes	Description
RowDecompressed	bigint	Counter	Number of rows decompressed
RowPageDecompressed	bigint	Counter	Number of page-compressed rows decompressed to be row-compressed
ColDecompressed	bigint	Counter	Number of columns decompressed
RowCompNoneed	int	Counter	Number of rows not compressed because their compressed row length exceeded their normal row length
PageCompNoneed	bigint	Counter	Number of pages that are not suitable for page-level compression because the SAP ASE server cannot generate a dictionary or index
PagesCompressed	bigint	Counter	Number of pages compressed at the page-level
AvgBytesSavedPageLevel	bigint	Counter	Number of bytes page level compression saved
TableName	varchar	None	Name of the compressed table

3.94 monTableParameters

Provides a description for all columns in a monitoring table used to optimize query performance for the monitoring tables.

The metadata view for this table is identical for all instances in a shared-disk cluster.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monTableParameters` are:

Name	Datatype	Attributes	Description
TableID	int	None	Unique identifier for the table
ParameterID	int	None	Position of the parameter
TypeID	int	None	Identifier of the datatype of the parameter

Name	Datatype	Attributes	Description
Precision	tinyint	None	Precision of the parameter, if numeric
Scale	tinyint	None	Scale of the parameter, if numeric
Length	smallint	None	Maximum length of the parameter (in bytes)
TableName	varchar (30)	None	Name of the table
ParameterName	varchar (30)	None	Name of the parameter
TypeName	varchar (20)	None	Name of the datatype of the parameter
Description	varchar (255)	None	Description of the parameter

3.95 monTables

Provides a description of all monitoring tables. You can join `monTables` with `monTableColumns` for a description of each monitoring table and the columns it contains.

The metadata view for this table is identical for all instances in a shared-disk cluster.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Table 4: monTables Columns

Name	Datatype	Attributes	Description
TableID	int	None	Unique identifier for the table
Columns	tinyint	None	Total number of columns in the table
Parameters	tinyint	None	Total number of optional parameters you can specify

Name	Datatype	Attributes	Description
Indicators	int	None	<p>Indicators for specific table properties (for example, if the table retains session context)</p> <p>The <code>Indicators</code> column is a bit map. Use a bitmask to determine which bits are turned on. A value of 1 indicates the table is a historical table.</p> <p>To display all tables that are historical:</p> <pre>Select TableName from master..monTables where Indicators & 1 != 0</pre>
Size	int	None	Maximum row size (in bytes)
TableName	varchar (30)	None	Table name
Description	varchar (368)	None	Table description. Supports 512 characters.
Language	varchar (30)	None	<p>Allows you to specify the language in which the SAP ASE server returns the values of the <code>Description</code> column.</p> <p>By default, the SAP ASE server returns US English. Queries must use the the ISO-639 and ISO-3166 naming conventions.</p>

3.96 monTableTransfer

Provides historical transfer information for tables in the SAP ASE server's active memory. It does not store information for completed transfers. `MonTableTransfer` provides transfer information on currently ongoing transfers of all tables, whether they are marked for incremental transfer or not, and on previous transfers on tables marked for incremental transfer.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monTableTransfer` are:

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) Holds the instance ID of the server in which the command is running. In non-clustered servers, always holds zero.
DBID	smallint	None	Database ID of table
TableID	int	None	Unique identifier of table
TableName	varchar (255)	None	Name of table
SequenceID	int	None	Internal tracking ID generated by the SAP ASE server
TrackingID	int	None	User-specified tracking ID
PercentDone	smallint	None	Percentage of transfer work done, expressed as an integer between 0 – 100 (all completed transfers show 100)
BeginTime	datetime	None	Date and time at which transfer begins
EndTime	datetime	None	Date and time at which transfer ends. Ongoing transfers show NULL.
EndCode	smallint	None	Ending status of transfer. <ul style="list-style-type: none"> • 0 – successful transfer. • NULL – ongoing transfer. • Error code – failed transfer.
TransferFloor	bigint	None	Timestamp at which data can be sent
TransferCeiling	bigint	None	Timestamp at which data is uncommitted and cannot be sent
RowsSent	bigint	None	Number of rows sent
BytesSent	bigint	None	Number of bytes sent
Format	varchar (8)	None	Contains the name of the destination format: one of ase, bcp, csv, or iq.

3.97 monTask

Specific to the SAP ASE server in threaded mode, contains one row for each task.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monTask` are:

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster
KTID	int	None	ID of the kernel task
ThreadPoolID	int	None	ID of the thread pool this task is associated with
ThreadID	int	None	ID of the thread running this task
Name	varchar (30)	None	Name of the task
ThreadPoolName	varchar (30)	None	Name of the thread pool this task is associated with

3.98 monTempdbActivity

Applies to cluster environments only. Provides statistics for all open local temporary databases, including global system `tempdb` when the instance is started in `tempdb` configuration mode.

`monTempdbActivity` requires the `enable monitoring, per object statistics active, and object lockwait timing` configuration parameters to collect data.

Columns

Table 5: monTempdbActivity Columns

Name	Datatype	Attributes	Description
DBID	int	None	Unique identifier for the database
InstanceID	tinyint	None	ID of the instance within the cluster
DBName	varchar (30)	None	Name of the database
AppendLogRequests	int	Counter	Number of semaphore requests from an instance attempting to append to the database transaction log
AppendLogWaits	int	Counter	Number of times a task waits for the append log semaphore to be granted
LogicalReads	int	Counter	Total number of buffers read
PhysicalReads	int	Counter	Number of buffers read from disk
APFReads	int	Counter	Number of asynchronous prefetch (APF) buffers read
PagesRead	int	Counter	Total number of pages read
PhysicalWrites	int	Counter	Total number of buffers written to disk
PagesWritten	int	Counter	Total number of pages written to disk
LockRequests	int	Counter	Number of requests for an object lock in this temporary database
LockWaits	int	Counter	Number of times a task waited for an object lock in this temporary database
CatLockRequests	int	Counter	Number of requests for a lock on the system catalog
CatLockWaits	int	Counter	Number of times a task waited for a lock for system table
AssignedCnt	int	Counter	Number of times this temporary database was assigned to a user task
SharableTabCnt	int	Counter	Number of sharable tables created

