



Sybase® Adaptive Server™ Enterprise
Reference Manual
Volume 3: Datatypes and System Tables

Adaptive Server™

Document ID: 32416-01-1150

September 1997

Copyright Information

Copyright © 1989–1997 by Sybase, Inc. All rights reserved.

Sybase, Inc., 6475 Christie Avenue, Emeryville, CA 94608.

Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, the Sybase logo, APT-FORMS, Certified SYBASE Professional, Data Workbench, First Impression, InfoMaker, PowerBuilder, Powersoft, Replication Server, S-Designer, SQL Advantage, SQL Debug, SQL SMART, SQL Solutions, Transact-SQL, VisualWriter, and VQL are registered trademarks of Sybase, Inc. Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Monitor, ADA Workbench, AnswerBase, Application Manager, AppModeler, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, APT Workbench, Backup Server, BayCam, Bit-Wise, ClearConnect, Client-Library, Client Services, CodeBank, Column Design, Connection Manager, DataArchitect, Database Analyzer, DataExpress, Data Pipeline, DataWindow, DB-Library, dbQ, Developers Workbench, DirectConnect, Distribution Agent, Distribution Director, Dynamo, Embedded SQL, EMS, Enterprise Client/Server, Enterprise Connect, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, EWA, Formula One, Gateway Manager, GeoPoint, ImpactNow, InformationConnect, InstaHelp, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, Logical Memory Manager, MainframeConnect, Maintenance Express, MAP, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, MethodSet, Net-Gateway, NetImpact, Net-Library, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT-Execute, PC DB-Net, PC Net Library, Power++, Power AMC, PowerBuilt, PowerBuilt with PowerBuilder, PowerDesigner, Power J, PowerScript, PowerSite, PowerSocket, Powersoft Portfolio, Power Through Knowledge, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Quickstart Datamart, Replication Agent, Replication Driver, Replication Server Manager, Report-Execute, Report Workbench, Resource Manager, RW-DisplayLib, RW-Library, SAFE, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILLS, smart.partners, smart.parts, smart.script, SQL Anywhere, SQL Central, SQL Code Checker, SQL Edit, SQL Edit/TPU, SQL Modeler, SQL Remote, SQL Server, SQL Server/CFT, SQL Server/DBM, SQL Server Manager, SQL Server SNMP SubAgent, SQL Station, SQL Toolset, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Gateways, Sybase IQ, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase Synergy Program, Sybase Virtual Server Architecture, Sybase User Workbench, SybaseWare, SyBooks, System 10, System 11, the System XI logo, SystemTools, Tabular Data Stream, The Architecture for Change, The Enterprise Client/Server Company, The Model for Client/Server Solutions, The Online Information Center, Translation Toolkit, Turning Imagination Into Reality, Unibom, Unilib, Uninull, Unisep, Unistring, Viewer, Visual Components, VisualSpeller, WarehouseArchitect, WarehouseNow, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, and XA-Server are trademarks of Sybase, Inc. 6/97

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Table of Contents

Chapters 1 through 6 are in Volumes 1 and 2 of the *Adaptive Server Reference Manual*.

About This Book

How to Use This Book	xi
----------------------------	----

7. System and User-Defined Datatypes

Datatype Categories	7-1
Range and Storage Size	7-1
Declaring the Datatype of a Column, Variable, or Parameter	7-3
Datatype of Mixed-Mode Expressions	7-5
Converting One Datatype to Another	7-7
Standards and Compliance	7-8
<i>Exact Numeric Datatypes</i>	7-9
<i>Approximate Numeric Datatypes</i>	7-13
<i>Money Datatypes</i>	7-15
<i>timestamp Datatype</i>	7-17
<i>Date/time Datatypes</i>	7-19
<i>Character Datatypes</i>	7-24
<i>Binary Datatypes</i>	7-28
<i>bit Datatype</i>	7-31
<i>sysname Datatype</i>	7-32
<i>text and image Datatypes</i>	7-33
<i>User-Defined Datatypes</i>	7-39

8. System Tables

Lists of System Tables	8-1
<i>sysalternates</i>	8-5
<i>sysattributes</i>	8-6
<i>sysauditoptions</i>	8-9
<i>sysaudits_01, sysaudits_02...sysaudits_08</i>	8-10
<i>syscharsets</i>	8-27
<i>syscolumns</i>	8-29
<i>syscomments</i>	8-31

<i>sysconfigures</i>	8-33
<i>sysconstraints</i>	8-34
<i>syscurconfigs</i>	8-35
<i>sysdatabases</i>	8-37
<i>sysdepends</i>	8-40
<i>sysdevices</i>	8-41
<i>sysengines</i>	8-43
<i>sysgams</i>	8-44
<i>sysindexes</i>	8-45
<i>syskeys</i>	8-48
<i>syslanguages</i>	8-50
<i>syslisteners</i>	8-52
<i>syslocks</i>	8-53
<i>sysloginroles</i>	8-55
<i>syslogins</i>	8-56
<i>syslogs</i>	8-58
<i>syslogshold</i>	8-59
<i>sysmessages</i>	8-61
<i>sysmonitors</i>	8-62
<i>sysobjects</i>	8-63
<i>syspartitions</i>	8-66
<i>sysprocedures</i>	8-67
<i>sysprocesses</i>	8-68
<i>sysprotects</i>	8-71
<i>sysreferences</i>	8-73
<i>sysremotelogins</i>	8-75
<i>sysresourcelimits</i>	8-76
<i>sysroles</i>	8-77
<i>syssecmechs</i>	8-78
<i>syssegments</i>	8-79
<i>syservers</i>	8-80
<i>sysrvroles</i>	8-82
<i>systhresholds</i>	8-83
<i>systimeranges</i>	8-84
<i>systypes</i>	8-85
<i>sysusages</i>	8-87
<i>sysusermessages</i>	8-88
<i>sysusers</i>	8-89

A. Expressions, Identifiers, and Wildcard Characters

Expressions	A-1
Identifiers	A-11
Pattern Matching with Wildcard Characters	A-16

B. Reserved Words

Transact-SQL Keywords	B-1
SQL92 Keywords	B-3
Potential SQL92 Reserved Words	B-5

C. SQLSTATE Codes and Messages

Warnings	C-1
Exceptions	C-1

Index

List of Tables

Table 7-1:	Datatype categories	7-1
Table 7-2:	Range and storage size for system datatypes	7-2
Table 7-3:	Precision and scale after arithmetic operations.....	7-6
Table 7-4:	Automatic conversion of fixed-length datatypes.....	7-7
Table 7-5:	Integer datatypes	7-9
Table 7-6:	Valid integer values	7-9
Table 7-7:	Invalid integer values.....	7-10
Table 7-8:	Valid decimal values	7-11
Table 7-9:	Invalid decimal values	7-12
Table 7-10:	Approximate numeric datatypes.....	7-14
Table 7-11:	Money datatypes	7-15
Table 7-12:	Transact-SQL datatypes for storing dates and times.....	7-19
Table 7-13:	Date formats for datetime and smalldatetime datatypes	7-20
Table 7-14:	Examples of datetime entries.....	7-22
Table 7-15:	Character datatypes.....	7-24
Table 7-16:	Storage of text and image data	7-35
Table 7-17:	text and image global variables.....	7-37
Table 8-1:	Object types for attributes	8-7
Table 8-2:	Items in the extrainfo field	8-11
Table 8-3:	Values in event and extrainfo column	8-12
Table 8-4:	status control bits in the sysdatabases table	8-38
Table 8-5:	status2 control bits in the sysdatabases table	8-38
Table 8-6:	status control bits in the sysdevices table	8-41
Table 8-7:	status2 control bits in the sysindexes table	8-46
Table 8-8:	status control bits in the sysindexes table	8-47
Table 8-9:	type control bits in the syslocks table	8-53
Table 8-10:	fid column values in the syslocks table	8-54
Table 8-11:	context column values in the syslocks table	8-54
Table 8-12:	status control bits in the syslogins table	8-57
Table 8-13:	sysstat2 control bits in the sysobjects table	8-64
Table 8-14:	type control bits in the sysprocedures table	8-67
Table 8-15:	sysprocesses status column values	8-69
Table 8-16:	status control bits in the syssservers table.....	8-80
Table 8-17:	Server categories in the syssservers table	8-80
Table 8-18:	Datatype names, hierarchy, types, and usertypes.....	8-86
Table A-1:	Types of expressions used in syntax statements	A-1
Table A-2:	Arithmetic operators	A-3
Table A-3:	Truth tables for bitwise operations.....	A-3

Table A-4:	Examples of bitwise operations.....	A-4
Table A-5:	Comparison operators	A-5
Table A-6:	Truth tables for logical expressions.....	A-9
Table A-7:	Wildcard characters used with like.....	A-18
Table A-8:	Using square brackets to search for wildcard characters.....	A-21
Table A-9:	Using the escape clause	A-22
Table B-1:	Transact-SQL keywords.....	B-1
Table B-2:	SQL92 keywords.....	B-3
Table B-3:	Potential SQL92 reserved words	B-5
Table C-1:	SQLSTATE warnings.....	C-1
Table C-2:	Cardinality violations.....	C-2
Table C-3:	Data exceptions.....	C-3
Table C-4:	Integrity constraint violations.....	C-3
Table C-5:	Invalid cursor states	C-4
Table C-6:	Syntax errors and access rule violations.....	C-5
Table C-7:	Transaction rollbacks.....	C-6
Table C-8:	with check option violation.....	C-7

About This Book

The *Adaptive Server Reference Manual* is a three-volume guide to Sybase® Adaptive Server™ Enterprise and the Transact-SQL® language. This volume includes information about datatypes, system tables, reserved words, SQLSTATE errors, and an index for all three volumes. Volume 1, *Commands and Functions*, contains information about Transact-SQL commands and built-in functions. Volume 2, *Procedures*, includes information about system procedures, catalog stored procedures, system extended stored procedures, and `dbcc` stored procedures.

For information about the intended audience of this book, related documents, other sources of information, conventions used in this manual, and help, refer to “About This Book” in Volume 1.

How to Use This Book

This manual consists of the following:

- Chapter 7, “System and User-Defined Datatypes,” describes the system and user-defined datatypes that are supplied with Adaptive Server and indicates how to use them to create user-defined datatypes.
- Chapter 8, “System Tables,” contains information about all system tables in the *master* database, the auditing database, and any user databases (such as *pubs2*).
- Appendix A, “Expressions, Identifiers, and Wildcard Characters” contains information about using the Transact-SQL language.
- Appendix B, “Reserved Words,” contains information about the Transact-SQL and SQL92 keywords.
- Appendix C, “SQLSTATE Codes and Messages,” contains information about Adaptive Server’s SQLSTATE status codes and the associated messages.
- The Index contains entries for all three volumes of the *Adaptive Server Reference Manual*.

System and User- Defined Datatypes

7

System and User-Defined Datatypes

This chapter describes the Transact-SQL

datatypes. Datatypes specify the type, size, and storage format of columns, stored procedure parameters, and local variables.

Datatype Categories

Adaptive Server provides a number of system datatypes, as well as the user-defined datatypes *timestamp* and *sysname*.

Adaptive Server datatypes fall into the categories listed in Table 7-1. Each category is described in a section of this chapter.

Table 7-1: Datatype categories

Category	Used For
Exact Numeric Datatypes	Numeric values (both integers and numbers with a decimal portion) that must be represented exactly
Approximate Numeric Datatypes	Numeric data that can tolerate rounding during arithmetic operations
Money Datatypes	Monetary data
timestamp Datatype	Tables that are browsed in Client-Library™ applications
Date/time Datatypes	Date and time information
Character Datatypes	Strings consisting of letters, numbers, and symbols
Binary Datatypes	Raw binary data, such as pictures, in a hexadecimal-like notation
bit Datatype	True/false and yes/no type data
sysname Datatype	System tables
text and image Datatypes	Printable characters or hexadecimal-like data that requires more than 255 bytes of storage
User-Defined Datatypes	Defining objects that inherit the rules, default, null type, IDENTITY property, and base datatype

Range and Storage Size

Table 7-2 lists the system-supplied datatypes and their synonyms and provides information about the range of valid values and storage size for each. For simplicity, the datatypes are printed in

lowercase characters, although Adaptive Server allows you to use either uppercase or lowercase characters for system datatypes. (User-defined datatypes, such as *timestamp*, are **case sensitive**.) Most Adaptive Server-supplied datatypes are not reserved words and can be used to name other objects.

Table 7-2: Range and storage size for system datatypes

Datatypes	Synonyms	Range	Bytes of Storage
Exact numeric datatypes			
<i>tinyint</i>		0 to 255	1
<i>smallint</i>		-2^{15} (-32,768) to $2^{15}-1$ (32,767)	2
<i>int</i>	<i>integer</i>	-2^{31} (-2,147,483,648) to $2^{31}-1$ (2,147,483,647)	4
<i>numeric (p, s)</i>		-10^{38} to $10^{38}-1$	2 to 17
<i>decimal (p, s)</i>	<i>dec</i>	-10^{38} to $10^{38}-1$	2 to 17
Approximate numeric datatypes			
<i>float (precision)</i>		Machine dependent	4 or 8
<i>double precision</i>		Machine dependent	8
<i>real</i>		Machine dependent	4
Money datatypes			
<i>smallmoney</i>		-214,748.3648 to 214,748.3647	4
<i>money</i>		-922,337,203,685,477.5808 to 922,337,203,685,477.5807	8
Date/time datatypes			
<i>smalldatetime</i>		January 1, 1900 to June 6, 2079	4
<i>datetime</i>		January 1, 1753 to December 31, 9999	8
Character datatypes			
<i>char(n)</i>	<i>character</i>	255 characters or fewer	<i>n</i>
<i>varchar(n)</i>	<i>char[acter] varying</i>	255 characters or fewer	actual entry length
<i>nchar(n)</i>	<i>national char[acter]</i>	255 characters or fewer	<i>n</i> * @@ <i>ncharsize</i>
<i>nvarchar(n)</i>	<i>nchar varying, national char[acter] varying</i>	255 characters or fewer	<i>n</i>

Table 7-2: Range and storage size for system datatypes (continued)

Datatypes	Synonyms	Range	Bytes of Storage
Binary datatypes			
<i>binary(n)</i>		255 bytes or fewer	<i>n</i>
<i>varbinary(n)</i>		255 bytes or fewer	actual entry length
Bit datatype			
<i>bit</i>		0 or 1	1 (1 byte holds up to 8 <i>bit</i> columns)
Text and image datatypes			
<i>text</i>		2 ³¹ - 1 (2,147,483,647) bytes or fewer	0 until initialized, then a multiple of 2K
<i>image</i>		2 ³¹ - 1 (2,147,483,647) bytes or fewer	0 until initialized, then a multiple of 2K

Declaring the Datatype of a Column, Variable, or Parameter

You must declare the datatype for a column, local variable, or parameter. The datatype can be any of the system-supplied datatypes or any user-defined datatype in the database.

Declaring the Datatype for a Column in a Table

Use the following syntax to declare the datatype of a new column in an alter table or create table statement:

```
create table [[database.]owner.]table_name
  (column_name datatype [identity | not null | null]
    [, column_name datatype [identity | not null |
    null]]...)

alter table [[database.]owner.]table_name
  add column_name datatype [identity | null
    [, column_name datatype [identity | null]]...
```

For example:

```
create table sales_daily
  (stor_id char(4)not null,
   ord_num numeric(10,0)identity,
   ord_amt money null)
```

Declaring the Datatype for a Local Variable in a Batch or Procedure

Use the following syntax to declare the datatype for a local variable in a batch or stored procedure:

```
declare @variable_name datatype
        [, @variable_name datatype]...
```

For example:

```
declare @hope money
```

Declaring the Datatype for a Parameter in a Stored Procedure

Use the following syntax to declare the datatype for a parameter in a stored procedure:

```
create procedure [owner.]procedure_name [;number]
    [[(@parameter_name datatype [= default] [output]
      [,@parameter_name datatype [= default]
      [output]]...[])]
    [with recompile]
    as SQL_statements
```

For example:

```
create procedure auname_sp @auname varchar(40)
as
    select au_lname, title, au_ord
    from authors, titles, titleauthor
    where @auname = au_lname
    and authors.au_id = titleauthor.au_id
    and titles.title_id = titleauthor.title_id
```

Determining the Datatype of a Literal

You cannot declare the datatype of a literal. Adaptive Server treats all character literals as *varchar*. Numeric literals entered with E notation are treated as *float*; all others are treated as exact numerics:

- Literals between $2^{31} - 1$ and -2^{31} with no decimal point are treated as *integer*.
- Literals that include a decimal point, or that fall outside the range for integers, are treated as *numeric*.

► Note

To preserve backward compatibility, use E notation for numeric literals that should be treated as *float*.

Datatype of Mixed-Mode Expressions

When you perform concatenation or mixed-mode arithmetic on values with different datatypes, Adaptive Server must determine the datatype, length, and precision of the result.

Determining the Datatype Hierarchy

Each system datatype has a **datatype hierarchy**, which is stored in the *systypes* system table. User-defined datatypes inherit the hierarchy of the system datatype on which they are based.

The following query ranks the datatypes in a database by hierarchy. In addition to the information shown below, your query results will include information about any user-defined datatypes in the database:

```
select name,hierarchy
from systypes
order by hierarchy
```

name	hierarchy
-----	-----
floatn	1
float	2
datetimn	3
datetime	4
real	5
numericn	6
numeric	7
decimaln	8
decimal	9
moneyn	10
money	11
smallmoney	12
smalldatetime	13
intn	14
int	15
smallint	16
tinyint	17

bit	18
varchar	19
sysname	19
nvarchar	19
char	20
nchar	20
varbinary	21
timestamp	21
binary	22
text	23
image	24
(28 rows affected)	

The datatype hierarchy determines the results of computations using values of different datatypes. The result value is assigned the datatype that is closest to the top of the list.

In the following example, *qty* from the *sales* table is multiplied by *royalty* from the *roysched* table. *qty* is a *smallint*, which has a hierarchy of 16; *royalty* is an *int*, which has a hierarchy of 15. Therefore, the datatype of the result is an *int*.

```
smallint(qty) * int(royalty) = int
```

Determining Precision and Scale

For *numeric* and *decimal* datatypes, each combination of precision and scale is a distinct Adaptive Server datatype. If you perform arithmetic on two *numeric* or *decimal* values:

- *n1* with precision *p1* and scale *s1*, and
- *n2* with precision *p2* and scale *s2*

Adaptive Server determines the precision and scale of the results as shown in Table 7-3:

Table 7-3: Precision and scale after arithmetic operations

Operation	Precision	Scale
$n1 + n2$	$\max(s1, s2) + \max(p1 - s1, p2 - s2) + 1$	$\max(s1, s2)$
$n1 - n2$	$\max(s1, s2) + \max(p1 - s1, p2 - s2) + 1$	$\max(s1, s2)$
$n1 * n2$	$s1 + s2 + (p1 - s1) + (p2 - s2) + 1$	$s1 + s2$
$n1 / n2$	$\max(s1 + p2 + 1, 6) + p1 - s1 + p2$	$\max(s1 + p2 - s2 + 1, 6)$

Converting One Datatype to Another

Many conversions from one datatype to another are handled automatically by Adaptive Server. These are called implicit conversions. Other conversions must be performed explicitly with the `convert`, `intohex`, and `hextoint` functions. See “Datatype Conversion Functions” in Chapter 2, “Transact-SQL Functions,” for details about datatype conversions supported by Adaptive Server.

Automatic Conversion of Fixed-Length NULL Columns

Only columns with variable-length datatypes can store null values. When you create a NULL column with a fixed-length datatype, Adaptive Server automatically converts it to the corresponding variable-length datatype. Adaptive Server does not inform the user of the datatype change.

Table 7-4 lists the fixed- and variable-length datatypes to which they are converted. Certain variable-length datatypes, such as *money*n, are reserved datatypes; you cannot use them to create columns, variables, or parameters:

Table 7-4: Automatic conversion of fixed-length datatypes

Original Fixed-Length Datatype	Converted To
<i>char</i>	<i>varchar</i>
<i>nchar</i>	<i>nvarchar</i>
<i>binary</i>	<i>varbinary</i>
<i>datetime</i>	<i>datetime</i>
<i>float</i>	<i>floatn</i>
<i>int</i> , <i>smallint</i> , and <i>tinyint</i>	<i>intn</i>
<i>decimal</i>	<i>decimaln</i>
<i>numeric</i>	<i>numericn</i>
<i>money</i> and <i>smallmoney</i>	<i>moneyn</i>

Handling Overflow and Truncation Errors

The `arithabort` option determines how Adaptive Server behaves when an arithmetic error occurs. The two `arithabort` options, `arith_abort` and `arith_abort numeric_truncation`, handle different types of arithmetic errors. You can set each option independently, or set both options with a single `set arithabort on` or `set arithabort off` statement.

- `arithabort arith_overflow` specifies behavior following a divide-by-zero error or a loss of precision during either an explicit or an implicit datatype conversion. This type of error is considered serious. The default setting, `arithabort arith_overflow on`, rolls back the entire transaction in which the error occurs. If the error occurs in a batch that does not contain a transaction, `arithabort arith_overflow on` does not roll back earlier commands in the batch, but Adaptive Server does not execute any statements that follow the error-generating statement in the batch.

If you set `arithabort arith_overflow off`, Adaptive Server aborts the statement that causes the error, but continues to process other statements in the transaction or batch.

- `arithabort numeric_truncation` specifies behavior following a loss of scale by an exact numeric datatype during an implicit datatype conversion. (When an explicit conversion results in a loss of scale, the results are truncated without warning.) The default setting, `arithabort numeric_truncation on`, aborts the statement that causes the error but continues to process other statements in the transaction or batch. If you set `arithabort numeric_truncation off`, Adaptive Server truncates the query results and continues processing.

The `arithignore` option determines whether Adaptive Server prints a warning message after an overflow error. By default, the `arithignore` option is turned off. This causes Adaptive Server to display a warning message after any query that results in numeric overflow. To ignore overflow errors, use `set arithignore on`.

► **Note**

The `arithabort` and `arithignore` options were redefined for release 10.0. If you use these options in your applications, examine them to be sure they still produce the desired effects.

Standards and Compliance

Standard	Compliance Level
SQL92	Transact-SQL provides the <i>smallint</i> , <i>int</i> , <i>numeric</i> , <i>decimal</i> , <i>float</i> , <i>double precision</i> , <i>real</i> , <i>char</i> , and <i>varchar</i> SQL92 datatypes. The <i>tinyint</i> , <i>binary</i> , <i>varbinary</i> , <i>image</i> , <i>bit</i> , <i>datetime</i> , <i>smalldatetime</i> , <i>money</i> , <i>smallmoney</i> , <i>nchar</i> , <i>nvarchar</i> , <i>sysname</i> , <i>text</i> , <i>timestamp</i> , and user-defined datatypes are Transact-SQL extensions.

Exact Numeric Datatypes

Function

Use the exact numeric datatypes when it is important to represent a value exactly. Adaptive Server provides exact numeric types for both integers (whole numbers) and numbers with a decimal portion.

Integer Types

Adaptive Server provides three exact numeric datatypes, *tinyint*, *smallint*, and *int* (or *integer*), to store integers. Choose among the integer types based on the expected size of the numbers to be stored. Internal storage size varies by type, as shown in Table 7-5:

Table 7-5: Integer datatypes

Datatype	Stores	Bytes of Storage
<i>tinyint</i>	Whole numbers between 0 and 255, inclusive. (Negative numbers are not permitted.)	1
<i>smallint</i>	Whole numbers between -2^{15} and $2^{15} - 1$ (-32,768 and 32,767), inclusive.	2
<i>int[eger]</i>	Whole numbers between -2^{31} and $2^{31} - 1$ (-2,147,483,648 and 2,147,483,647), inclusive.	4

Entering Integer Data

Enter integer data as a string of digits without commas. Integer data can include a decimal point as long as all digits to the right of the decimal point are zeros. The *smallint* and *integer* types can be preceded by an optional plus or minus sign; the *tinyint* type can be preceded by an optional plus sign.

Table 7-6 shows some valid entries for a column with a datatype of *integer* and indicates how isql displays these values:

Table 7-6: Valid integer values

Value Entered	Value Displayed
2	2
+2	2
-2	-2
2.	2
2.000	2

Table 7-7 lists some invalid entries for an *integer* column:

Table 7-7: Invalid integer values

Value Entered	Type of Error
2,000	Commas not allowed.
2-	Minus sign should precede digits.
3.45	Digits to the right of the decimal point are nonzero digits.

Decimal Datatypes

Adaptive Server provides two other exact numeric datatypes, *numeric* and *dec[imal]*, for numbers that include decimal points. Data stored in *numeric* and *decimal* columns is packed to conserve disk space, and preserves its accuracy to the least significant digit after arithmetic operations. The *numeric* and *decimal* datatypes are identical in all respects but one: only *numeric* datatypes with a scale of 0 can be used for the IDENTITY column.

Specifying Precision and Scale

The *numeric* and *decimal* datatypes accept two optional parameters, *precision* and *scale*, enclosed in parentheses and separated by a comma:

datatype [(*precision* [, *scale*])]

Adaptive Server treats each combination of precision and scale as a distinct datatype. For example, *numeric*(10,0) and *numeric*(5,0) are two separate datatypes. The *precision* and *scale* determine the range of values that can be stored in a decimal or numeric column:

- The precision specifies the maximum number of decimal digits that can be stored in the column. It includes **all** digits, both to the right and to the left of the decimal point. You can specify precisions ranging from 1 digit to 38 digits or use the default precision of 18 digits.
- The scale specifies the maximum number of digits that can be stored to the right of the decimal point. The scale must be less than or equal to the precision. You can specify a scale ranging from 0 digits to 38 digits or use the default scale of 0 digits.

Storage Size

The storage size for a *numeric* or *decimal* column depends on its precision. The minimum storage requirement is 2 bytes for a 1- or 2-digit column. Storage size increases by approximately 1 byte for each additional 2 digits of precision, up to a maximum of 17 bytes.

Use the following formula to calculate the exact storage size for a *numeric* or *decimal* column:

$$\text{ceiling} (\text{precision} / \log 256) + 1$$

For example, the storage size for a *numeric*(18,4) column is 9 bytes.

Entering Decimal Data

Enter *decimal* and *numeric* data as a string of digits preceded by an optional plus or minus sign and including an optional decimal point. If the value exceeds either the precision or scale specified for the column, Adaptive Server returns an error message. Exact numeric types with a scale of 0 are displayed without a decimal point.

Table 7-8 shows some valid entries for a column with a datatype of *numeric*(5,3) and indicates how these values are displayed by *isql*:

Table 7-8: Valid decimal values

Value Entered	Value Displayed
12.345	12.345
+12.345	12.345
-12.345	-12.345
12.345000	12.345
12.1	12.100
12	12.000

Table 7-9 shows some invalid entries for a column with a datatype of *numeric(5,3)*:

Table 7-9: Invalid decimal values

Value Entered	Type of Error
1,200	Commas not allowed.
12-	Minus sign should precede digits.
12.345678	Too many nonzero digits to the right of the decimal point.

Standards and Compliance

Standard	Compliance Level
SQL92	Transact-SQL provides the <i>smallint</i> , <i>int</i> , <i>numeric</i> , and <i>decimal</i> SQL92 exact numeric datatypes. The <i>tinyint</i> type is a Transact-SQL extension.

Approximate Numeric Datatypes

Function

Use the approximate numeric types, *float*, *double precision*, and *real*, for numeric data that can tolerate rounding during arithmetic operations. The approximate numeric types are especially suited to data that covers a wide range of values. They support all aggregate functions and all arithmetic operations except *modulo*.

Understanding Approximate Numeric Datatypes

Approximate numeric datatypes, used to store floating-point numbers, are inherently slightly inaccurate in their representation of real numbers—hence the name “approximate numeric”. In order to use these datatypes, you must understand and accept their limitations.

Any time a floating-point number is printed or displayed, the printed representation is not quite the same as the stored number, and the stored number is not quite the same as the number that the user entered. Most of the time, the stored representation is close enough, and software makes the printed output look just like the original input, but you must understand the inaccuracy if you plan to use floating-point numbers for calculations, particularly if you will be doing repeated calculations using approximate numeric datatypes—the results can be surprisingly and unexpectedly inaccurate.

The inaccuracy occurs because floating-point numbers are stored in the computer as binary fractions (that is, as a representative number divided by a power of 2), but the numbers we use are decimal (powers of 10). This means that only a very small set of numbers can be stored accurately: 0.75 (3/4) can be stored accurately because it is a binary fraction (4 is a power of 2); 0.2 (2/10) can not (10 is not a power of 2).

Some numbers contain too many digits to store accurately. *double precision* is stored as 8 binary bytes and can represent about 17 digits with reasonable accuracy; *real* is stored as 4 binary bytes and can represent only about 6 digits with reasonable accuracy.

As you can see, if you begin with numbers that are almost right, and do computations with them using other numbers that are almost right, you can easily end up with a result that is not even close. If these considerations are important to your application, consider using an exact numeric datatype.

Range, Precision, and Storage Size

The *real* and *double precision* types are built on types supplied by the operating system. The *float* type accepts an optional binary precision in parentheses. *float* columns with a precision of 1–15 are stored as *real*; those with higher precision are stored as *double precision*.

The range and storage precision for all three types is machine dependent.

Table 7-10 shows the range and storage size for each approximate numeric type. Note that *isql* displays only 6 significant digits after the decimal point and rounds the remainder:

Table 7-10: Approximate numeric datatypes

Datatype	Bytes of Storage
<i>float</i> [(default precision)]	4 for <i>default precision</i> < 16 8 for <i>default precision</i> >= 16
<i>double precision</i>	8
<i>real</i>	4

Entering Approximate Numeric Data

Enter approximate numeric data as a mantissa followed by an optional exponent:

- The mantissa is a signed or unsigned number, with or without a decimal point. The column's binary precision determines the maximum number of binary digits allowed in the mantissa.
- The exponent, which begins with the character "e" or "E," must be a whole number.

The value represented by the entry is the following product:

$$\text{mantissa} * 10^{\text{EXPONENT}}$$

For example, 2.4E3 represents the value 2.4 times 10³, or 2400.

Standards and Compliance

Standard	Compliance Level
SQL92	The <i>float</i> , <i>double precision</i> , and <i>real</i> datatypes are entry level compliant.

Money Datatypes

Function

Use the *money* and *smallmoney* datatypes to store monetary data. You can use these types for U.S. dollars and other decimal currencies, but Adaptive Server provides no means to convert from one currency to another. You can use all arithmetic operations except *modulo*, and all aggregate functions, with *money* and *smallmoney* data.

Accuracy

Both *money* and *smallmoney* are accurate to one ten-thousandth of a monetary unit, but they round values up to two decimal places for display purposes. The default print format places a comma after every three digits.

Range and Storage Size

Table 7-11 summarizes the range and storage requirements for money datatypes:

Table 7-11: Money datatypes

Datatype	Range	Bytes of Storage
<i>money</i>	Monetary values between +922,337,203,685,477.5807 and -922,337,203,685,477.5808	8
<i>smallmoney</i>	Monetary values between +214,748.3647 and -214,748.3648	4

Entering Monetary Values

Monetary values entered with E notation are interpreted as *float*. This may cause an entry to be rejected or to lose some of its precision when it is stored as a *money* or *smallmoney* value.

money and *smallmoney* values can be entered with or without a preceding currency symbol, such as the dollar sign (\$), yen sign (¥), or pound sterling sign (£). To enter a negative value, place the minus sign after the currency symbol. Do not include commas in your entry.

Standards and Compliance

Standard	Compliance Level
SQL92	The <i>money</i> and <i>smallmoney</i> datatypes are Transact-SQL extensions.

timestamp Datatype

Function

Use the user-defined datatype *timestamp* in tables that are to be browsed in Client-Library™ applications (see “Browse Mode” for more information). Adaptive Server updates the *timestamp* column each time its row is modified. A table can have only one column of *timestamp* datatype.

Datatype Definition

timestamp is an Adaptive Server-supplied, user-defined datatype that is defined as *varbinary(8)* NULL. It requires 8 bytes of storage. Because *timestamp* is a user-defined datatype, you cannot use it to define other user-defined datatypes. You cannot use the aggregate functions *sum* or *avg* with the *timestamp* datatype.

Unlike the SQL standard *timestamp* datatype, the Transact-SQL *timestamp* datatype does not hold date and time information, and cannot be converted to a date and time. *timestamp* holds binary-type data like that shown below:

```
timestamp
-----
0x000100000000e51
```

Creating a *timestamp* Column

If you create a column named *timestamp* without specifying a datatype, Adaptive Server defines the column as a *timestamp* datatype:

```
create table testing
(c1 int, timestamp, c2 int)
```

You can also explicitly assign the *timestamp* datatype to a column named *timestamp*:

```
create table testing
(c1 int, timestamp timestamp, c2 int)
```

or to a column with another name:

```
create table testing
(c1 int, t_stamp timestamp, c2 int)
```

You can create a column named *timestamp* and assign it another datatype (although this could be confusing to other users and would

not allow the use of the **browse** functions in Open Client™ or with the **tsequal** function):

```
create table testing
(c1 int, timestamp datetime)
```

Standards and Compliance

Standard	Compliance Level
SQL92	The <i>timestamp</i> datatype is a Transact-SQL extension.

Date/time Datatypes

Function

Use the *datetime* and *smalldatetime* datatypes to store absolute date and time information.

► Note

Adaptive Server also provides a *timestamp* datatype, which stores binary-type information.

Range and Storage Requirements

Table 7-12 summarizes the range and storage requirements for the *datetime* and *smalldatetime* datatypes:

Table 7-12: Transact-SQL datatypes for storing dates and times

Datatype	Range	Bytes of Storage
<i>datetime</i>	January 1, 1753 through December 31, 9999	8
<i>smalldatetime</i>	January 1, 1900 through June 6, 2079	4

Entering *datetime* and *smalldatetime* Data

The *datetime* and *smalldatetime* datatypes consist of a date portion either followed by or preceded by a time portion. (You can omit either the date or the time, or both.) Both *datetime* and *smalldatetime* values must be enclosed in single or double quotes.

- *datetime* columns hold dates between January 1, 1753 and December 31, 9999. *datetime* values are accurate to 1/300 of a second on platforms that support this level of granularity. Storage size is 8 bytes: 4 bytes for the number of days since the base date of January 1, 1900 and 4 bytes for the time of day.
- *smalldatetime* columns hold dates from January 1, 1900 to June 6, 2079, with accuracy to the minute. Storage size is 4 bytes: 2 bytes for the number of days since January 1, 1900 and 2 bytes for the number of minutes since midnight.

Entering the Date Portion of a *datetime* or *smalldatetime* Value

Dates consist of a month, day, and year and can be entered in a variety of formats:

- You can enter the entire date as an unseparated string of 4, 6, or 8 digits, or use slash(/), hyphen (-), or period(.) separators between the date parts.
 - When entering dates as unseparated strings, use the appropriate format for that string length. Use leading zeros for single-digit years, months, and days. Dates entered in the wrong format may be misinterpreted or result in errors.
 - When entering dates with separators, use the set `dateformat` option to determine the expected order of date parts. If the first date part in a separated string is four digits, Adaptive Server interprets the string as `yyyy-mm-dd` format.
- Some date formats accept 2-digit years (*yy*). Dates greater than or equal to 50 are interpreted as 19*yy*; those less than 50 are interpreted as 20*yy*.
- You can specify the month as either a number or a name. Month names and their abbreviations are language-specific and can be entered in uppercase, lowercase, or mixed case.
- If you omit the date portion of a *datetime* or *smalldatetime* value, Adaptive Server uses the default date of January 1, 1900.

Table 7-13 describes the acceptable formats for entering the date portion of a *datetime* or *smalldatetime* value:

Table 7-13: Date formats for *datetime* and *smalldatetime* datatypes

Date Format	Interpretation	Sample Entries	Meaning
4-digit string with no separators	Interpreted as <i>yyyy</i> . Date defaults to Jan 1 of the specified year.	"1947"	Jan 1 1947
6-digit string with no separators	Interpreted as <i>yymmdd</i> . For <i>yy</i> < 50, year is 20 <i>yy</i> . For <i>yy</i> >= 50, year is 19 <i>yy</i> .	"450128" "520128"	Jan 28 2045 Jan 28 1952
8-digit string with no separators	Interpreted as <i>yyyymmdd</i> .	"19940415"	Apr 15 1994

Table 7-13: Date formats for *datetime* and *smalldatetime* datatypes (continued)

Date Format	Interpretation	Sample Entries	Meaning
String consisting of 2-digit month, day, and year separated by slashes, hyphens, or periods, or a combination of the above.	The <i>dateformat</i> and <i>language</i> set options determine the expected order of date parts. For <i>us_english</i> , the default order is <i>mdy</i> . For <i>yy</i> < 50, year is interpreted as 20 <i>yy</i> . For <i>yy</i> >= 50, year is interpreted as 19 <i>yy</i> .	"4/15/94" "4.15.94" "4-15-94" "04.15/94"	All of these entries are interpreted as Apr 15 1994 when the <i>dateformat</i> option is set to <i>mdy</i> .
String consisting of 2-digit month, 2-digit day, and 4-digit year separated by slashes, hyphens, or periods, or a combination of the above.	The <i>dateformat</i> and <i>language</i> set options determine the expected order of date parts. For <i>us_english</i> , the default order is <i>mdy</i> .	"04/15.1994"	Interpreted as Apr 15 1994 when the <i>dateformat</i> option is set to <i>mdy</i> .
Month is entered in character form (either full month name or its standard abbreviation), followed by an optional comma.	If 4-digit year is entered, date parts can be entered in any order.	"April 15, 1994" "1994 15 apr" "1994 April 15" "15 APR 1994"	All of these entries are interpreted as Apr 15 1994.
	If day is omitted, all 4 digits of year must be specified. Day defaults to the first day of the month.	"apr 1994"	Apr 1 1994
	If year is only 2 digits (<i>yy</i>), it is expected to appear after the day. For <i>yy</i> < 50, year is interpreted as 20 <i>yy</i> . For <i>yy</i> >= 50, year is interpreted as 19 <i>yy</i> .	"mar 16 17" "apr 15 94"	Mar 16 2017 Apr 15 1994
The empty string, ""	Date defaults to Jan 1 1900.	""	Jan 1 1900

Entering the Time Portion of a *datetime* or *smalldatetime* Value

The time component of a *datetime* or *smalldatetime* value must be specified as follows:

hours[:*minutes*[:*seconds*[:*milliseconds*]]] [*AM* | *PM*]

- Use 12AM for midnight and 12PM for noon.
- A time value must contain either a colon or an AM or PM signifier. The AM or PM can be entered in uppercase, lowercase, or mixed case.

- The seconds specification can include either a decimal portion preceded by a decimal point or a number of milliseconds preceded by a colon. For example, “12:30:20:1” means twenty seconds and one millisecond past 12:30; “12:30:20.1” means twenty and one-tenth of a second past 12:30.
- If you omit the time portion of a *datetime* or *smalldatetime* value, Adaptive Server uses the default time of 12:00:00:000AM.

Display Formats for *datetime* and *smalldatetime* Values

The display format for *datetime* and *smalldatetime* values is “Mon dd yyyy hh:mmAM” (or “PM”); for example, “Apr 15 1988 10:23PM”. To display seconds and milliseconds, and to obtain additional date styles and date-part orders, use the *convert* function to convert the data to a character string. Adaptive Server may round or truncate millisecond values.

Table 7-14 lists some examples of *datetime* entries and their display values:

Table 7-14: Examples of *datetime* entries

Entry	Value Displayed
“1947”	Jan 1 1947 12:00AM
“450128 12:30:1PM”	Jan 28 2045 12:30PM
“12:30.1PM 450128”	Jan 28 2045 12:30PM
“14:30.22”	Jan 1 1900 2:30PM
“4am”	Jan 1 1900 4:00AM

Finding *datetime* Values That Match a Pattern

Use the *like* keyword to look for dates that match a particular pattern. If you use the equality operator (=) to search *datetime* values for a particular month, day, and year, Adaptive Server returns only those values for which the time is precisely 12:00:00:000AM.

For example, if you insert the value “9:20” into a column named *arrival_time*, Adaptive Server converts the entry into “Jan 1 1900 9:20AM”. If you look for this entry using the equality operator, it is not found:

```
where arrival_time = "9:20" /* does not match */
```

You can find the entry using the `like` operator:

```
where arrival_time like "%9:20%"
```

When using `like`, Adaptive Server first converts the dates to *datetime* format and then to *varchar*. The display format consists of the 3-character month in the current language, 2 characters for the day, 4 characters for the year, the time in hours and minutes, and “AM” or “PM.”

When searching with `like`, you cannot use the wide variety of input formats that are available for entering the date portion of *datetime* and *smalldatetime* values. Since the standard display formats do not include seconds or milliseconds, you cannot search for seconds or milliseconds with `like` and a match pattern, unless you are also using *style 9* or *109* and the `convert` function.

If you are using `like`, and the day of the month is a number between 1 and 9, insert 2 spaces between the month and the day to match the *varchar* conversion of the *datetime* value. Similarly, if the hour is less than 10, the conversion places 2 spaces between the year and the hour. The clause:

```
like May 2%
```

(with 1 space between “May” and “2”) finds all dates from May 20 through May 29, but not May 2. You do not need to insert the extra space with other date comparisons, only with `like`, since the *datetime* values are converted to *varchar* only for the `like` comparison.

Manipulating Dates

You can do some arithmetic calculations on *datetime* values with the built-in date functions. See “Date Functions” in Chapter 2, “Transact-SQL Functions.”

Standards and Compliance

Standard	Compliance Level
SQL92	The <i>datetime</i> and <i>smalldatetime</i> datatypes are Transact-SQL extensions.

Character Datatypes

Function

Use the character datatypes to store strings consisting of letters, numbers, and symbols. Use the fixed-length datatype, *char(n)*, and the variable-length datatype, *varchar(n)*, for single-byte character sets such as *us_english*. Use the fixed-length datatype, *nchar(n)*, and the variable-length datatype, *nvarchar(n)*, for multibyte character sets such as Japanese. The character datatypes can store a maximum of 255 characters; use the *text* datatype (described in “text and image Datatypes”) for strings longer than 255 characters.

Length and Storage Size

Use *n* to specify the length in characters for the fixed-length datatypes, *char(n)* and *nchar(n)*. Entries shorter than the assigned length are blank-padded; entries longer than the assigned length are truncated without warning, unless the *string_truncation* option to the *set* command is set to *on*. Fixed-length columns that allow nulls are internally converted to variable-length columns.

Use *n* to specify the maximum length in characters for the variable-length datatypes, *varchar(n)* and *nvarchar(n)*. Data in variable-length columns is stripped of trailing blanks; storage size is the actual length of the data entered. Data in variable-length variables and parameters retains all trailing blanks, but is not padded to the defined length. Character literals are treated as variable-length datatypes.

Fixed-length columns tend to take more storage space than variable-length columns, but are accessed somewhat faster. Table 7-15 summarizes the storage requirements of the different character datatypes:

Table 7-15: Character datatypes

Datatype	Stores	Bytes of Storage
<i>char(n)</i>	Fixed-length data, such as social security numbers or postal codes, in single-byte character sets.	<i>n</i>
<i>nchar(n)</i>	Fixed-length data in multibyte character sets	<i>n * @@ncharsize</i>
<i>varchar(n)</i>	Variable-length data, such as names, in single-byte character sets.	Actual number of characters entered

Table 7-15: Character datatypes (continued)

Datatype	Stores	Bytes of Storage
<i>nvarchar(n)</i>	Variable-length data in multibyte character sets	Actual number of characters * @@ncharsize

Determining Column Length with System Functions

Use the `char_length` string function and `datalength` system function to determine column length:

- `char_length` returns the number of characters in the column, stripping trailing blanks for variable-length datatypes.
- `datalength` returns the number of bytes, stripping trailing blanks for data stored in variable-length columns.

When a *char* value is declared to allow NULLS, SQL Server stores it internally as a *varchar*.

Entering Character Data

Character strings must be enclosed in single or double quotes. If you use `set quoted_identifier on`, use single quotes for character strings; otherwise, Adaptive Server treats them as identifiers.

Strings that include the double-quote character should be surrounded by single quotes. Strings that include the single-quote character should be surrounded by double quotes. For example:

```
'George said, "There must be a better way."'
"Isn't there a better way?"
```

An alternative is to enter two quotation marks for each quotation mark you want to include in the string. For example:

```
"George said, ""There must be a better way.""
'Isn''t there a better way?'
```

To continue a character string onto the next line of your screen, enter a backslash (\) before going to the next line.

Treatment of Blanks

The following example creates a table named *spaces* that has both fixed- and variable-length character columns:

```
create table spaces (cnot char(5) not null,
                   cnull char(5) null,
                   vnot varchar(5) not null,
                   vnull varchar(5) null,
                   explanation varchar(25) not null)

insert spaces values ("a", "b", "c", "d",
                    "pads char-not-null only")
insert spaces values ("1  ", "2   ", "3    ",
                    "4     ", "truncates trailing blanks")
insert spaces values ("  e", "   f", "    g",
                    "     h", "leading blanks, no change")
insert spaces values ("  w ", "   x ", "    y ",
                    "     z ", "truncates trailing blanks")
insert spaces values ("", "", "", "",
                    "empty string equals space" )

select "[" + cnot + "]",
       "[" + cnull + "]",
       "[" + vnot + "]",
       "[" + vnull + "]",
       explanation from spaces
```

					explanation
[a]	[b]	[c]	[d]		pads char-not-null only
[1]	[2]	[3]	[4]		truncates trailing blanks
[e]	[f]	[g]	[h]		leading blanks, no change
[w]	[x]	[y]	[z]		truncates trailing blanks
[]	[]	[]	[]		empty string equals space

(5 rows affected)

This example illustrates how the column's datatype and null type interact to determine how blank spaces are treated:

- Only *char* not null and *nchar* not null columns are padded to the full width of the column; *char* null columns are treated like *varchar* and *nchar* null columns are treated like *nvarchar*.
- Preceding blanks are not affected.
- Trailing blanks are truncated except for *char* and *nchar* not null columns.

- The empty string ("") is treated as a single space. In *char* and *nchar* not null columns, the result is a column-length field of spaces.

Manipulating Character Data

You can use the *like* keyword to search character strings for particular characters and the built-in string functions to manipulate their contents. Strings consisting of numbers can be used for arithmetic after being converted to exact and approximate numeric datatypes with the *convert* function.

Standards and Compliance

Standard	Compliance Level
SQL92	Transact-SQL provides the <i>char</i> and <i>varchar</i> SQL92 datatypes. The <i>nchar</i> and <i>nvarchar</i> datatypes are Transact-SQL extensions.

Binary Datatypes

Function

Use the binary datatypes, *binary(n)* and *varbinary(n)*, to store up to 255 bytes of raw binary data, such as pictures, in a hexadecimal-like notation.

Valid Binary and Varbinary Entries

Binary data begins with the characters “0x” and can include any combination of digits and the uppercase and lowercase letters A through F.

Use *n* to specify the column length in bytes, or use the default length of 1 byte. Each byte stores 2 binary digits. If you enter a value longer than *n*, Adaptive Server truncates the entry to the specified length without warning or error.

Use the fixed-length binary type, *binary(n)*, for data in which all entries are expected to be approximately equal in length.

Use the variable-length binary type, *varbinary(n)*, for data that is expected to vary greatly in length.

Because entries in *binary* columns are zero-padded to the column length (*n*), they may require more storage space than those in *varbinary* columns, but they are accessed somewhat faster.

Use the *image* Datatype for Entries of More Than 255 Bytes

Use the *image* datatype to store larger blocks of binary data (up to 2,147,483,647 bytes) on external data pages. You cannot use the *image* datatype for variables or for parameters in stored procedures. See the section “text and image Datatypes” for more information.

Treatment of Trailing Zeros

All *binary* not null columns are padded with zeros to the full width of the column. Trailing zeros are truncated in all *varbinary* data and in *binary* null columns, since columns that accept null values must be treated as variable-length columns.

The following example creates a table with all four variations of *binary* and *varbinary* datatypes, NULL and NOT NULL. The same data is inserted in all four columns and is padded or truncated according to the datatype of the column.

```

create table zeros (bnot binary(5) not null,
                   bnull binary(5) null,
                   vnot varbinary(5) not null,
                   vnull varbinary(5) null)

insert zeros values (0x12345000, 0x12345000,
                   0x12345000, 0x12345000)
insert zeros values (0x123, 0x123, 0x123, 0x123)
select * from zeros

```

bnot	bnull	vnot	vnull
-----	-----	-----	-----
0x1234500000	0x123450	0x123450	0x123450
0x0123000000	0x0123	0x0123	0x0123

Because each byte of storage holds 2 binary digits, Adaptive Server expects binary entries to consist of the characters “0x” followed by an even number of digits. When the “0x” is followed by an odd number of digits, Adaptive Server assumes that you omitted the leading 0 and adds it for you.

Input values “0x00” and “0x0” are stored as “0x00” in variable-length binary columns (*binary null*, *image* and *varbinary* columns). In fixed-length binary (*binary not null*) columns, the value is padded with zeros to the full length of the field:

```

insert zeros values (0x0, 0x0,0x0, 0x0)
select * from zeros where bnot = 0x00

```

bnot	bnull	vnot	vnull
-----	-----	-----	-----
0x0000000000	0x00	0x00	0x00

If the input value does not include the “0x”, Adaptive Server assumes that the value is an ASCII value and converts it. For example:

```

create table sample (col_a binary(8))

insert sample values ('002710000000aeb1b')

select * from sample

```

col_a

0x3030323731303030

Platform Dependence

The exact form in which you enter a particular value depends upon the platform you are using, so **calculations involving binary data can produce different results on different machines.**

You cannot use the aggregate functions `sum` or `avg` with the binary datatypes.

For platform-independent conversions between hexadecimal strings and integers, use the `inttohex` and `hextoint` functions rather than the platform-specific `convert` function. (See “Datatype Conversion Functions” in Chapter 2, “Transact-SQL Functions,” for details.)

Standards and Compliance

Standard	Compliance Level
SQL92	The <i>binary</i> and <i>varbinary</i> datatypes are Transact-SQL extensions.

bit Datatype

Function

Use *bit* columns for true/false and yes/no types of data. The *status* column in the *syscolumns* system table indicates the unique offset position for *bit* columns.

Entering Data into *bit* Columns

bit columns hold either 0 or 1. Integer values other than 0 or 1 are accepted, but are always interpreted as 1.

Storage Size

Storage size is 1 byte. Multiple *bit* datatypes in a table are collected into bytes. For example, 7 *bit* columns fit into 1 byte; 9 *bit* columns take 2 bytes.

Restrictions

Columns with a datatype of *bit* cannot be NULL and cannot have indexes on them.

Standards and Compliance

Standard	Compliance Level
SQL92	Transact-SQL extension

sysname Datatype

Function

sysname is a user-defined datatype that is distributed on the Adaptive Server installation tape and used in the system tables. Its definition is:

```
varchar(30) "not null"
```

Using the *sysname* Datatype

You cannot declare a column, parameter, or variable to be of type *sysname*. It is possible, however, to create a user-defined datatype with a base type of *sysname*. You can then define columns, parameters, and variables with the user-defined datatype.

Standards and Compliance

Standard	Compliance Level
SQL92	All user-defined datatypes, including <i>sysname</i> , are Transact-SQL extensions.

text and image Datatypes

Function

text columns are variable-length columns that can hold up to 2,147,483,647 ($2^{31} - 1$) bytes of printable characters.

image columns are variable-length columns that can hold up to 2,147,483,647 ($2^{31} - 1$) bytes of hexadecimal-like data.

Defining a *text* or *image* Column

You define a *text* or *image* column as you would any other column, with a create table or alter table statement. *text* and *image* datatype definitions do not include lengths. They do permit null values. The column definition takes the form:

```
column_name {text | image} [null]
```

For example, the create table statement for the author's *blurbs* table in the *pubs2* database with a *text* column, *blurb*, that permits null values, is:

```
create table blurbs
(au_id id not null,
copy text null)
```

To create the *au_pix* table in the *pubs2* database with an *image* column:

```
create table au_pix
(au_id          char(11) not null,
pic            image null,
format_type    char(11) null,
bytesize      int null,
pixwidth_hor   char(14) null,
pixwidth_vert  char(14) null)
```

How Adaptive Server Stores *text* and *image* Data

Adaptive Server stores *text* and *image* data in a linked list of data pages that are separate from the rest of the table. Each *text* or *image* page stores a maximum of 1800 bytes of data. All *text* and *image* data for a table is stored in a single page chain, regardless of the number of *text* and *image* columns the table contains.

Putting Additional Pages on Another Device

You can place subsequent *text* and *image* data pages on a different logical device with *sp_placeobject*.

Zero Padding

image values of less than 255 bytes that have an odd number of hexadecimal digits are padded with a leading zero (an insert of “0xaaabb” becomes “0x0aaabb”).

► **Note**

It is an error to insert *image* values of more than 255 bytes that have an odd number of bytes.

Partitioning Has No Effect on How the Data Is Stored

You can use the `partition` option of the `alter table` command to partition a table that contains *text* and *image* columns. Partitioning the table creates additional page chains for the other columns in the table, but has **no** effect on the way the *text* and *image* columns are stored.

Initializing *text* and *image* Columns

text and *image* columns are not initialized until you update them or insert a non-null value. Initialization allocates at least one data page for each non-null *text* or *image* data value. It also creates a pointer in the table to the location of the *text* or *image* data.

For example, the following statements create the table *testtext* and initialize the *blurb* column by inserting a non-null value. The column now has a valid text pointer, and the first 2K data page has been allocated.

```
create table testtext
(title_id varchar(6), blurb text null, pub_id
char(4))

insert testtext values
("BU7832", "Straight Talk About Computers is an
annotated analysis of what computers can do for
you: a no-hype guide for the critical user.",
"1389")
```

The following statements create a table for *image* values and initialize the *image* column:

```
create table imagetest
(image_id varchar(6), imagecol image null,
graphic_id char(4))
```



```
insert imagetest values
("94732", 0x00000083000000000010000000013c,
"1389")
```

► **Note**

Remember to surround *text* values with quotation marks and precede *image* values with the characters "0x".

See the *Client-Library/C Reference Manual* for information on inserting and updating *text* and *image* data with Client-Library programs.

Saving Space by Allowing Nulls

To save storage space for empty *text* or *image* columns, define them to permit null values and insert nulls until you use the column. Inserting a null value does not initialize a *text* or *image* column and, therefore, does not create a text pointer or allocate 2K bytes of storage. For example, the following statement inserts values into the *title_id* and *pub_id* columns of the *testtext* table created above, but does not initialize the *blurb* text column:

```
insert texttest
(title_id, pub_id) values ("BU7832", "1389")
```

Once a *text* or *image* row is given a non-null value, it always contains at least one data page. Resetting the value to null does not deallocate its data page.

Getting Information from *sysindexes*

Each table with *text* or *image* columns has an additional row in *sysindexes* that provides information about these columns. The *name* column in *sysindexes* uses the form "*ttablename*"; the *indid* is always 255. These columns provide information about text storage:

Table 7-16: Storage of text and image data

Column	Description
<i>ioampg</i>	Pointer to the allocation page for the text page chain
<i>first</i>	Pointer to the first page of text data
<i>root</i>	Pointer to the last page
<i>segment</i>	Number of the segment where the object resides

You can query the *sysindexes* table for information about these columns. For example, the following query reports the number of data pages used by the *blurbs* table in the *pubs2* database:

```
select name, data_pgs(object_id("blurbs"), ioampg)
from sysindexes
where name = "tblurbs"
```

```
name
-----
tblurbs                                7
```

Using *readtext* and *writetext*

Before you can use *writetext* to enter *text* data or *readtext* to read it, you must initialize the *text* column. See *readtext* and *writetext* for more details.

Using *update* to replace existing *text* and *image* data with NULL reclaims all allocated data pages except the first page, which remains available for future use of *writetext*. To deallocate all storage for the row, use *delete* to remove the entire row.

Determining How Much Space a Column Uses

The system procedure *sp_spaceused* provides information about the space used for text data as *index_size*:

```
sp_spaceused blurbs
name          rowtotal  reserved  data    index_size  unused
-----
tblurbs       6          32 KB    2 KB    14 KB      16 KB
```

Restrictions on *text* and *image* Columns

text and *image* columns cannot be used:

- As parameters to stored procedures or as values passed to these parameters
- As local variables
- In *order by*, *compute*, *group by*, and *union* clauses
- In an index
- In subqueries or joins
- In a *where* clause, except with the keyword *like*
- With the *+* concatenation operator
- In the *if update* clause of a trigger

Selecting *text* and *image* Data

The following global variables return information on *text* and *image* data:

Table 7-17: text and image global variables

Variable	Explanation
<code>@@textptr</code>	The text pointer of the last <i>text</i> or <i>image</i> column inserted or updated by a process. Do not confuse this global variable with the Open Client <code>textptr()</code> function.
<code>@@textcolid</code>	ID of the column referenced by <code>@@textptr</code> .
<code>@@textdbid</code>	ID of a database containing the object with the column referenced by <code>@@textptr</code> .
<code>@@textobjid</code>	ID of the object containing the column referenced by <code>@@textptr</code> .
<code>@@textsize</code>	Current value of the set <code>textsize</code> option, which specifies the maximum length, in bytes, of <i>text</i> or <i>image</i> data to be returned with a <code>select</code> statement. It defaults to 32K. The maximum size for <code>@@textsize</code> is 231 - 1 (that is, 2,147,483,647).
<code>@@textts</code>	Text timestamp of the column referenced by <code>@@textptr</code> .

Converting the *text* and *image* Datatypes

You can explicitly convert *text* values to *char* or *varchar* and *image* values to *binary* or *varbinary* with the `convert` function, but you are limited to the maximum length of the character and binary datatypes, 255 bytes. If you do not specify the length, the converted value has a default length of 30 bytes. Implicit conversion is not supported.

Pattern Matching in *text* Data

Use the `patindex` function to search for the starting position of the first occurrence of a specified pattern in a *text*, *varchar*, or *char* column. The % wildcard character must precede and follow the pattern (except when you are searching for the first or last character).

You can also use the `like` keyword to search for a particular pattern. The following example selects each *text* data value from the *copy* column of the *blurbs* table that contains the pattern "Net Etiquette".

```
select copy from blurb
where copy like "%Net Etiquette%"
```

Duplicate Rows Are Prohibited

Because the pointer to the *text* or *image* data uniquely identifies each row, a table that contains *text* or *image* data cannot contain duplicate rows unless the pointer has not been initialized; that is, unless all *text* and *image* data is NULL.

Standards and Compliance

Standard	Compliance Level
SQL92	The <i>text</i> and <i>image</i> datatypes are Transact-SQL extensions.

User-Defined Datatypes

Function

User-defined datatypes are built from the system datatypes and from the *sysname* user-defined datatype. Once you create a user-defined datatype, you can use it to define columns, parameters, and variables. Objects that are created from user-defined datatypes inherit the rules, defaults, null type, and IDENTITY property of the user-defined datatype, as well as inheriting the defaults and null type of the system datatypes on which the user-defined datatype is based.

Creating Frequently Used Datatypes in the *model* Database

A user-defined datatype must be created in each database in which it will be used. It is a good practice to create frequently used types in the *model* database. These types are automatically added to each new database (including *tempdb*, which is used for temporary tables) as it is created.

Creating a User-Defined Datatype

Adaptive Server allows you to create user-defined datatypes, based on any system datatype, with the `sp_addtype` system procedure. You cannot create a user-defined datatype based on another user-defined datatype, such as *timestamp* or the *tid* datatype in the *pubs2* database.

The *sysname* datatype is an exception to this rule. Though *sysname* is a user-defined datatype, you can use it to build user-defined datatypes.

User-defined datatypes are database objects. Their names are case-sensitive and must conform to the rules for identifiers.

You can bind rules to user-defined datatypes with `sp_bindrule` and bind defaults with `sp_bindefault`.

By default, objects built on a user-defined datatype inherit the user-defined datatype's null type or IDENTITY property. You can override the null type or IDENTITY property in a column definition.

Renaming a User-Defined Datatype

Use `sp_rename` to rename a user-defined datatype.

Dropping a User-Defined Datatype

Use `sp_droptype` to remove a user-defined datatype from a database.

► Note

You cannot drop a datatype that is already in use in a table.

Getting Help on Datatypes

Use the `sp_help` system procedure to display information about the properties of a system datatype or a user-defined datatype. You can also use `sp_help` to display the datatype, length, precision, and scale for each column in a table.

Standards and Compliance

Standard	Compliance Level
SQL92	User-defined datatypes are a Transact-SQL extension.

System Tables

8

System Tables

This chapter describes the Adaptive Server system tables.

All 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.

Lists of System Tables

The following system tables occur in all databases:

System Table	Contents
<i>sysalternates</i>	One row for each Adaptive Server user mapped to a database user
<i>sysattributes</i>	One row for each object attribute definition.
<i>syscolumns</i>	One row for each column in a table or view, and for each parameter in a procedure
<i>syscomments</i>	One or more rows for each view, rule, default, trigger, and procedure, giving SQL definition statement
<i>sysconstraints</i>	One row for each referential and check constraint associated with a table or column
<i>sysdepends</i>	One row for each procedure, view, or table that is referenced by a procedure, view, or trigger
<i>sysgams</i>	Allocation bitmaps for an entire database
<i>sysindexes</i>	One row for each clustered or nonclustered index, and one row for each table with no indexes, and an additional row for each table containing text or image data
<i>syskeys</i>	One row for each primary, foreign, or common key; set by user (not maintained by Adaptive Server)
<i>syslogs</i>	Transaction log
<i>sysobjects</i>	One row for each table, view, procedure, rule, trigger default, log, and (in <i>tempdb</i> only) temporary object
<i>syspartitions</i>	One row for each partition (page chain) of a partitioned table
<i>sysprocedures</i>	One row for each view, rule, default, trigger, and procedure, giving internal definition

System Table	Contents
<i>sysprotects</i>	User permissions information
<i>sysreferences</i>	One row for each referential integrity constraint declared on a table or column
<i>sysroles</i>	Maps server-wide roles to local database groups
<i>syssegments</i>	One row for each segment (named collection of disk pieces)
<i>systhresholds</i>	One row for each threshold defined for the database
<i>systypes</i>	One row for each system-supplied and user-defined datatype
<i>sysusermessages</i>	One row for each user-defined message
<i>sysusers</i>	One row for each user allowed in the database

The following system tables occur in the *master* database only:

System Table	Contents
<i>syscharsets</i>	One row for each character set or sort order
<i>sysconfigures</i>	One row for each configuration parameter that can be set by users
<i>syscurconfigs</i>	Information about configuration parameters currently being used by Adaptive Server
<i>sysdatabases</i>	One row for each database on Adaptive Server
<i>sysdevices</i>	One row for each tape dump device, disk dump device, disk for databases, and disk partition for databases
<i>sysengines</i>	One row for each Adaptive Server engine currently online
<i>syslanguages</i>	One row for each language (except U.S. English) known to the server
<i>syslisteners</i>	One row for each type of network connection used by current Adaptive Server
<i>syslocks</i>	Information about active locks
<i>sysloginroles</i>	One row for each server login that possesses a system role
<i>syslogins</i>	One row for each valid Adaptive Server user account
<i>syslogshold</i>	Information about the oldest active transaction and the Replication Server® truncation point for each database

System Table	Contents
<i>sysmessages</i>	One row for each system error or warning
<i>sysmonitors</i>	One row for each monitor counter
<i>sysprocesses</i>	Information about server processes
<i>sysremotelogins</i>	One row for each remote user
<i>sysresourcelimits</i>	One row for each resource limit
<i>syssecmechs</i>	Information about the security services available for each security mechanism that is available to Adaptive Server
<i>sysservers</i>	One row for each remote Adaptive Server
<i>sysserverroles</i>	One row for each server-wide role
<i>sys timeranges</i>	One row for each named time range
<i>sysusages</i>	One row for each disk piece allocated to a database

The following system tables occur in the *sybsecurity* database only:

System Table	Contents
<i>sysauditoptions</i>	One row for each global audit option
<i>sysaudits_01</i> , <i>sysaudits_02...sysaudits_08</i>	The audit trail. Each audit table contains one row for each audit record.

In the pages that follow, each system table is described in more detail, including a list of its columns and datatypes, as well as the indexes and Sybase-supplied procedures that reference a particular table are listed.

The word “reserved” in the column description means that the column is not currently used by Adaptive Server.

Note that aggregate functions cannot be used on virtual tables such as *syslocks* and *sysprocesses*.

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 Adaptive Server is installed, the *installmodel* script grants select access to “public” (all users) for most system tables and for most fields in the tables. However, no access is given for some system

tables, such as *systhresholds*, and no access is given for certain fields in other system tables. For example, all users, by default, can select all columns of *sysobjects* except *audflags*. To determine the current permissions for a particular system table, execute:

```
sp_helpprotect system_table_name
```

For example, to check the permissions of *systhresholds* in *your_database*, execute:

```
use your_database
go
sp_helpprotect systhresholds
go
```

Updating System Tables

All direct updates on system tables are by default not allowed — even for the database owner. Instead, Adaptive Server supplies system procedures to make any normally needed updates and additions to system tables.

You can allow direct updates to the system tables if it becomes necessary to modify them in a way that cannot be accomplished with a system procedure. To accomplish this, a System Security Officer must reset the configuration parameter called *allow updates to system tables* with the system procedure *sp_configure*. For information, see the *System Administration Guide*.

There are entries in some of the *master* database tables that should not be altered by any user under any circumstances. For example, do not attempt to modify *syslogs* with a delete, update, or insert command. In addition, an attempt to delete all rows from *syslogs* will put Adaptive Server into an infinite loop that eventually fills up the entire database.

Triggers on System Tables

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

sysalternates

(all databases)

Description

sysalternates contains one row for each Adaptive Server user mapped (or aliased) to a user of the current database. When a user tries to access a database, Adaptive 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*.

On the Adaptive Server distribution media, there are no entries in *sysalternates*.

Columns

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

Indexes

Unique clustered index on *suid*

Referenced by System Procedures

sp_addalias, *sp_adduser*, *sp_changedbowner*, *sp_dropalias*, *sp_dropuser*,
sp_helpuser

sysattributes

(all databases)

Description

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 Adaptive Server as a whole. It also stores attribute definitions for server-wide objects, such as databases and logins.

sysattributes should only be accessed indirectly using system procedures. The permissions required for modifying *sysattributes* depend on the system procedure you use.

Columns

Name	Datatype	Description
<i>class</i>	<i>smallint</i>	The attribute class ID. This describes the category of the attribute. In <i>master..sysattributes</i> , the special class 1 identifies all valid attributes for Adaptive Server. Class 0 identifies valid classes of attributes.
<i>attribute</i>	<i>smallint</i>	The attribute ID.
<i>object_type</i>	<i>char(2)</i>	The one- or two-letter character ID that defines the type of object to associate with the attribute.
<i>object_cinfo</i>	<i>varchar(30)</i>	A string identifier for the object (for example, the name of an application). This field is not used by all attributes.
<i>object</i>	<i>int null</i>	The object identifier. This may be an object ID, user ID, or database ID, depending on the type of object. If the object is a part of a table (for example, an index), then this column contains the object ID of the associated table.
<i>object_info1</i>	<i>int null</i>	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.

Name	Datatype	Description
<i>object_info2</i>	<i>int null</i>	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.
<i>object_info3</i>	<i>int null</i>	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.
<i>int_value</i>	<i>int null</i>	An integer value for the attribute (for example, the display level of a user).
<i>char_value</i>	<i>varchar(255)</i>	A character value for the attribute (for example, a cache name).
<i>text_value</i>	<i>text null</i>	A text value for the attribute.
<i>image_value</i>	<i>image null</i>	An image value for the attribute.
<i>comments</i>	<i>varchar(255)</i>	Comments or additional information about the attribute definition.

Table 8-1 describes the *object_type* values and their meanings:

Table 8-1: Object types for attributes

ID	Object Type
D	Database
EL	External Login (for Component Integration Services)
I	Index
L	Login name
OD	Object Definition (for Component Integration Services)
P	Procedure
T	Table
TP	Text Page (for Component Integration Services)
U	Username
UI	Upgrade Item (used internally during user database upgrades)

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*

Referenced by System Procedures

sp_activeroles, sp_addengine, sp_addexceclass, sp_addexternlogin,
sp_addobjectdef, sp_bindcache, sp_bindexceclass, sp_clearpsexex, sp_configure,
sp_displaylevel, sp_displayroles, sp_dropengine, sp_dropexceclass,
sp_dropexternlogin, sp_dropglockpromote, sp_droplogin, sp_dropobjectdef,
sp_dropserver, sp_dropuser, sp_forceonline_db, sp_forceonline_page, sp_help,
sp_helpdb, sp_helpexternlogin, sp_helpindex, sp_helpobjectdef, sp_helpprotect,
sp_listsuspect_db, sp_listsuspect_page, sp_logiosize, sp_setpglockpromote,
sp_setpsexex,, sp_setsuspect_granularity, sp_setsuspect_threshold,
sp_shmdumpconfig, sp_showcontrolinfo, sp_showexceclass, sp_unbindexceclass

sysauditoptions

(*sybsecurity* database)

Description

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.” *sysauditoptions* can be accessed only by System Security Officers.

Columns

Name	Datatype	Description
<i>num</i>	<i>smallint</i>	Number of the server-wide option.
<i>val</i>	<i>smallint</i>	Current value; one of the following: 0 = off 1 = pass 2 = fail 3 = on
<i>minval</i>	<i>smallint</i>	Minimum valid value for this option.
<i>maxval</i>	<i>smallint</i>	Maximum valid value for this option.
<i>name</i>	<i>varchar(30)</i>	Name of option.
<i>sval</i>	<i>varchar(30)</i>	String equivalent of the current value: for example, “on”, “off”, “nonfatal”.
<i>comment</i>	<i>varchar(255)</i>	Description of option.