3.99 monThread

Specific to the SAP ASE server in threaded mode: Contains one row for each thread.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Table 6: monThread Columns

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster
ThreadID	int	None	ID of the thread pool
KTID	int	None	Internal kernel thread ID
OSThreadID	bigint	None	ID of the operating system thread
AltOSThreadID	int	None	Alternate operating system thread ID (on some platforms this may be a lightweight process (LWP) ID)
ThreadPoolID	int	None	ID of the thread pool
State	varchar (30)	None	Current state of the thread
ThreadAffinity	int	None	CPU number to which the thread has affinity
ThreadPoolName	varchar (30)	None	Name of the thread pool
TaskRuns	bigint	Counter	Number of tasks this thread has run
TotalTicks	bigint	Counter	Total number of ticks for this thread
IdleTicks	bigint	Counter	Total number of ticks during which this thread was idle
SleepTicks	bigint	Counter	Total number of ticks during which this thread was sleeping
BusyTicks	bigint	Counter	Total number of ticks during which this thread was busy
UserTime	bigint	Counter	Total amount of thread user CPU time, in milliseconds
SystemTime	bigint	Counter	Total amount of thread system CPU time, in milliseconds
MinorFaults	bigint	Counter	Total number of minor page faults. Value is 0 on Windows

Name	Datatype	Attributes	Description
MajorFaults	bigint	Counter	Total number of major page faults. Value is 0 on Windows
VoluntaryCtxtSwitches	bigint	Counter	Total number of voluntary operating system context switches. Value is 0 on Windows
NonVoluntaryCtxtSwitches	bigint	Counter	Total number of nonvoluntary operating system context switches. Value is 0 on Windows

3.100 monThreadPool

Specific to the SAP ASE server in threaded mode: Contains one row for each thread pool.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monThreadPool` are:

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance
ThreadPoolID	int	None	ID of the thread pool
Size	int	None	Number of threads in the thread pool
TargetSize	int	None	Requested size (differs from <code>Size</code> only when you change pool sizes)
Tasks	int	None	Number of tasks associated with the thread pool
ThreadPoolName	varchar (30)	None	Name of the thread pool
ThreadPoolDescription	varchar (255)	None	(Optional) description of the thread pool
Type	varchar (30)	None	Thread pool type, Engine (multiplexed) or Run to Completion (RTC)

Name	Datatype	Attributes	Description
IdleTimeout	int	None	Amount of time, in microseconds, that threads in this pool search for runnable tasks before idling
InstanceName	varchar (30)	None	Name of instance

3.101 monThresholdEvent

The `monThresholdEvent` monitoring table includes one row for each event recorded by SAP ASE.

Enable the `allow resource limits` configuration parameter to enable resource limits collection. Enable the `enable monitoring, threshold event monitoring, and set threshold event max messages` configuration parameters for this monitoring table to collect data.

`monThresholdEvent` is a stateful historical monitoring table (see the *Performance and Tuning Guide: Monitoring Tables*). Determine the number of events `monThresholdEvent` stores with the `threshold event max messages` configuration parameter.

Columns

Table 7: `monThresholdEvent` Columns

Name	Datatype	Attribute	Description
SPID	int	None	Server process ID.
InstanceID	tinyint	None	ID of the instance within the cluster..
KPID	int	None	SAP ASE kernel process ID.
KTID	int	None	ID of the kernel task.
ServerUserID	int	None	Server user identifier (SUID) of the user who executed this SQL text. The <code>ServerUserID</code> matches the value in <code>syslogins.suid</code> . Use the <code>suser_name</code> function to obtain the corresponding name.
FamilyID	int	None	spid of the parent process.
Login	varchar (30)	None	Login user name.
Application	varchar (30)	None	Application name.

Name	Datatype	Attribute	Description
HostName	varchar (30)	None	Client host name.
ClientName	varchar (30)	None	Client name set with <code>set clientname</code> .
ClientHostName	varchar (30)	None	Value of the <code>clienthostname</code> property set by the application.
ClientApplName	varchar (30)	None	Value of the <code>clientapplname</code> property set by the application.
ClientIP	varchar (64)	None	IP address of the client.
Command	varchar (30)	None	Category of process or command the process is currently executing.
DBID	int	None	Unique identifier for the database currently being used by the process.
DBName	varchar (30)	None	Name of the database running the process.
ProcedureID	int	None	Unique identifier for the procedure.
BatchID	int	None	Unique identifier for the SQL batch containing the statement being executed.
LineNumber	int	None	Line number of the current statement within the SQL batch.
BlockingSPID	int	None	Session process identifier of the process holding the lock this process requested, if waiting for a lock.
TempDbObjects	int	Counter	Total number of temporary tables created by the process.
RangeID	smallint	None	Range ID of the limit.
LimitType	varchar (30)	None	Limit type.
LimitID	smallint	None	Limit identifier.
LimitValue	int	None	Value of the limit that was violated.
Enforced	tinyint	None	Determines if the limit is enforced prior to, or during, query execution.
Action	varchar (30)	None	Action to perform when the limit is exceeded.
Scope	varchar (30)	None	Scope of the limit.
ReportDatetime	datetime	None	Date and time the report was issued due to the limit violation.
SQLText	varchar (255)	None	SQL text of the event.

Indexes

- Unique clustered index on `<jid>`
- Unique nonclustered index on `<jname>`

3.102 monWaitClassInfo

Provides a textual description for all of the wait classes (for example, waiting for a disk read to complete). All wait events (see the description for `monWaitEventInfo`) have been grouped into wait classes that classify the type of event for which a process is waiting.

This table displays the same information for all instances in a shared-disk cluster

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monWaitClassInfo` are:

Name	Datatype	Attributes	Description
<code>WaitClassID</code>	<code>smallint</code>	None	Unique identifier for the wait event class
<code>Description</code>	<code>varchar (50)</code>	None	Description of the wait event class
<code>Language</code>	<code>varchar (30)</code>	None	Allows you to specify the language in which the SAP ASE server returns the values of the <code>Description</code> column. By default, the SAP ASE server returns US English. Queries must use the the ISO-639 and ISO-3166 naming conventions.

3.103 monWaitEventInfo

Provides a textual description of wait conditions reported in the `monSysWaits` and `monProcessWaits` tables.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monWaitEventInfo` are:

Name	Datatype	Attributes	Description
WaitEventID	smallint	None	Unique identifier for the wait event type
WaitClassID	smallint	None	Unique identifier for the wait event class
Description	varchar (50)	None	Description of the wait event type
Language	varchar (3)	None	Allows you to specify the language in which the SAP ASE server returns the values of the <code>Description</code> column. By default, the SAP ASE server returns US English. Queries must use the the ISO-639 and ISO-3166 naming conventions.

Join `monWaitEventInfo` with `monProcessWaits` or `monSysWaits` on the `WaitEventID` column to obtain the wait event descriptions listed in those tables.

3.104 monWorkload

(Cluster environments only) Displays the workload score for each logical cluster on each instance according to its load profile.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monWorkload` are:

Name	Datatype	Attributes	Description
LCID	int	None	Logical cluster ID
InstanceID	tinyint	None	ID of the instance within the cluster
LoadProfileID	tinyint	None	ID of the load profile used to generate the load score
LoadScore	real	None	Load score for this instance or logical cluster
ConnectionsScore	real	None	Weighted value for the <code>user_connections</code> metric

Name	Datatype	Attributes	Description
CpuScore	real	None	Weighted value for the <code>cpu</code> utilization metric
RunQueueScore	real	None	Weighted value for the <code>run</code> queue metric
IoLoadScore	real	None	Weighted value for the <code>io</code> load metric
EngineScore	real	None	Weighted value for the <code>engine</code> deficit metric
UserScore	real	None	Weighted value for the <code>user</code> metric
LogicalClusterName	<code>varchar(30)</code>	None	Logical cluster name
InstanceName	<code>varchar(30)</code>	None	Instance name
LoadProfileName	<code>varchar(30)</code>	None	Name of the load profile used to generate the load score

3.105 monWorkloadPreview

(Cluster environments only) Provides an estimate of how a load profile impacts the workload score without enabling the profile.

`monWorkload` includes one row for each logical cluster and instance on which this logical cluster is running. The load score and components are based on the current profile for that logical cluster. The `monWorkloadPreview` table has one row for each combination of instance and load profile configured on the system, allowing the administrator to see how workload scoring would be done for each profile. You need not have the `mon_role` role to query this monitor table.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monWorkloadPreview` are:

Name	Datatype	Attributes	Description
InstanceID	<code>tinyint</code>	None	ID of the instance within the cluster
LoadProfileID	<code>smallint</code>	None	Load profile ID

Name	Datatype	Attributes	Description
LoadScore	int	None	Load score for this instance or logical cluster
ConnectionScore	float	None	Weighted value for the user connections metric
CpuScore	float	None	Weighted value for the cpu utilization metric
RunQueueScore	float	None	Weighted value for the run queue metric
IoLoadScore	float	None	Weighted value for the io load metric
EngineScore	float	None	Weighted value for the engine deficit metric
UserScore	float	None	Weighted value for the user metric
InstanceName	varchar (30)	None	Instance name
LoadProfileName	varchar (30)	None	Name of load profile used to generate the load score

3.106 monWorkloadProfile

Applies to cluster environments only. Displays currently configured workload profiles. You need not have the `mon_role` role to query this monitor table.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

Table 8: monWorkloadProfile Columns

Name	Datatype	Attributes	Description
ProfileID	int	None	Workload profile ID
ConnectionsWeight	tinyint	None	Weight associated with the active connections metric
CpuWeight	tinyint	None	Weight associated with the cpu utilization metric
RunQueueWeight	tinyint	None	Weight associated with the run queue metric
IoLoadWeight	tinyint	None	Weight associated with the io load metric

Name	Datatype	Attributes	Description
EngineWeight	tinyint	None	Weight associated with the <code>engine_deficit</code> metric
UserWeight	tinyint	None	Weight associated with the <code>user</code> metric
LoginThreshold	tinyint	None	Threshold for the login load distribution.
DynamicThreshold	tinyint	None	Threshold for dynamic load distribution (that is, post-login migration for load purposes)
Hysteresis	tinyint	None	Minimum load score that enables redirection.
Name	varchar(30)	None	Workload profile name
Type	varchar(30)	None	Type of workload profile. Indicates whether the profile is defined by a user or the system. Values are: <ul style="list-style-type: none"> User System

3.107 monWorkloadRaw

Applies to cluster environments only. Provides the raw workload statistics for each instance. You need not have the `mon_role` role to query this monitor table.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monWorkloadRaw` are:

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	ID of the instance within the cluster
ConnectionsRaw	real	None	Raw value for the <code>user_connections</code> metric
CpuRaw	real	None	Raw value for the <code>cpu_utilization</code> metric
RunQueueRaw	real	None	Raw value for the <code>run_queue</code> metric
IoLoadRaw	real	None	Raw value for the <code>io_load</code> metric

Name	Datatype	Attributes	Description
EngineRaw	real	None	Raw value for the <code>engine_deficit</code> metric
UserRaw	real	None	Raw value for the <code>user</code> metric
InstanceName	varchar (30)	None	Instance name

3.108 monWorkQueue

Provides information on work queues.

You need not enable any configuration parameters for this monitoring table to collect data.

Columns

The columns for `monWorkQueue` are:

Name	Datatype	Attributes	Description
InstanceID	tinyint	None	(Cluster environments only) ID of an instance in a shared-disk cluster.
CurrentLength	int	None	Current number of queued items
MaxLength	int	None	Maximum number of queued items
TotalRequests	int	Counter	Total number of requests
QueuedRequests	int	Counter	Total number of requests that waited for another request to finish
WaitTime	int	Counter	Amount of time, in milliseconds, requests waited
Name	varchar (30)	None	Name of the work queue

4 sybpcidb Tables

The `sybpcidb` database stores configuration information for the Java PCI Bridge and the PCA/JVM plug-in. This chapter describes the `sybpcidb` tables in alphabetical order.

You create `sybpcidb`, install its tables, and create its system stored procedures when you configure the server for Java. See the installation guide for your platform. See also *Java in Adaptive Server Enterprise* for information about how to use the `sp_jreconfig` and `sp_pciconfig` stored procedures that let you configure and display information in `sybpcidb`.

4.1 pca_jre_arguments

Stores information about the arguments used to configure the PCA/JVM plug-in.