Indexes

None

Referenced by System Procedures

sp_addauditrecord, sp_audit

sysaudits_01, sysaudits_02...sysaudits_08

(*sybsecurity* database)

Description

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 up to 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

Name	Datatype	Description
<i>event</i>	<i>smallint</i>	Type of event being audited. See Table 8-3 on page 8-12.
<i>eventmod</i>	<i>smallint</i>	Further information about the event. Possible values are: 0 = no modifier for this event 1 = the event passed permission checking 2 = the event failed permission checking
<i>spid</i>	<i>smallint</i>	Server process ID of the process that caused the audit record to be written.
<i>eventtime</i>	<i>datetime</i>	Date and time of the audited event.
<i>sequence</i>	<i>smallint</i>	Sequence number of the record within a single event; some events require more than one audit record.
<i>suid</i>	<i>smallint</i>	Server login ID of the user who performed the audited event.
<i>dbid</i>	<i>int null</i>	Database ID in which the audited event occurred or the object/stored procedure/trigger resides, depending on the type of event.
<i>objid</i>	<i>int null</i>	ID of the accessed object or stored procedure/trigger.
<i>xactid</i>	<i>binary(6) null</i>	ID of the transaction containing the audited event. For a multi-database transaction, this is the transaction ID from the database where the transaction originated.
<i>loginname</i>	<i>varchar(30) null</i>	Login name corresponding to the <i>suid</i> .
<i>dbname</i>	<i>varchar(30) null</i>	Database name corresponding to the <i>dbid</i> .

Name	Datatype	Description
<i>objname</i>	<i>varchar(30) null</i>	Object name corresponding to the <i>objid</i> .
<i>objowner</i>	<i>varchar(30) null</i>	Name of the owner of <i>objid</i> .
<i>extrainfo</i>	<i>varchar(255) null</i>	Additional information about the audited event. This field contains a sequence of items separated by semicolons. See Table 8-2.

The *extrainfo* column contains a sequence of items separated by semicolons. Table 8-2 lists the items in the *extrainfo* column:

Table 8-2: Items in the *extrainfo* field

Item	Contents
Roles	Lists the roles that are active. The roles are separated by blanks.
Subcommand	The name of the subcommand or command option that was used for the event. For example, for the alter table command, the options “add column” or “drop constraint” might be used. Multiple subcommands 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.
Proxy information	The original login name, if the event occurred while a set proxy 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 into Adaptive Server via unified login. The value of this field is NULL, if the secure default login is not being used.

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 *extrainfo* column 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). The other columns in the audit record give other pertinent information. For example, the record contains the server user id (*suid*) and the login name (*loginname*).

The *event* column values that pertain to each audit event are listed in Table 8-3.

Table 8-3: Values in event and extrainfo column

Audit Option	<i>event</i>	Command or Access Audited	<i>extrainfo</i>
adhoc	1	User-defined audit record	<i>extrainfo</i> is filled by the <i>text</i> parameter of <code>sp_addauditrecord</code>
alter	2	alter database	Roles: Current active roles Subcommand: "ALTER SIZE" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	3	alter table	Roles: Current active roles Subcommand: "ADD COLUMN", "REPLACE COLUMN", "ADD CONSTRAINT", or "DROP CONSTRAINT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
bcp	4	bcp in	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
bind	6	sp_bindefault	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: Name of default Proxy information: Original login name, if a set proxy is in effect
	7	sp_bindmsg	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: Message ID Proxy information: Original login name, if a set proxy is in effect
	8	sp_bindrule	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: Name of the rule Proxy information: Original login name, if a set proxy is in effect
create	9	create database	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	10	create table	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	11	create procedure	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
create (continued)	12	create trigger	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	13	create rule	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	14	create default	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	15	sp_addmessage	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: Message Number Proxy information: Original login name, if a set proxy is in effect
	16	create view	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
dbaccess	17	Any access to the database by any user	Roles: Current active roles Subcommand: "USE CMD" or "OUTSIDE REFERENCE" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
dbcc	81	dbcc	Roles: Current active roles Subcommand: The dbcc subcommand name Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
delete	18	delete from a table	Roles: Current active roles Subcommand: "DELETE" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	19	delete from a view	Roles: Current active roles Subcommand: "DELETE" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
disk	20	disk init	Roles: Current active roles Subcommand: "disk init" Previous value: NULL Current value: NULL Other Information: Name of the disk Proxy information: Original login name, if a set proxy is in effect
	21	disk refit	Roles: Current active roles Subcommand: "disk refit" Previous value: NULL Current value: NULL Other Information: Name of the disk Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
disk (continued)	22	disk reinit	Roles: Current active roles Subcommand: "disk reinit" Previous value: NULL Current value: NULL Other Information: Name of the disk Proxy information: Original login name, if a set proxy is in effect
	23	disk mirror	Roles: Current active roles Subcommand: "disk mirror" Previous value: NULL Current value: NULL Other Information: Name of the disk Proxy information: Original login name, if a set proxy is in effect
	24	disk unmirror	Roles: Current active roles Subcommand: "disk unmirror" Previous value: NULL Current value: NULL Other Information: Name of the disk Proxy information: Original login name, if a set proxy is in effect
	25	disk remirror	Roles: Current active roles Subcommand: "disk remirror" Previous value: NULL Current value: NULL Other Information: Name of the disk Proxy information: Original login name, if a set proxy is in effect
drop	26	drop database	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	27	drop table	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
drop (continued)	28	drop procedure	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	29	drop trigger	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	30	drop rule	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	31	drop default	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	32	sp_dropmessage	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other Information: Message number Proxy information: Original login name, if a set proxy is in effect
	33	drop view	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
dump	34	dump database	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	35	dump transaction	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
errors	36	Fatal error	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: Error number.Severity.State Proxy information: Original login name, if a set proxy is in effect
	37	Non-fatal error	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: Error number.Severity.State Proxy information: Original login name, if a set proxy is in effect
exec_procedure	38	Execution of a procedure	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other Information: All input parameters Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
exec_trigger	39	Execution of a trigger	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
func_obj_access, func_dbaccess	85	Accesses to objects and databases via Transact-SQL functions	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
grant	40	grant	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
insert	41	insert into a table	Roles: Current active roles Subcommand: If insert: "INSERT" If select into: "INSERT INTO" followed by the fully qualified object name Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	42	insert into a view	Roles: Current active roles Subcommand: "INSERT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
load	43	load database	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	44	load transaction	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
login	45	Any login to Adaptive Server	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other Information: Host name of the machine from which login was done Proxy information: Original login name, if a set proxy is in effect
logout	46	Any logouts from Adaptive Server	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other Information: Host name of the machine from which login was done Proxy information: Original login name, if a set proxy is in effect
revoke	47	revoke	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
rpc	48	Remote procedure call from another server	Roles: Current active roles Subcommand: Name of client program Previous value: NULL Current value: NULL Other information: Server name, host name of the machine from which the RPC was done. Proxy information: Original login name, if a set proxy is in effect
	49	Remote procedure call to another server	Roles: Current active roles Subcommand: Procedure name Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
security	50	Server start	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other Information: -dmasterdevicename -iinterfaces file path -Sservername -errorfilename Proxy information: Original login name, if a set proxy is in effect
	51	Server shutdown	Roles: Current active roles Subcommand: "shutdown" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	55	Role toggling	Roles: Current active roles Subcommand: NULL Previous Value: "on" or "off" Current Value: "on" or "off" Other Information: Name of the role being set Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
security (continued)	82	sp_configure	Roles: Current active roles Subcommand: Name of the configuration parameter Previous Value: The old parameter value if the command is setting a new value Current Value: The new parameter value if the command is setting a new value Other Information: Number of configuration parameter, if a parameter is being set; Name of the configuration file, if a configuration file is being used to set parameters Proxy information: Original login name, if a set proxy is in effect
	83	online database	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	76	Regeneration of a password by a System Security Officer (SSO)	Roles: Current active roles Subcommand: Setting SSO password Previous value: NULL Current value: NULL Other information: Login name Proxy information: Original login name, if a set proxy is in effect
	80	proc_role within a system procedure	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other Information: Required roles Proxy information: Original login name, if a set proxy is in effect
	85	valid_user	Roles: Current active roles Subcommand: "valid_user" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
security (continued)	88	set proxy or set session authorization	Roles: Current active roles Subcommand: NULL Previous value: Previous <i>suid</i> Current value: New <i>suid</i> Other information: NULL Proxy information: Original login name, if set proxy or set session authorization had no parameters; otherwise, NULL.
select	62	select from a table	Roles: Current active roles Subcommand: "SELECT INTO", "SELECT", or "READTEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	63	select from a view	Roles: Current active roles Subcommand: "SELECT INTO", "SELECT", or "READTEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
setuser	84	setuser	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other Information: Name of the user being set Proxy information: Original login name, if a set proxy is in effect
table_access	62	select	Roles: Current active roles Subcommand: "SELECT INTO", "SELECT", or "READTEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
table_access (continued)	18	delete	Roles: Current active roles Subcommand: "DELETE" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	70	update	Roles: Current active roles Subcommand: "UPDATE" or "WRITETEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	41	insert	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
truncate	64	truncate table	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
unbind	67	sp_unbindefault	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	68	sp_unbindrule	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
unbind (continued)	69	sp_unbindmsg	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
update	70	update to a table	Roles: Current active roles Subcommand: "UPDATE" or "WRITETEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	71	update to a view	Roles: Current active roles Subcommand: "UPDATE" or "WRITETEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
view_access	63	select	Roles: Current active roles Subcommand: "SELECT INTO" "SELECT", or "READTEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	19	delete	Roles: Current active roles Subcommand: "DELETE" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Table 8-3: Values in event and extrainfo column (continued)

Audit Option	event	Command or Access Audited	extrainfo
view_access (continued)	42	insert	Roles: Current active roles Subcommand: "INSERT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
	71	update	Roles: Current active roles Subcommand: "UPDATE" or "WRITETEXT" Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
Note: This event is audited automatically. It is not controlled by an audit option.	73	Turning the auditing parameter on with sp_configure	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect
Note: This event is audited automatically. It is not controlled by an audit option.	74	Turning the auditing parameter off with sp_configure	Roles: Current active roles Subcommand: NULL Previous value: NULL Current value: NULL Other information: NULL Proxy information: Original login name, if a set proxy is in effect

Indexes

None

Referenced by System Procedures

sp_addauditrecord, sp_audit

syscharsets

(*master* database only)

Description

syscharsets contains one row for each character set and sort order defined for use by Adaptive 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

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

Indexes

Unique clustered index on *id*, *csid*, *type*
 Unique nonclustered index on *name*

Referenced by System Procedures

sp_checkreswords, sp_configure, sp_helpsort, sp_serverinfo

syscolumns

(all databases)

Description

syscolumns contains one row for every column in every table and view, and a row for each parameter in a procedure.

Columns

Name	Datatype	Description
<i>id</i>	<i>int</i>	ID of table to which this column belongs or of procedure with which this parameter is associated
<i>number</i>	<i>smallint</i>	Sub-procedure number when the procedure is grouped (0 for non-procedure entries)
<i>colid</i>	<i>tinyint</i>	Column ID
<i>status</i>	<i>tinyint</i>	Indicates unique position for <i>bit</i> columns, whether NULL values are legal in this column, and if more than one check constraint exists for the column
<i>type</i>	<i>tinyint</i>	Physical storage type; copied from <i>systypes</i>
<i>length</i>	<i>tinyint</i>	Physical length of data; copied from <i>systypes</i> or supplied by user
<i>offset</i>	<i>smallint</i>	Offset into the row where this column appears; if negative, this is a variable-length column
<i>usertype</i>	<i>smallint</i>	User type ID; copied from <i>systypes</i>
<i>cdefault</i>	<i>int</i>	ID of the procedure that generates default value for this column
<i>domain</i>	<i>int</i>	Constraint ID of the first rule or check constraint for this column
<i>name</i>	<i>sysname</i>	Column name
<i>printfmt</i>	<i>varchar(255)</i>	Reserved
<i>prec</i>	<i>tinyint</i>	Number of significant digits
<i>scale</i>	<i>tinyint</i>	Number of digits to the right of the decimal point

Name	Datatype	Description
<i>remote_type</i>	<i>int</i>	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 <i>access_server</i> .
<i>remote_name</i>	<i>varchar(30)</i>	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.

Indexes

Unique clustered index on *id, number, colid*

Referenced by System Procedures

sp_bindefault, sp_bindrule, sp_changegroup, sp_checkreswords, sp_column_privileges, sp_commonkey, sp_droptype, sp_dropuser, sp_estspace, sp_foreignkey, sp_help, sp_helpjoins, sp_helprotect, sp_primarykey, sp_rename, sp_statistics, sp_unbindefault, sp_unbindrule

syscomments

(all databases)

Description

syscomments contains entries for each view, rule, default, trigger, table constraint, and procedure. The *text* column contains the original definition statements. If the *text* column is longer than 255 bytes, the entries will span rows. Each object can occupy up to 65,025 rows.

Columns

Name	Datatype	Description
<i>id</i>	<i>int</i>	Object ID to which this text applies
<i>number</i>	<i>smallint</i>	Sub-procedure number when the procedure is grouped (0 for non-procedure entries)
<i>colid</i>	<i>tinyint</i>	Sequence of 255 rows for the object
<i>texttype</i>	<i>smallint</i>	0 for system-supplied comment (for views, rules, defaults, triggers, and procedures); 1 for user-supplied comment (users can add entries that describe an object or column)
<i>language</i>	<i>smallint</i>	Reserved
<i>text</i>	<i>varchar(255)</i>	Actual text of SQL definition statement
<i>colid2</i>	<i>tinyint</i>	Indicates next sequence of rows for the object (see <i>colid</i> above); object can have up to 255 sequences of 255 rows each
<i>status</i>	<i>smallint</i>	

► Note

Do not delete the definition statements from the *text* column of *syscomments*. These statements are required for the Adaptive Server upgrade process. To encrypt a definition statement, run the system procedure *sp_hidetext*. To see if a statement created in release 11.5 or later was deleted, run *sp_checksource*. If the statement was deleted, you must either recreate the object that created the statement or reinstall the application that created the object, which will re-create 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 in order to run Adaptive Server in the evaluated configuration. To enact this restriction, a System Security Officer must reset the parameter called *allow select* on *syscomments.text* column with the system procedure *sp_configure*. For information, see the *System Administration Guide*.

Indexes

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

Referenced by System Procedures

sp_checksource, *sp_helpconstraint*, *sp_helptext*, *sp_hidetext*

sysconfigures

(*master database only*)

Description

sysconfigures contains one row for each configuration parameter that can be set by the user.

Columns

Name	Datatype	Description
<i>config</i>	<i>smallint</i>	Configuration parameter number.
<i>value</i>	<i>int</i>	The user-modifiable value for the parameter with <i>integer</i> datatype. Its value is 0 for the parameters with <i>character</i> datatype.
<i>comment</i>	<i>varchar(255)</i>	Name of the configuration parameter.
<i>status</i>	<i>smallint</i>	Either 1 (dynamic) or 0 (parameter takes effect when Adaptive Server is restarted).
<i>name</i>	<i>varchar(80)</i>	Name of the configuration parameter (the same value as <i>comment</i>).
<i>parent</i>	<i>smallint</i>	Configuration parameter number of the parent; if more than one parent, the additional parent numbers are stored in <i>sysattributes</i> .
<i>value2</i>	<i>varchar(255)</i>	The user-modified value for the parameter with the character datatype. Its value is NULL for parameters with <i>integer</i> datatype. It is also used to store the pool size of a buffer pool.
<i>value3</i>	<i>int</i>	Stores the wash size of a buffer pool.
<i>value4</i>	<i>int</i>	Stores the asynchronous prefetch percents of a buffer pool.

Indexes

Unique clustered index on *config, name, parent*

Nonclustered index on *config, parent*

Nonclustered index on *config*

Referenced by System Procedures

sp_configure

sysconstraints

(all databases)

Description

The *sysconstraints* table has one row for each referential constraint and check constraint associated with a table or column.

Whenever a user declares a new check constraint or referential constraint using *create table* or *alter table*, Adaptive 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.

Columns

Name	Datatype	Description
<i>colid</i>	<i>tinyint</i>	Column number in the table
<i>spare1</i>	<i>tinyint</i>	Unused
<i>constrid</i>	<i>int</i>	Object ID of the constraint
<i>tableid</i>	<i>int</i>	ID of the table on which the constraint is declared
<i>error</i>	<i>int</i>	Constraint specific error message
<i>status</i>	<i>int</i>	The type of constraint: 0x0040 = a referential constraint 0x0080 = a check constraint
<i>spare2</i>	<i>int</i>	Unused

Indexes

Clustered index on *tableid*, *colid*

Unique nonclustered index on *constrid*

Referenced by System Procedures

sp_bindmsg, *sp_bindrule*, *sp_helpconstraint*, *sp_unbindmsg*, *sp_unbindrule*

syscurconfigs

(*master database only*)

Description

syscurconfigs is built dynamically when queried. It contains an entry for each of the configuration parameters, as does *sysconfigs*, but with the current values rather than the default values. In addition, it contains four rows that describe the configuration structure.

Columns

Name	Datatype	Description
<i>config</i>	<i>smallint</i>	Configuration parameter number.
<i>value</i>	<i>int</i>	The current run value for the parameter with <i>integer</i> datatype. Its value is 0 for the parameters with character datatype.
<i>comment</i>	<i>varchar(255)</i>	Amount of memory used by each configuration parameter, represented in a string format. Values marked with a hash mark (#) share memory with other parameters.
<i>status</i>	<i>smallint</i>	Either 1 (dynamic) or 0 (parameter takes effect when Adaptive Server is restarted).
<i>value2</i>	<i>varchar(255)</i>	The current run value for the parameter with the <i>character</i> datatype. Its value is NULL for parameters with the <i>integer</i> datatype.
<i>defvalue</i>	<i>varchar(255)</i>	Default value of the configuration parameter.
<i>minimum_value</i>	<i>int</i>	Minimum value of the configuration parameter.
<i>maximum_value</i>	<i>int</i>	Maximum value of the configuration parameter.
<i>memory_used</i>	<i>int</i>	Integer value for the amount of memory used by each configuration parameter.
<i>display_level</i>	<i>int</i>	Display level of the configuration parameter (the values are 1, 5, and 10).
<i>datatype</i>	<i>int</i>	Datatype of the configuration parameter.
<i>message_num</i>	<i>int</i>	Unused.
<i>apf_percent</i>	<i>int</i>	The current run value for the asynchronous prefetch percent for a buffer pool. Valid only for rows that represent buffer pools.

Indexes

None

Referenced by System Procedures

sp_configure, sp_countmetadata, sp_helpconfig, sp_helpserver, sp_helpsort,
sp_helptext, sp_procqmode, sp_serverinfo

sysdatabases

(*master* database only)

Description

sysdatabases contains one row for each database in Adaptive Server. When Adaptive Server is installed, *sysdatabases* contains entries for the *master* database, the *model* database, the *sybssystemprocs* database, and the *tempdb* database. If you have installed auditing, it also contains an entry for the *sybsecurity* database.

Columns

Name	Datatype	Description
<i>name</i>	<i>sysname</i>	Name of the database
<i>dbid</i>	<i>smallint</i>	Database ID
<i>suid</i>	<i>smallint</i>	Server user ID of database owner
<i>status</i>	<i>smallint</i>	Control bits; those that the user can set with <code>sp_dboption</code> are so indicated in Table 8-4
<i>version</i>	<i>smallint</i>	Unused
<i>logptr</i>	<i>int</i>	Pointer to transaction log
<i>crdate</i>	<i>datetime</i>	Creation date
<i>dumptrdate</i>	<i>datetime</i>	Date of the last dump transaction
<i>status2</i>	<i>intn</i>	Additional control bits (see Table 8-5)
<i>audflags</i>	<i>intn</i>	Audit settings for database
<i>deftabaud</i>	<i>intn</i>	Bit-mask that defines default audit settings for tables
<i>defvwaud</i>	<i>intn</i>	Bit-mask that defines default audit settings for views
<i>defpraud</i>	<i>intn</i>	Bit-mask that defines default audit settings for stored procedures
<i>def_remote_type</i>	<i>smallint</i>	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
<i>def_remote_loc</i>	<i>varchar(255)</i>	Identifies the default storage location to be used for remote tables if no storage location is provided via the stored procedure <i>sp_addobjectdef</i>

Table 8-4 lists the bit representations for the *status* column.

Table 8-4: *status* control bits in the sysdatabases table

Decimal	Hex	Status
4	0x04	<i>select into/bulkcopy</i> ; can be set by user
8	0x08	<i>trunc log on chkpt</i> ; can be set by user
16	0x10	<i>no chkpt on recovery</i> ; can be set by user
32	0x20	Database created with <i>for load</i> option, or crashed while loading database, instructs recovery not to proceed
256	0x100	Database suspect; not recovered; cannot be opened or used; can be dropped only with <i>dbcc dbrepair</i>
512	0x200	<i>ddl in tran</i> ; can be set by user
1024	0x400	<i>read only</i> ; can be set by user
2048	0x800	<i>dbo use only</i> ; can be set by user
4096	0x1000	<i>single user</i> ; can be set by user
8192	0x2000	<i>allow nulls by default</i> ; can be set by user

Table 8-5 lists the bit representations for the *status2* column.

Table 8-5: *status2* control bits in the sysdatabases table

Decimal	Hex	Status
1	0x0001	<i>abort tran on log full</i> ; can be set by user
2	0x0002	<i>no free space acctg</i> ; can be set by user
4	0x0004	<i>auto identity</i> ; can be set by user
8	0x0008	<i>identity in nonunique index</i> ; can be set by user
16	0x0010	Database is offline
32	0x0020	Database is offline until recovery completes
64	0x0040	Database is being recovered (internal use)
32768	0x8000	Database does not have a dedicated log device

Indexes

Unique clustered index on *name*
 Unique nonclustered index on *dbid*

Referenced by System Procedures

sp_addlogin, *sp_addsegment*, *sp_addtype*, *sp_audit*, *sp_changedbowner*,
sp_checknames, *sp_checkreswords*, *sp_databases*, *sp_dboption*, *sp_dbremap*,

**sp_dropdevice, sp_dropsegment, sp_extendsegment, sp_helpdb, sp_logdevice,
sp_renamedb, sp_tables**

sysdepends

(all databases)

Description

sysdepends contains one row for each procedure, view, or table that is referenced by a procedure, view, or trigger.

Columns

Name	Datatype	Description
<i>id</i>	<i>int</i>	Object ID
<i>number</i>	<i>smallint</i>	Procedure number
<i>depid</i>	<i>int</i>	Dependent object ID
<i>depnumber</i>	<i>smallint</i>	Dependent procedure number
<i>status</i>	<i>smallint</i>	Internal status information
<i>selall</i>	<i>bit</i>	On if object is used in select * statement
<i>resultobj</i>	<i>bit</i>	On if object is being updated
<i>readobj</i>	<i>bit</i>	On if object is being read

Indexes

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

Referenced by System Procedures

sp_audit, *sp_depends*

sysdevices

(*master database only*)

Description

sysdevices contains one row for each tape dump device, disk dump device, disk for databases, and disk partition for databases. On the Adaptive Server distribution media, there are four entries in *sysdevices*: one for the master device (for databases), one for a disk dump device, and two for tape dump devices.

Columns

Name	Datatype	Description
<i>low</i>	<i>int</i>	First virtual page number on database device (not used for dump devices)
<i>high</i>	<i>int</i>	Last virtual page number on database device or dump device
<i>status</i>	<i>smallint</i>	Bitmap indicating type of device, default and mirror status (see Table 8-6)
<i>cntrltype</i>	<i>smallint</i>	Controller type (0 if database device, 2 if disk dump device or streaming tape, 3–8 if tape dump device)
<i>name</i>	<i>sysname</i>	Logical name of dump device or database device
<i>phyname</i>	<i>varchar(127)</i>	Name of physical device
<i>mirrorname</i>	<i>varchar(127)</i>	Name of mirror device

The bit representations for the *status* column, shown in Table 8-6, are additive. For example, “3” indicates a physical disk that is also a default.

Table 8-6: status control bits in the *sysdevices* table

Decimal	Hex	Status
1	0x01	Default disk
2	0x02	Physical disk
4	0x04	Logical disk (not used)
8	0x08	Skip header
16	0x10	Dump device
32	0x20	Serial writes
64	0x40	Device mirrored

Table 8-6: status control bits in the sysdevices table (continued)

Decimal	Hex	Status
128	0x80	Reads mirrored
256	0x100	Secondary mirror side only
512	0x200	Mirror enabled
1024	0x400	Master device is mirrored
2048	0x800	Mirror disabled (used internally)
4096	0x1000	Primary device needs to be unmirrored (used internally)
8192	0x2000	Secondary device needs to be unmirrored (used internally)

Indexes

Unique clustered index on *name*

Referenced by System Procedures

sp_addsegment, sp_addumpdevice, sp_checknames, sp_checkreswords,
 sp_configure, sp_diskdefault, sp_dropdevice, sp_dropsegment,
 sp_extendsegment, sp_helpdb, sp_helpdevice, sp_helplog, sp_helpsegment,
 sp_logdevice, sp_volchanged

sysengines

(*master database only*)

Description

sysengines contains one row for each Adaptive Server engine currently online.

Columns

Name	Datatype	Description
<i>engine</i>	<i>smallint</i>	Engine number
<i>osprocid</i>	<i>int</i>	Operating system process ID (may be NULL)
<i>osprocname</i>	<i>char</i>	Operating system process name (may be NULL)
<i>status</i>	<i>char</i>	One of: online, offline, in create, in destroy, debug
<i>affinitied</i>	<i>int</i>	Number of Adaptive Server processes with affinity to this engine
<i>cur_kpid</i>	<i>int</i>	Kernel process ID of process currently running on this engine, if any
<i>last_kpid</i>	<i>int</i>	Kernel process ID of process that previously ran on this engine
<i>idle_1</i>	<i>tinyint</i>	Reserved
<i>idle_2</i>	<i>tinyint</i>	Reserved
<i>idle_3</i>	<i>tinyint</i>	Reserved
<i>idle_4</i>	<i>tinyint</i>	Reserved
<i>starttime</i>	<i>datetime</i>	Date and time engine came online

Indexes

None

Referenced by System Procedures

sp_monitor

sysgams

(all databases)

Description

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*.

Columns

None

Indexes

None

Referenced by System Procedures

None

sysindexes

(all databases)

Description

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.

Columns

Name	Datatype	Description
<i>name</i>	<i>sysname</i>	Index or table name
<i>id</i>	<i>int</i>	ID of table, or ID of table to which the index belongs
<i>indid</i>	<i>smallint</i>	0 if table, 1 if clustered index, >1 if nonclustered, 255 if text chain
<i>doampg</i>	<i>int</i>	Page number for the object allocation map of a table or clustered index
<i>ioampg</i>	<i>int</i>	Page number for the allocation map of a nonclustered index
<i>oampgtrips</i>	<i>int</i>	Ratio of OAM page to data page residency in cache
<i>status2</i>	<i>int</i>	Internal system status information (see Table 8-7)
<i>ipgtrips</i>	<i>int</i>	Ratio of index page to data page residency in cache
<i>first</i>	<i>int</i>	Page number of the first data or leaf page
<i>root</i>	<i>int</i>	Page number of the root page if entry is an index; page number of the last page if entry is an unpartitioned table or text chain; unused if entry is a partitioned table (see <i>syspartitions</i>)
<i>distribution</i>	<i>int</i>	Page number of the distribution page (if entry is an index)
<i>usagecnt</i>	<i>smallint</i>	Reserved
<i>segment</i>	<i>smallint</i>	Number of segment in which this object resides
<i>status</i>	<i>smallint</i>	Internal system status information (see Table 8-8)

Name	Datatype	Description
<i>maxrowsperpage</i>	<i>smallint</i>	Maximum number of rows per page
<i>minlen</i>	<i>smallint</i>	Minimum size of a row
<i>maxlen</i>	<i>smallint</i>	Maximum size of a row
<i>maxirow</i>	<i>smallint</i>	Maximum size of a non-leaf index row
<i>keycnt</i>	<i>smallint</i>	Number of keys for a clustered index; number of keys+1 for a nonclustered index
<i>keys1</i>	<i>varbinary(255)</i>	Description of key columns (if entry is an index)
<i>keys2</i>	<i>varbinary(255)</i>	Description of key columns (if entry is an index)
<i>soid</i>	<i>tinyint</i>	Sort order ID that the index was created with; 0 if there is no character data in the keys
<i>csid</i>	<i>tinyint</i>	Character set ID that the index was created with; 0 if there is no character data in the keys
<i>base_partition</i>	<i>int</i>	Partition number, incremented by <code>alter table...unpartition</code> commands

The *doampg* column is used only if the row describes a table or clustered index; this column and the *ioampg* column are used by the system functions `data_pgs`, `reserved_pgs`, and `used_pgs`.

Table 8-7 lists the bit representations for the *status2* column.

Table 8-7: status2 control bits in the sysindexes table

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	User did not specify a constraint name
16	0x10	Large I/Os (prefetch) enabled for table, index, or text chain
32	0x20	MRU cache strategy enabled for table, index, or text chain
64	0x40	Ascending inserts turned on for the table

Table 8-8 lists the bit representations for the *status* column.

Table 8-8: status control bits in the sysindexes table

Decimal	Hex	Status
1	0x1	Abort current command or trigger if an attempt is made to insert duplicate key
2	0x2	Unique index
4	0x4	Abort current command or trigger if an attempt is made to insert duplicate row
16	0x10	Clustered index
64	0x40	Index allows duplicate rows
128	0x80	Sorted object; not set for tables without clustered indexes or for text objects
512	0x200	sorted data option used in create index statement
1024	0x400	Index being created
2048	0x800	Index on primary key
32768	0x8000	Suspect index; index was created under another sort order

Indexes

Unique clustered index on *id*, *indid*

Referenced by System Procedures

sp_cachestrategy, sp_checknames, sp_checkreswords, sp_dropsegment, sp_estspace, sp_help, sp_helpconstraint, sp_helpindex, sp_helplog, sp_helpsegment, sp_indsuspect, sp_pkeys, sp_placeobject, sp_relimit, sp_rename, sp_spaceused, sp_special_columns, sp_statistics

syskeys

(all databases)

Description

syskeys contains one row for each primary, foreign, or common key.

Columns

Name	Datatype	Description
<i>id</i>	<i>int</i>	Object ID
<i>type</i>	<i>smallint</i>	Record type
<i>depid</i>	<i>int null</i>	Dependent object ID
<i>keycnt</i>	<i>int null</i>	Number of non-null keys
<i>size</i>	<i>int null</i>	Reserved
<i>key1</i>	<i>int null</i>	Column ID
<i>key2</i>	<i>int null</i>	Column ID
<i>key3</i>	<i>int null</i>	Column ID
<i>key4</i>	<i>int null</i>	Column ID
<i>key5</i>	<i>int null</i>	Column ID
<i>key6</i>	<i>int null</i>	Column ID
<i>key7</i>	<i>int null</i>	Column ID
<i>key8</i>	<i>int null</i>	Column ID
<i>depkey1</i>	<i>int null</i>	Column ID
<i>depkey2</i>	<i>int null</i>	Column ID
<i>depkey3</i>	<i>int null</i>	Column ID
<i>depkey4</i>	<i>int null</i>	Column ID
<i>depkey5</i>	<i>int null</i>	Column ID
<i>depkey6</i>	<i>int null</i>	Column ID
<i>depkey7</i>	<i>int null</i>	Column ID
<i>depkey8</i>	<i>int null</i>	Column ID

Indexes

Clustered index on *id*

Referenced by System Procedures

sp_commonkey, sp_dropkey, sp_foreignkey, sp_helpjoins, sp_helpkey,
sp_primarykey

syslanguages

(*master* database only)

Description

syslanguages contains one row for each language known to Adaptive Server. *us_english* is not in *syslanguages*, but it is always available to Adaptive Server.

Columns

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

Referenced by System Procedures

sp_addlanguage, sp_addmessage, sp_checkreswords, sp_configure,
sp_droplanguage, sp_dropmessage, sp_getmessage, sp_helplanguage,
sp_setlangalias

syslisteners

(*master database only*)

Description

syslisteners contains a row for each network protocol available for connecting with the current Adaptive Server. Adaptive Server builds *syslisteners* dynamically when a user or client application queries the table.

Columns

Name	Datatype	Description
<i>net_type</i>	<i>char(32)</i>	Network protocol
<i>address_info</i>	<i>char(255)</i>	Information that uniquely identifies this Adaptive Server on the network, usually the name of the current Adaptive Server and an identifying number, such as the server's port number for the protocol

Indexes

None

Referenced by System Procedures

None

syslocks

(*master database only*)

Description

syslocks contains information about active locks, but it is not a normal table. Rather, it is built dynamically when queried by a user. No updates to *syslocks* are allowed.

Columns

Name	Datatype	Description
<i>id</i>	<i>int</i>	Table ID
<i>dbid</i>	<i>smallint</i>	Database ID
<i>page</i>	<i>int</i>	Page number
<i>type</i>	<i>smallint</i>	Type of lock (bit values for the <i>type</i> column are listed in Table 8-9)
<i>spid</i>	<i>smallint</i>	ID of process that holds the lock
<i>class</i>	<i>char(30)</i>	Name of the cursor this lock is associated with, if any
<i>fid</i>	<i>smallint</i>	The family (coordinating process and its worker processes) to which the lock belongs. <i>fid</i> values are listed in Table 8-10.
<i>context</i>	<i>tinyint</i>	Context type of lock request. <i>context</i> values are listed in Table 8-11.

Table 8-9 lists the bit representations for the *type* column.

Table 8-9: *type* control bits in the *syslocks* table

Decimal	Hex	Status
1	0x1	Exclusive table lock
2	0x2	Shared table lock
3	0x3	Exclusive intent lock (will do page locking on indicated pages)
4	0x4	Shared intent lock
5	0x5	Exclusive page lock
6	0x6	Shared page lock
7	0x7	Update page lock (changes to exclusive if page is modified)
256	0x100	Lock is blocking another process
512	0x200	Demand lock

Table 8-10 lists the values for the *fid* column:

Table 8-10: fid column values in the syslocks table

Value	Interpretation
0	The task represented by the <i>spid</i> is a single task executing a statement in serial.
Nonzero value	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.

Table 8-11 lists the values for the *context* column:

Table 8-11: context column values in the syslocks table

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) under the following conditions: <ul style="list-style-type: none"> • The lock is a table lock held as part of a parallel query • The lock is held by a worker process at transaction isolation level 3 • The lock is held by a worker process in a parallel query and must be held for the duration of the transaction

Indexes

None

Referenced by System Procedures

sp_familylock, sp_lock

sysloginroles

(master database only)

Description

sysloginroles contains a row for each instance of a server login possessing a system 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*).

Columns

Name	Datatype	Description
<i>suid</i>	<i>smallint</i>	Server user ID
<i>srid</i>	<i>smallint</i>	Server role ID; one of the following: 0 = <i>sa_role</i> 1 = <i>sso_role</i> 2 = <i>oper_role</i> 4 = <i>navigator_role</i> 5 = <i>replication_role</i>
<i>status</i>	<i>smallint</i>	Reserved

Indexes

Clustered index on *suid*

Referenced by System Procedures

sp_displaylogin, *sp_droplogin*, *sp_locklogin*, *sp_role*

syslogins

(master database only)

Description

syslogins contains one row for each valid Adaptive Server user account.