Columns

Located in `sybpcidb`. The columns for `pca_jre_arguments` are:

Name	Datatype	Description
<code>jre_args_directive_index</code>	<code>int</code>	The index of the directive to which the argument belongs.
<code>jre_args_name</code>	<code>varchar(255)</code>	The name of the argument.
<code>jre_args_units</code>	<code>varchar(255)</code>	The argument type. Values are: <ul style="list-style-type: none">• switch• string• number• array
<code>jre_args_number_value</code>	<code>int</code>	If <code>units=number</code> , holds the number associated with the argument.
<code>jre_args_string_value</code>	<code>varchar(255)</code>	If <code>units=string</code> or <code>units=array</code> , holds the string value associated with the argument.

Name	Datatype	Description
jre_args_description	varchar(255)	A brief text description of the argument.
jre_args_enabled	int	Values are: <ul style="list-style-type: none"> • 0 – not enabled • 1 – enabled (default)
jre_args_status	int	Reserved for future use.

Indexes

Unique clustered index on jre_args_directive_index, jre_args_name, jre_args_string_value

4.2 pca_jre_directives

Stores information about the directives used to configure the PCA/JVM.

Columns

Located in sybpcidb. The columns for pca_jre_directives are:

Name	Datatype	Description
jre_directives_index	int	The index of the directive.
jre_directives_name	varchar(255)	The name of the directive.
jre_directives_description	varchar(255)	A text description of the directive.
jre_directives_enabled	int	Values are: <ul style="list-style-type: none"> • 0 – not enabled • 1 – enabled (default)
jre_directives_status	int	Reserved for future use.

Indexes

- Unique clustered index on `jre_directives_name`.
- Unique nonclustered index on `jre_directives_index`.

4.3 pci_arguments

Stores information that defines each of the arguments used to configure the PCI Bridge.

Columns

Located in `sybpcidb`. The columns for `pci_arguments` are:

Name	Datatype	Description
<code>pci_args_directive_index</code>	<code>int</code>	The index of the directive to which the argument belongs.
<code>pci_args_name</code>	<code>varchar(255)</code>	The name of the argument.
<code>pci_args_units</code>	<code>varchar(255)</code>	The units type. Values are: <ul style="list-style-type: none">• switch• number
<code>pci_args_number_value</code>	<code>int</code>	When <code>units=number</code> , the value of number. If <code>units=switch</code> , the value is zero (0).
<code>pci_args_string_value</code>	<code>varchar(255)</code>	Reserved for future use.
<code>pci_args_description</code>	<code>varchar(255)</code>	Brief text description of the argument and its purpose.
<code>pci_args_enabled</code>	<code>int</code>	Values are: <ul style="list-style-type: none">• 0 – not enabled• 1 – enabled (default)
<code>pci_args_status</code>	<code>int</code>	Reserved for future use.

Indexes

Unique clustered index on `pci_args_directive_index`, `pci_args_name`.

4.4 pci_directives

Stores the directives that configure the PCI Bridge.

Columns

Located in `sybpcidb`. The columns for `pci_directives` are:

Name	Datatype	Description
<code>pci_directives_index</code>	<code>int</code>	The index of the directive.
<code>pci_directives_name</code>	<code>varchar(255)</code>	The name of the directive.
<code>pci_directives_description</code>	<code>varchar(255)</code>	A description of the directive.
<code>pci_directives_enabled</code>	<code>int</code>	Values are: <ul style="list-style-type: none">• 0 – not enabled• 1 – enabled (default)
<code>pci_directives_status</code>	<code>int</code>	Reserved for future use.

Indexes

- Unique clustered index on `pci_directives_name`
- Unique nonclustered index on `pci_directives_index`

4.5 pci_slotinfo

Contains information describing each slot, including table names for the slot's directives and arguments.

Columns

Located in `sybpcidb`. The columns for `pci_slotinfo` are:

Name	Datatype	Description
<code>slot_number</code>	<code>int</code>	The number of the slot.
<code>slot_name</code>	<code>varchar(255)</code>	The name of the slot, such as JVM.
<code>slot_pca_directives_table_name</code>	<code>varchar(255)</code>	The name of the PCA directives table, such as <code>pca_jre_directives</code> .
<code>slot_pca_arguments_table_name</code>	<code>varchar(255)</code>	The name of the PCA arguments table, such as <code>pca_jre_arguments</code> .
<code>slot_status</code>	<code>varchar(255)</code>	Reserved for future use.

Indexes

- Unique clustered index on `slot_name`
- Unique nonclustered index on `slot_number`

4.6 pci_slot_syscalls

Contains the runtime system call configuration information for the runtime dispatching model used by the PCI Bridge.

Columns

Located in `sybpcidb`. The columns for `pci_slot_syscalls` are:

Name	Datatype	Description
<code>syscall_slot_number</code>	<code>int</code>	The slot number associated with the system call.
<code>syscall_system_call</code>	<code>varchar(255)</code>	The name of the system call.
<code>syscall_dispatch_name</code>	<code>varchar(255)</code>	The name of the dispatch function for the system call.
<code>syscall_enabled</code>	<code>int</code>	Values are: <ul style="list-style-type: none">• 0 – not enabled• 1 – enabled (default)
<code>syscall_status</code>	<code>int</code>	Reserved for future use.

Indexes

Unique clustered index on `syscall_slot_number`, `syscall_system_call`

5 Workload Profiler Tables

The workload profiler uses a series of tables to provide metrics and analysis for the in-memory row storages. These tables are installed when you install the workload profiler with the `installwlp_profiler` script (located in `$SYBASE/$SYBASE_ASE/scripts`).

By default, the workload profiler tables are located in the `sybdsamdb` database, and the accompanying system procedures are located in `sybsystemprocs`, although you can specify different databases by editing the installation script.

5.1 wlp_tables

Lists and describes the tables available to the workload profiler.

Columns

The columns for `wlp_tables` are:

Column name	Datatype	Null	Description
<code>wlpt_tablename</code>	<code>varchar(255)</code>	No	Name of the workload profiler table.
<code>wlpt_colprefix</code>	<code>varchar(8)</code>	No	Prefix used for all columns in this table.
<code>wlpt_label</code>	<code>varchar(30)</code>	No	Description of table's usage.
<code>wlpt_description</code>	<code>varchar(80)</code>	Yes	Long description of table's schema.

Indexes

`wlp_tables` includes the `wlpt_tables_ui` unique index on `wlp_tables` (`wlpt_tablename`, `wlpt_colprefix`)

5.2 wlp_table_columns

Describes each column in every table in the workload profiler.

Columns

The columns for `wlp_table_columns` are:

Name	Datatype	Null	Description
<code>wlptc_tablename</code>	<code>varchar(255)</code>	No	Name of the table
<code>wlptc_colname</code>	<code>varchar(255)</code>	No	Name of the column
<code>wlptc_label</code>	<code>varchar(30)</code>	No	Short description of columns.
<code>wlptc_description</code>	<code>varchar(80)</code>	Yes	Long description of columns.

Indexes

`wlp_table_columns` includes the `wlptd_table_colname_ui` unique index on `wlp_table_defs` (`wlpt_tablename, wlptd_colname`)

5.3 wlp_exec_defaults

Includes execution-time defaults for workload profiling properties (for example, the name of metrics database, the sampling interval to gather MDA metrics, and so on).

Columns

The columns for `wlp_exec_defaults` are:

Name	Datatype	Null	Identity	Description
<code>wlx_def_tabname</code>	<code>varchar(255)</code>	No		Name of the table.
<code>wlx_def_colname</code>	<code>varchar(255)</code>	No		Name of the column.
<code>wlx_def_int_value</code>	<code>int</code>	Yes		(Optional) default integer value.
<code>wlx_def_vc_value</code>	<code>varchar(30)</code>	Yes		(Optional) default character value.
<code>wlx_def_value_units</code>	<code>varchar(10)</code>	Null		Unit-specifier for default values, if applicable. NULL for values (for example, path names, database names) that have no units.
<code>wlx_def_description</code>	<code>varchar(30)</code>	Yes		Long description for columns and default value semantics

Indexes

`wlp_exec_defaults` unique index `wlx_def_values_ui` on `wlp_exec_defaults` (`wlx_def_tabname` and `wlx_def_colname`).

5.4 wlp_exec_control

Contains one row for each user-defined workload profiling activity.

Columns

The columns for `wlp_exec_control` are:

Name	Datatype	Null	Identity	Description
<code>wlx_runid</code>	<code>int</code>	No	Yes	ID of the workload profiling activity. Used as the join-key value across all tables holding data relating to this profile (for example, metrics collected, recommended configuration, and so on).
<code>wlx_name</code>	<code>varchar(255)</code>	Yes		(Optional) user-specified name identifying a workload. If not provided, the profiling tool generates a name in the form <code>Workload ID-<number></code> , where <code><number></code> is the value of <code>wlx_runid</code> .
<code>wlx_dbname</code>	<code>varchar(30)</code>	No		Target database name of the workload profiler. The default is the name of the database in which <code>sp_wlprofiler</code> is running.
<code>wlx_metric_s_dbname</code>	<code>varchar(30)</code>	No		Name of the database where the workload profiler metrics are collected and archived. The default is <code>sybdsamdb</code> .
<code>wlx_features</code>	<code>varchar(255)</code>	No		Comma-separated list of SAP ASE features the workload profiler is evaluating. The default is <code>IMRS</code> .
<code>wlx_aseversion</code>	<code>varchar(255)</code>	No		<code><@@version></code> string for the SAP ASE against which you are executing the workload profiler.
<code>wlx_traceflags</code>	<code>varchar(255)</code>	Yes		Trace flag numbers that were active at the start of the workload profiler activity.
<code>wlx_prepare_date</code>	<code>bigdatetime</code>	No		Date and time you executed a <code>sp_wlprofiler...prepare</code> parameter.
<code>wlx_begin_date</code>	<code>bigdatetime</code>	No		Date and time you executed a <code>sp_wlprofiler...begin</code> parameter.
<code>wlx_last_sampled_date</code>	<code>bigdatetime</code>	No		Date and time you last sampled the monitoring table metrics.

Name	Datatype	Null	Identity	Description
wlx_end_date	bigdatetime	No		Date and time you executed a <code>sp_wlprofiler...end</code> parameter
wlx_monitoring_interval_seconds	int	No		Workload monitoring interval, in seconds, specified with the <code>sp_wlprofiler...prepare</code> parameter, if included. The default is 300 seconds.
wlx_sample_interval_secs	int	No		Sampling interval, in seconds, for periodically archiving relevant monitoring metrics whose state can change during the profiling interval. The default is 120 seconds.
wlx_end_traceflags	varchar(255)	Yes		Traceflag numbers that were active at the end of the workload profiler activity (that is, when you issued <code>sp_wlprofiler...end</code>). This is distinguished from <code>wlx_traceflags</code> , which lists only traceflags that were enabled by the workload profiler.
wlx_auto_reconfig_attempted	smallint	No		Number of times you attempted to run <code>sp_wlprofiler...reconfigure</code> .
wlx_begin_done	tinyint	No		Indicates if the <code>sp_wlprofiler...begin</code> parameter successfully completed.
wlx_end_done	tinyint	No		Indicates if the <code>sp_wlprofiler...end</code> parameter successfully completed.
wlx_auto_reconfig_done	tinyint	No		Indicates if the <code>sp_wlprofiler...reconfigure</code> parameter successfully completed. If reconfiguration succeeds on the first attempt, the value for <code>wlx_auto_reconfig_attempted</code> is 1, and the value for <code>wlx_auto_reconfig_done</code> is 1. If it is unsuccessful, <code>wlx_auto_reconfig_attempted</code> is a non-zero value, and the value for <code>wlx_auto_reconfig_done</code> is 0.
wlx_insby	varchar(30)	Yes		Indicates how the row describing the workload profiler was inserted into <code>wlp_exec_control</code> . The default is <code>isql</code> . If the workload profiler generates the row describing a new interval, <code>wlx_insby</code> contains the value <code>sp_wlp_prepare</code> , indicating that it was inserted by the <code>sp_wlprofiler ... prepare</code> command.

Indexes

`wlp_exec_control` includes the `wlx_runid_pk_ui` primary key index on `wlp_exec_control` (`wlx_runid`), and a unique index `wlx_name_ui` on `wlp_exec_control` (`wlx_name`)

5.5 wlp_exec_commands

Contains one row for each T-SQL command that archives data.