Columns

Name	Datatype	Description
<i>suid</i>	<i>smallint</i>	Server user ID
<i>status</i>	<i>smallint</i>	Status of the account (see Table 8-12)
<i>accdate</i>	<i>datetime</i>	Date <i>totcpu</i> and <i>totio</i> were last cleared
<i>totcpu</i>	<i>int</i>	CPU time accumulated by login
<i>totio</i>	<i>int</i>	I/O accumulated by login
<i>spacelimit</i>	<i>int</i>	Reserved
<i>timelimit</i>	<i>int</i>	Reserved
<i>resultlimit</i>	<i>int</i>	Reserved
<i>dbname</i>	<i>sysname</i>	Name of database in which to put user when connection established
<i>name</i>	<i>sysname</i>	Login name of user
<i>password</i>	<i>varbinary</i>	Password of user (encrypted)
<i>language</i>	<i>varchar(30)</i>	User's default language
<i>pwdate</i>	<i>datetime</i>	Date the password was last changed
<i>audflags</i>	<i>int</i>	User's audit settings
<i>fullname</i>	<i>varchar(30)</i>	Full name of the user
<i>srvname</i>	<i>varchar(30)</i>	Name of server to which a passthrough connection must be established if the AUTOCONNECT flag is turned on.

On the Adaptive Server 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 Adaptive Server.

Table 8-12 lists the bit representations for the *status* column:

Table 8-12: status control bits in the syslogins table

Decimal	Hex	Status
1	0x1	Password contains fewer than 6 characters or is NULL
2	0x2	Account is locked
4	0x4	Password has expired

Indexes

Unique clustered index on *suid*

Unique nonclustered index on *name*

Referenced by System Procedures

sp_addalias, sp_addlogin, sp_addremotelogin, sp_adduser, sp_audit,
 sp_changedbowner, sp_checknames, sp_checkreswords, sp_clearstats,
 sp_displaylogin, sp_droplogin, sp_helpdb, sp_helpuser, sp_locklogin,
 sp_modifylogin, sp_reportstats, sp_role

syslogs

(all databases)

Description

syslogs contains the transaction log. It is used by Adaptive 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

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

Indexes

None

Referenced by System Procedures

None

syslogshold

(*master database only*)

Description

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.

Columns

Name	Datatype	Description
<i>dbid</i>	<i>smallint</i>	Database ID.
<i>reserved</i>	<i>int</i>	Unused.
<i>spid</i>	<i>smallint</i>	Server process ID of the user that owns the oldest active transaction (always 0 for Replication Server).
<i>page</i>	<i>int</i>	Starting page number of active portion in <i>syslogs</i> defined by oldest transaction (or the truncation page in <i>syslogs</i> for Replication Server).
<i>xactid</i>	<i>char(6)</i>	ID of the oldest active transaction (always 0x000000 for Replication Server).
<i>masterxactid</i>	<i>char(6)</i>	ID of the transaction's master transaction (if any) for multi-database transactions; otherwise 0x000000 (always 0x000000 for Replication Server).
<i>starttime</i>	<i>datetime</i>	Date and time the transaction started (or when the truncation point was set for Replication Server).

Name	Datatype	Description
<i>name</i>	<i>char(67)</i>	Name of the oldest active transaction. It is the name defined with begin transaction , “\$user_transaction” if no value is specified with begin transaction , or “\$chained_transaction” for implicit transactions started by the ANSI chained mode. Internal transactions started by Adaptive Server have names that begin with the dollar sign (\$) and are named for the operation, or are named “\$replication_truncation_point” for Replication Server.

Indexes

None

Referenced by System Procedures

None

sysmessages

(*master database only*)

Description

sysmessages contains one row for each system error or warning that can be returned by Adaptive Server. Adaptive Server displays the error description on the user's screen.

Columns

Name	Datatype	Description
<i>error</i>	<i>int</i>	Unique error number
<i>severity</i>	<i>smallint</i>	Severity level of error
<i>dlevel</i>	<i>smallint</i>	Reserved
<i>description</i>	<i>varchar(255)</i>	Explanation of error with placeholders for parameters
<i>langid</i>	<i>smallint</i>	Language; null for us_english
<i>sqlstate</i>	<i>varchar(5)</i>	SQLSTATE value for the error

Indexes

Clustered index on *error*, *dlevel*

Unique nonclustered index on *error*, *dlevel*, *langid*

Referenced by System Procedures

sp_configure, *sp_dboption*, *sp_depends*, *sp_droplanguage*, *sp_getmessage*, *sp_help*, *sp_helpdb*, *sp_helpdevice*, *sp_helpremotelogin*, *sp_remotoption*

sysmonitors

(*master* database only)

Description

sysmonitors contains one row for each monitor counter.

Columns

Name	Datatype	Description
<i>field_name</i>	<i>char(35)</i>	Name of the counter
<i>group_name</i>	<i>char(25)</i>	Group this counter belongs to
<i>field_id</i>	<i>smallint</i>	Unique identifier for the row
<i>value</i>	<i>int</i>	Current value of the counter
<i>description</i>	<i>char(255)</i>	Description of the counter; not used

Indexes

None

Referenced by System Procedures

sp_sysmon

sysobjects

(all databases)

Description

sysobjects contains one row for each table, view, stored procedure, extended stored procedure, log, rule, default, trigger, check constraint, referential constraint, and (in *tempdb* only) temporary object.

Columns

Name	Datatype	Description
<i>name</i>	<i>sysname</i>	Object name
<i>id</i>	<i>int</i>	Object ID
<i>uid</i>	<i>smallint</i>	User ID of object owner
<i>type</i>	<i>char(2)</i>	One of the following object types: D = default L = log P = procedure PR = prepare objects (created by Dynamic SQL) R = rule RI = referential constraint S = system table TR = trigger U = user table V = view XP = extended stored procedure
<i>userstat</i>	<i>smallint</i>	Application-dependent type information (32768 decimal [0x8000 hex] indicates to Data Workbench® that a procedure is a report)
<i>sysstat</i>	<i>smallint</i>	Internal status information (256 decimal [0x100 hex] indicates that table is read-only)
<i>indexdel</i>	<i>smallint</i>	Index delete count (incremented if an index is deleted)
<i>schemacnt</i>	<i>smallint</i>	Count of changes in the schema of an object (incremented if a rule or default is added)
<i>sysstat2</i>	<i>smallint</i>	Additional internal status information (see Table 8-13)
<i>crdate</i>	<i>datetime</i>	Date the object was created
<i>expdate</i>	<i>datetime</i>	Reserved

Name	Datatype	Description
<i>deltrig</i>	<i>int</i>	Stored procedure ID of a delete trigger if the entry is a table. Table ID if the entry is a trigger.
<i>instrig</i>	<i>int</i>	Stored procedure ID of a table's insert trigger if the entry is a table
<i>updtrig</i>	<i>int</i>	Stored procedure ID of a table's update trigger if the entry is a table
<i>seltrig</i>	<i>int</i>	Reserved
<i>ckfirst</i>	<i>int</i>	ID of first check constraint on the table
<i>cache</i>	<i>smallint</i>	Reserved
<i>audflags</i>	<i>int</i>	Object's audit settings
<i>objspare</i>	<i>int</i>	Spare
<i>versions</i>	<i>binary</i>	

Table 8-13 lists the bit representations for the *sysstat2* column:

Table 8-13: *sysstat2* control bits in the *sysobjects* table

Decimal	Hex	Status
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
512	0x200	Table does not contain variable-length columns

Indexes

Unique clustered index on *id*

Unique nonclustered index on *name*, *uid*

Referenced by System Procedures

sp_addmessage, *sp_addthreshold*, *sp_audit*, *sp_bindefault*, *sp_bindmsg*, *sp_bindrule*, *sp_checknames*, *sp_checkreswords*, *sp_column_privileges*, *sp_columns*, *sp_commonkey*, *sp_depends*, *sp_dropgroup*, *sp_dropkey*, *sp_dropsegment*, *sp_droptreshold*, *sp_droptype*, *sp_dropuser*, *sp_estspace*, *sp_fkeys*, *sp_foreignkey*, *sp_help*, *sp_helpconstraint*, *sp_helpindex*, *sp_helpjoins*, *sp_helpkey*, *sp_helpprotect*, *sp_helpthreshold*, *sp_indsuspect*, *sp_modifythreshold*,

sp_pkeys, sp_placeobject, sp_primarykey, sp_procxmode, sp_recompile, sp_relimit,
sp_remap, sp_rename, sp_spaceused, sp_sproc_columns, sp_statistics,
sp_stored_procedures, sp_table_privileges, sp_tables, sp_unbindefault,
sp_unbindmsg, sp_unbindrule

syspartitions

(all databases)

Description

syspartitions contains one row for each partition (page chain) of a partitioned table.

Columns

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

Indexes

Unique clustered index on *id*, *partitionid*

Referenced by System Procedures

sp_help, *sp_helppartition*, *sp_placeobject*

sysprocedures

(all databases)

Description

sysprocedures contains entries for each view, default, rule, trigger, procedure, declarative default, and check constraint. The plan or sequence tree for each object 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

Name	Datatype	Description
<i>type</i>	<i>smallint</i>	Object type (see Table 8-14)
<i>id</i>	<i>int</i>	Object ID
<i>sequence</i>	<i>smallint</i>	Sequence number if more than one row is used to describe this object
<i>status</i>	<i>smallint</i>	Internal system status
<i>number</i>	<i>smallint</i>	Sub-procedure number when the procedure is grouped (0 for non-procedure entries)
<i>version</i>	<i>int</i>	

Table 8-14 lists the bit representations for the *type* column.

Table 8-14: *type* control bits in the *sysprocedures* table

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

Indexes

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

Referenced by System Procedures

sp_bindefault, *sp_bindrule*, *sp_remap*, *sp_sproc_columns*, *sp_stored_procedures*, *sp_unbindefault*, *sp_unbindrule*

sysprocesses

(*master database only*)

Description

sysprocesses contains information about Adaptive Server processes, but it is not a normal table. Rather, it is built dynamically when queried by a user. No updates to *sysprocesses* are allowed.

Use the `kill` statement to kill a process.

Columns

Name	Datatype	Description
<i>spid</i>	<i>smallint</i>	Process ID
<i>kpid</i>	<i>int</i>	Kernel process ID
<i>enginenum</i>	<i>int</i>	Number of engine on which process is being executed
<i>status</i>	<i>char(12)</i>	Process ID status (see Table 8-15)
<i>suid</i>	<i>smallint</i>	Server user ID of user who issued command
<i>hostname</i>	<i>char(10)</i>	Name of host computer
<i>program_name</i>	<i>char(16)</i>	Name of front-end module
<i>hostprocess</i>	<i>char(8)</i>	Host process ID number
<i>cmd</i>	<i>char(16)</i>	Command currently being executed
<i>cpu</i>	<i>int</i>	Cumulative CPU time for process in ticks
<i>physical_io</i>	<i>int</i>	Number of disk reads and writes for current command
<i>memusage</i>	<i>int</i>	Amount of memory allocated to process
<i>blocked</i>	<i>smallint</i>	Process ID of blocking process, if any
<i>dbid</i>	<i>smallint</i>	Database ID
<i>uid</i>	<i>smallint</i>	ID of user who executed command
<i>gid</i>	<i>smallint</i>	Group ID of user who executed command
<i>tran_name</i>	<i>varchar(64)</i>	Name of the active transaction
<i>time_blocked</i>	<i>int</i>	Time blocked in seconds
<i>network_pktsz</i>	<i>int</i>	Current connection's network packet size
<i>fid</i>	<i>smallint</i>	Process ID of the worker process' parent
<i>execlass</i>	<i>varchar(30)</i>	Execution class that the process is bound to

Name	Datatype	Description
<i>priority</i>	<i>varchar(10)</i>	Base priority associated with the process
<i>affinity</i>	<i>varchar(30)</i>	Name of the engine to which the process has affinity
<i>id</i>	<i>int</i>	Object ID of the currently running procedure (or 0 if no procedure is running)
<i>stmtnum</i>	<i>int</i>	The current statement number within the running procedure (or the SQL batch statement number if no procedure is running)
<i>linenum</i>	<i>int</i>	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)
<i>origsuid</i>	<i>smallint</i>	Original server user ID. If this value is not NULL, a user with an <i>suid</i> of <i>origsuid</i> executed set proxy or set session authorization to impersonate the user who executed the command.

Table 8-15 lists the values for the *status* column:

Table 8-15: sysprocesses status column values

Status	Meaning
alarm sleep	Waiting for alarm to wake process up (user executed a waitfor delay command)
background	A process, such as a threshold procedure, run by Adaptive Server rather than by a user process
infected	Server has detected a serious error condition; extremely rare
lock sleep	Waiting on a lock acquisition
log suspend	Processes suspended by reaching the last-chance threshold on the log
recv sleep	Waiting on a network read
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

Indexes

None

Referenced by System Procedures

sp_dboption, sp_droplogin, sp_locklogin, sp_role, sp_showplan, sp_who

sysprotects

(all databases)

Description

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

Columns

Name	Datatype	Description
<i>id</i>	<i>int</i>	ID of the object to which this permission applies.
<i>uid</i>	<i>smallint</i>	ID of the user, group, or role to which this permission applies.
<i>action</i>	<i>tinyint</i>	One of the following permissions: 167 = set proxy or set session authorization 193 = select 195 = insert 196 = delete 197 = update 224 = execute 151 = references 203 = create database 233 = create default 222 = create procedure 236 = create rule 198 = create table 207 = create view 228 = dump database 235 = dump transaction
<i>protecttype</i>	<i>tinyint</i>	One of the following values: 0 = grant with grant 1 = grant 2 = revoke
<i>columns</i>	<i>varbinary(32)</i>	Bitmap of columns to which this select or update permission applies. Bit 0 indicates all columns; 1 means permission applies to that column; NULL means no information.
<i>grantor</i>	<i>smallint</i>	User ID of the grantor (or of object owner if grantor is a System Administrator).

Indexes

Unique clustered index on *id, action, grantor, uid, protecttype*

Referenced by System Procedures

*sp_changegroup, sp_dropgroup, sp_dropuser, sp_helprotect,
sp_stored_procedures, sp_tables*

sysreferences

(all databases)

Description

sysreferences contains one row for each referential integrity constraint declared on a table or column.

Columns

Name	Datatype	Description
<i>indexid</i>	<i>smallint</i>	ID of the unique index on referenced columns
<i>constrid</i>	<i>int</i>	Object ID of the constraint from <i>sysobjects</i>
<i>tableid</i>	<i>int</i>	Object ID of the referencing table
<i>reftabid</i>	<i>int</i>	Object ID of the referenced table
<i>keycnt</i>	<i>tinyint</i>	Number of columns in the foreign key
<i>status</i>	<i>smallint</i>	Reserved
<i>frgnbid</i>	<i>smallint</i>	Reserved
<i>frgndbname</i>	<i>varchar(30)</i>	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
<i>pmrydbid</i>	<i>smallint</i>	Reserved
<i>pmrydbname</i>	<i>varchar(30)</i>	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
<i>spare2</i>	<i>int</i>	Reserved
<i>fokey1</i>	<i>tinyint</i>	Column ID of the first referencing column
.		
.		
<i>fokey16</i>	<i>tinyint</i>	Column ID of the sixteenth referencing column
<i>refkey1</i>	<i>tinyint</i>	Column ID of the first referenced column
.		
.		
<i>refkey16</i>	<i>tinyint</i>	Column ID of the sixteenth referenced column

Indexes

Clustered index on *tableid, frgndbname*

Unique nonclustered index on *frgnbid, constrid*

Nonclustered index on *reftabid, indexid, pmrydbname*

Referenced by System Procedures

sp_fkeys, sp_helpconstraint

sysremotelogins

(*master database only*)

Description

sysremotelogins contains one row for each remote user that is allowed to execute remote procedure calls on this Adaptive Server.

Columns

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

Indexes

Unique clustered index on *remoteserverid*, *remoteusername*

Referenced by System Procedures

sp_addrmotelogin, *sp_checknames*, *sp_checkreswords*, *sp_droptremotelogin*, *sp_dropserver*, *sp_helpremotelogin*, *sp_remotoption*

sysresourcelimits

(*master* database only)

Description

sysresourcelimits contains a row for each resource limit defined by Adaptive Server. Resource limits specify the maximum amount of server resources that can be used by a Adaptive Server login or an application to execute a query, query batch, or transaction.

Columns

Name	Datatype	Description
<i>name</i>	<i>varchar(30) null</i>	Login name
<i>appname</i>	<i>varchar(30) null</i>	Application name
<i>rangeid</i>	<i>smallint</i>	<i>id</i> column from <i>systimeranges</i>
<i>limitid</i>	<i>smallint</i>	<i>id</i> column from <i>spt_limit_types</i>
<i>limitvalue</i>	<i>int</i>	Value of limit
<i>enforced</i>	<i>tinyint</i>	Subset of the <i>enforced</i> column from <i>spt_limit_types</i> : 1 = prior to execution 2 = during execution 3 = both
<i>actiontotake</i>	<i>tinyint</i>	Action to take on a violation: 1 = issue warning 2 = abort query batch 3 = abort transaction 4 = kill session
<i>scope</i>	<i>tinyint</i>	Scope of user limit (a bitmap indicating one or more of the following): 1 = query 2 = query batch 4 = transaction
<i>spare</i>	<i>tinyint</i>	Reserved

Indexes

Clustered index on *name*, *appname*

Referenced by System Procedures

sp_add_resource_limit, *sp_drop_resource_limit*, *sp_help_resource_limit*,
sp_modify_resource_limit

sysroles

(all databases)

Description

sysroles maps server role IDs to local role IDs.

Columns

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

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

Indexes

Unique clustered index on *lrid*

Referenced by System Procedures

None

syssecmechs

(*master database only*)

Description

syssecmechs contains information about the security services supported by each security mechanism that is available to Adaptive Server, but it is not a normal table. Rather, it is built dynamically when queried by a user.

Columns

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

Indexes

None

Referenced by System Procedures

None

syssegments

(all databases)

Description

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.

Columns

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

Indexes

None

Referenced by System Procedures

sp_addsegment, *sp_addthreshold*, *sp_checknames*, *sp_checkreswords*,
sp_dropsegment, *sp_droptreshold*, *sp_dropuser*, *sp_extendsegment*, *sp_helpdb*,
sp_helpindex, *sp_helpsegment*, *sp_helpthreshold*, *sp_modifythreshold*,
sp_placeobject

sys.servers

(master database only)

Description

sys.servers contains one row for each remote Adaptive Server, Backup Server™, or Open Server™ on which this Adaptive Server can execute remote procedure calls.

Columns

Name	Datatype	Description
<i>srvid</i>	<i>smallint</i>	ID number (for local use only) of the remote server
<i>srvstatus</i>	<i>smallint</i>	Bitmap of options (see Table 8-14)
<i>srvname</i>	<i>varchar(30)</i>	Server name
<i>srvnetname</i>	<i>varchar(32)</i>	Interfaces file name for the server
<i>srvclass</i>	<i>smallint</i>	Server category defined by the class parameter of <i>sp_addserver</i> . See Table 8-17.
<i>srvsecmech</i>	<i>varchar(30)</i>	Security mechanism

Table 8-16 lists the bit representations for the *srvstatus* column:

Table 8-16: status control bits in the *sys.servers* table

Decimal	Hex	Status
0	0x0	Timeouts are enabled
1	0x1	Timeouts are disabled
2	0x2	Network password encryption is enabled
4	0x4	Remote server is read only
8	0x8	Use rpc security model A

Table 8-17 lists the server categories for the *srvclass* column:

Table 8-17: Server categories in the *sys.servers* table

srvclass	Server category
0	Local server (this server)
1	Another Adaptive Server or Component Integration Services server
3	Server coded to the DirectCONNECT specification
4	Server accessible by Net-Gateway or MDI Database Gateway
5	Server coded to the Generic Access Module specification

Indexes

Unique clustered index on *srv_id*

Unique nonclustered index on *srvname*

Referenced by System Procedures

sp_addremotelogin, *sp_addserver*, *sp_checknames*, *sp_checkreswords*,
sp_configure, *sp_droptremotelogin*, *sp_dropserver*, *sp_helpremotelogin*,
sp_helpserver, *sp_remotoption*, *sp_serveroption*

sysrvroles

(*master* database only)

Description

sysrvroles contains a row for each system or user-defined role.

Columns

Name	Datatype	Description
<i>srid</i>	<i>smallint</i>	Server role ID
<i>name</i>	<i>varchar(30)</i>	Name of the role
<i>password</i>	<i>varinary(30)</i>	Password for the role (encrypted)

Indexes

Unique clustered index on *srid*

Referenced by System Procedures

sp_adduser, *sp_changegroup*, *sp_displaylogin*, *sp_dropgroup*, *sp_helpgroup*,
sp_role

systhresholds

(all databases)

Description

systhresholds contains one row for each threshold defined for the database.

Columns

Name	Datatype	Description
<i>segment</i>	<i>smallint</i>	Segment number for which free space is being monitored.
<i>free_space</i>	<i>int</i>	Size of threshold, in 2K pages (4K for Stratus).
<i>status</i>	<i>smallint</i>	Bit 1 equals 1 for the logsegment's last-chance threshold, 0 for all other thresholds.
<i>proc_name</i>	<i>varchar(255)</i>	Name of the procedure that is executed when the number of unused pages on <i>segment</i> falls below <i>free_space</i> .
<i>suid</i>	<i>smallint</i>	The server user ID of the user who added the threshold or modified it most recently.
<i>currauth</i>	<i>varbinary(255)</i>	A bit mask that indicates which roles were active for <i>suid</i> at the time the threshold was added or most recently modified. When the threshold is crossed, <i>proc_name</i> executes with this set of roles, less any that have been deactivated since the threshold was added or last modified.

Indexes

Unique clustered index on *segment*, *free_space*

Referenced by System Procedures

sp_addthreshold, *sp_dropsegment*, *sp_dropthreshold*, *sp_dropuser*,
sp_helpthreshold, *sp_modifythreshold*

systimeranges

(*master* database only)

Description

systimeranges stores named time ranges, which are used by Adaptive Server to control when a resource limit is active.

Columns

Name	Datatype	Description
<i>name</i>	<i>varchar(30)</i>	Unique name of the time range.
<i>id</i>	<i>smallint</i>	Unique identifier for the time range. 1 represents the “at all times” limit.
<i>startday</i>	<i>tinyint</i>	Day of week (1–7) for the beginning of the range. Monday = 1, Sunday = 7.
<i>endday</i>	<i>tinyint</i>	Day of week (1–7) for the end of the range. Monday = 1, Sunday = 7.
<i>starttime</i>	<i>varchar(10)</i>	Time of day for the beginning of the range.
<i>endtime</i>	<i>varchar(10)</i>	Time of day for the end of the range.

Indexes

Clustered index on *id*

Referenced by System Procedures

sp_add_resource_limit, *sp_add_time_range*, *sp_drop_resource_limit*,
sp_drop_time_range, *sp_help_resource_limit*, *sp_modify_resource_limit*,
sp_modify_time_range

systypes

(all databases)

Description

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

The rows that describe system-supplied datatypes cannot be altered.

Columns

Name	Datatype	Description
<i>uid</i>	<i>smallint</i>	User ID of datatype creator
<i>usertype</i>	<i>smallint</i>	User type ID
<i>variable</i>	<i>bit</i>	1 if datatype is variable length; 0 otherwise
<i>allownulls</i>	<i>bit</i>	Indicates whether nulls are allowed for this datatype
<i>type</i>	<i>tinyint</i>	Physical storage datatype
<i>length</i>	<i>tinyint</i>	Physical length of datatype
<i>tdefault</i>	<i>int</i>	ID of system procedure that generates default for this datatype
<i>domain</i>	<i>int</i>	ID of system procedure that contains integrity checks for this datatype
<i>name</i>	<i>sysname</i>	Datatype name
<i>printfmt</i>	<i>varchar(255)</i>	Reserved
<i>prec</i>	<i>tinyint</i>	Number of significant digits
<i>scale</i>	<i>tinyint</i>	Number of digits to the right of the decimal point
<i>ident</i>	<i>tinyint</i>	1 if column has the IDENTITY property, 0 if it does not
<i>hierarchy</i>	<i>tinyint</i>	Precedence of the datatype in mixed mode arithmetic

Table 8-18 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:

Table 8-18: Datatype names, hierarchy, types, and usertypes

Name	<i>hierarchy</i>	<i>type</i>	<i>usertype</i>
<i>floatn</i>	1	109	14
<i>float</i>	2	62	8
<i>datetimn</i>	3	111	15
<i>datetime</i>	4	61	12
<i>real</i>	5	59	23
<i>numericn</i>	6	108	28
<i>numeric</i>	7	63	10
<i>decimaln</i>	8	106	27
<i>decimal</i>	9	55	26
<i>moneyn</i>	10	110	17
<i>money</i>	11	60	11
<i>smallmoney</i>	12	122	21
<i>smalldatetime</i>	13	58	22
<i>intn</i>	14	38	13
<i>int</i>	15	56	7
<i>smallint</i>	16	52	6
<i>tinyint</i>	17	48	5
<i>bit</i>	18	50	16
<i>varchar</i>	19	39	2
<i>sysname</i>	19	39	18
<i>nvarchar</i>	19	39	25
<i>char</i>	20	47	1
<i>nchar</i>	20	47	24
<i>varbinary</i>	21	37	4
<i>timestamp</i>	21	37	80
<i>binary</i>	22	45	3
<i>text</i>	23	35	19
<i>image</i>	24	34	20

Indexes

Unique clustered index on *name*

Unique nonclustered index on *usertype*

Referenced by System Procedures

sp_addtype, *sp_bindefault*, *sp_bindrule*, *sp_checknames*, *sp_checkreswords*,
sp_columns, *sp_datatype_info*, *sp_droptype*, *sp_dropuser*, *sp_help*, *sp_rename*,
sp_special_columns, *sp_sproc_columns*, *sp_unbindefault*, *sp_unbindrule*

sysusages

(*master database only*)

Description

sysusages contains one row for each **disk allocation piece** assigned to a database. Each database contains a specified number of database (logical) page numbers. Each disk piece includes the segments on the Adaptive Server distribution media, segments 0 and 1.

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*.

Columns

Name	Datatype	Description
<i>dbid</i>	<i>smallint</i>	Database ID
<i>segmap</i>	<i>int</i>	Bitmap of possible segment assignments
<i>lstart</i>	<i>int</i>	First database (logical) page number
<i>size</i>	<i>int</i>	Number of contiguous database (logical) pages
<i>vstart</i>	<i>int</i>	Starting virtual page number
<i>pad</i>	<i>smallint</i>	Unused
<i>unreservedpgs</i>	<i>int</i>	Free space not part of an allocated extent

Indexes

Unique clustered index on *dbid*, *lstart*

Unique nonclustered index on *vstart*

Referenced by System Procedures

`sp_addsegment`, `sp_addthreshold`, `sp_databases`, `sp_dropdevice`,
`sp_dropsegment`, `sp_extendsegment`, `sp_helpdb`, `sp_helplog`, `sp_helpsegment`,
`sp_logdevice`, `sp_modifythreshold`, `sp_spaceused`

sysusermessages

(all databases)

Description

sysusermessages contains one row for each user-defined message that can be returned by Adaptive Server.

Columns

Name	Datatype	Description
<i>error</i>	<i>int</i>	Unique error number. Must be 20,000 or higher.
<i>uid</i>	<i>smallint</i>	Server user ID (<i>suser_id</i>) of the message creator.
<i>description</i>	<i>varchar(255)</i>	User-defined message with optional placeholders for parameters.
<i>langid</i>	<i>smallint</i>	Language ID for this message; null for <i>us_english</i> .
<i>dlevel</i>	<i>smallint</i>	Stores the <i>with_log</i> bit, which is used to call the appropriate routine to log a message.

Indexes

Clustered index on *error*

Unique nonclustered index on *error*, *langid*

Referenced by System Procedures

sp_addmessage, *sp_bindmsg*, *sp_dropmessage*, *sp_getmessage*,
sp_helpconstraint

sysusers

(all databases)

Description

sysusers contains one row for each user allowed in the database, and one row for each group or role.

Columns

Name	Datatype	Description
<i>suid</i>	<i>smallint</i>	Server user ID, copied from <i>syslogins</i> .
<i>uid</i>	<i>smallint</i>	User ID, unique in this database, is used for granting and revoking permissions. User ID 1 is "dbo".
<i>gid</i>	<i>smallint</i>	Group ID to which this user belongs. If <i>uid = gid</i> , this entry defines a group. The group "public" has <i>suid = -2</i> ; all other groups have <i>suid = - gid</i> .
<i>name</i>	<i>sysname</i>	User or group name, unique in this database.
<i>environ</i>	<i>varchar(255)</i>	Reserved.

On the Adaptive Server distribution media, *master..sysusers* contains some initial users: "dbo", whose *suid* is 1 and whose *uid* is 1; "guest", whose *suid* is -1 and whose *uid* is 2; and "public", whose *suid* is -2 and whose *uid* is 0. In addition, both system-defined and user-defined roles (*sa_role*, *sso_role*, *role_name*) is listed in *sysusers*.

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

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.

Indexes

Unique clustered index on *suid*
 Unique nonclustered index on *name*
 Unique nonclustered index on *uid*

Referenced by System Procedures

sp_addalias, sp_addgroup, sp_adduser, sp_changedbowner, sp_changegroup,
sp_checknames, sp_checkreswords, sp_column_privileges, sp_depends,
sp_dropgroup, sp_droptype, sp_dropuser, sp_helpgroup, sp_helprotect,
sp_helpuser, sp_indsuspect, sp_stored_procedures, sp_table_privileges, sp_tables

Appendixes

A

Expressions, Identifiers, and Wildcard Characters

This appendix describes Transact-SQL expressions, valid identifiers, and wildcard characters.

Expressions

An expression is a combination of one or more constants, literals, functions, column identifiers and/or variables, separated by operators, that returns a single value. Expressions can be of several types, including **arithmetic**, **relational**, **logical** (or **Boolean**), and **character string**. In some Transact-SQL clauses, a subquery can be used in an expression. A case expression can be used in an expression.

Table A-1 lists the types of expressions that are used in Adaptive Server syntax statements.

Table A-1: Types of expressions used in syntax statements

Usage	Definition
<i>expression</i>	Can include constants, literals, functions, column identifiers, variables, or parameters
<i>logical expression</i>	An expression that returns TRUE, FALSE, or UNKNOWN
<i>constant expression</i>	An expression that always returns the same value, such as "5+3" or "ABCDE"
<i>float_expr</i>	Any floating-point expression or an expression that implicitly converts to a floating value
<i>integer_expr</i>	Any integer expression or an expression that implicitly converts to an integer value
<i>numeric_expr</i>	Any numeric expression that returns a single value
<i>char_expr</i>	Any expression that returns a single character-type value
<i>binary_expression</i>	An expression that returns a single <i>binary</i> or <i>varbinary</i> value

Arithmetic and Character Expressions

The general pattern for arithmetic and character expressions is:

```
{constant | column_name | function | (subquery)
 | (case_expression)}
  [{arithmetic_operator | bitwise_operator |
   string_operator | comparison_operator }
 {constant | column_name | function | (subquery)
 | case_expression}]...
```

Relational and Logical Expressions

A logical expression or relational expression returns TRUE, FALSE, or UNKNOWN. The general patterns are:

```
expression comparison_operator [any | all] expression
expression [not] in expression
[not]exists expression
expression [not] between expression and expression
expression [not] like "match_string"
  [escape "escape_character"]
not expression like "match_string"
  [escape "escape_character"]
expression is [not] null
not logical_expression
logical_expression {and | or} logical_expression
```

Operator Precedence

Operators have the following precedence levels, where 1 is the highest level and 6 is the lowest:

1. unary (single argument) - + ~
2. * / %
3. binary (two argument) + - & | ^
4. not
5. and
6. or

When all operators in an expression are at the same level, the order of execution is left to right. You can change the order of execution with parentheses—the most deeply nested expression is processed first.

Arithmetic Operators

Adaptive Server uses the following arithmetic operators:

Table A-2: Arithmetic operators

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo (Transact-SQL extension)

Addition, subtraction, division, and multiplication can be used on exact numeric, approximate numeric, and money type columns.

The modulo operator cannot be used on *smallmoney*, *money*, *float* or *real* columns. Modulo finds the integer remainder after a division involving two whole numbers. For example, $21 \% 11 = 10$ because 21 divided by 11 equals 1 with a remainder of 10.

When you perform arithmetic operations on mixed datatypes, for example *float* and *int*, Adaptive Server follows specific rules for determining the type of the result. See Chapter 7, “System and User-Defined Datatypes,” for more information.

Bitwise Operators

The bitwise operators are a Transact-SQL extension for use with integer type data. These operators convert each integer operand into its binary representation, then evaluate the operands column by column. A value of 1 corresponds to true; a value of 0 corresponds to false.

Table A-3 summarizes the results for operands of 0 and 1. If either operand is NULL, the bitwise operator returns NULL:

Table A-3: Truth tables for bitwise operations

& (and)	1	0
1	1	0
0	0	0

Table A-3: Truth tables for bitwise operations (continued)

(or)	1	0
1	1	1
0	1	0
<hr/>		
^ (exclusive or)	1	0
1	0	1
0	1	0
<hr/>		
~ (not)		
1	FALSE	
0	0	

The examples in Table A-4 use two *tinyint* arguments, A = 170 (10101010 in binary form) and B = 75 (01001011 in binary form).

Table A-4: Examples of bitwise operations

Operation	Binary Form	Result	Explanation
(A & B)	10101010 01001011 ----- 00001010	10	Result column equals 1 if both A and B are 1. Otherwise, result column equals 0.
(A B)	10101010 01001011 ----- 11101011	235	Result column equals 1 if either A or B, or both, is 1. Otherwise, result column equals 0
(A ^ B)	10101010 01001011 ----- 11100001	225	Result column equals 1 if either A or B, but not both, is 1
(~A)	10101010 ----- 01010101	85	All 1's are changed to 0's and all 0's to 1's

The String Concatenation Operator

The string operator + can be used to concatenate two or more character or binary expressions. For example:

```
1. select Name = (au_lname + ", " + au_fname)
   from authors
```

Displays author names under the column heading *Name* in last-name first-name order, with a comma after the last name; for example, "Bennett, Abraham."

```
2. select "abc" + " " + "def"
```

Returns the string "abc def". The empty string is interpreted as a single space in all *char*, *varchar*, *nchar*, *nvarchar*, and *text* concatenation, and in *varchar* insert and assignment statements.

When concatenating non-character, non-binary expressions, always use *convert*:

```
select "The date is " +
       convert(varchar(12), getdate())
```

A string concatenated with NULL evaluates to the value of the string. This is an exception to the SQL standard, which states that a string concatenated with a NULL should evaluate to NULL.

The Comparison Operators

Adaptive Server uses the comparison operators listed in Table A-5:

Table A-5: Comparison operators

Operator	Meaning
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to
!=	Not equal to (Transact-SQL extension)
!>	Not greater than (Transact-SQL extension)
!<	Not less than (Transact-SQL extension)

In comparing character data, < means closer to the beginning of the server's sort order and > means closer to the end of the sort order. Uppercase and lowercase letters are equal in a case-insensitive sort order. Use *sp_helpsort* to see the sort order for your Adaptive Server.

Trailing blanks are ignored for comparison purposes. So, for example, "Dirk" is the same as "Dirk ".

In comparing dates, < means earlier and > means later.