Columns

The columns for `wlp_exec_commands` are:

Name	Datatype	Null	Identity	Description
<code>wlxcmd_ctr</code>	<code>int</code>	No	Yes	Unique ID used to sequence the commands or operations for the command specified by <code>wlxcmd_name</code> .
<code>wlxcmd_feature</code>	<code>varchar(30)</code>	No		Feature (specified in <code>wlp_exec_control.wlx_features</code>) for which the command or operation is applied. By default, data archival is done for all features indicated by <code>default</code> .
<code>wlxcmd_name</code>	<code>varchar(80)</code>	No		Internally specified name of a command or SQL operation understood by the workload profiler.
<code>wlxcmd_sql_stmt</code>	<code>varchar(512)</code>	No		SQL statement or stored procedure call that performs the required activity, such as data archival, and so on.
<code>wlxcmd_obj_name</code>	<code>varchar(255)</code>	Yes		Name of the object on which the SQL statement executes.

Indexes

`wlp_exec_commands` includes a unique index, `wlx_def_values_ui`, on `wlp_exec_defaults` (`wlx_def_tabname` and `wlx_def_colname`).

5.6 wlp_control

Logs activities that occur as part of profiling.

Columns

The columns for `wlp_control` are:

Name	Datatype	Null	Identity	Description
<code>wlpc_id</code>	<code>int</code>	No	Yes	Unique ID representing the event.
<code>wlpc_name</code>	<code>varchar(30)</code>	No		Name of the event occurring in the workload profiler.
<code>wlpc_value</code>	<code>int</code>	Yes		Value indicating whether the event occurred or not. A value of 1 indicates it occurred, 0 that it has not.
<code>wlpc_date</code>	<code>datetime</code>	Yes		Time the event occurred.

5.7 wlp_valid_features

Lists the features supported by the workload profiler

Columns

The columns for `wlp_valid_features` are:

Name	Datatype	Null	Identity	Description
<code>wlvf_code</code>	<code>smallint</code>	No	Yes	Unique ID representing the feature.
<code>wlvf_ftag</code>	<code>varchar(8)</code>	No		Tag for the feature tag. For example, DRC (datarow columns), MVCC (multiversion concurrency control), or LFB (latch-free btree).
<code>wlvf_fname</code>	<code>varchar(30)</code>	No		Descriptive name of the feature..
<code>wlvf_description</code>	<code>datetime</code>	Yes		Feature description.

5.8 wlp_exec_cmd_timings

Tracks execution time metrics for SQL operation and commands run to collect and archive metrics.

Columns

The columns for `wlp_exec_cmd_timings` are:

Name	Datatype	Null	Identity	Description
<code>wlxct_runid</code>	<code>int</code>	No		Unique ID identifying a profiling activity, and used as the join-key value across all other tables holding data relating to this profiling activity (for example, metrics collected, configuration recommended, and so on).
<code>wlxct_runctr</code>	<code>int</code>	No		Unique ID for sequencing the commands or operations for a specific command (specified by <code>wlp_exec_commands.wlxcmd_name</code>).
<code>wlxct_sqlstmt</code>	<code>varchar(512)</code>	No		SQL statement or stored procedure that performs the required activity (data archiving and so on).
<code>wlxct_objname</code>	<code>varchar(255)</code>	Yes		Name of the object on which the SQL statement is executed.
<code>wlxct_elapsed_us</code>	<code>int</code>	No		Elapsed amount of time, in microseconds, for the currently running SQL statement or stored procedure call.

Indexes

`wlp_exec_cmd_timings` includes a unique index, `wlxct_runid_ctr_ui`, on `wlp_exec_cmd_timings` (`wlxct_runid` and `wlxct_runctr`).

5.9 wlp_exec_cmd_show

Stores column names from archived tables for the `sp_wlprofiler ... show` parameter to display.

Columns

The columns for `wlp_exec_cmd_show` are:

Name	Datatype	Null	Identity	Description
<code>wlxc_show_colctr</code>	<code>int</code>	No	Yes	Unique ID value for the columns in the table.
<code>wlxc_show_montable</code>	<code>varchar(30)</code>	No		Name of the metrics table that contains the column.
<code>wlxc_show_label</code>	<code>varchar(80)</code>	Yes		Short name assigned to the column.
<code>wlxc_show_colname</code>	<code>varchar(30)</code>	No		User-visible column name.
<code>wlxc_show_orderby_no</code>	<code>tinyint</code>	Yes		Number specified by the <code>order by</code> command for the column list.
<code>wlxc_show_orderby_asc</code>	<code>tinyint</code>	Yes		Indicates the column's ascending or descending order.
<code>wlxc_show_for</code>	<code>tinyint</code>	No		Indicates which subcommands display this column.

Indexes

`wlp_exec_commands` contains unique indexes on:

- `wlxc_show_montable_cols_ui` on `wlp_exec_cmd_show` (`wlxc_show_montable` and `wlxc_show_colctr`)
- `wlxc_show_montable_colnames_ui` on `wlp_exec_cmd_show` 4 (`wlxc_show_montable` and `wlxc_show_colname`)

5.10 wlp_unique_collist_for_mda_archive

Lists the columns for the monitoring tables that provide metrics for the workload profiler.

Columns

The columns for `wlp_unique_collist_for_mda_archive` are:

Name	Datatype	Null	Identity	Description
<code>wlp_uc_mda_ctr</code>	<code>int</code>	No	Yes	Unique ID for each entry in the table.
<code>wlp_uc_mda_name</code>	<code>varchar(30)</code>	No		Name of the metrics table in which the column exists.
<code>wlp_uc_mda_uniq_col1</code>	<code>varchar(30)</code>	Yes		Name of the metrics table column.
<code>wlp_uc_mda_uniq_col2</code>	<code>varchar(30)</code>	Yes		Name of the metrics table column.
<code>wlp_uc_mda_uniq_col3</code>	<code>varchar(30)</code>	Yes		Name of the metrics table column.
<code>wlp_uc_mda_uniq_col4</code>	<code>varchar(30)</code>	Yes		Name of the metrics table column.
<code>wlp_uc_mda_uniq_col5</code>	<code>varchar(30)</code>	Yes		Name of the metrics table column.
<code>wlp_uc_mda_where_clause</code>	<code>varchar(256)</code>	Yes		where clause used to select the rows.
<code>wlp_uc_mda_periodic_sample</code>	<code>bit</code>	No		Indicates if the monitoring table is periodically monitored.

Indexes

- Unique clustered index on `<jid>`
- Unique nonclustered index on `<jname>`

Referenced by

5.11 wlp_exec_cmd_generate_deltas

Lists the archive table columns that generate changes during a sampling period.

Columns

The columns for `wlp_exec_cmd_generate_deltas` are:

Name	Datatype	Null	Identity	Description
<code>wlxc_gd_colctr</code>	<code>int</code>	No	Yes	Unique ID for each entry in the table.
<code>wlxc_gd_montable</code>	<code>varchar(30)</code>	No		Name of the metrics table in which the column exists.
<code>wlxc_gd_colname</code>	<code>varchar(30)</code>	No		Name of the metrics table column.
<code>wlxc_gd_delta_colname</code>	<code>varchar(30)</code>	No		Name of the changed column, derived from the original column name by appending <code>_delta</code> .
<code>wlxc_gd_datatype</code>	<code>varchar(30)</code>	No		Column data type.

Indexes

`wlp_exec_cmd_generate_deltas` includes a unique index, `wlxc_gd_1_montable_colnames_ui`, on `wlp_exec_cmd_generate_deltas (wlxc_gd_montable and wlxc_gd_colname)`.

5.12 wlp_afs_inputs

Stores the input values from columns that provide metrics for workload profiling.

Columns

The columns for `wlp_afs_inputs` are:

Name	Datatype	Null	Identity	Description
<code>wlpsi_identity</code>	<code>int</code>	No	Yes	Unique ID for each entry in the table.
<code>wlpsi_wlp_id</code>	<code>int</code>	No		Unique ID of the workload profiling session.
<code>wlpsi_feature</code>	<code>smallint</code>	No		Code name for the feature.
<code>wlpsi_value1 ... wlpsi_value32</code>	<code>float</code>	No		Contains up to 32 input column values from a monitoring table.

5.13 wlp_plan_objscores

Contains the the scores computed for each table, determining if they are suitable for a feature.

Columns

The columns for `wlp_plan_objscores` are:

Name	Datatype	Null	Identity	Description
<code>wlps_identity</code>	<code>int</code>	No	Yes	Unique ID for each entry in the table.
<code>wlps_wlp_id</code>	<code>int</code>	No		Unique ID of the workload profiler.
<code>wlps_feature_remap</code>	<code>int</code>	No		Bit map encoding of the feature code

Name	Datatype	Null	Identity	Description
wlps_drc_score	float	Yes		Score for datarow caching
wlps_mvcc_score	float	Yes		Score for multiversion concurrency control.
wlps_odmvc_score	float	Yes		Score for on-disk multiversion concurrency control.
wlps_lfb_score	float	Yes		Score for latch free btree
wlps_hcb_score	float	Yes		Score for hash-cache btree index.
wlps_nvc_score	float	Yes		Score for non-volatile cache
wlps_ftr07_score	float	Yes		Score for additional feature
wlps_ftr08_score	float	Yes		Score for additional feature
wlps_ftr09_score	float	Yes		Score for additional feature
wlps_ftr10_score	float	Yes		Score for additional feature
	float	Yes		Score for additional feature

5.14 WLP_spaceusage_object

Tracks space growth rates and usage that are important for evaluating the target feature.

Columns

The columns for `WLP_spaceusage_object` are:

Name	Datatype	Null	Identity	Description
VersionNum	int	No	Yes	Version number from <code><@@version></code> output that identifies the server used for archiving the metrics.
ESDNum	int	No		ESD number from <code><@@version></code> output that identifies the server used for archiving the metrics
EBFNum	int	No		EBF number from <code><@@version></code> output that identifies the server used for archiving the metrics
ArchiveDate Time	datetime	No		Date and time the archive was generated
DBName	varchar(30)	No		Database name (this is the target database of the workload Profiler).
OwnerName	varchar(30)	No		Name of the object owner.
TableName	varchar(255)	No		Name of the table.
IndexName	varchar(255)	Yes		Name of the index (can be NULL for the table's data layer).
PtnName	varchar(255)	No		Name of the partition
Id	int	No		Object ID
IndId	int	No		Index ID
DataPtnId	int	No		Data partition ID
PtnId	int	No		Partition ID. Uses the value from <code>DataPtnId</code> for the data layer, and the index partition ID for index rows.
WLP_ID	int	No		Workload profiler ID. Identifies all metrics gathered and results generated during a single execution of the workload profiler. Followed by an analysis phase.

Name	Datatype	Null	Identity	Description
WLP_Ctr	int	No		Running counter to archive metrics rows from periodic sampling for a specific workload profiler ID. Useful for generating changes in metrics between consecutive WLP_Ctr rows.

Indexes

WLP_spaceusage_object contains unique indexes on:

- wlp_spusage_obj_ids_ui ON WLP_spaceusage_object (WLP_ID, DBName, Id, IndId, PtnId, WLP_Ctr)
- wlp_spusage_obj_names_ui ON WLP_spaceusage_object (WLP_ID, DBName, TableName, PartitionName, WLP_Ctr)

5.15 WLP_spaceusage_syslogs

Tracks log space usage in the syslogs system table.

Columns

The columns for WLP_spaceusage_syslogs are:

Name	Datatype	Null	Identity	Description
VersionNum	int	No		Version number from <@@version> output that identifies the server used for archiving the metrics.
ESDNum	int	No		ESD number from <@@version> output that identifies the server used for archiving the metrics
EBFNum	int	No		EBF number from <@@version> output that identifies the server used for archiving the metrics
ArchiveDateTime	datetime	No		Date and time the archive was generated
DBName	varchar(30)	No		Database name (this is the target database of the workload Profiler).

Name	Datatype	Null	Identity	Description
OwnerName	varchar(30)	No		Name of the object owner (by default, dbo)
TableName	varchar(255)	No		Name of the table (by default, syslogs).
Id	int	No		Object ID (a value of 8 for syslogs).
WLP_ID	int	No		Workload profiler ID. Identifies all metrics gathered and results generated during a single execution of the workload profiler. Followed by an analysis phase.
WLP_Ctr	int	No		Running counter to archive metrics rows from periodic sampling for a specific workload profiler ID. Useful for generating changes in metrics between consecutive WLP_Ctr rows.