Put single or double quotes around all character and *datetime* data used with a comparison operator:

```
= "Bennet"  
> "May 22 1947"
```

Nonstandard Operators

The following operators are Transact-SQL extensions:

- Modulo operator: %
- Negative comparison operators: !>, !<, !=
- Bitwise operators: ~, ^, |, &
- Join operators: *= and =*

Using *any*, *all* and *in*

any is used with <, >, or = and a subquery. It returns results when any value retrieved in the subquery matches the value in the *where* or *having* clause of the outer statement. See Chapter 5, "Subqueries: Using Queries Within Other Queries," in the *Transact-SQL User's Guide* for more information.

all is used with < or > and a subquery. It returns results when all values retrieved in the subquery are less than (<) or greater than (>) the value in the *where* or *having* clause of the outer statement. See Chapter 5, "Subqueries: Using Queries Within Other Queries," in the *Transact-SQL User's Guide* for more information.

in returns results when any value returned by the second expression matches the value in the first expression. The second expression must be a subquery or a list of values enclosed in parentheses. *in* is equivalent to = *any*. See "where Clause" for details.

Negating and Testing

not negates the meaning of a keyword or logical expression.

Use *exists*, followed by a subquery, to test for the existence of a particular result.

Ranges

`between` is the range-start keyword; `and` is the range-end keyword. The range:

```
where column1 between x and y
```

is inclusive.

The range:

```
where column1 > x and column1 < y
```

is not inclusive.

Using Nulls in Expressions

Use `is null` or `is not null` in queries on columns defined to allow null values.

An expression with a bitwise or arithmetic operator evaluates to NULL if any of the operands are null. For example:

```
1 + column1
```

evaluates to NULL if `column1` is NULL.

Comparisons That Return TRUE

In general, the result of comparing null values is UNKNOWN, since it is not possible to determine whether NULL is equal (or not equal) to a given value or to another NULL. However, the following cases return TRUE when *expression* is any column, variable or literal, or combination of these, which evaluates as NULL:

- *expression* is null
- *expression* = null
- *expression* = @x, where @x is a variable or parameter containing NULL. This exception facilitates writing stored procedures with null default parameters.
- *expression* != n, where n is a literal that does not contain NULL, and *expression* evaluates to NULL.

The negative versions of these expressions return TRUE when the expression does not evaluate to NULL:

- *expression* is not null
- *expression* != null

- *expression* != @x

Note that the far right side of these exceptions is a literal null, or a variable or parameter containing NULL. If the far right side of the comparison is an expression (such as @nullvar + 1), the entire expression evaluates to NULL.

Following these rules, null column values do not join with other null column values. Comparing null column values to other null column values in a where clause always returns UNKNOWN for null values, regardless of the comparison operator, and the rows are not included in the results. For example, this query returns no result rows where column1 contains NULL in both tables (although it may return other rows):

```
select column1
from table1, table2
where table1.column1 = table2.column1
```

Difference Between FALSE and UNKNOWN

Although neither FALSE nor UNKNOWN returns values, there is an important logical difference between FALSE and UNKNOWN, because the opposite of false (“not false”) is true. For example, “1 = 2” evaluates to false and its opposite, “1 != 2”, evaluates to true. But “not unknown” is still unknown. If null values are included in a comparison, you cannot negate the expression to get the opposite set of rows or the opposite truth value.

Using “NULL” As a Character String

Only columns for which NULL was specified in the create table statement and into which you have explicitly entered NULL (no quotes), or into which no data has been entered, contain null values. Avoid entering the character string “NULL” (with quotes) as data for a character column. It can only lead to confusion. Use “N/A”, “none”, or a similar value instead. When you want to enter the value NULL explicitly, do **not** use single or double quotes.

NULLs Compared to the Empty String

The empty string (“ ” or ‘ ’) is always stored as a single space in variables and column data. This concatenation statement:

```
"abc" + " " + "def"
```

is equivalent to “abc def”, not to “abcdef”. The empty string is never evaluated as NULL.

Connecting Expressions

and connects two expressions and returns results when both are true. **or** connects two or more conditions and returns results when either of the conditions is true.

When more than one logical operator is used in a statement, **and** is evaluated before **or**. You can change the order of execution with parentheses.

Table A-6 shows the results of logical operations, including those that involve null values:

Table A-6: Truth tables for logical expressions

and	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
NULL	UNKNOWN	FALSE	UNKNOWN

or	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
NULL	TRUE	UNKNOWN	UNKNOWN

not	
TRUE	FALSE
FALSE	TRUE
NULL	UNKNOWN

The result UNKNOWN indicates that one or more of the expressions evaluates to NULL, and that the result of the operation cannot be determined to be either TRUE or FALSE. See “Using Nulls in Expressions” on page A-7 for more information.

Using Parentheses in Expressions

Parentheses can be used to group the elements in an expression. When “expression” is given as a variable in a syntax statement, a simple expression is assumed. “Logical expression” is specified when only a logical expression is acceptable.

Comparing Character Expressions

Character constant expressions are treated as *varchar*. If they are compared with non-*varchar* variables or column data, the datatype precedence rules are used in the comparison (that is, the datatype with lower precedence is converted to the datatype with higher precedence). If implicit datatype conversion is not supported, you must use the `convert` function.

Comparison of a *char* expression to a *varchar* expression follows the datatype precedence rule; the “lower” datatype is converted to the “higher” datatype. All *varchar* expressions are converted to *char* (that is, trailing blanks are appended) for the comparison.

Using the Empty String

The empty string (“”) or (‘’) is interpreted as a single blank in insert or assignment statements on *varchar* data. In concatenation of *varchar*, *char*, *nchar*, *nvarchar* data, the empty string is interpreted as a single space; for example:

```
"abc" + "" + "def"
```

is stored as “abc def”. The empty string is never evaluated as NULL.

Including Quotation Marks in Character Expressions

There are two ways to specify literal quotes within a *char* or *varchar* entry. The first method is to double the quotes. For example, if you

begin a character entry with a single quote and you want to include a single quote as part of the entry, use two single quotes:

```
'I don't understand.'
```

With double quotes:

```
"He said, "It's not really confusing.""
```

The second method is to enclose a quote in the opposite kind of quote mark. In other words, surround an entry containing a double quote with single quotes (or vice versa). Here are some examples:

```
'George said, "There must be a better way."
"Isn't there a better way?"
'George asked, "Isn't there a better way?"'
```

Using the Continuation Character

To continue a character string to the next line on your screen, enter a backslash (\) before going to the next line.

Identifiers

Identifiers are names for database objects such as databases, tables, views, columns, indexes, triggers, procedures, defaults, rules, and cursors.

Adaptive Server identifiers can be a maximum of 30 bytes in length, whether single-byte or multibyte characters are used. The first character of an identifier must be either an alphabetic character, as defined in the current character set, or the underscore (_) character.

► Note

Temporary table names, which begin with the pound sign (#), and local variable names, which begin with the at sign(@), are exceptions to this rule.

Subsequent characters can include letters, numbers, the symbols #, @, _, and currency symbols such as \$ (dollars), ¥ (yen), and £ (pound sterling). Identifiers cannot include special characters such as !, %, ^, &, *, and . or embedded spaces.

You cannot use a reserved word, such as a Transact-SQL command, as an identifier. For a complete list of reserved words, see Appendix B, "Reserved Words."

Tables Beginning with # (Temporary Tables)

Tables whose names begin with the pound sign (#) are temporary tables. You cannot create other types of objects whose names begin with the pound sign.

Adaptive Server performs special operations on temporary table names to maintain unique naming on a per-session basis. Long temporary table names are truncated to 13 characters (including the pound sign); short names are padded to 13 characters with underscores (_). A 17-digit numeric suffix that is unique for an Adaptive Server session is appended.

Case Sensitivity and Identifiers

Sensitivity to the case (upper or lower) of identifiers and data depends on the sort order installed on your Adaptive Server. Case sensitivity can be changed for single-byte character sets by reconfiguring Adaptive Server's sort order (see the *System Administration Guide* for more information). Case is significant in utility program options.

If Adaptive Server is installed with a case-insensitive sort order, you cannot create a table named *MYTABLE* if a table named *MyTable* or *mytable* already exists. Similarly, this command:

```
select * from MYTABLE
```

will return rows from *MYTABLE*, *MyTable*, or *mytable*, or any combination of uppercase and lowercase letters in the name.

Uniqueness of Object Names

Object names need not be unique in a database. However, column names and index names must be unique within a table, and other object names must be unique for each **owner** within a **database**. Database names must be unique on Adaptive Server.

Using Delimited Identifiers

Delimited identifiers are object names enclosed in double quotes. Using delimited identifiers allows you to avoid certain restrictions on object names. Table, view, and column names can be delimited by quotes; other object names cannot.

Delimited identifiers can be reserved words, can begin with non-alphabetic characters, and can include characters that would not otherwise be allowed. They cannot exceed 28 bytes.

◆ **WARNING!**

Delimited identifiers may not be recognized by all front-end applications and should not be used as parameters to system procedures.

Before creating or referencing a delimited identifier, you must execute:

```
set quoted_identifier on
```

Each time you use the delimited identifier in a statement, you must enclose it in double quotes. For example:

```
create table "lone"(coll char(3))
create table "include spaces" (coll int)

create table "grant"("add" int)
insert "grant"("add") values (3)
```

While the `quoted_identifier` option is turned on, do not use double quotes around character or date strings; use single quotes instead. Delimiting these strings with double quotes causes Adaptive Server to treat them as identifiers. For example, to insert a character string into *coll* of *ltable*, use:

```
insert "lone"(coll) values ('abc')
```

not:

```
insert "lone"(coll) values ("abc")
```

To insert a single quote into a column, use two consecutive single quotation marks. For example, to insert the characters "a'b" into *coll* use:

```
insert "lone"(coll) values('a''b')
```

Using Qualified Object Names

You can uniquely identify a table or column by adding other names that qualify it—the database name, owner's name, and (for a column) the table or view name. Each qualifier is separated from the next one by a period. For example:

```
database.owner.table_name.column_name
```

database.owner.view_name.column_name

The naming conventions are:

[[database.]owner.]table_name

[[database.]owner.]view_name

Using Delimited Identifiers Within an Object Name

If you use `set quoted_identifier on`, you can use double quotes around individual parts of a qualified object name. Use a separate pair of quotes for each qualifier that requires quotes. For example, use:

database.owner."table_name"."column_name"

rather than:

database.owner."table_name.column_name"

Omitting the Owner Name

You can omit the intermediate elements in a name and use dots to indicate their positions, as long as the system is given enough information to identify the object:

database..table_name

database..view_name

Referencing Your Own Objects in the Current Database

You need not use the database name or owner name to reference your own objects in the current database. The default value for *owner* is the current user, and the default value for *database* is the current database.

If you reference an object without qualifying it with the database name and owner name, Adaptive Server tries to find the object in the current database among the objects you own.

Referencing Objects Owned by the Database Owner

If you omit the owner name and you do not own an object by that name, Adaptive Server looks for objects of that name owned by the Database Owner. You must qualify objects owned by the Database Owner only if you own an object of the same name, but you want to use the object owned by the Database Owner. However, you must

qualify objects owned by other users with the user's name, whether or not you own objects of the same name.

Using Qualified Identifiers Consistently

When qualifying a column name and table name in the same statement, be sure to use the same qualifying expressions for each; they are evaluated as strings and must match; otherwise, an error is returned. The second of the following examples is incorrect because the syntax style for the column name does not match the syntax style used for the table name.

1.

```
select demo.mary.publishers.city
   from demo.mary.publishers

   city
   -----
   Boston
   Washington
   Berkeley
```
2.

```
select demo.mary.publishers.city
   from demo..publishers
```

The column prefix "demo.mary.publishers" does not match a table name or alias name used in the query.

Determining Whether an Identifier Is Valid

Use the system function `valid_name`, after changing character sets or before creating a table or view, to determine whether the object name is acceptable to Adaptive Server. Here is the syntax:

```
select valid_name("Object_name")
```

If *object_name* is not a valid identifier (for example, if it contains illegal characters or is more than 30 bytes long), Adaptive Server returns 0. If *object_name* is a valid identifier, Adaptive Server returns a nonzero number.

Renaming Database Objects

Rename user objects (including user-defined datatypes) with `sp_rename`.

◆ WARNING!

After you rename a table or column, be sure to redefine any procedures, triggers, and views that depend on the renamed object.

Using Multibyte Character Sets

In multibyte character sets, a wider range of characters is available for use in identifiers. For example, on a server with the Japanese language installed, the following types of characters may be used as the first character of an identifier: Zenkaku or Hankaku Katakana, Hiragana, Kanji, Romaji, Greek, Cyrillic, or ASCII.

Although Hankaku Katakana characters are legal in identifiers on Japanese systems, they are not recommended for use in heterogeneous systems. These characters cannot be converted between the EUC-JIS and Shift-JIS character sets.

The same is true for some 8-bit European characters. For example, the character “Œ,” the OE ligature, is part of the Macintosh character set (codepoint 0xCE). This character does not exist in the ISO 8859-1 (iso_1) character set. If “Œ” exists in data being converted from the Macintosh to the ISO 8859-1 character set, it causes a conversion error.

If an object identifier contains a character that cannot be converted, the client loses direct access to that object.

Pattern Matching with Wildcard Characters

Wildcard characters represent one or more characters, or a range of characters, in a *match_string*. A *match_string* is a character string containing the pattern to find in the expression. It can be any combination of constants, variables, and column names or a concatenated expression, such as:

```
like @variable + "%".
```

If the match string is a constant, it must always be enclosed in single or double quotes.

Use wildcard characters with the keyword `like` to find character and date strings that match a particular pattern. You cannot use `like` to search for seconds or milliseconds (see “Using Wildcard Characters with datetime Data” on page A-22).

Use wildcard characters in *where* and *having* clauses to find character or date/time information that is *like*—or *not like*—the match string:

```
{where | having} [not]
  expression [not] like match_string
  [escape "escape_character"]
```

expression can be any combination of column names, constants, or functions with a character value.

Wildcard characters used without *like* have no special meaning. For example, this query finds any phone numbers that start with the four characters "415%":

```
select phone
from authors
where phone = "415%"
```

Using *not like*

Use *not like* to find strings that do not match a particular pattern. These two queries are equivalent: they find all the phone numbers in the *authors* table that do not begin with the 415 area code.

```
select phone
from authors
where phone not like "415%"
```

```
select phone
from authors
where not phone like "415%"
```

For example, this query finds the system tables in a database whose names begin with "sys":

```
select name
from sysobjects
where name like "sys%"
```

To see all the objects that are **not** system tables, use

```
not like "sys%"
```

If you have a total of 32 objects and *like* finds 13 names that match the pattern, *not like* will find the 19 objects that do not match the pattern.

not like and the negative wildcard character [^] may give different results (see "The Caret (^) Wildcard Character" on page A-20). You cannot always duplicate *not like* patterns with *like* and ^. This is because *not like* finds the items that do not match the entire *like*

pattern, but like with negative wildcard characters is evaluated one character at a time.

A pattern such as like "[^s][^y][^s]%" may not produce the same results. Instead of 19, you might get only 14, with all the names that begin with "s" or have "y" as the second letter or have "s" as the third letter eliminated from the results, as well as the system table names. This is because match strings with negative wildcard characters are evaluated in steps, one character at a time. If the match fails at any point in the evaluation, it is eliminated.

Case and Accent Insensitivity

If your Adaptive Server uses a case-insensitive sort order, case is ignored when comparing *expression* and *match_string*. For example, this clause:

```
where col_name like "Sm%"
```

would return "Smith," "smith," and "SMITH" on a case-insensitive Adaptive Server.

If your Adaptive Server is also accent-insensitive, it treats all accented characters as equal to each other and to their unaccented counterparts, both uppercase and lowercase. The `sp_helpsort` system procedure displays the characters that are treated as equivalent, displaying an "=" between them.

Using Wildcard Characters

You can use the match string with a number of wildcard characters, which are discussed in detail in the following sections. Table A-7 summarizes the wildcard characters:

Table A-7: Wildcard characters used with like

Symbol	Meaning
%	Any string of 0 or more characters
_	Any single character
[]	Any single character within the specified range ([a-f]) or set ([abcdef])
[^]	Any single character not within the specified range ([^a-f]) or set ([^abcdef])

Enclose the wildcard character and the match string in single or double quotes (like "[dD]eFr_nce").

The Percent Sign (%) Wildcard Character

Use the % wildcard character to represent any string of zero or more characters. For example, to find all the phone numbers in the *authors* table that begin with the 415 area code:

```
select phone
from authors
where phone like "415%"
```

To find names that have the characters “en” in them (Bennet, Green, McBaden):

```
select au_lname
from authors
where au_lname like "%en%"
```

Trailing blanks following “%” in a like clause are truncated to a single trailing blank. For example, “%” followed by two spaces matches “X ” (one space); “X ” (two spaces); “X ” (three spaces), or any number of trailing spaces.

The Underscore (_) Wildcard Character

Use the _ wildcard character to represent any single character. For example, to find all six-letter names that end with “heryl” (for example, Cheryl):

```
select au_fname
from authors
where au_fname like "_heryl"
```

Bracketed ([]) Characters

Use brackets to enclose a range of characters, such as [a-f], or a set of characters such as [a2Br]. When ranges are used, all values in the sort order between (and including) *rangespec1* and *rangespec2* are returned. For example, “[0-z]” matches 0-9, A-Z and a-z (and several punctuation characters) in 7-bit ASCII.

To find names ending with “inger” and beginning with any single character between M and Z:

```
select au_lname
from authors
where au_lname like "[M-Z]inger"
```

To find both “DeFrance” and “deFrance”:

```
select au_lname
from authors
where au_lname like "[dD]eFrance"
```

The Caret (^) Wildcard Character

The caret is the negative wildcard character. Use it to find strings that do not match a particular pattern. For example, “[^a-f]” finds strings that are not in the range a-f and “[^a2bR]” finds strings that are not “a,” “2,” “b,” or “R.”

To find names beginning with “M” where the second letter is not “c”:

```
select au_lname
from authors
where au_lname like "M[^c]%"
```

When ranges are used, all values in the sort order between (and including) *rangespec1* and *rangespec2* are returned. For example, “[0-z]” matches 0-9, A-Z, a-z, and several punctuation characters in 7-bit ASCII.

Using Multibyte Wildcard Characters

If the multibyte character set configured on your Adaptive Server defines equivalent double-byte characters for the wildcard characters `_`, `%`, `-`, `[`, `]`, and `^`, you can substitute the equivalent character in the match string. The underscore equivalent represents either a single- or double-byte character in the match string.

Using Wildcard Characters As Literal Characters

To search for the occurrence of `%`, `_`, `[`, `]`, or `^` within a string, you must use an escape character. When a wildcard character is used in conjunction with an escape character, Adaptive Server interprets the wildcard character literally, rather than using it to represent other characters.

Adaptive Server provides two types of escape characters:

- Square brackets (a Transact-SQL extension)
- Any single character that immediately follows an escape clause (compliant with the SQL standards)

Using Square Brackets As Escape Characters

Use square brackets as escape characters for the percent sign, the underscore, and the left bracket. The right bracket does not need an escape character; use it by itself. If you use the dash as a literal character, it must be the first character inside a set of square brackets.

Table A-8 shows some examples of square brackets as escape characters:

Table A-8: Using square brackets to search for wildcard characters

<i>like</i> Predicate	Meaning
<i>like</i> "5%"	5 followed by any string of 0 or more characters
<i>like</i> "5[%]"	5%
<i>like</i> "_n"	an, in, on (and so on)
<i>like</i> "[_n]"	_n
<i>like</i> "[a-cdf]"	a, b, c, d, or f
<i>like</i> "[-acdf]"	-, a, c, d, or f
<i>like</i> "[["	[
<i>like</i> "]"]
<i>like</i> "[[]ab]"	[]ab

Using the *escape* Clause

Use the *escape* clause to specify an escape character. Any single character in the server's default character set can be used as an escape character. If you try to use more than one character as an escape character, Adaptive Server generates an exception.

Do not use existing wildcard characters as escape characters because:

- If you specify the underscore (`_`) or percent sign (`%`) as an escape character, it loses its special meaning within that *like* predicate and acts only as an escape character.
- If you specify the left or right bracket (`[` or `]`) as an escape character, the Transact-SQL meaning of the bracket is disabled within that *like* predicate.
- If you specify the hyphen or caret (`-` or `^`) as an escape character, it loses its special meaning and acts only as an escape character.

An escape character retains its special meaning within square brackets, unlike wildcard characters such as the underscore, the percent sign, and the open bracket.

The escape character is valid only within its *like* predicate and has no effect on other *like* predicates contained in the same statement. The

only characters that are valid following an escape character are the wildcard characters (`_`, `%`, `[`, `]`, or `[^]`), and the escape character itself. The escape character affects only the character following it, and subsequent characters are not affected by it.

If the pattern contains two literal occurrences of the character that happens to be the escape character, the string must contain four consecutive escape characters. If the escape character does not divide the pattern into pieces of one or two characters, Adaptive Server returns an error message.

Following are examples of `like` predicates with escape clauses:

Table A-9: Using the escape clause

like Predicate	Meaning
<code>like "5@%" escape "@"</code>	5%
<code>like "**_n" escape ""</code>	<code>_n</code>
<code>like "%80@%" escape "@"</code>	String containing 80%
<code>like "**_sql*" escape ""</code>	String containing <code>_sql*</code>
<code>like "%####_#%" escape "#"</code>	String containing <code>##_%</code>

To enforce standard behavior and disable the special meaning of the square brackets, use `set fipsflagger on`.

Using Wildcard Characters with *datetime* Data

When you use `like` with *datetime* values, Adaptive Server converts the dates to the standard *datetime* format, and then to *varchar*. Since the standard storage format does not include seconds or milliseconds, you cannot search for seconds or milliseconds with `like` and a pattern.

It is a good idea to use `like` when you search for *datetime* values, since *datetime* entries may contain a variety of date parts. For example, if you insert the value "9:20" and the current date into a column named *arrival_time*, the clause:

```
where arrival_time = '9:20'
```

would not find the value, because Adaptive Server converts the entry into "Jan 1 1900 9:20AM." However, the following clause would find this value:

```
where arrival_time like '%9:20%'
```

B

Reserved Words

Keywords, also known as reserved words, are words that have special meanings. Transact-SQL and SQL92 keywords are listed in this appendix.

Transact-SQL Keywords

The words in Table B-1 are reserved by Adaptive Server as keywords (part of SQL command syntax) and cannot be used as names of database objects such as databases, tables, rules, and defaults. They can be used as names of local variables and as stored procedure parameter names. You can use the system procedure `sp_checkreswords` to find the names of existing objects that are reserved words.

Table B-1: Transact-SQL keywords

activation	char_convert	dbcc	exclusive
add	check	deallocate	exec
all	checkpoint	declare	execute
alter	close	default	exists
and	clustered	delete	exit
any	commit	desc	external
arith_overflow	compute	disk	fetch
as	confirm	distinct	fillfactor
asc	connect	double	for
at	constraint	dummy	foreign
authorization	consumers	dump	from
avg	continue	else	goto
begin	controlrow	end	grant
between	convert	endtran	group
break	count	errlvl	having
browse	create	errordata	holdlock
bulk	current	errorexit	identity_insert
by	cursor	escape	identity_start
cascade	database	except	if

Table B-1: Transact-SQL keywords (continued)

in	off	read	temp
index	offsets	readtext	temporary
insert	on	reconfigure	textsize
intersect	once	references	to
into	online	replace	tran
is	only	return	transaction
isolation	open	revoke	trigger
key	option	role	truncate
kill	or	rollback	tsequal
level	order	rowcount	union
like	over	rows	unique
lineno	partition	rule	unpartition
load	passwd	save	update
max	perm	schema	use
max_rows_per_page	permanent	select	user
membership	plan	session	user_option
min	precision	set	using
mirror	prepare	setuser	values
mirrorexit	primary	shared	varying
national	print	shutdown	view
noholdlock	privileges	some	waitfor
nonclustered	proc	statistics	where
not	procedure	stripe	while
null	processexit	sum	with
lineno	proxy	syb_identity	work
numeric_transaction	public	syb_restree	writetext
of	raiserror	table	

SQL92 Keywords

Adaptive Server includes entry-level SQL92 features. Full SQL92 implementation includes the words listed in the following tables as command syntax. Since upgrading identifiers can be a complex process, we are providing this list for your convenience. The publication of this information does not commit Sybase to providing all of these SQL92 features in subsequent releases. In addition, subsequent releases may include keywords not included in this list. The words in Table B-2 are SQL92 keywords that are not reserved words in Transact-SQL.

Table B-2: SQL92 keywords

absolute	corresponding	float
action	cross	found
allocate	current_date	full
are	current_time	get
assertion	current_timestamp	global
bit	current_user	go
bit_length	date	hour
both	day	immediate
cascaded	dec	indicator
case	decimal	initially
cast	deferrable	inner
catalog	deferred	input
char	describe	insensitive
char_length	descriptor	int
character	diagnostics	integer
character_length	disconnect	interval
coalesce	domain	join
collate	end-exec	language
collation	exception	last
column	extract	leading
connection	false	left
constraints	first	local

Table B-2: SQL92 keywords (continued)

lower	prior	timestamp
match	real	timezone_hour
minute	relative	timezone_minute
module	restrict	trailing
month	right	translate
names	scroll	translation
natural	second	trim
nchar	section	true
next	session_user	unknown
no	size	upper
nullif	smallint	usage
numeric	space	value
octet_length	sql	varchar
outer	sqlcode	when
output	sqlerror	whenever
overlaps	sqlstate	write
pad	substring	year
partial	system_user	zone
position	then	
preserve	time	

Potential SQL92 Reserved Words

If you are using the ISO/IEC 9075:1989 standard, also avoid using the words listed in Table B-3, as these words may become SQL92 reserved words in the future.

Table B-3: Potential SQL92 reserved words

after	loop	returns
alias	modify	routine
async	new	row
before	none	savepoint
boolean	object	search
breadth	oid	sensitive
call	old	sequence
completion	operation	signal
cycle	operators	similar
data	others	sqlexception
depth	parameters	structure
dictionary	pendant	test
each	preorder	there
elseif	private	type
equals	protected	under
general	recursive	variable
ignore	ref	virtual
leave	referencing	visible
less	resignal	wait
limit	return	without

C

SQLSTATE Codes and Messages

This appendix describes Adaptive Server's SQLSTATE status codes and their associated messages. SQLSTATE codes are required for entry level SQL92 compliance. They provide diagnostic information about two types of conditions:

- **Warnings** – conditions that require user notification but are not serious enough to prevent a SQL statement from executing successfully
- **Exceptions** – conditions that prevent a SQL statement from having any effect on the database

Each SQLSTATE code consists of a 2-character class followed by a 3-character subclass. The class specifies general information about error type; the subclass specifies more specific information.

SQLSTATE codes are stored in the *sysmessages* system table, along with the messages that display when these conditions are detected. Not all Adaptive Server error conditions are associated with a SQLSTATE code—only those mandated by SQL92. In some cases, multiple Adaptive Server error conditions are associated with a single SQLSTATE value.

Warnings

Adaptive Server currently detects only one SQLSTATE warning condition, which is described in Table C-1:

Table C-1: SQLSTATE warnings

Message	Value	Description
Warning - null value eliminated in set function.	01003	Occurs when you use an aggregate function (avg, max, min, sum, or count) on an expression with a null value.

Exceptions

Adaptive Server detects the following types of exceptions:

- Cardinality violations
- Data exceptions

- Integrity constraint violations
- Invalid cursor states
- Syntax errors and access rule violations
- Transaction rollbacks
- with check option violations

Exception conditions are described in Table C-2 through Table C-8. Each class of exceptions appears in its own table. Within each table, conditions are sorted alphabetically by message text.

Cardinality Violations

Cardinality violations occur when a query that should return only a single row returns more than one row to an Embedded SQL™ application.

Table C-2: Cardinality violations

Message	Value	Description
Subquery returned more than 1 value. This is illegal when the subquery follows =, !=, <, <=, >, >=, or when the subquery is used as an expression.	21000	Occurs when: <ul style="list-style-type: none"> • A scalar subquery or a row subquery returns more than one row. • A select into parameter_list query in Embedded SQL returns more than one row.

Data Exceptions

Data exceptions occur when an entry:

- Is too long for its datatype,
- Contains an illegal escape sequence, or
- Contains other format errors.

Table C-3: Data exceptions

Message	Value	Description
Arithmetic overflow occurred.	22003	Occurs when: <ul style="list-style-type: none"> An exact numeric type would lose precision or scale as a result of an arithmetic operation or <code>sum</code> function. An approximate numeric type would lose precision or scale as a result of truncation, rounding, or a <code>sum</code> function.
Data exception - string data right truncated.	22001	Occurs when a <code>char</code> or <code>varchar</code> column is too short for the data being inserted or updated and non-blank characters must be truncated.
Divide by zero occurred.	22012	Occurs when a numeric expression is being evaluated and the value of the divisor is zero.
Illegal escape character found. There are fewer bytes than necessary to form a valid character.	22019	Occurs when you are searching for strings that match a given pattern if the escape sequence does not consist of a single character.
Invalid pattern string. The character following the escape character must be percent sign, underscore, left square bracket, right square bracket, or the escape character.	22025	Occurs when you are searching for strings that match a particular pattern when: <ul style="list-style-type: none"> The escape character is not immediately followed by a percent sign, an underscore, or the escape character itself, or The escape character partitions the pattern into substrings whose lengths are other than 1 or 2 characters.

Integrity Constraint Violations

Integrity constraint violations occur when an insert, update, or delete statement violates a primary key, foreign key, check, or unique constraint or a unique index.

Table C-4: Integrity constraint violations

Message	Value	Description
Attempt to insert duplicate key row in object <code>object_name</code> with unique index <code>index_name</code>	23000	Occurs when a duplicate row is inserted into a table that has a unique constraint or index.

Table C-4: Integrity constraint violations (continued)

Message	Value	Description
Check constraint violation occurred, dbname = <i>database_name</i> , table name = <i>table_name</i> , constraint name = <i>constraint_name</i>	23000	Occurs when an update or delete would violate a check constraint on a column.
Dependent foreign key constraint violation in a referential integrity constraint. dbname = <i>database_name</i> , table name = <i>table_name</i> , constraint name = <i>constraint_name</i>	23000	Occurs when an update or delete on a primary key table would violate a foreign key constraint.
Foreign key constraint violation occurred, dbname = <i>database_name</i> , table name = <i>table_name</i> , constraint name = <i>constraint_name</i>	23000	Occurs when an insert or update on a foreign key table is performed without a matching value in the primary key table.

Invalid Cursor States

Invalid cursor states occur when:

- A **fetch** uses a cursor that is not currently open, or
- An **update where current of** or **delete where current of** affects a cursor row that has been modified or deleted, or
- An **update where current of** or **delete where current of** affects a cursor row that not been fetched.

Table C-5: Invalid cursor states

Message	Value	Description
Attempt to use cursor <i>cursor_name</i> which is not open. Use the system stored procedure <i>sp_cursorinfo</i> for more information.	24000	Occurs when an attempt is made to fetch from a cursor that has never been opened or that was closed by a commit statement or an implicit or explicit rollback . Reopen the cursor and repeat the fetch .
Cursor <i>cursor_name</i> was closed implicitly because the current cursor position was deleted due to an update or a delete. The cursor scan position could not be recovered. This happens for cursors which reference more than one table.	24000	Occurs when the join column of a multitable cursor has been deleted or changed. Issue another fetch to reposition the cursor.

Table C-5: Invalid cursor states (continued)

Message	Value	Description
The cursor <i>cursor_name</i> had its current scan position deleted because of a DELETE/UPDATE WHERE CURRENT OF or a regular searched DELETE/UPDATE. You must do a new FETCH before doing an UPDATE or DELETE WHERE CURRENT OF.	24000	Occurs when a user issues an update/delete where current of whose current cursor position has been deleted or changed. Issue another fetch before retrying the update/delete where current of .
The UPDATE/DELETE WHERE CURRENT OF failed for the cursor <i>cursor_name</i> because it is not positioned on a row.	24000	Occurs when a user issues an update/delete where current of on a cursor that: <ul style="list-style-type: none"> • Has not yet fetched a row • Has fetched one or more rows after reaching the end of the result set

Syntax Errors and Access Rule Violations

Syntax errors are generated by SQL statements that contain unterminated comments, implicit datatype conversions not supported by Adaptive Server or other incorrect syntax.

Access rule violations are generated when a user tries to access an object that does not exist or one for which he or she does not have the correct permissions.

Table C-6: Syntax errors and access rule violations

Message	Value	Description
<i>command</i> permission denied on object <i>object_name</i> , database <i>database_name</i> , owner <i>owner_name</i> .	42000	Occurs when a user tries to access an object for which he or she does not have the proper permissions.
Implicit conversion from datatype ' <i>datatype</i> ' to ' <i>datatype</i> ' is not allowed. Use the CONVERT function to run this query.	42000	Occurs when the user attempts to convert one datatype to another but Adaptive Server cannot do the conversion implicitly.
Incorrect syntax near <i>object_name</i> .	42000	Occurs when incorrect SQL syntax is found near the object specified.
Insert error: column name or number of supplied values does not match table definition.	42000	Occurs during inserts when an invalid column name is used or when an incorrect number of values is inserted.
Missing end comment mark <i>*/</i> .	42000	Occurs when a comment that begins with the <i>/*</i> opening delimiter does not also have the <i>*/</i> closing delimiter.

Table C-6: Syntax errors and access rule violations (continued)

Message	Value	Description
<i>object_name</i> not found. Specify owner.objectname or use sp_help to check whether the object exists (sp_help may produce lots of output).	42000	Occurs when a user tries to reference an object that he or she does not own. When referencing an object owned by another user, be sure to qualify the object name with the name of its owner.
The size (<i>size</i>) given to the <i>object_name</i> exceeds the maximum. The largest size allowed is <i>size</i> .	42000	Occurs when: <ul style="list-style-type: none"> The total size of all the columns in a table definition exceeds the maximum allowed row size. The size of a single column or parameter exceeds the maximum allowed for its datatype.

Transaction Rollbacks

Transaction rollbacks occur when the transaction isolation level is set to 3, but Adaptive Server cannot guarantee that concurrent transactions can be serialized. This type of exception generally results from system problems such as disk crashes and offline disks.

Table C-7: Transaction rollbacks

Message	Value	Description
Your server command (process id # <i>process_id</i>) was deadlocked with another process and has been chosen as deadlock victim. Re-run your command.	40001	Occurs when Adaptive Server detects that it cannot guarantee that two or more concurrent transactions can be serialized.

with check option Violation

This class of exception occurs when data being inserted or updated through a view would not be visible through the view.

Table C-8: with check option violation

Message	Value	Description
The attempted insert or update failed because the target view was either created WITH CHECK OPTION or spans another view created WITH CHECK OPTION. At least one resultant row from the command would not qualify under the CHECK OPTION constraint.	44000	Occurs when a view, or any view on which it depends, was created with a with check option clause.

Index

Index

This index pertains to all three volumes of the *Adaptive Server Reference Manual*. It is divided into three sections:

- Symbols
Indexes entries that begin with symbols.
- Numerics
Indexes entries that begin numerically.
- Subjects
Indexes subjects alphabetically.

Page numbers in **bold** are primary references.