Indexes

WLP_spaceusage_syslogs contains unique indexes on:

- wlp_spusage_logs_id_ui ON WLP_spaceusage_syslogs (WLP_ID, DBName, WLP_Ctr)
- wlp_spusage_logs_name_ui ON WLP_spaceusage_syslogs (WLP_ID, DBName, WLP_Ctr)

5.16 WLP_spaceusage_sysimrslogs

Tracks log space usage in the sysimrslogs system table.

Columns

The columns for WLP_spaceusage_sysimrslogs are:

Name	Datatype	Null	Identity	Description
VersionNum	int	No		Version number from <@@version> output that identifies the server used for archiving the metrics.
ESDNum	int	No		ESD number from <@@version> output that identifies the server used for archiving the metrics

Name	Datatype	Null	Identity	Description
EBFNum	int	No		EBF number from <@@version> output that identifies the server used for archiving the metrics
ArchiveDateTime	datetime	No		Date and time the archive was generated
DBName	varchar(30)	No		Database name (this is the target database of the workload Profiler).
OwnerName	varchar(30)	No		Name of the object owner (by default, dbo)
TableName	varchar(255)	No		Name of the table (by default, sysimrslogs).
Id	int	No		Object ID (a value of 59 for sysimrslogs).
WLP_ID	int	No		Workload profiler ID. Identifies all metrics gathered and results generated during a single execution of the workload profiler. Followed by an analysis phase.
WLP_Ctr	int	No		Running counter to archive metrics rows from periodic sampling for a specific workload profiler ID. Useful for generating changes in metrics between consecutive WLP_Ctr rows.

Indexes

WLP_spaceusage_syslogs contains unique indexes on:

- wlp_spusage_imrslogs_id_ui ON WLP_spaceusage_sysimrslogs (WLP_ID, DBName, WLP_Ctr)
- wlp_spusage_imrslogs_name_ui ON WLP_spaceusage_sysimrslogs (WLP_ID, DBName, WLP_Ctr)

5.17 WLP_monOpenObjectActivity

Provides metrics for open objects to the workload profiler.

Columns

The columns for `WLP_monOpenObjectActivity` are:

Name	Datatype	Null	Identity	Description
VersionNum	int	No		Version number from <code><@@version></code> output that identifies the server used for archiving the metrics.
SPNum	int	No		Service pack number from <code><@@version></code> output that identifies the server used for archiving the metrics
PLNum	int	No		PL number from <code><@@version></code> output that identifies the server used for archiving the metrics
EBFNum	datetime	No		EBF number from <code><@@version></code> output that identifies the server used for archiving the metrics
ArchiveDate	varchar(30)	No		Date and time the archive was generated
WLP_ID	int	No		Workload profiler ID. Identifies all metrics gathered and results generated during a single execution of the workload profiler. Followed by an analysis phase.
WLP_Ctr	int	No		Running counter to archive metrics rows from periodic sampling for a specific workload profiler ID. Useful for generating changes in metrics between consecutive <code>WLP_Ctr</code> rows.

Indexes

`WLP_monOpenObjectActivity` contains unique indexes on:

- `wlp_monOOA_ids_ui` ON `WLP_monOpenObjectActivity` (`WLP_ID`, `DBID`, `ObjectID`, `IndexID`, `WLP_Ctr`)
- `wlp_monOOA_names_ui` ON `WLP_monOpenObjectActivity` (`WLP_ID`, `DBName`, `ObjectName`, `IndexName`, `WLP_Ctr`)

5.18 WLP_monDataCache

Provides metrics for data caches to the workload profiler.

Columns

The columns for `WLP_monDataCache` are:

Name	Datatype	Null	Identity	Description
<code>VersionNum</code>	<code>int</code>	No		Version number from <code><@@version></code> output that identifies the server used for archiving the metrics.
<code>SPNum</code>	<code>int</code>	No		Service pack number from <code><@@version></code> output that identifies the server used for archiving the metrics
<code>PLNum</code>	<code>int</code>	No		PL number from <code><@@version></code> output that identifies the server used for archiving the metrics
<code>EBFNum</code>	<code>datetime</code>	No		EBF number from <code><@@version></code> output that identifies the server used for archiving the metrics
<code>ArchiveDate</code>	<code>varchar(30)</code>	No		Date and time the archive was generated
<code>WLP_ID</code>	<code>int</code>	No		Workload profiler ID. Identifies all metrics gathered and results generated during a single execution of the workload profiler. Followed by an analysis phase.
<code>WLP_Ctr</code>	<code>int</code>	No		Running counter to archive metrics rows from periodic sampling for a specific workload profiler ID. Useful for generating changes in metrics between consecutive <code>WLP_Ctr</code> rows.

Indexes

`WLP_monDataCache` contains unique indexes on:

- `wlp_monDC_ids_ui` ON `WLP_monDataCache` (`WLP_ID`, `CacheID`, `WLP_Ctr`)
- `wlp_monDC_names_ui` ON `WLP_monDataCache` (`WLP_ID`, `CacheName`, `WLP_Ctr`)

5.19 WLP_monSysWaits

Provides a server-wide view of the statistics for events on which processes have waited to the workload profiler.

Columns

The columns for `WLP_monSysWaits` are:

Name	Datatype	Null	Identity	Description
VersionNum	int	No		Version number from <code><@@version></code> output that identifies the server used for archiving the metrics.
SPNum	int	No		Service pack number from <code><@@version></code> output that identifies the server used for archiving the metrics
PLNum	int	No		PL number from <code><@@version></code> output that identifies the server used for archiving the metrics
EBFNum	datetime	No		EBF number from <code><@@version></code> output that identifies the server used for archiving the metrics
ArchiveDate	varchar(30)	No		Date and time the archive was generated
WLP_ID	int	No		Workload profiler ID. Identifies all metrics gathered and results generated during a single execution of the workload profiler. Followed by an analysis phase.
WLP_Ctr	int	No		Running counter to archive metrics rows from periodic sampling for a specific workload profiler ID. Useful for generating changes in metrics between consecutive <code>WLP_Ctr</code> rows.

Indexes



`WLP_monSysWaits` contains a unique index, `wlp_monSW_ids_ui`, on `WLP_monSysWaits` (`WLP_ID`, `WaitEventID`, `WLP_Ctr`) on:

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.