Symbols

- & (ampersand)
 - “and” bitwise operator V3 A-3
- * (asterisk)
 - multiplication operator V3 A-3
 - for overlength numbers V1 2-144
 - select and V1 1-133
- @ (at sign)
 - local variable name V1 1-146 to V1 1-147
 - procedure parameters and V1 1-224, V2 3-12
 - rule arguments and V1 1-91
- \ (backslash)
 - character string continuation with V1 1-409, V3 A-11
- ^ (caret)
 - “exclusive or” bitwise operator V3 A-4
 - wildcard character V3 A-18, V3 A-20
- : (colon)
 - preceding milliseconds V1 2-20, V1 2-66
- , (comma)
 - in default print format for money values V3 7-15
 - not allowed in money values V3 7-15
 - in SQL statements V1 xviii
 - in user-defined datatypes V2 3-79
- { } (curly braces)
 - in SQL statements V1 xviii
- \$ (dollar sign)
 - in identifiers V3 A-11
 - in money datatypes V3 7-15
- .. (dots) in database object names V2 3-46, V3 A-14
- ... (ellipsis) in SQL statements V1 xx
- = (equals sign)
 - for assigning variables V1 1-340
 - comparison operator V3 A-5
 - for renaming column headings V1 1-339
- ! (exclamation point)
 - error message placeholder V1 1-304
- > (greater than)
 - comparison operator V3 A-5
- >= (greater than or equal to) comparison operator V3 A-5
- < (less than)
 - comparison operator V3 A-5

- <= (less than or equal to) comparison operator V3 A-5
 - (minus sign)
 - arithmetic operator V3 A-3
 - in integer data V3 7-9
 - for negative monetary values V3 7-15
 - != (not equal to) comparison operator V3 A-5
 - <> (not equal to) comparison operator V3 A-5
 - !> (not greater than) comparison operator V3 A-5
 - !< (not less than) comparison operator V3 A-5
 - () (parentheses)
 - in expressions V3 A-10
 - in SQL statements V1 xviii
 - in user-defined datatypes V2 3-79
 - % (percent sign)
 - arithmetic operator (modulo) V3 A-3
 - error message literal V1 1-306
 - error message placeholder V1 1-304
 - wildcard character V3 A-18
 - . (period)
 - preceding milliseconds V1 2-20, V1 2-66
 - separator for qualifier names V3 A-13
 - | (pipe)
 - “or” bitwise operator V3 A-4
 - + (plus)
 - arithmetic operator V3 A-3
 - in integer data V3 7-9
 - null values and V3 A-5
 - string concatenation operator V3 A-5
 - # (pound sign), temporary table identifier prefix V1 1-98
 - £ (pound sterling sign)
 - in identifiers V3 A-11
 - in money datatypes V3 7-15
 - ?? (question marks)
 - for partial characters V1 1-316
 - " " (quotation marks)
 - comparison operators and V3 A-6
 - enclosing constant values V1 2-23
 - enclosing *datetime* values V3 7-19
 - enclosing empty strings V3 A-9, V3 A-10
 - enclosing parameter values V2 3-11, V2 4-2
 - enclosing reserved words V2 3-155
 - in expressions V3 A-10
 - literal specification of V3 A-10, V1 1-409
 - single, and `quoted_identifier` V2 3-162
 - / (slash)
 - arithmetic operator (division) V3 A-3
 - [] (square brackets)
 - character set wildcard V3 A-18, V3 A-19
 - in SQL statements V1 xviii
 - [^] (square brackets and caret) character set wildcard V3 A-18
 - ~ (tilde)
 - “not” bitwise operator V3 A-4
 - _ (underscore)
 - character string wildcard V3 A-18, V3 A-19
 - object identifier prefix V1 2-169, V3 A-11
 - in temporary table names V3 A-12
 - ¥ (yen sign)
 - in identifiers V3 A-11
 - in money datatypes V3 7-15
- ## Numerics
- 0 isolation level (read uncommitted) V1 1-345
 - 0 return status V2 3-10, V2 4-2
 - stored procedures V1 1-84
 - “0x” V1 2-17, V3 7-28, V3 7-29
 - in defaults V1 1-59
 - in rules V1 1-91
 - `writetext` command and *image* data V1 1-414
 - 1 isolation level (read committed) V1 1-345
 - 21st century numbers V3 7-19

- 3 isolation level (serializable) V1 1-346
 - 7-bit ASCII characters, checking with `sp_checknames` V2 3-148
 - 7-bit terminal, `sp_helpsort` output V2 3-355
 - 8-bit terminal, `sp_helpsort` output V2 3-355
- A**
- Abbreviations
 - chars for characters, `patindex` V1 2-103
 - chars for characters, `readtext` V1 1-314
 - date parts V1 2-20, V1 2-66
 - exec for execute V1 1-223
 - out for output V1 1-78, V1 1-224
 - tran for transaction, `rollback` command V1 1-330
 - abort tran on log full database option V2 3-193
 - abs absolute value mathematical function V1 2-26
 - Accent sensitivity
 - compute and V1 1-48
 - dictionary sort order and V1 1-299
 - group by and V1 1-257
 - wildcard characters and V3 A-18
 - Access
 - ANSI restrictions on tapes V1 1-220
 - Access, object. *See* Permissions; Users
 - Accounting, chargeback
 - `sp_clearstats` V2 3-172
 - `sp_reportstats` V2 3-462 to V2 3-463
 - Accounts. *See* Logins
 - acos mathematical function V1 2-27
 - Actions
 - modifying for resource limits V2 3-393
 - resource limit information on V2 3-343
 - specifying for resource limits V2 3-57
 - activation keyword, `alter role` V1 1-10
 - Adding
 - aliases V2 3-16 to V2 3-50
 - columns to a table V1 1-13
 - date strings V2 3-35 to V2 3-39
 - dump devices V2 3-85 to V2 3-87
 - engine groups V2 3-24
 - engines to a group V2 3-24
 - execution classes V2 3-26
 - foreign keys V2 3-288 to V2 3-290
 - group to a database V2 3-33 to V2 3-34
 - an interval to a date V1 2-60
 - limits V2 3-55
 - logins to Server V2 3-40 to V2 3-42
 - messages to `sysusermessages` V1 1-306, V2 3-43 to V2 3-45
 - mirror device V1 1-164 to V1 1-167
 - mutually exclusive user-defined roles V1 1-10
 - named time ranges V2 3-75
 - objects to `tempdb` V1 1-107
 - passwords to roles V1 1-10
 - remote logins V2 3-51 to V2 3-54
 - resource limits V2 3-55
 - roles V1 1-89
 - rows to a table or view V1 1-261 to V1 1-269
 - segments V2 3-62 to V2 3-64
 - servers V2 3-65 to V2 3-68
 - space to a database V1 1-6 to V1 1-9
 - thresholds V2 3-69 to V2 3-74
 - time ranges V2 3-75
 - `timestamp` column V1 2-159
 - user-defined datatypes V2 3-79 to V2 3-84, V3 7-39
 - user-defined roles V1 1-89
 - users to a database V2 3-88 to V2 3-90
 - users to a group V2 3-88 to V2 3-90, V2 3-146 to V2 3-147
 - Addition operator (+) V3 A-3
 - add keyword
 - `alter role` V1 1-10
 - `alter table` V1 1-14
 - Aggregate-free expression, grouping by V1 1-246
 - Aggregate functions V1 2-6 to V1 2-11

- See also* Row aggregates; *individual function names*
- cursors and V1 2-9
 - difference from row aggregates V1 2-10
 - group by clause and V1 1-245, V1 1-248, V1 2-6, V1 2-8
 - having clause and V1 1-246, V1 1-248, V1 2-6
 - not used on virtual tables V3 8-3
 - scalar aggregates V1 1-248, V1 2-6
 - vector aggregates V1 2-7
 - vector aggregates, **group by** and V1 1-248
- Aliases
- server V2 3-65
 - table correlation names V1 1-341
- Aliases, column
- compute clauses allowing V1 1-45
 - prohibited after **group by** V1 1-246, V1 1-247
- Aliases, language
- assigning V2 3-472 to V2 3-473
 - defining V2 3-35 to V2 3-39
 - syslanguages* table V3 8-50
- Aliases, user
- See also* Logins; Users
- assigning V2 3-16 to V2 3-17
 - assigning different names compared to V2 3-88
 - database ownership transfer and V2 3-143
 - dropping V2 3-223 to V2 3-224, V2 3-270
 - help on V2 3-362
 - sysalternates* table V2 3-17, V2 3-223, V3 8-5
- all keyword
- grant V1 1-232, V1 1-241
 - group by V1 1-245
 - negated by having clause V1 1-246
 - revoke V1 1-322
 - select V1 1-339, V1 1-348
 - subqueries including V3 A-6
- union V1 1-381
 - where V1 1-407
- Allocation map. *See* Object Allocation Map (OAM)
- Allocation units
- sysusages* table V3 8-87
- allow_dup_row option, create index V1 1-68
- allow nested triggers configuration
- parameter V1 1-126
- allow nulls by default database option V2 3-194
- allow updates to system tables configuration
- parameter V3 8-4
- alter database command V1 1-6 to V1 1-9
- for load option V1 1-7
 - sp_dbremap and V2 3-200
 - with override option V1 1-7
- Alternate identity. *See* Alias, user
- Alternate languages. *See* Languages, alternate
- alter role command V1 1-10 to V1 1-12
- alter table command V1 1-13 to V1 1-24
- adding *timestamp* column V1 2-159
- And (&)
- bitwise operator V3 A-3
- and keyword
- in expressions V3 A-9
 - number allowed in search conditions V1 1-409
 - range-end V1 1-406, V3 A-7
 - in search conditions V1 1-407
- Angles, mathematical functions for V1 2-27
- ansinull option, set V1 1-355
- ansi_permissions option, set V1 1-355
- ANSI tape label
- dumpvolume option to dump database V1 1-197
 - dumpvolume option to dump transaction V1 1-211
 - listonly option to load database V1 1-275
 - listonly option to load transaction V1 1-284
- ANYENGINE engine group V2 3-24

- any keyword
 - in expressions V3 A-6
 - where clause V1 1-407
- Applications
 - applying resource limits to V2 3-55
 - dropping resource limits from V2 3-254
 - modifying resource limits for V2 3-392
 - resource limit information on V2 3-342
- Approximate numeric datatypes **V3 7-13**
- Arguments
 - See also* Logical expressions
 - numbered placeholders for, in print command V1 1-304, V1 1-305
 - in user-defined error messages V1 1-309
 - where clause, number allowed V1 1-410
- arithabort option, set
 - arith_overflow and V1 1-356, V1 2-16, V3 7-8
 - mathematical functions and arith_overflow V1 2-21
 - mathematical functions and numeric_truncation V1 2-17, V1 2-21
- arithignore option, set
 - arith_overflow and V1 1-356, V1 2-16
 - mathematical functions and arith_overflow V1 2-22
- Arithmetic errors V1 2-21
- Arithmetic expressions V3 A-2
- Arithmetic operations
 - approximate numeric datatypes and V3 7-13
 - exact numeric datatypes and V3 7-9
 - money datatypes and V3 7-15
- Arithmetic operators
 - in expressions V3 A-3
- Ascending order, asc keyword V1 1-296, V1 1-344
- ASCII characters
 - ascii string function and V1 2-28
 - checking for with sp_checknames V2 3-148
 - ascii string function V1 2-28
 - asin mathematical function **V1 2-30**
 - as keyword for renaming column headings V1 1-339
- Asterisk (*)
 - multiplication operator V3 A-3
 - overlength numbers V1 2-144
 - select and V1 1-133
- Asynchronous prefetch
 - configuring limits V2 3-429
- atan mathematical function **V1 2-31**
- @@char_convert global variable V1 1-372
- @@connections global variable
 - sp_monitor and V2 3-407
- @@cpu_busy global variable
 - sp_monitor and V2 3-406
- @@error global variable
 - select into and V1 1-350
 - stored procedures and V1 1-82
 - user-defined error messages and V1 1-306, V1 1-312
- @@identity global variable V1 1-266
- @@idle global variable
 - sp_monitor and V2 3-406
- @@io_busy global variable
 - sp_monitor and V2 3-406
- @@isolation global variable V1 1-372
- @@langid global variable V1 1-308
- @@ncharsize global variable
 - sp_addtype and V2 3-81
- @@nestlevel global variable V1 1-227
 - nested procedures and V1 1-87
 - nested triggers and V1 1-126
- @@options global variable V1 1-372
- @@packet_errors global variable
 - sp_monitor and V2 3-406
- @@pack_received global variable
 - sp_monitor and V2 3-406
- @@pack_sent global variable
 - sp_monitor and V2 3-406

- `@@parallel_degree` global variable V1 1-372
 - set `parallel_degree` and V1 1-360
 - `@@rowcount` global variable V1 1-372
 - cursors and V1 1-230
 - set `nocount` and V1 1-372
 - triggers and V1 1-125
 - `@@scan_parallel_degree` global variable V1 1-372
 - set `scan_parallel_degree` and V1 1-361
 - `@@sqlstatus` global variable
 - fetch and V1 1-229
 - `@@textcolid` global variable V3 7-37
 - `@@textdbid` global variable V3 7-37
 - `@@textobjid` global variable V3 7-37
 - `@@textptr` global variable V3 7-37
 - `@@textsize` global variable V1 1-373
 - readtext and V1 1-315
 - set `textsize` and V1 1-363, V3 7-37
 - `@@textts` global variable V3 7-37
 - `@@thresh_hysteresis` global variable
 - threshold placement and V2 3-70
 - `@@total_errors` global variable
 - `sp_monitor` and V2 3-407
 - `@@total_read` global variable
 - `sp_monitor` and V2 3-406
 - `@@total_write` global variable
 - `sp_monitor` and V2 3-406
 - `@@tranchained` global variable V1 1-373
 - `@@version` global variable V1 1-305
 - `atn2` mathematical function V1 2-32
 - at option
 - dump database V1 1-196
 - dump transaction V1 1-210
 - load database V1 1-274
 - load transaction V1 1-283
 - At sign (@)
 - local variable name V1 1-146 to V1 1-147
 - procedure parameters and V1 1-224, V2 3-12
 - rule arguments and V1 1-91
 - Attributes
 - execution classes V2 3-26
 - remote tables V1 1-63
 - server (`sp_server_info`) V2 4-21
 - `sp_addobjectdef` and V2 3-48
 - Auditing
 - adding an audit table V2 3-21
 - `sysauditoptions` table V3 8-9
 - `sysaudits_01...sysaudits_08` tables V3 8-10
 - Audit options
 - displaying V2 3-211
 - examples V2 3-99
 - setting V2 3-93
 - Audit trail
 - adding comments V2 3-18
 - Authority. *See* Permissions
 - Authorizations. *See* Permissions
 - auto identity database option V2 3-194
 - Automatic operations
 - checkpoints V1 1-35
 - datatype conversion V1 1-106
 - `timestamp` columns V3 7-17
 - triggers V1 1-118
 - avg aggregate function V1 2-33
- ## B
- Backslash (\)
 - for character string continuation V1 1-409, V3 A-11
 - Backups
 - See also* Dump, database; Dump, transaction log; Load, database; Load, transaction log
 - disk mirroring and V1 1-165, V1 1-175
 - disk remirroring and V1 1-171
 - master database V1 1-8
 - Backup Server
 - See also* Utility Programs manual information about V2 3-353
 - multiple V2 3-67
 - volume handling messages V2 3-527 to V2 3-534
 - Base 10 logarithm function V1 2-90
 - Base date V3 7-19

- Base tables. *See* Tables
- Basic display level for configuration parameters V2 3-216
- Batch processing
 create default and V1 1-60
 execute V1 1-223, V1 1-226
 return status V1 1-318 to V1 1-321
 set options for V1 1-368
- bcp (bulk copy utility)
 select into/bulkcopy/pilsort and V2 3-196
- begin...end commands V1 1-25 to V1 1-26
 if...else and V1 1-258
 triggers and V1 1-119
- begin transaction command V1 1-27 to V1 1-28
 commit and V1 1-39
 rollback to V1 1-331
- between keyword V3 A-7
 check constraint using V1 1-115
 where V1 1-406
- binary datatype V3 7-28 to V3 7-30
- Binary datatypes V3 7-28 to V3 7-30
 "0x" prefix V1 1-59, V1 1-91, V3 7-28
 trailing zeros in V3 7-28
- Binary expressions V1 xxi, V3 A-1
 concatenating V3 A-5
- Binary operation, union V1 1-382
- Binary representation of data for bitwise operations V3 A-3
- Binary sort order of character sets V2 3-356
 order by and V1 1-299
- Binding
 data caches V2 3-109 to V2 3-113
 defaults V1 1-59, V2 3-114 to V2 3-117
 objects to data caches V2 3-109 to V2 3-113
 rules V1 1-93, V2 3-125 to V2 3-128
 unbinding and V1 1-179, V2 3-511 to V2 3-513, V2 3-516
 user messages to constraints V2 3-122 to V2 3-124
- bit datatype V3 7-31
- Bitwise operators V3 A-3 to V3 A-4
- Blanks
See also Spaces, character catalog stored procedure parameter values V2 4-2
 character datatypes and V1 1-263, V1 1-390, V3 7-24 to V3 7-27
 in comparisons V3 A-6
 empty string evaluated as V3 A-10
 like and V3 A-19
 removing leading with ltrim function V1 2-92
 removing trailing with rtrim function V1 2-132
 in system procedure parameter values V2 3-11
- Blocking process V1 1-271, V3 8-68
 sp_lock report on V2 3-281, V2 3-371
 sp_who report on V2 3-537
- Block size
 database device V1 1-161
- blocksize option
 dump database V1 1-196
 dump transaction V1 1-210
 load database V1 1-274
 load transaction V1 1-283
- Boolean (logical) expressions V3 A-1
 select statements in V1 1-259
- Brackets. *See* Square brackets []
- Branching V1 1-231
- break command V1 1-29 to V1 1-30, V1 1-411 to V1 1-412
- Browse mode
 select V1 1-346
 timestamp datatype and V1 2-158, V3 7-17
- B-trees, index
 fillfactor and V1 1-66
- Built-in functions V1 2-1 to V1 2-171
See also individual function names
 aggregate V1 2-6
 conversion V1 2-11
 date V1 2-19
 image V1 2-25
 mathematical V1 2-20

security V1 2-22
 string V1 2-22
 system V1 2-23
 text V1 2-25
 type conversion V1 2-47 to V1 2-49
 Bulk copying. *See* bcp (bulk copy utility)
 by row aggregate subgroup V1 1-40, V1
 2-10
 Bytes V3 7-19
 See also Size
 per row V1 1-20, V1 1-105
 bytes option, readtext V1 1-314

C

Caches, data
 binding objects to V2 3-109
 configuring V2 3-129 to V2 3-138
 dropping V2 3-136
 information about V2 3-132, V2 3-306
 logonly type V2 3-136
 memory pools V2 3-425 to V2 3-431
 overhead V2 3-136, V2 3-306
 recovery and V2 3-132
 status V2 3-134
 unbinding all objects from V2 3-514
 unbinding objects from V2 3-511
 Calculating dates V1 2-62
 caldayofweek date part V1 2-66
 calweekofyear date part V1 2-66
 calyearofweek date part V1 2-66
 Canceling
 See also rollback command
 command at rowcount V1 1-361
 duplicate updates or inserts V1 1-68
 queries with adjusted plans V1 1-360
 transactions with arithmetic
 errors V1 1-356
 triggers V1 1-333
 capacity option
 dump database V1 1-196
 dump transaction V1 1-210
 cascade option, revoke V1 1-324, V1 1-327
 Cascading changes (triggers) V1 1-121

case expressions **V1 1-31 to V1 1-34**
 null values and V1 1-33
 Case sensitivity V3 A-12
 in comparison expressions V3 A-5,
 V3 A-18
 compute and V1 1-46
 group by and V1 1-256
 sort order and V1 1-299
 in SQL V1 xix
 Catalog stored procedures V2 4-1 to V2
 4-39
 list of V2 4-1
 return status V2 4-2
 syntax V2 4-2 to V2 4-3
 cdw. *See* caldayofweek date part
 ceiling mathematical function **V1 2-35**
 chained option, set V1 1-357
 Chained transaction mode
 commit and V1 1-39
 delete and V1 1-157
 fetch and V1 1-229
 insert and V1 1-265
 open and V1 1-294
 sp_procxmode and V2 3-441
 update and V1 1-389
 Chains of pages
 partitions V1 1-19, V1 1-22
 text or image data V3 7-33
 unpartitioning V1 1-19
 Changes, canceling. *See* rollback
 command
 Changing
 See also Updating
 database options V2 3-191 to V2 3-199
 Database Owners V2 3-143 to V2
 3-145
 database size V1 1-6
 dbccdb workspace size V2 6-4
 language alias V2 3-472
 memory pools within data caches V2
 3-425
 object names V2 3-454 to V2 3-457
 passwords for login accounts V2
 3-415 to V2 3-417

- passwords for user-defined roles V1 1-12
- resource limits V2 3-392
- system tables, dangers of V3 8-4
- tables V1 1-13 to V1 1-24
- thresholds V2 3-400 to V2 3-404
- time ranges V2 3-396
- user-defined roles V1 1-10
- user's group V2 3-146 to V2 3-147
- view definitions V1 1-133
- Character data
 - avoiding "NULL" in V3 A-8
- Character datatypes V3 7-24 to V3 7-27
- Character expressions V1 xxi, V3 A-1, V3 A-2
 - blanks or spaces in V3 7-24 to V3 7-27
- Characters
 - See also* Spaces, character
 - "0x" V1 1-91, V1 2-17, V3 7-28, V3 7-29
 - not converted with `char_convert` V1 1-357
 - number of V1 2-41
 - stuff function for deleting V1 2-146
 - wildcard V3 A-16 to V3 A-22
- Character sets
 - changing names of V2 3-159, V2 3-161
 - checking with `sp_checknames` V2 3-148
 - checking with `sp_checkreswords` V2 3-154
 - conversion between client and server V1 1-357
 - conversion errors V3 A-16
 - `fix_text` upgrade after change in V1 1-140
 - `iso_1` V3 A-16
 - multibyte V2 3-356, V3 A-16
 - multibyte, changing to V1 1-140
 - object identifiers and V3 A-16
 - set `char_convert` V1 1-357
 - `sp_helpsort` display of V2 3-355
 - `syscharsets` system table V3 8-27
- Character strings
 - continuation with backslash (\) V3 A-11
 - empty V1 1-263, V3 A-10
 - specifying quotes within V3 A-10
 - truncation V1 1-263, V1 1-363
 - wildcards in V3 A-16
 - `@char_convert` global variable V1 1-372
 - `char_convert` option, set V1 1-357
 - `char` datatype V3 7-24
 - in expressions V3 A-10
 - row sort order and V1 1-300
 - Chargeback accounting
 - `sp_clearstats` procedure V2 3-172 to V2 3-173
 - `sp_reportstats` procedure V2 3-462 to V2 3-463
 - `charindex` string function V1 2-39
 - `char_length` string function V1 2-41
 - `chars` or `characters` option, `readtext` V1 1-314
 - `char` string function V1 2-37
 - `checkalloc` option, `dbcc` V1 1-138
 - `checkcatalog` option, `dbcc` V1 1-139
 - Check constraints
 - binding user messages to V2 3-122
 - column definition conflict with V1 1-115
 - displaying source text of V2 3-357
 - insert and V1 1-263
 - renaming V2 3-454 to V2 3-457
 - `sysconstraints` table V3 8-34
 - system tables entries for V3 8-63 to V3 8-64, V3 8-67
 - `checkdb` option, `dbcc` V1 1-139
 - Checker, consistency. *See* `dbcc` (Database Consistency Checker)
 - Checking passwords. *See* Passwords; `sp_remotoption` system procedure
 - `check` option
 - alter table V1 1-18
 - create table V1 1-102
 - checkpoint command V1 1-35 to V1 1-36
 - setting database options and V2 3-193
 - Checkpoint process V1 1-35 to V1 1-36

- See also* Recovery; Savepoints
- checkstorage option, dbcc V1 1-139
- checktable option, dbcc V1 1-139 to V1 1-140
- cis_rpc_handling option, set command V1 1-357
- Clearing accounting statistics V2 3-172 to V2 3-173
- Client
 - character set conversion V1 1-357
 - host computer name V1 2-81
- close command **V1 1-37**
- close on endtran option, set V1 1-357
- Closing cursors V1 1-37
- clustered constraint
 - alter table V1 1-15
 - create table V1 1-100
- Clustered indexes
 - See also* Indexes
 - creating V1 1-65 to V1 1-66
 - fillfactor and V1 1-66
 - migration of tables to V1 1-70, V1 1-107
 - number of total pages used V1 2-163
 - pages allocated to V3 8-46
 - segments and V1 1-68, V1 1-71
 - used_pgs system function and V1 2-163
- cntrtype option
 - disk init V1 1-161
 - disk reinit V1 1-169
- coalesce keyword, case V1 1-31
- Codes
 - datatype V2 4-14
 - ODBC datatype V2 4-4
 - soundex V1 2-139
- Collating sequence. *See* Sort order
- col_length system function **V1 2-43**
- Collision of database creation
 - requests V1 1-56
- col_name system function **V1 2-45**
- Colon (:)
- preceding milliseconds V1 2-66
- Column data. *See* Datatypes
- Column identifiers. *See* Identifiers
- Column name V1 2-45
 - aliasing V1 1-310, V1 1-339
 - changing V2 3-157, V2 3-454 to V2 3-457
 - checking with sp_checknames V2 3-148
 - grouping by V1 1-246, V1 1-247
 - in parentheses V1 2-9
 - as qualifier V3 A-13
 - union result set V1 1-383
 - views and V1 1-129
- Column pairs. *See* Joins; Keys
- Columns
 - adding data with insert V1 1-262
 - adding to table V1 1-13
 - check constraints conflict with definitions of V1 1-115
 - common key V2 3-174 to V2 3-176
 - creating indexes on V1 1-65 to V1 1-75
 - datatypes V2 4-9 to V2 4-11
 - defaults for V1 1-59 to V1 1-61, V1 1-263, V2 3-114 to V2 3-117
 - dependencies, finding V2 3-157
 - foreign keys V2 3-288 to V2 3-290, V2 4-16 to V2 4-18
 - group by and V1 1-246
 - identifying V3 A-13
 - joins and V2 3-330
 - length definition V1 2-43
 - length of V1 2-43
 - list and insert V1 1-261
 - null values and check constraints V1 1-115
 - null values and default V1 1-61, V1 1-93
 - numeric, and row aggregates V1 2-9
 - order by V1 1-344
 - permissions on V1 1-233, V2 4-5 to V2 4-7
 - permissions revoked V1 1-323
 - per table V1 1-20
 - primary key V2 3-432
 - reserved V3 8-3
 - rules V1 1-263, V2 3-125 to V2 3-128

- rules conflict with definitions of V1 1-93
- sizes of (list) V3 7-2 to V3 7-3
- unbinding defaults from V2 3-516 to V2 3-518
- unbinding rules with `sp_unbindrule` V2 3-524 to V2 3-526
- union of V1 1-383
- variable-length, and sort order V1 1-300
- views and V1 1-129
- Columns padding. *See* Padding, data
- Comma (,)
 - default print format for money values V3 7-15
 - not allowed in money values V3 7-15
 - in SQL statements V1 xviii
 - in user-defined datatypes V2 3-79
- Command execution delay. *See* `waitfor` command
- Command permissions **V1 1-237 to V1 1-239**
 - See also* Object permissions; Permissions
 - `grant all` V1 1-241
 - `grant assignment of` V1 1-232 to V1 1-244
 - levels V1 1-236
 - revoking V1 1-323
- Commands V1 1-1 to V1 1-416
 - display syntax of V2 3-501 to V2 3-503
 - order-sensitive V1 1-239, V1 1-327
 - `rowcount range for` V1 1-361
 - `statistics io for` V1 1-362
 - `statistics time information on` V1 1-362
 - Transact-SQL, summary table V1 1-1 to V1 1-5
- Comments
 - adding to audit trail V2 3-18
- `commit` command **V1 1-38 to V1 1-39**
 - `begin transaction and` V1 1-27, V1 1-39
 - `rollback and` V1 1-39, V1 1-331
- `commit work` command. *See* `commit` command
- Common keys
 - See also* Foreign keys; Joins; Primary keys
 - defining V2 3-174 to V2 3-176
 - dropping V2 3-240
 - join candidates and V2 3-330
 - reporting V2 3-332 to V2 3-334
 - `syskeys` table V3 8-48
- Comparing values
 - datatype conversion for V1 1-409
 - difference string function V1 2-72
 - in expressions V3 A-6
 - null-valued operands V1 1-355
 - for sort order V1 1-300
 - `timestamp` V1 2-158
 - in `where` clause V1 1-409
- Comparison operators
 - See also* Relational expressions
 - in expressions V3 A-5
 - symbols V3 A-5
 - `where` clause V1 1-405
- Compatibility, data
 - `create default and` V1 1-60
 - of rule to column datatype V1 1-92
- Compiled objects
 - checking for source text of V2 3-164
 - displaying source text of V2 3-357
 - hiding source text of V2 3-363
- Compiling
 - `exec with recompile and` V1 1-224
 - joins and table count V1 1-363
 - `sp_recompile and` V2 3-443 to V2 3-444
 - time (statistics time) V1 1-362
 - without execution (`noexec`) V1 1-359
- Composite indexes V1 1-65, V1 1-73
- Comprehensive display level for configuration parameters V2 3-216
- `compute` clause **V1 1-40 to V1 1-48**
 - `order by and` V1 1-298, V1 1-344
 - `select` V1 1-344
 - using row aggregates V1 2-8
 - `without by` V1 1-44
- Computing dates V1 2-62

- Concatenation
 - null values V3 A-5
 - using + operator V3 A-5
- Conceptual (logical) tables V1 1-121, V1 1-123
- Configuration parameters V1 1-4, V1 1-317
 - changing V2 3-177 to V2 3-183
 - display levels V2 3-216
 - help information on V2 3-308
 - system tables for V3 8-33, V3 8-35
- Conflicting roles V1 1-11
- @connections* global variable
 - sp_monitor* and V2 3-407
- connect to** command **V1 1-49**
- Consistency check. *See* **dbcc** (Database Consistency Checker)
- Constants V1 xx, V3 A-1
 - in expressions V3 A-10
 - return parameters in place of V1 1-226
 - in string functions V1 2-23
- constraint keyword
 - alter table V1 1-15
 - create table V1 1-99
- Constraints
 - binding user messages to V2 3-122
 - changing table V1 1-13
 - create table V1 1-108
 - cross-database V1 1-114, V1 1-190
 - displaying source text of V2 3-357
 - error messages V1 1-110
 - indexes created by and
 - max_rows_per_page* V1 1-17
 - information about V2 3-303, V2 3-312
 - referential integrity V1 1-112
 - renaming V2 3-454 to V2 3-457
 - sysconstraints* table V3 8-34
 - sysreferences* table V3 8-73
 - system tables entries for V3 8-31, V3 8-63 to V3 8-64
 - unbinding messages with
 - sp_unbindmsg* V2 3-522 to V2 3-523
 - unique V1 1-110
- Consumer process V1 1-69
- contiguous option (OpenVMS)
 - disk init V1 1-161
 - disk mirror V1 1-164
- Continuation lines, character string V1 1-409, V3 A-11
- continue command **V1 1-52 to V1 1-53**
 - while loop V1 1-411 to V1 1-412
- Controller, device
 - sp_helpdevice* and number V2 3-321
- Control-of-flow language
 - begin...end* and V1 1-25
 - create procedure* and V1 1-78
- Control pages for partitioned tables V1 1-22
 - syspartitions* and V3 8-66
 - updating statistics on V1 1-396
- Conventions
 - See also* Syntax
 - identifier name V3 A-14
 - Transact-SQL syntax V1 xviii
 - used in manuals V1 xviii
- Conversion
 - automatic values V3 7-7
 - between character sets V3 A-16
 - character value to ASCII code V1 2-28
 - columns V1 1-106
 - dates used with *like* V1 1-405, V3 7-22
 - degrees to radians V1 2-112
 - implicit V3 7-7, V3 A-10
 - integer value to character value V1 2-37
 - lowercase to uppercase V1 2-161
 - lower to higher datatypes V3 A-10
 - null values and automatic V1 1-106, V3 7-7
 - radians to degrees V1 2-71
 - string concatenation V3 A-5
 - styles for dates V1 2-47
 - uppercase to lowercase V1 2-91
 - where clause and datatype V1 1-409
- convert** function **V1 2-47 to V1 2-49**
 - concatenation and V3 A-5
 - date styles V1 2-47

- text values V3 7-37
- truncating values V1 2-14
- Copying
 - databases with create database V1 1-56 to V1 1-58
 - the *model* database V1 1-56
 - rows with insert...select V1 1-262
 - tables with select into V1 1-349
- Correlation names
 - table names V1 1-341
- Corrupt databases
 - listing V2 3-367
 - recovery fault isolation mode V2 3-481
- Corrupt indexes. *See* *reindex* option, *dbcc*
- Corrupt pages
 - bringing online V2 3-285 to V2 3-287
 - isolating on recovery V2 3-481 to V2 3-484, V2 3-485 to V2 3-487
 - listing V2 3-368
- cos mathematical function V1 2-50
- cot mathematical function V1 2-51
- count(*) aggregate function V1 2-53
- count aggregate function V1 2-52
- Counters, while loop. *See* while loop
- @*cpu_busy* global variable
 - sp_monitor* and V2 3-406
- CPU usage
 - monitoring V2 3-406
- create database command V1 1-54 to V1 1-58
 - disk init and V1 1-162
 - log on option V1 1-54
 - log on option compared to *sp_logdevice* V2 3-377
 - permission V1 1-241
- create default command V1 1-59 to V1 1-61
 - batches and V1 1-60
- create existing table command V1 1-62 to V1 1-64
 - datatype conversions and V1 1-64
- create index command V1 1-65 to V1 1-75
 - insert and V1 1-263
 - sp_extendsegment* and V2 3-277
- create procedure command V1 1-76 to V1 1-88
 - See also* Stored procedures; Extended stored procedures (ESPs)
 - order of parameters in V1 1-224, V1 1-226
 - return status and V1 1-84 to V1 1-85
 - select * in V1 1-82
- create role command V1 1-89
 - grant all and V1 1-90
- create rule command V1 1-91 to V1 1-94
- create schema command V1 1-95 to V1 1-96
- create table command V1 1-97 to V1 1-117
 - column order and V1 1-300
 - null values and V1 1-99, V3 A-8
 - sp_extendsegment* and V2 3-277
- create trigger command V1 1-118 to V1 1-137, V1 1-240, V1 1-327
- create view command V1 1-129 to V1 1-137
 - union prohibited in V1 1-384
- Creating
 - databases V1 1-54 to V1 1-58
 - datatypes V2 3-79 to V2 3-84
 - dbccdb* workspaces V2 6-9
 - defaults V1 1-59 to V1 1-61
 - execution classes V2 3-26
 - extended stored procedures V1 1-76 to V1 1-88, V2 3-28 to V2 3-29
 - indexes V1 1-65 to V1 1-75
 - limits V2 3-55
 - named time ranges V2 3-75
 - resource limits V2 3-55
 - rules V1 1-91 to V1 1-94
 - schemas V1 1-95 to V1 1-96
 - stored procedures V1 1-76 to V1 1-88
 - tables V1 1-97 to V1 1-117, V1 1-340
 - thresholds V2 3-69 to V2 3-74
 - time ranges V2 3-75
 - triggers V1 1-118 to V1 1-128, V1 1-240, V1 1-327
 - user aliases V2 3-16 to V2 3-17

- user-defined roles V1 1-89
 - user groups V2 3-33
 - views V1 1-129 to V1 1-137
 - Curly braces ({})
 - in SQL statements V1 xviii
 - Currency symbols V3 7-15, V3 A-11
 - Current database
 - changing V1 1-400
 - information from `sp_helpdb` V2 3-318
 - space used by V2 3-498 to V2 3-500
 - Current date V1 2-77
 - Current locks, `sp_lock` system
 - procedure V1 1-272, V2 3-370
 - Current processes. *See* Processes (Server tasks)
 - Current usage statistics V2 3-462 to V2 3-463
 - Current user
 - roles of V1 2-133
 - `user_id` system function V1 2-151
 - `user_name` system function V1 2-152
 - `user_id` system function V1 2-165
 - `user_name` system function V1 2-167
 - `user` system function V1 2-164
 - Cursor result set V1 1-151
 - datatypes and V1 1-228
 - returning rows V1 1-228
 - cursor rows option, set V1 1-358
 - Cursors
 - aggregate functions and V1 2-9
 - closing V1 1-37
 - compute clause and V1 1-45
 - datatype compatibility V1 1-228
 - deallocating V1 1-145
 - declaring V1 1-148 to V1 1-153
 - deleting rows V1 1-158
 - fetching V1 1-228 to V1 1-230
 - grant and V1 1-239
 - group by and V1 1-248
 - Halloween problem V1 1-152
 - information about V2 3-187
 - opening V1 1-294
 - order by and V1 1-298
 - read-only V1 1-151
 - scans V1 1-151
 - scope V1 1-149
 - select and V1 1-349
 - union prohibited in updatable V1 1-384
 - updatable V1 1-151
 - updating rows V1 1-390
 - `curunreservedpgs` system function V1 2-54
 - Custom datatypes. *See* User-defined datatypes
 - `cwk`. *See* `calweekofyear` date part
 - `cyr`. *See* `calyearofweek` date part
 - Cyrillic characters V3 A-16
- ## D
- Damaged database, removing and repairing V1 1-140
 - Database design
 - dropping keys V2 3-240
 - logical relationships in V2 3-174, V2 3-288
 - Database devices
 - alter database and V1 1-6
 - defaulton or defaultoff status V2 3-209 to V2 3-210
 - dropping V2 3-225 to V2 3-226
 - dropping segments from V2 3-259 to V2 3-261
 - last device reference for V2 3-261
 - listing of V2 3-320
 - new database V1 1-54
 - `sp_helpdevice` system procedure V2 3-320
 - status V2 3-209
 - `sysdevices` table V3 8-41
 - system table entries for V3 8-41
 - transaction logs on separate V1 1-166, V1 1-172
 - Database dump. *See* Dump, database; Dump devices
 - Database files. *See* Files
 - Database object owners
 - See also* Database Owners; Ownership

- identifiers and V3 A-14
- sp_depends system procedure and V2 3-206
- Database objects
 - See also individual object names*
 - adding to tempdb V1 1-106
 - binding defaults to V2 3-114 to V2 3-117
 - binding rules to V2 3-125
 - binding to caches V2 3-109
 - dependencies of V2 3-206 to V2 3-208, V3 8-40
 - finding V2 3-207, V2 3-301
 - identifier names V3 A-11
 - ID number (object_id) V1 2-99
 - listings of V2 3-298
 - permissions on V1 1-238, V2 3-346
 - permissions when creating procedures V1 1-87
 - permissions when creating triggers V1 1-128
 - permissions when creating views V1 1-136
 - permissions when executing procedures V1 1-88
 - permissions when executing triggers V1 1-128
 - permissions when invoking views V1 1-136
 - referencing, create procedure and V1 1-82
 - remapping V2 3-445 to V2 3-447
 - renaming V2 3-454 to V2 3-457
 - select_list V1 1-309 to V1 1-310, V1 1-339 to V1 1-340
 - space used by V2 3-498 to V2 3-500
 - sp_tables list of V2 4-38 to V2 4-39
 - sysobjects table V3 8-63 to V3 8-64
 - user-defined datatypes as V3 7-39
- Database options V2 3-193 to V2 3-198
 - See also individual option names*
 - listing V2 3-191 to V2 3-199
 - showing settings V2 3-193, V2 3-317
- Database Owners
 - See also Database object owners; Permissions*
 - adding users V2 3-88
 - changing V2 3-143
 - dbo use only database option V2 3-194
 - information on V2 3-361 to V2 3-362
 - name as qualifier V3 A-13, V3 A-14
 - objects and identifiers V3 A-14
 - permissions granted by V1 1-232
 - transferring ownership V2 3-143
 - use of setuser V1 1-236
- Databases
 - See also Database objects*
 - adding groups V2 3-33
 - adding users V2 3-88
 - backing up V1 1-195 to V1 1-207
 - binding to data caches V2 3-109, V2 3-110
 - changing user's default V2 3-389
 - checkalloc option (dbcc) V1 1-138
 - checkdb option (dbcc) V1 1-139
 - checking with sp_checknames V2 3-148
 - checkstorage option (dbcc) V1 1-139
 - creating V1 1-54
 - creating with separate log segment V1 1-216
 - creation permission V1 1-58
 - default size V1 1-55
 - dropping V1 1-177
 - dropping segments from V2 3-259 to V2 3-261
 - dropping users from V2 3-270
 - dumping V1 1-195 to V1 1-207
 - help on V2 3-317
 - ID number, db_id function V1 2-69
 - increasing size of V1 1-6
 - information on storage space used V2 3-318, V2 3-498
 - listing suspect V2 3-367
 - listing suspect pages in V2 3-368
 - listing with sp_databases V2 4-12
 - listing with sp_helpdb V2 3-317
 - loading V1 1-273 to V1 1-281

- lock promotion thresholds for V2 3-475
- name V1 2-70
- number of Server V1 1-56
- options V2 3-191 to V2 3-199
- ownership V2 3-143
- recovering V1 1-273 to V1 1-281
- removing and repairing damaged V1 1-140
- renaming V2 3-458 to V2 3-461
- running out of space in V2 3-508
- selecting V1 1-400
- storage information V2 3-498
- system tables entries for V3 8-37
- thresholds V2 3-508
- unbinding from data caches V2 3-511
- upgrading database dumps V1 1-278, V1 1-288
- use command V1 1-400
- Data caches
 - binding objects to V2 3-109
 - configuring V2 3-129 to V2 3-138
 - dropping V2 3-136
 - information about V2 3-132, V2 3-306
 - logonly type V2 3-136
 - memory pools V2 3-425 to V2 3-431
 - overhead V2 3-136, V2 3-306
 - recovery and V2 3-132
 - status V2 3-134
 - unbinding all objects from V2 3-514
 - unbinding objects from V2 3-511
- Data dependency. *See* Dependencies, database object
- Data dictionary. *See* System tables
- Data integrity V1 1-263
 - See also* Referential integrity constraints
 - dbcc check for V1 1-138
- datalength system function **V1 2-58**
 - compared to col_length V1 2-43
- Data modification
 - text and image with writetext V1 1-414
 - update V1 1-385
- Data padding. *See* Padding, data
- data_pgs system function **V1 2-56**
- dataserver utility command
 - See also* Utility Programs manual
 - disk mirror and V1 1-166
 - disk remirror and V1 1-172
- Datatype conversion functions V1 2-11 to V1 2-19
- Datatype conversions
 - binary and numeric data V1 2-18
 - bit information V1 2-18
 - character information V1 2-13, V1 2-14
 - column definitions and V1 1-106
 - convert function V1 2-49
 - date and time information V1 2-15
 - domain errors V1 2-17, V1 2-49
 - hexadecimal-like information V1 2-17
 - hextoint function V1 2-78
 - image V1 2-18, V1 2-49
 - implicit V1 2-11
 - intohex function V1 2-84
 - money information V1 2-14
 - numeric information V1 2-14, V1 2-15
 - overflow errors V1 2-16
 - rounding during V1 2-14
 - scale errors V1 2-17
- Datatype precedence. *See* Precedence
- Datatypes **V3 7-1 to V3 7-40**
 - See also* User-defined datatypes; individual datatype names
 - approximate numeric V3 7-13
 - binary V3 7-28 to V3 7-30
 - bit V3 7-31
 - codes V2 4-4, V2 4-14
 - comparison in union operations V1 1-383
 - compatibility of column and default V1 1-60
 - cursor result set and V1 1-228
 - date and time V3 7-19 to V3 7-23
 - datetime values comparison V3 A-6
 - decimal V3 7-10
 - defaults and V2 3-114 to V2 3-117

- dropping user-defined V2 3-268, V3 7-39
- exact numeric V3 7-9 to V3 7-10
- extended V2 4-4
- hierarchy V2 3-81, V3 7-5, V3 8-85
- integer V3 7-9 to V3 7-10
- invalid in *group by* and *having* clauses V1 1-248
- list of V3 7-2, V3 8-85
- local variables and V1 1-146
- mixed, arithmetic operations on V3 A-3
- ODBC V2 4-4
- physical V2 3-79
- sp_datatype_info* information on V2 4-14 to V2 4-15
- sp_help* information on V2 3-298 to V2 3-302
- summary of V3 7-2
- systypes* table V3 8-85 to V3 8-86
- trailing zeros in *binary* V3 7-28
- unbinding defaults from V2 3-516 to V2 3-518
- unbinding rules with *sp_unbindrule* V2 3-524 to V2 3-526
- Datatypes, custom. *See* User-defined datatypes
- dateadd* function V1 2-60
- datediff* function V1 2-62 to V1 2-63
- datefirst* option, set V1 1-358, V1 2-64, V1 2-67
- dateformat* option, set V1 1-358, V3 7-21
- Date formats V3 7-20
- Date functions V1 2-19 to V1 2-20
See also individual function names
- datename* function V1 2-64
- datepart* function V1 2-66
- Date parts
 - abbreviation names and values V1 2-20, V1 2-66
 - entering V3 7-19
 - order of V1 1-358, V2 3-35, V3 7-21
- Dates
 - comparing V3 A-6
 - datatypes V3 7-19 to V3 7-23
 - display formats V1 1-358
 - display formats, *waitfor* command V1 1-402
 - earliest allowed V1 2-19, V1 2-60, V3 7-19
 - pre-1753 datatypes for V1 2-19, V1 2-60
 - datetime* datatype V3 7-19 to V3 7-23
See also set command
 - comparison of V3 A-6
 - conversion V3 7-23
 - date functions and V1 2-66
 - values and comparisons V3 7-23
 - day date part V1 2-20, V1 2-66
 - Day-long time ranges V2 3-75
 - dayofyear* date part abbreviation and values V1 2-20, V1 2-66
 - Days
 - alternate language V2 3-35
 - date style V1 2-48
 - in time ranges V2 3-75
 - dbcc* (Database Consistency Checker) V1 1-138 to V1 1-144
See also individual dbcc options
 - readtext* and V1 1-316
 - scripts and *sp_checkreswords* V2 3-156
 - space allocation and V2 3-419
 - dbccdb* database
 - changing workspace size in V2 6-4
 - creating workspaces in V2 6-9
 - deleting *dbcc* checkstorage history from V2 6-15
 - deleting target database information from V2 6-13
 - reporting allocation statistics from V2 6-30
 - reporting comprehensive information from V2 6-26
 - reporting configuration information from V2 6-7, V2 6-23, V2 6-26
 - reporting fault information from V2 6-17, V2 6-23
 - reporting full details from V2 6-26

- reporting I/O statistics from V2 6-17
- stored procedures for use with V2 6-1
- dbcc traceon V1 1-142
- dbcc tune V1 1-142
- dbid column, sysusages table V3 8-87
- db_id system function V1 2-69
- DB-Library programs
 - browse mode V1 1-346
 - changing identifier names and V2 3-156
 - dbwritetext and dbmoretext, writetext compared to V1 1-416
 - overflow errors V1 2-34, V1 2-150
 - prepare transaction V1 1-303
 - set options for V1 1-359, V1 1-367
 - waitfor mirrorexit and V1 1-402
- db_name system function V1 2-70
- dbo use only database option
 - setting with sp_dboption V2 3-194
- dbrepair option, dbcc V1 1-140
- dd. *See* day date part
- ddl in tran database option V2 3-195
- Deactivation of disk mirroring V1 1-174 to V1 1-176
- Deadlocks
 - descending scans and V1 1-301
- deallocate cursor command V1 1-145
- Deallocating cursors V1 1-145
- Debugging aids
 - set showplan on V1 1-362
 - set sort_resources on V1 1-362
 - set statistics io on V1 1-362
 - triggers and V1 1-126
- decimal datatype V3 7-10
- Decimal numbers
 - round function and V1 2-128
 - str function, representation of V1 2-143
- Decimal points
 - datatypes allowing V3 7-10
 - in integer data V3 7-9
- declare command V1 1-146 to V1 1-147
- declare cursor command V1 1-148 to V1 1-153
- Declaring
 - local variables V1 1-146
 - parameters V1 1-77
- Default database
 - See also* sysdevices table
 - assigning with sp_addlogin V2 3-40
 - changing user's V2 3-389
- Default database devices
 - setting status with sp_diskdefault V2 3-209
 - sp_helpdevice and V2 3-320
- default database size configuration
 - parameter in sysconfigures V1 1-55
- default keyword
 - alter database V1 1-6
 - alter table V1 1-14
 - create table V1 1-98
- default language id configuration
 - parameter V2 3-40
- defaulton | defaultoff option, sp_diskdefault V2 3-209
- Defaults V1 1-263
 - binding V2 3-114 to V2 3-117
 - checking name with sp_checkreswords V2 3-153
 - column V1 1-14
 - creating V1 1-59 to V1 1-61
 - definitions and create default V1 1-59 to V1 1-61
 - displaying source text of V2 3-357
 - dropping V1 1-179
 - IDENTITY columns and V1 1-23
 - remapping V2 3-445 to V2 3-447
 - renaming V2 3-157, V2 3-454 to V2 3-457
 - rules and V1 1-60, V1 1-93
 - system tables and V2 3-115
 - system tables entries for V3 8-31, V3 8-63 to V3 8-64, V3 8-67
 - unbinding V2 3-516 to V2 3-518
- default segment
 - dropping V2 3-260
 - extending V1 1-9

- mapping V2 3-63
- Default settings
 - changing login V2 3-41, V2 3-389
 - configuration parameters V2 3-179
 - date display format V3 7-22
 - language V2 3-40
 - parameters for stored procedures V1 1-77
 - set command options V1 1-367
 - weekday order V1 1-367, V1 2-67
- Default values
 - datatype length V1 2-47
 - datatype precision V1 2-47
 - datatype scale V1 2-47
 - datatypes when no length specified V1 1-77
- Defining local variables V1 1-146 to V1 1-147
- defncopy utility command V2 3-155
- Degree of parallelism
 - select and parallel V1 1-341
- Degrees, conversion to radians V1 2-112
- degrees mathematical function **V1 2-71**
- Delayed execution (waitfor) V1 1-401
- delete command **V1 1-154 to V1 1-159**
 - text row V3 7-36
 - triggers and V1 1-122
 - truncate table compared to V1 1-379
- deleted table
 - triggers and V1 1-121, V1 1-123
- Deleting
 - See also* Dropping
 - dbcc checkstorage history from *dbccdb* V2 6-15
 - files V2 3-225
 - target database information from *dbccdb* V2 6-13
- Delimited identifiers
 - testing V2 3-155
 - using V2 3-154, V2 3-161 to V2 3-162
- density option
 - dump database V1 1-196
 - dump transaction V1 1-210
 - load database V1 1-274
 - load transaction V1 1-283
- Denying access to a user V2 3-374
- Dependencies, database object
 - changing names of V2 3-155
 - recompilation and V2 3-455
 - sp_depends system procedure V1 1-106, V2 3-206 to V2 3-208
 - sysdepends table V3 8-40
- Descending order (desc keyword) V1 1-296, V1 1-344
- Descending scans V1 1-300
 - deadlocks and V1 1-301
 - overflow pages and V1 1-301
- detail option, sp_helpconstraint V2 3-312
- Device failure
 - dumping transaction log after V1 1-212, V1 1-215
- Device fragments
 - number of V1 1-8, V1 1-56
 - sp_helpdb report on V2 3-317
- Device initialization. *See* Initializing
- Devices
 - See also* sysdevices table
 - changing names of V2 3-158, V2 3-161
 - disk mirroring to V1 1-164 to V1 1-167
 - information on log V2 3-337
 - numbering V1 1-160, V1 1-169
 - secondary V1 1-165
 - system tables entries for V3 8-41
- Dictionary sort order V1 1-299
- difference string function **V1 2-72**
- Direct updates
 - to system tables V2 3-158, V3 8-4
- Dirty pages
 - updating V1 1-35 to V1 1-36
- Disabling mirroring. *See* Disk mirroring
- disconnect command **V1 1-49**
- Disk allocation pieces V3 8-87
- Disk controllers V1 1-161, V1 1-169
- Disk devices
 - adding V1 1-160 to V1 1-163, V2 3-85 to V2 3-87
 - mirroring V1 1-164 to V1 1-167
 - sysdevices table V3 8-41

- unmirroring V1 1-174 to V1 1-176
- disk init command V1 1-160 to V1 1-163
 - master database backup after V1 1-162
- disk mirror command V1 1-164 to V1 1-167
- Disk mirroring V1 1-164 to V1 1-167
 - database dump and V1 1-206
 - database load and V1 1-280
 - restarting V1 1-171 to V1 1-173
 - sp_who report on V2 3-536
 - status in sysdevices table V3 8-41
 - transaction log dump and V1 1-221
 - transaction log load and V1 1-290
 - unmirroring and V1 1-174 to V1 1-176
 - waitfor mirrorexist V1 1-401
- disk option, sp_addumpdevice V2 3-85
- disk refit command V1 1-168
 - create database and V1 1-56
- disk reinit command V1 1-169 to V1 1-170
 - See also disk init command
- disk remirror command V1 1-171 to V1 1-173
 - See also Disk mirroring
- disk unmirror command V1 1-174 to V1 1-176
 - See also Disk mirroring
- dismount option
 - dump database V1 1-197
 - dump transaction V1 1-211
 - load database V1 1-274
 - load transaction V1 1-283
- Display
 - character sets V2 3-355
 - create procedure statement text V1 1-86
 - database options V2 3-191 to V2 3-199
 - procedures for information V1 1-79
 - setting for command-affected
 - rows V1 1-359
 - source text of compiled objects V2 3-357
 - syntax of modules V2 3-501
- distinct keyword
 - create view V1 1-130
 - select V1 1-339, V1 1-348
- Dividing tables into groups. See group by clause
- Division operator (/) V3 A-3
- Dollar sign (\$)
 - in identifiers V3 A-11
 - in money datatypes V3 7-15
- Domain rules V1 1-263
 - create rule command V1 1-91
 - mathematical functions errors in V1 2-21
 - violations V1 1-263
- “don’t recover” status of databases
 - created for load V1 1-57
- Dots (..) for omitted name elements V2 3-46, V3 A-14
- Double-byte characters. See Multibyte character sets
- double precision datatype V3 7-14
- Double-precision floating-point values V3 7-14
- Doubling quotes
 - in character strings V1 1-409, V3 7-25
 - in expressions V3 A-10
- drop database command V1 1-177 to V1 1-178
 - damaged databases and V1 1-140
- dropdb option, dbcc dbrepair V1 1-140
- drop default command V1 1-179 to V1 1-180
- drop index command V1 1-181 to V1 1-182
- drop keyword
 - alter role V1 1-10
 - alter table V1 1-18
- drop logins option, sp_dropserver V2 3-262
- dropmessages option, sp_droplanguage V2 3-243
- Dropping
 - See also Deleting
 - aliased user V2 3-223 to V2 3-224
 - character with stuff function V1 2-146
 - columns from a table V1 1-20
 - corrupt indexes V1 1-141
 - damaged database V1 1-140
 - database devices V2 3-225 to V2 3-226

- databases V1 1-177 to V1 1-178
- dbcc dbrepair database V1 1-140
- defaults V1 1-60, V1 1-179
- grouped procedures V1 1-76
- groups V2 3-238 to V2 3-239
- indexes V1 1-181 to V1 1-182
- leading or trailing blanks V1 2-92
- lock promotion thresholds V2 3-235
- passwords from roles V1 1-10
- procedures V1 1-183 to V1 1-184, V2 3-231 to V2 3-232
- remote logins V2 3-252 to V2 3-253, V2 3-262
- remote servers V2 3-262 to V2 3-263
- resource limits V2 3-254
- roles in a mutually exclusive relationship V1 1-10
- rows from a table V1 1-154 to V1 1-159, V1 1-190
- rows from a table using truncate table V1 1-379
- rules V1 1-187
- segment from a database V2 3-259 to V2 3-261
- tables V1 1-189 to V1 1-191
- tables with triggers V1 1-123
- time ranges V2 3-266
- triggers V1 1-123, V1 1-192
- user-defined datatype V2 3-268 to V2 3-269
- user-defined messages V2 3-247 to V2 3-248
- user-defined roles V1 1-185
- user from a database V2 3-270 to V2 3-272
- user from a group V2 3-146
- views V1 1-194
- drop procedure command **V1 1-183 to V1 1-184**
 - grouped procedures and V1 1-183, V1 1-224
- drop role command **V1 1-185**
- drop rule command **V1 1-187 to V1 1-188**
- drop table command **V1 1-189 to V1 1-191**
- drop trigger command **V1 1-192 to V1 1-193**
- drop view command **V1 1-194**
- Dump, database
 - across networks V1 1-201
 - appending to volume V1 1-205 to V1 1-206
 - Backup Server, remote V1 1-196
 - Backup Server and V1 1-203
 - block size V1 1-196
 - commands used for V1 1-200, V1 1-215
 - dismounting tapes V1 1-197
 - dump devices V1 1-196, V1 1-202
 - dump striping V1 1-197
 - dynamic V1 1-201
 - expiration date V1 1-197
 - file name V1 1-198, V1 1-203
 - initializing/appending V1 1-198
 - interrupted V2 3-200
 - loading V1 1-57, V1 1-273 to V1 1-281
 - master database V1 1-202
 - message destination V1 1-198
 - new databases and V1 1-201
 - overwriting V1 1-197, V1 1-205 to V1 1-206
 - remote V1 1-203
 - rewinding tapes after V1 1-197
 - scheduling V1 1-200 to V1 1-202
 - successive V1 1-204, V1 1-219
 - system databases V1 1-202
 - tape capacity V1 1-196
 - tape density V1 1-196
 - thresholds and V1 1-202
 - volume changes V1 1-204
 - volume name V1 1-197, V1 1-204
- Dump, transaction log
 - across networks V1 1-216
 - appending dumps V1 1-212
 - appending to volume V1 1-220 to V1 1-221
 - Backup Server, remote V1 1-218
 - command used for V1 1-215
 - dismounting tapes V1 1-211

- dump striping V1 1-211
 - expiration date V1 1-211
 - file name V1 1-212, V1 1-218 to V1 1-219
 - initializing tape V1 1-212
 - initializing volume V1 1-220 to V1 1-221
 - insufficient log space V1 1-216
 - loading V1 1-282 to V1 1-291
 - message destination V1 1-212
 - permissions problems V1 1-214
 - remote V1 1-218, V1 1-219
 - rewinding tapes after V1 1-211
 - scheduling V1 1-216 to V1 1-217
 - tape capacity V1 1-210
 - thresholds and V1 1-217
 - volume name V1 1-211, V1 1-219
 - dump database command **V1 1-195 to V1 1-207**
 - See also* Dump, database
 - after using create database V1 1-57
 - after using disk init V1 1-162
 - after using dump transaction with `no_log` V1 1-209
 - dump transaction and V1 1-201
 - master database and V1 1-201
 - restrictions V1 1-200
 - select into and V1 1-350
 - Dump devices
 - See also* Database devices; Log device
 - adding V2 3-85 to V2 3-87
 - dropping V2 3-225 to V2 3-226
 - dump, database and V1 1-196
 - dump, transaction log and V1 1-210
 - listing V2 3-320
 - naming V1 1-196, V1 1-210, V1 1-217 to V1 1-218
 - number required V1 1-279
 - permission and ownership problems V2 3-86
 - `sysdevices` table and V3 8-41
 - system tables entries for V3 8-41
 - Dump striping
 - database dumps and V1 1-197
 - transaction dumps and V1 1-211
 - dump transaction command **V1 1-208 to V1 1-222**
 - See also* Dump, transaction log
 - after using disk init V1 1-162
 - permissions for execution V1 1-222
 - select into/bulkcopy/pllsort and V1 1-214
 - `sp_logdevice` and V2 3-378
 - trunc log on chkpt and V1 1-214
 - with `no_log` option V1 1-216
 - with `no_truncate` option V1 1-212, V1 1-215
 - with `truncate_only` option V1 1-215
 - dumpvolume option
 - dump database V1 1-197, V2 3-527
 - dump transaction V1 1-211
 - load database V1 1-274
 - load transaction V1 1-283
 - Duplicate rows
 - indexes and V1 1-65, V1 1-68
 - removing with union V1 1-381
 - text or image V3 7-37
 - Duplication
 - of space for a new database V1 1-57
 - of a table with no data V1 1-350
 - Duplication of text. *See* replicate string function
 - dw. *See* weekday date part
 - dy. *See* dayofyear date part
 - Dynamic dumps V1 1-201, V1 1-216
 - Dynamic Link Libraries
 - unloading V2 3-291 to V2 3-292
- ## E
- 8-bit terminal, `sp_helpsort` output V2 3-355
 - Ellipsis (...) in SQL statements V1 xx
 - else keyword. *See* if...else conditions
 - Embedded spaces. *See* Spaces, character
 - Empty string (" ") or (' ')
 - not evaluated as null V3 A-9
 - as a single space V1 1-263, V3 7-27, V3 A-10

- updating an V1 1-390
- Enclosing quotes in expressions V3 A-10
- Encryption
 - compiled object source text V2 3-363
 - role passwords V3 8-82
 - user passwords V3 8-56
- Ending days of named time ranges V2 3-75
- Ending times of named time ranges V2 3-75
- end keyword V1 1-25
- Enforcing resource limits V2 3-56
- Engines
 - sysengines* table V3 8-43
 - system tables entries for V3 8-43
- English language, U.S. *See* *us_english* language
- e or E exponent notation
 - approximate numeric datatypes V3 7-14
 - float* datatype V3 7-4
 - money datatypes V3 7-15
- Equal to. *See* Comparison operators
- errorexit keyword, waitfor V1 1-401
- @error global variable
 - select into and V1 1-350
 - stored procedures and V1 1-82
 - user-defined error messages and V1 1-306, V1 1-312
- Error handling
 - in character set conversion V1 1-357
 - dbcc and V1 1-144
 - domain or range V1 2-21
 - triggers and V1 1-126
- Error messages
 - See also* SQLSTATE codes
 - Backup Server V2 3-531
 - character conversion V1 1-357
 - printing user-defined V1 1-306
 - system tables entries for V3 8-61
 - user-defined V1 1-308 to V1 1-313
- Errors
 - See also* Error messages
- allocation V1 1-139, V1 1-141, V1 1-142
- arithmetic overflow V1 2-16
- convert function V1 2-14 to V1 2-17, V1 2-49
- datatype conversion V1 1-99
- divide-by-zero V1 2-16
- domain V1 2-17, V1 2-49
- number of V2 3-406
- numbers for user-defined V1 1-308
- return status values V1 1-319
- scale V1 2-17
- trapping mathematical V1 2-21
- Escape characters V3 A-21
 - wildcard characters and V3 A-22
- escape keyword V3 A-21 to V3 A-22
 - where V1 1-406
- ESPs. *See* Extended stored procedures
- European characters in object
 - identifiers V3 A-16
- Evaluation order V1 1-382
- Exact numeric datatypes V3 7-9 to V3 7-10
 - arithmetic operations and V3 7-9
- Exception report, dbcc tablealloc V1 1-141, V1 1-142
- Exclamation point (!)
 - error message placeholder V1 1-304
- exclusive keyword, alter role V1 1-10
- Exclusive locks V2 3-281, V2 3-371
- execute command V1 1-223 to V1 1-227
 - create procedure and V1 1-82
- Execution
 - operating system commands V2 5-3
 - procedures V1 1-223
 - specifying times for V1 1-401
- Execution delay. *See* waitfor command
- exists keyword
 - in expressions V3 A-6
 - where V1 1-407
- Exit
 - unconditional, and return command V1 1-318 to V1 1-321
 - waitfor command V1 1-401

expand_down parameter
 sp_activeroles V2 3-14
 sp_displayroles V2 3-221
 sp_displayroles V2 3-221
 Explicit null value V3 A-8
 Explicit values for IDENTITY
 columns V1 1-266, V1 1-358
 exp mathematical function V1 2-74
 Exponent, datatype (e or E)
 approximate numeric types V3 7-14
 float datatype V3 7-4
 money types V3 7-15
 Exponential value V1 2-74
 Expressions
 definition of V3 A-1
 enclosing quotes in V3 A-10
 evaluation order in V1 1-382
 grouping by V1 1-247
 including null values V3 A-7
 insert and V1 1-261
 name and table name qualifying V3
 A-15
 summary values for V1 1-45
 types of V1 xx, V3 A-1
 Extended columns, Transact-SQL V1
 1-250, V1 1-252
 Extended datatypes, ODBC V2 4-4
 Extended stored procedures
 creating V1 1-76 to V1 1-88, V2 3-28 to
 V2 3-29
 C run-time signals not allowed V1
 1-83
 displaying V2 3-322
 dropping V1 1-183, V2 3-231
 executing V1 1-223 to V1 1-227
 system tables entries for V3 8-31, V3
 8-63 to V3 8-64
 Extending database storage V1 1-6
 Extending segments V2 3-277
 Extensions, Transact-SQL V1 1-250
 Extents V1 1-70
 create table and V1 1-105
 dbcc indexalloc report on index V1 1-141
 dbcc report on table V1 1-142

F

Failures, media
 See also Recovery
 automatic failover and V1 1-174
 disk remirror and V1 1-172
 trunc log on chkpt database option
 and V2 3-197
 Family of worker processes
 fid reported by sp_lock V2 3-372
 sp_familylock report on fid V2 3-280
 fast option
 dbcc indexalloc V1 1-140
 dbcc tablealloc V1 1-141, V1 1-142
 fetch command V1 1-228 to V1 1-230
 Fetching cursors V1 1-228 to V1 1-230
 fid (family ID) number V2 3-280
 sp_lock report V2 3-372
 File names
 configuration file V2 3-177
 database dumps V1 1-203
 DLL V1 1-78, V2 3-291
 listing database dump with listonly V1
 1-275
 listing transaction log with listonly V1
 1-284
 transaction log dumps V1 1-212, V1
 1-283
 file option
 dump database V1 1-198
 dump transaction V1 1-212
 load database V1 1-274
 load transaction V1 1-283
 Files
 See also Tables; Transaction log
 contiguous (OpenVMS) V1 1-161, V1
 1-164
 deleting V2 3-225
 inaccessible after sp_dropdevice V2
 3-225
 interfaces, and server names V2 3-65
 localization V2 3-161
 mirror device V1 1-164
 Fillfactor
 alter table V1 1-16

- create index and V1 1-66
- fillfactor option
 - alter table V1 1-16
 - create index V1 1-66
 - create table V1 1-100
- Finding
 - active roles V1 2-133
 - cache bindings V2 3-129, V2 3-306
 - character sets V2 3-355, V3 8-27
 - configuration parameters V2 3-308, V3 8-33, V3 8-35
 - constraints V2 3-312, V3 8-34
 - current date V1 2-77
 - database ID V1 2-69, V3 8-37
 - database name V1 2-70, V3 8-37
 - database objects V2 3-301, V3 8-63
 - database options V2 3-191
 - database settings V2 3-317, V3 8-37
 - datatypes V2 3-298, V3 8-85
 - device names V3 8-41
 - devices V2 3-320
 - languages V2 3-335, V3 8-50
 - object definitions V3 8-31, V3 8-67
 - object dependencies V2 3-206, V2 3-207, V3 8-40
 - object information V2 3-298
 - partition information V2 3-303, V3 8-66
 - permission information V3 8-71
 - permissions V2 3-346
 - reserved words V2 3-150
 - resource limits V2 3-342, V3 8-76
 - roles V3 8-77
 - segments V2 3-350
 - server names V2 3-353
 - server user ID V1 2-151
 - server user name V1 2-152
 - starting position of an expression V1 2-39
 - thresholds V2 3-359
 - user aliases V1 2-171, V3 8-5
 - user IDs V1 2-165
 - user names V1 2-164, V1 2-167
 - users in a database V2 3-361, V3 8-89
 - valid identifiers V1 2-169
- FIPS flagger
 - insert extension not detected by V1 1-269
 - set option for V1 1-358
 - update extensions not detected by V1 1-393
- fipsflagger option, set V1 1-358
- First column parameter. *See* Keys
- First-of-the-months, number of V1 2-63
- First page
 - log device V2 3-337
 - partition, displaying with `sp_helppartition` V2 3-303
 - text pointer V1 2-154
- Fixed-length columns
 - binary datatypes for V3 7-28
 - character datatypes for V3 7-24
 - null values in V3 7-7
 - stored order of V1 1-300
- fix option
 - `dbcc` V1 1-139, V1 1-141, V1 1-142
 - `dbcc indexalloc` V1 1-140
 - `dbcc tablealloc` V1 1-138
- `fix_text` option, `dbcc` V1 1-140, V1 1-144
- float* datatype V3 7-14
- Floating-point data V1 xx, V3 A-1
 - `str` character representation of V1 2-143
- floor mathematical function V1 2-76
- flushmessage option, set V1 1-358
- for browse option, select V1 1-346
 - union prohibited in V1 1-384
- foreign key constraint
 - alter table V1 1-18
 - create table V1 1-102
- Foreign keys V1 1-110
 - dropping V2 3-240
 - inserting V2 3-288 to V2 3-290
 - `sp_fkeys` information on V2 4-16 to V2 4-18
 - `sp_helpkey` and V2 3-332
 - `syskeys` table V3 8-48
- for load option

- alter database V1 1-7
 - create database V1 1-55, V1 1-57
 - with override V1 1-54
 - Formats
 - dates V3 7-20
 - times in named time ranges V2 3-75
 - Formats, date. *See* Dates
 - Format strings
 - print V1 1-304
 - raiserror V1 1-308
 - in user-defined error messages V1 1-308, V2 3-44
 - Formulas
 - max_rows_per_page of nonclustered indexes V2 3-168
 - for read only option, declare cursor V1 1-148
 - for update option, declare cursor V1 1-148
 - Fragments, device space
 - sp_placeobject and V2 3-419
 - Free pages, curunreservedpgs system
 - function V1 2-55
 - from keyword
 - delete V1 1-154
 - grant V1 1-236
 - load database V1 1-274
 - load transaction V1 1-283
 - select V1 1-340
 - sp_tables list of objects appearing in clause V2 4-38 to V2 4-39
 - update V1 1-386
 - Front-end applications, browse mode
 - and V1 2-158
 - Full name
 - changing with sp_modifylogin V2 3-389
 - specifying with sp_addlogin V2 3-41
 - full option
 - dbcc indexalloc V1 1-140
 - dbcc tablealloc V1 1-141, V1 1-142
 - Functions **V1 2-1**
 - aggregate V1 2-6
 - conversion V1 2-11
 - date V1 2-19
 - image V1 2-25
 - mathematical V1 2-20
 - security V1 2-22
 - string V1 2-22
 - system V1 2-23
 - text V1 2-25
 - futureonly option
 - sp_bindefault V2 3-114
 - sp_bindrule V2 3-125, V2 3-127
 - sp_unbindefault V2 3-516, V2 3-517
 - sp_unbindrule V2 3-524
 - Future space allocation. *See* Space allocation; sp_placeobject system procedure
- ## G
- German language print message
 - example V1 1-304
 - getdate date function **V1 2-77**
 - Getting messages. *See* sp_getmessage system procedure
 - Global allocation map pages V3 8-44
 - Global audit options, sysauditoptions system table V3 8-9
 - Global variables
 - See also individual variable names*
 - sp_monitor report on V2 3-405
 - goto keyword **V1 1-231**
 - Grammatical structure, numbered
 - placeholders and V1 1-304
 - Grand totals
 - compute V1 1-44
 - order by V1 1-298
 - grant command V1 1-50, **V1 1-232 to V1 1-244**
 - all keyword V1 1-232
 - drop role permission not included in V1 1-186
 - "public" group and V1 1-234
 - roles and V1 1-242
 - sysprotects table V3 8-71
 - Granting
 - create trigger permission V1 1-127, V1 1-240, V1 1-327
 - grant option

- sp_helprotect V2 3-346
 - sp_role V2 3-466
 - grant option for option, revoke V1 1-324
 - Greater than. *See* Comparison operators
 - Greek characters V3 A-16
 - group by clause **V1 1-245 to V1 1-257**
 - aggregate functions and V1 1-245, V1 1-248, V1 2-6, V1 2-8
 - having clause and V1 1-245 to V1 1-257
 - having clause and, in standard SQL V1 1-249
 - having clause and, in Transact-SQL V1 1-250
 - having clause and, sort orders V1 1-256
 - select V1 1-342 to V1 1-343
 - views and V1 1-134
 - without having clause V1 1-256
 - Grouping
 - multiple trigger actions V1 1-119
 - procedures of the same name V1 1-76, V1 1-183, V1 1-224
 - table rows V1 1-248
 - Groups
 - See also* "public" group
 - changing V2 3-146 to V2 3-147
 - dropping V2 3-238 to V2 3-239
 - grant and V1 1-243
 - information on V2 3-326
 - revoke and V1 1-328
 - sp_addgroup V2 3-33 to V2 3-34
 - sp_adduser procedure V2 3-88
 - sysusers table entries for V3 8-89
 - table rows V1 1-245
 - Windows NT domain V2 5-7
 - Guest users V1 2-165
 - master access required V2 3-271
 - permissions V1 1-243
 - sybssystemprocs database V2 3-10
 - tempdb access required V2 3-271
- H**
- Halloween problem V1 1-152
 - having clause **V1 1-245 to V1 1-257**
 - aggregate functions and V1 1-246, V1 1-248, V1 2-6
 - group by and V1 1-245 to V1 1-257
 - group by extensions in Transact-SQL and V1 1-250
 - negates all V1 1-246
 - select V1 1-343
 - Headings, column V1 1-246
 - in views V1 1-129
 - Help
 - sp_syntax display V2 3-501
 - sp_sysmon display V2 3-504
 - Help reports
 - See also* Information (Server); System procedures
 - constraints V2 3-312
 - database devices V2 3-320
 - database object V2 3-298
 - databases V2 3-317
 - datatypes V2 3-298
 - dump devices V2 3-320
 - extended stored procedures V2 3-322
 - groups V2 3-326
 - indexes V2 3-328
 - joins V2 3-330
 - keys V2 3-332
 - language, alternate V2 3-335
 - logins V2 3-340
 - permissions V2 3-346
 - remote servers V2 3-353
 - resource limits V2 3-342
 - segments V2 3-350
 - source text for compiled objects V2 3-357
 - system procedures V2 3-298 to V2 3-362
 - thresholds V2 3-359
 - users V2 3-361 to V2 3-362
 - Hexadecimal numbers
 - "0x" prefix for V1 1-59
 - converting V1 2-17
 - hextoint function V1 2-78, **V1 2-78**
 - hh. *See* hour date part
 - Hierarchy

- See also* Precedence
 data cache bindings V2 3-110
 datatype V3 8-85
 lock promotion thresholds V2 3-475
 operators V3 A-2
 roles, displaying with `sp_activeroles` V2 3-14
 user-defined datatypes V2 3-81
 Hierarchy of permissions. *See* Permissions
 Hierarchy of roles. *See* Role hierarchies
 Historic dates, pre-1753 V1 2-19, V1 2-60
 holdlock keyword
 readtext V1 1-314
 select V1 1-341, V2 3-371
 Host computer name V1 2-81
 host_id system function V1 2-80
 host_name system function V1 2-81
 Host process ID, client process V1 2-80
 hour date part V1 2-20, V1 2-66
 Hour values date style V1 2-48
- I**
- I/O
- configuring size V2 3-425
 - devices, disk mirroring to V1 1-164
 - displaying total actual cost (statistics io) V1 1-362
 - limiting V2 3-56
 - log size V2 3-385
 - prefetch and delete V1 1-155
 - prefetch and select V1 1-341
 - prefetch and update V1 1-386
 - usage statistics V2 3-462
- Identifiers V3 A-11 to V3 A-16
- delimited V2 3-154
 - quoted V2 3-154
 - renaming V2 3-155, V3 A-15
 - reserved words and V2 3-150 to V2 3-163
 - select V1 1-348
 - set quoted_identifier on V2 3-154, V2 3-161 to V2 3-162
 - `sp_checkreswords` and V2 3-154
 - system functions and V1 2-169
- Identities
- alternate V2 3-16
 - sa_role and Database Owner V1 1-369, V1 2-165
 - server user (suser_id) V1 2-152
 - set proxy and V1 1-371
 - set session authorization and V1 1-371
 - setuser command V1 1-374
 - user (user_id) V1 2-165
- identity burning set factor configuration parameter V1 1-265
- IDENTITY columns
- automatic V2 3-194, V2 3-197
 - creating tables with V1 1-116
 - database options using V2 3-195
 - defaults and V1 1-23
 - inserting values into V1 1-261
 - inserts into tables with V1 1-265 to V1 1-266
 - maximum value of V1 1-266
 - nonunique indexes V2 3-195
 - null values and V1 1-266
 - selecting V1 1-266, V1 1-351
 - updates not allowed V1 1-391
 - views and V1 1-134
- @@identity global variable V1 1-266
- identity in nonunique index database option setting with `sp_dboption` V2 3-195
- identity_insert option, set V1 1-358
- identity keyword
- alter table V1 1-14
 - create table V1 1-99
 - sp_addtype and V2 3-79
- Identity of user. *See* Aliases; Logins; Users
- @@idle global variable
- sp_monitor and V2 3-406
- IDs, server role
- role_id V1 2-125
 - sysroles table V3 8-77
- IDs, time range V2 3-76
- IDs, user

- See also* Logins
- database (**db_id**) V1 2-69
 - server user V1 2-152
 - stored procedure (**procid**) V1 1-360
 - user_id** function for V1 2-151
- if...else** conditions V1 1-258 to V1 1-260
- continue and V1 1-52
 - local variables and V1 1-147
- if update** clause, create trigger V1 1-118, V1 1-119, V1 1-125
- ignore_dup_key** option, create index V1 1-68
- ignore_dup_row** option, create index V1 1-68
- image** datatype V3 7-28, V3 7-33 to V3 7-38
- “0x” prefix for V3 7-37
 - initializing V3 7-34
 - length of data returned V1 1-349, V1 1-363
 - null values in V3 7-35
 - order by not allowed V1 1-298
 - pointer values in **readtext** V1 1-314
 - prohibited actions on V3 7-36
 - size of V2 3-499
 - storage on separate device V1 1-314
 - sysindexes** table and V3 8-45
 - triggers and V1 1-122
 - union not allowed on V1 1-384
 - writetext** to V1 1-414
- Image** functions V1 2-25
- Immediate** shutdown V1 1-376
- Impersonating** a user. *See* **setuser** command
- Implicit** conversion (of datatypes) V3 7-7, V3 A-10
- Inactive** transaction log space V1 1-209
- Included** groups, **group by** query V1 1-250
- indexalloc** option, **dbcc** V1 1-140
- index_col** system function V1 2-82
- Indexes**
- See also* Clustered indexes; Database objects; Nonclustered indexes
 - binding to data caches V2 3-109
 - checking name with **sp_checknames** V2 3-148
 - checking name with **sp_checkreswords** V2 3-153
 - composite V1 1-73
 - creating V1 1-65 to V1 1-75
 - dbcc indexalloc** and V1 1-140
 - dropping V1 1-181 to V1 1-182
 - estimating space and time requirements V2 3-273
 - IDENTITY** columns in nonunique V2 3-195
 - information about V2 3-328
 - integrity checks (**dbcc**) V1 1-141
 - joins and V1 1-69
 - key values V1 1-398
 - listing V1 1-181
 - max_rows_per_page** and V1 1-17, V1 1-101
 - naming V1 1-66
 - nonclustered V1 1-66
 - number allowed V1 1-70
 - Object Allocation Maps of V1 1-141
 - page allocation check V1 1-140
 - renaming V2 3-156, V2 3-454 to V2 3-457
 - space used by V2 3-500
 - sp_placeobject** space allocation for V2 3-418 to V2 3-420
 - sp_statistics** information on V2 4-30 to V2 4-32
 - suspect V2 3-365 to V2 3-366
 - sysindexes** table V3 7-35
 - system tables entries for V3 8-45
 - truncate** table and V1 1-379
 - types of V1 1-65 to V1 1-66
 - unbinding from data caches V2 3-511
 - update** statistics on V1 1-69, V1 1-398
 - views and V1 1-70
- Index** pages
- allocation of V1 2-118
 - fillfactor effect on V1 1-16, V1 1-66, V1 1-100

- leaf level V1 1-16, V1 1-65, V1 1-66, V1 1-100
- system functions V1 2-118
- total of table and V1 2-118
- Infected processes
 - removal with kill V2 3-537
 - waitfor errexit and V1 1-402
- Information (Server)
 - alternate languages V2 3-335
 - cache bindings V2 3-111
 - configuration parameters V3 8-33, V3 8-35
 - current locks V2 3-370
 - database devices V2 3-320
 - database objects V2 3-298
 - Database Owners V2 3-361 to V2 3-362
 - databases V2 3-317, V3 8-37 to V3 8-39
 - data caches V2 3-132
 - datatypes V2 3-298
 - display procedures V1 1-79
 - dump devices V2 3-320
 - extended stored procedures V2 3-322
 - first page of log V2 3-337
 - groups V2 3-326, V2 3-361 to V2 3-362
 - indexes V2 3-328
 - join columns V2 3-330
 - keys V2 3-332
 - languages V2 3-335
 - log device V2 3-337
 - logins V2 3-535 to V2 3-537
 - monitor statistics V2 3-405
 - performance V2 3-504
 - permissions V2 3-346
 - remote server logins V2 3-340
 - remote servers V2 3-353
 - resource limits V2 3-342
 - segments V2 3-350
 - server logins V2 3-535 to V2 3-537
 - server users V2 3-218
 - source text for compiled objects V2 3-357
 - space usage V1 1-73, V2 3-498
 - suspect indexes V2 3-365 to V2 3-366
 - text V1 1-86
 - thresholds V2 3-359
 - users, database V2 3-361 to V2 3-362
- Information messages (Server). *See* Error messages; Severity levels, error
- Initializing
 - disk reinit and V1 1-162, V1 1-169 to V1 1-170
 - disk space V1 1-160 to V1 1-163
 - text* or *image* columns V3 7-36
- init option
 - dump database V1 1-198
 - dump transaction V1 1-212
- in keyword
 - alter table and V1 1-18
 - check constraint using V1 1-115
 - in expressions V3 A-6
 - where V1 1-407
- In-memory map V1 1-8
- Input packets, number of V2 3-406
- insert command **V1 1-261 to V1 1-269**
 - create default and V1 1-59
 - IDENTITY columns and V1 1-265 to V1 1-266
 - null/not null columns and V1 1-133
 - triggers and V1 1-122, V1 1-125
 - update and V1 1-262
 - views and V1 1-133, V1 1-267 to V1 1-268
- inserted* table
 - triggers and V1 1-121, V1 1-123
- Inserting
 - automatic leading zero V3 7-29
 - spaces in text strings V1 2-141
- int* datatype **V3 7-9**
 - aggregate functions and V1 2-34, V1 2-150
- Integer data V3 7-9
 - in SQL V1 xxi, V3 A-1
- Integer datatypes, converting to V1 2-17
- Integer remainder. *See* Modulo operator (%)
- Integrity. *See* dbcc (Database Consistency Checker); Referential integrity

- Integrity of data
 - constraints V1 1-108
 - methods V1 1-109
 - Intent table locks V2 3-281, V2 3-371
 - Interfaces file
 - changing server names in V2 3-161
 - sp_addserver and V2 3-65
 - Intermediate display level for
 - configuration parameters V2 3-216
 - Internal datatypes of null columns V1 1-106, V3 7-7
 - See also* Datatypes
 - Internal structures
 - pages used for V1 2-118
 - Interval, automatic checkpoint V1 1-35
 - into keyword
 - fetch V1 1-228
 - insert V1 1-261
 - select V1 1-340, V1 1-349
 - union V1 1-381
 - inttohex function **V1 2-84**
 - @@io_busy* global variable
 - sp_monitor and V2 3-406
 - is not null keyword in expressions V3 A-7
 - is null keyword
 - in expressions V3 A-7
 - where V1 1-406
 - isnull system function V1 2-85
 - insert and V1 1-264
 - print and V1 1-306
 - select and V1 1-348
 - iso_1 character set V3 A-16
 - @@isolation* global variable V1 1-372
 - Isolation levels
 - catalog stored procedures V2 4-2
 - identity in nonunique index database option and V2 3-195
 - system procedures V2 3-10
 - isql utility command
 - See also* *Utility Programs* manual
 - approximate numeric datatypes and V3 7-14
 - is_sec_service_on security function V1 2-86
- J**
- Japanese character sets
 - object identifiers and V3 A-16
 - print message example V1 1-304
 - Joins
 - count or count(*) with V1 2-53
 - indexes and V1 1-69
 - information on V2 3-330
 - null values and V3 A-8
 - number of tables considered by optimizer V1 1-363
 - sp_commonkey V2 3-174
 - table groups and V1 1-252
- K**
- Keys, table V1 1-110
 - See also* Common keys; Indexes
 - dropping V2 3-240
 - information on V2 3-332
 - syskeys table V2 3-175, V2 3-288, V2 3-432, V3 8-48
 - Key values V1 1-398
 - Keywords **V3 B-1 to V3 B-5**
 - as identifiers V2 3-150
 - Transact-SQL V3 A-11, V3 B-1 to V3 B-2
 - kill command **V1 1-270 to V1 1-272**
 - sp_who and V2 3-536, V2 3-537
- L**
- Labels
 - dump volumes V1 1-204, V1 1-279, V1 1-290
 - goto label V1 1-231
 - @@langid* global variable V1 1-308
 - Language defaults V2 3-40
 - adding V2 3-35 to V2 3-39
 - changing user's V2 3-41

- language option, set V1 1-359
- Languages, alternate
 - alias for V2 3-472
 - changing names of V2 3-159, V2 3-161
 - checking with `sp_checkreswords` V2 3-154
 - date formats in V2 3-35
 - dropping V2 3-243 to V2 3-244
 - dropping messages in V2 3-247
 - information on V2 3-335
 - installing V2 3-35
 - official name V2 3-472
 - structure and translation V1 1-304
 - `syslanguages` table V2 3-335, V3 8-50
 - system messages and V1 1-359, V2 3-293
 - system tables entries for V3 8-50
 - user-defined messages V2 3-43
 - weekday order and V1 1-367, V1 2-67
 - without Language Modules V2 3-35
- Last-chance thresholds V1 2-88, V2 3-70, V2 3-401, V2 3-403
- `LASTONLINE` engine group V2 3-24
- `lct_admin` system function **V1 2-88**
- Leading blanks, removal with `ltrim` function V1 2-92
- Leading zeros, automatic insertion of V3 7-29
- Leaf levels of indexes
 - clustered index V1 1-16, V1 1-65, V1 1-66, V1 1-100
- Leaving a procedure. *See* `return` command
- Length
 - See also* Size
 - of columns V1 2-43
 - of expressions in bytes V1 2-58
- Less than. *See* Comparison operators
- Levels
 - nested procedures and V1 1-87, V1 1-227
 - nesting triggers V1 1-126
 - `@@nestlevel` V1 1-87
 - permission assignment V1 1-236
- like keyword
 - alter table and V1 1-18
 - check constraint using V1 1-115
 - searching for dates with V3 7-22
 - where V1 1-405
 - wildcard characters used with V3 A-18
- Limited days
 - modifying for time ranges V2 3-396
 - resource limit information on V2 3-342
 - specifying for time ranges V2 3-75
- Limited times
 - modifying for time ranges V2 3-396
 - resource limit information on V2 3-342
 - specifying for time ranges V2 3-75
- Limit types V2 3-55
 - elapsed time V2 3-55
 - I/O cost V2 3-55
 - modifying values V2 3-393
 - number of rows returned V2 3-55
 - specifying values V2 3-55
- Linkage, page. *See* Pages, data
- Linking users. *See* Alias, user
- Listing
 - database options V2 3-191
 - datatypes with types V3 7-5 to V3 7-6
 - devices V2 3-320
 - existing defaults V1 1-179
 - user group members V1 1-243
- listonly option
 - load database V1 1-275
 - load transaction V1 1-284
- Lists
 - catalog stored procedures V2 4-1
 - commands V1 1-1 to V1 1-5
 - datatypes V3 7-2
 - `dbcc` stored procedures V2 6-1
 - error return values V1 1-320
 - functions V1 2-2 to V1 2-5
 - reserved return status value V1 1-320
 - sort order choices and effects V1 1-299

- system extended stored
 - procedures V2 5-1
 - system procedures V2 3-1 to V2 3-10
 - system tables V3 8-1 to V3 8-3
- Literal character specification
 - like match string V3 A-20
 - quotes (" ") V3 A-10
- Literal values
 - datatypes of V3 7-4
 - null V3 A-8
- Load, database V1 1-273 to V1 1-281
 - across networks V1 1-279
 - Backup Server and V1 1-279
 - block size V1 1-274
 - commands used for V1 1-276
 - cross-platform not supported V1 1-277, V1 1-286
 - disk mirroring and V1 1-280
 - dismounting tapes after V1 1-274
 - file name, listing V1 1-275
 - header, listing V1 1-275
 - load striping V1 1-274
 - message destination V1 1-275, V1 1-290
 - new database V1 1-57
 - remote V1 1-279
 - restricting use V1 1-278, V1 1-288
 - restrictions V1 1-277
 - rewinding tapes after V1 1-274
 - size required V1 1-277
 - updates prohibited during V1 1-278
 - volume name V1 1-274
- Load, transaction log V1 1-282 to V1 1-291
 - commands used for V1 1-285
 - disk mirroring and V1 1-290
 - dismounting tape after V1 1-283
 - dump devices V1 1-283
 - file name, listing V1 1-284
 - header, listing V1 1-284
 - load striping V1 1-283
 - message destination V1 1-284
 - point-in-time recovery V1 1-285
 - rewinding tape after V1 1-283
 - until_time V1 1-285
 - volume name V1 1-283
- load database command V1 1-273 to V1 1-281
 - restrictions V1 1-277
- load transaction command V1 1-282 to V1 1-291
 - restrictions V1 1-286
- Local alias, language V2 3-472
- Localization
 - changing language names and files V2 3-161
- local option, sp_addserver V2 3-65
- Local servers V2 3-65
 - See also Remote servers; Servers
- Local variables
 - declare (name and datatype) V1 1-146
 - raiserror and V1 1-309
 - in screen messages V1 1-305
 - in user-defined error messages V1 1-309
- Location of new database V1 1-54
- lock|unlock option, sp_locklogin V2 3-374
- Locking
 - cache binding and V2 3-111
 - cache unbinding and V2 3-512
 - control over V2 3-370 to V2 3-373
 - logins V2 3-374
 - text for reads V1 1-314
- Lock promotion thresholds V2 3-474
- Locks
 - exclusive page V2 3-281
 - exclusive table V2 3-281
 - exclusive table and page V2 3-371
 - "FAM DUR" status V2 3-282
 - intent table V2 3-281, V2 3-371
 - page V2 3-281, V2 3-371
 - shared page V2 3-281, V2 3-371
 - shared table V2 3-281, V2 3-371
 - sp_familylock system procedure V2 3-280 to V2 3-282
 - sp_lock system procedure V2 3-370 to V2 3-373
 - system tables entries for V3 8-53

- types of V2 3-281, V2 3-371
- log10** mathematical function **V1 2-90**
- Logarithm, base 10 V1 2-90
- Log device
 - See also* Transaction logs
 - information V2 3-337
 - purging a V1 1-202
 - space allocation V1 1-57, V1 1-144, V1 1-161
- Logging
 - messages V2 3-91 to V2 3-92
 - select into V1 1-350
 - text or *image* data V1 1-414
 - triggers and unlogged operations V1 1-124
 - user-defined events V2 5-11
 - user-defined messages V2 3-43
 - writetext command V1 1-414
- Logical (conceptual) tables V1 1-121, V1 1-123
- Logical consistency. *See* dbcc (Database Consistency Checker)
- Logical device name V2 3-85, V2 3-209
 - disk mirroring V1 1-164
 - disk remirroring V1 1-171
 - disk unmirroring V1 1-174
 - new database V1 1-54
 - syslogs table V2 3-377
- Logical expressions V1 xx, V3 A-1
 - if...else V1 1-258
 - syntax V1 1-29, V3 A-2
 - truth tables for V3 A-9 to V3 A-10
 - when...then V1 1-31
- Logical reads (statistics io) V1 1-362
- Logins
 - See also* Remote logins; Users
 - accounting statistics V2 3-172, V2 3-462
 - adding to Servers V2 3-40 to V2 3-42
 - alias V2 3-16, V2 3-223
 - applying resource limits to V2 3-55
 - changing current database owner V2 3-143
 - char_convert setting for V1 1-357
 - disabling V1 1-377
 - dropping V2 3-245, V2 3-262
 - dropping resource limits from V2 3-254
 - information on V2 3-218, V2 3-340
 - locking V2 3-374 to V2 3-376
 - modifying accounts V2 3-389 to V2 3-391
 - modifying resource limits for V2 3-392
 - number of V2 3-407
 - options for remote V2 3-448
 - password change V2 3-415 to V2 3-417
 - “probe” V2 3-463, V3 8-56
 - remote V2 3-252 to V2 3-253, V2 3-262
 - resource limit information on V2 3-342
 - syslogins table V3 8-56 to V3 8-57
 - sysremotelogins table V2 3-51 to V2 3-54, V2 3-252, V2 3-262, V2 3-340, V3 8-75
 - unlocking V2 3-374 to V2 3-376
- log** mathematical function **V1 2-89**
- log on option
 - alter database V1 1-6
 - create database V1 1-54
 - create database, and sp_logdevice V2 3-377
- Logs. *See* Segments; Transaction logs
- Log segment
 - dbcc checktable report on V1 1-139
 - not on its own device V1 1-140
 - sp_helplog report on V2 3-337
 - sp_helpthreshold report on V2 3-359
- logsegment log storage
 - dropping V2 3-260
- log10** mathematical function **V1 2-90**
- Loops
 - break and V1 1-29
 - continue and V1 1-52
 - goto label V1 1-231
 - syslogs changes and infinite V3 8-58
 - trigger chain infinite V1 1-126

- while V1 1-29, V1 1-411
 - Lower and higher datatypes. *See* Precedence
 - Lower and higher roles. *See* Role hierarchies
 - Lowercase letters, sort order and V1 1-299
 - See also* Case sensitivity
 - lower string function V1 2-91
 - ltrim string function V1 2-92
- M**
- Machine ticks V2 3-406
 - Macintosh character set V3 A-16
 - Mail messages, Server
 - deleting V2 5-5
 - processing V2 3-434 to V2 3-436
 - reading V2 5-13 to V2 5-16
 - sending V2 5-17 to V2 5-20
 - starting session V2 5-21
 - stopping session V2 5-23
 - Mapping
 - databases V2 3-200
 - remote users V2 3-51
 - system and default segments V1 1-9
 - sysusages* table V3 8-87
 - Markers, user-defined. *See* Placeholders; Savepoints
 - master* database
 - See also* Recovery of *master* database
 - alter database and V1 1-7
 - backing up V1 1-215
 - checking with *sp_checkreswords* V2 3-153
 - create database and V1 1-56
 - disk init and V1 1-162
 - disk mirror and V1 1-165
 - disk refit and V1 1-168
 - disk reinit and V1 1-169
 - disk remirror and V1 1-171
 - disk unmirror and V1 1-175
 - dropping databases and V1 1-177
 - sp_dboption* and V2 3-193
 - system procedure tables V2 3-13
 - system tables V3 8-2 to V3 8-3
 - thresholds and V2 3-71, V2 3-402
 - transaction log purging V1 1-202, V1 1-215
 - Master device V1 1-7
 - Matching
 - See also* Pattern matching
 - name and table name V3 A-15
 - Mathematical functions V1 2-20 to V1 2-22
 - max aggregate function V1 2-93
 - max_rows_per_page option
 - alter table and V1 1-16
 - changing with *sp_relimit* V2 3-167
 - create index V1 1-67
 - create table V1 1-101
 - membership keyword, alter role V1 1-10
 - Memory
 - See also* Space
 - freeing from XP Server V2 3-291 to V2 3-292
 - mapping V2 3-200
 - releasing with *deallocate cursor* V1 1-145
 - used by configuration parameters V2 3-308
 - Memory pools
 - configuring V2 3-425
 - configuring asynchronous prefetch limits V2 3-429
 - configuring wash percentage V2 3-428
 - defaults V2 3-130
 - minimum size of V2 3-427
 - sp_logiosize* and V2 3-385
 - transaction logs and V2 3-428
 - Message output parameter, *sp_getmessage* V2 3-293
 - Messages
 - adding user-defined V2 3-43 to V2 3-45
 - catalog stored procedure V2 4-3

- dropping system with
 - sp_droplanguage V2 3-243
- dropping user-defined V2 3-247 to V2 3-248
- language setting for V1 1-359, V2 3-247, V2 3-293
- logging V2 3-91 to V2 3-92
- mathematical functions and V1 2-22
- number for V2 3-43, V2 3-91, V2 3-247, V2 3-293
- printing user-defined V1 1-304 to V1 1-307
- revoke V1 1-327
- screen V1 1-304 to V1 1-307
- sp_getmessage procedure V2 3-293 to V2 3-294
- sp_volchanged list V2 3-530 to V2 3-534
- sysmessages table V3 8-61
- system procedure V2 3-13
- sysusermessages table V2 3-43 to V2 3-45, V3 8-88
- trigger V1 1-122, V1 1-192
- unbinding with sp_unbindmsg V2 3-522 to V2 3-523
- user-defined V3 8-88
- mi. *See* minute date part
- Midnights, number of V1 2-63
- Migration
 - of system log to another device V1 1-162
 - of tables to clustered indexes V1 1-71, V1 1-107
- millisecond date part V1 2-20, V1 2-66
- Millisecond values, datediff results in V1 2-62
- min aggregate function **V1 2-95**
- Minus sign (-)
 - in integer data V3 7-9
 - subtraction operator V3 A-3
- minute date part V1 2-20, V1 2-66
- mirrorexit keyword
 - waitfor V1 1-401
- Mirroring. *See* Disk mirroring
- mirror keyword, disk mirror V1 1-146
- Mistakes, user. *See* Errors
- Mixed datatypes, arithmetic operations
 - on V3 A-3
- mm. *See* month date part
- model database
 - changing database options V2 3-193
 - copying the V1 1-56
 - user-defined datatypes in V3 7-39
- mode option, disk unmirror V1 1-174
- Modifying
 - configuration parameter display level V2 3-216
 - configuration parameters V2 3-177
 - databases V1 1-6
 - login accounts V2 3-389
 - named time ranges V2 3-396
 - resource limits V2 3-392
 - roles V1 1-10
 - tables V1 1-13
 - thresholds V2 3-400
- Modules, display syntax of V2 3-501
- Modulo operator (%) V3 A-3
- Money
 - default comma placement V3 7-15
 - symbols V3 A-11
- money datatype **V3 7-15, V3 7-19**
 - arithmetic operations and V3 7-15
- Monitoring
 - space remaining V2 3-69, V2 3-70, V2 3-401
 - system activity V2 3-405
- month date part V1 2-20, V1 2-66
- Month values
 - alternate language V2 3-35, V3 8-50
 - date part abbreviation and V1 2-20, V1 2-66
 - date style V1 2-48
 - short (abbreviated) V3 8-50
 - syslanguages table V3 8-50
- Moving
 - indexes V2 3-418
 - tables V2 3-418
 - transaction logs V2 3-377
 - user to new group V2 3-146

- MRU replacement strategy
 - disabling V2 3-139
 - ms. *See* millisecond date part
 - Multibyte character sets
 - changing to V1 1-140
 - converting V1 2-14
 - fix_text upgrade for V1 1-140, V1 1-144
 - identifier names V3 A-16
 - nchar datatype for V3 7-24
 - readtext and V1 1-316
 - readtext using characters for V1 1-316
 - sort order V2 3-356
 - sp_helpsort output V2 3-356
 - wildcard characters and V3 A-20
 - writetext and V1 1-416
 - Multicolumn index. *See* Composite indexes
 - Multiple trigger actions V1 1-119
 - Multiplication operator (*) V3 A-3
 - Multitable views V1 1-392
 - See also* Views
 - delete and V1 1-133, V1 1-156
 - mut_excl_roles system function V1 2-97
 - Mutual exclusivity of roles V1 1-10
 - mut_excl_roles and V1 2-97
- N**
- “N/A”, using “NULL” or V3 A-8
 - Named time ranges
 - adding V2 3-75
 - “at all times” V2 3-76, V2 3-266
 - changing active time ranges V2 3-77
 - creating V2 3-75
 - dropping V2 3-266
 - entire day V2 3-75
 - IDs for V2 3-76
 - modifying V2 3-396
 - overlapping V2 3-77
 - systemranges system table V3 8-84
 - Name of device
 - disk mirroring and V1 1-164
 - disk remirroring and V1 1-171
 - disk unmirroring and V1 1-174
 - dump device V1 1-196, V1 1-210
 - physical, disk reinit and V1 1-169
 - remote dump device V1 1-279
 - name option
 - disk init V1 1-160
 - disk reinit V1 1-169
 - Names
 - See also* Identifiers
 - alias V2 3-16, V2 3-223, V2 3-270
 - alias for table V1 1-341
 - assigning different, compared to aliases V2 3-88
 - changing database object V2 3-454 to V2 3-457
 - changing identifier V2 3-155
 - character set V3 8-27
 - checking with sp_checknames V2 3-148
 - checking with sp_checkreswords V2 3-150
 - checking with valid_name V3 A-15
 - column, in views V1 1-129
 - date parts V1 2-20, V1 2-66
 - db_name function V1 2-70
 - DLL file V2 3-291
 - finding similar-sounding V1 2-139
 - host computer V1 2-81
 - index_col and index V1 2-82
 - object_name function V1 2-101
 - omitted elements of (..) V3 A-14
 - parameter, in create procedure V1 1-77
 - qualifying database objects V3 A-13, V3 A-16
 - remote user V2 3-252
 - segment V1 1-17, V1 1-68, V1 1-101, V1 1-103, V2 3-63
 - server V2 3-65
 - server attribute V2 4-21
 - setuser V1 1-374
 - sorting groups of V1 1-256
 - sort order V3 8-27
 - suser_name function V1 2-152
 - user_name function V1 2-167
 - user’s full V2 3-40
 - user system function V1 2-164

- view V1 1-194
- weekday numbers and V1 2-67
- Naming
 - columns in views V1 1-129
 - conventions V3 A-11 to V3 A-16
 - cursors V1 1-149
 - database device V1 1-160
 - database objects V3 A-11 to V3 A-16
 - file V1 1-160
 - groups V2 3-33
 - identifiers V3 A-11 to V3 A-16
 - indexes V1 1-66
 - stored procedures V1 1-82
 - tables V1 1-97
 - temporary tables V1 1-106
 - time ranges V2 3-75
 - triggers V1 1-118
 - user-defined datatypes V2 3-81, V3 7-39
 - views V1 1-129
- National Character. *See nchar* datatype
- Natural logarithm V1 2-89
- nchar* datatype V3 7-24
 - @*ncharsize* global variable
 - sp_addtype* and V2 3-81
- Negative sign (-) in money values V3 7-15
- Nested select statements. *See select* command; Subqueries
- Nesting
 - aggregate functions V1 2-7
 - begin...end blocks V1 1-25
 - cursors V2 3-187
 - if...else conditions V1 1-259
 - levels V1 1-87
 - levels of triggers V1 1-126
 - stored procedures V1 1-82, V1 1-227
 - string functions V1 2-23
 - triggers V1 1-126
 - while loops V1 1-412
 - while loops, break and V1 1-30
 - @*nestlevel* global variable V1 1-227
 - nested procedures and V1 1-87
 - nested triggers and V1 1-126
- net password encryption option
 - sp_serveroption* V2 3-468
- %*nn!* (placeholder format) V1 1-304
- no chkpt on recovery database option
 - setting with *sp_dboption* V2 3-196
- nocount option, set V1 1-359
- nodismount option
 - dump database V1 1-197
 - dump transaction V1 1-211
 - load database V1 1-274
 - load transaction V1 1-283
- noexec option, set V1 1-359
- nofix option, dbcc
 - checkalloc and V1 1-139
 - indexalloc and V1 1-141
 - tablealloc and V1 1-142
- no free space acctg database option
 - setting with *sp_dboption* V2 3-196
- noholdlock keyword, select V1 1-342
- noinit option
 - dump database V1 1-198
 - dump transaction V1 1-212
- no_log option, dump transaction V1 1-209
- nonclustered constraint
 - alter table V1 1-15
 - create table V1 1-100
- Nonclustered indexes V1 1-66
- "none", using "NULL" or V3 A-8
- noserial option, disk mirror V1 1-164
- notify option
 - dump database V1 1-198
 - dump transaction V1 1-212
 - load database V1 1-275
 - load transaction V1 1-284
- not keyword
 - in expressions V3 A-6
 - where V1 1-404
- not like keyword V3 A-17
- not null keyword
 - create table V1 1-99
- Not null values
 - dropping defaults for V1 1-179
 - insert and V1 1-264
 - select statements and V1 1-348

- spaces in V3 7-26
- sp_addtype and V2 3-80
 - for user-defined data V2 3-80
 - views and V1 1-133
- no_truncate option, dump transaction V1 1-212
- nounload option
 - dump database V1 1-197
 - dump transaction V1 1-211
 - load database V1 1-274
 - load transaction V1 1-283
- nowait option, shutdown V1 1-376
- nullif keyword V1 1-31
- null keyword
 - alter table V1 1-14
 - create table V1 1-99
 - in expressions V3 A-7
- Null string in character columns V1 2-145, V3 A-8
- Null values
 - check constraints and V1 1-115
 - column datatype conversion for V3 7-26
 - column defaults and V1 1-61, V1 1-93
 - comparing V1 1-355
 - default parameters as V3 A-7
 - defining V1 1-61, V1 1-105
 - dropping defaults for V1 1-179
 - in expressions V3 A-7
 - group by and V1 1-248
 - inserting substitute values for V1 1-264
 - new column V1 1-14, V1 1-61
 - new rules and column definition V1 1-93
 - null defaults and V1 1-61, V1 1-93
 - select statements and V1 1-348
 - sort order of V1 1-299
 - sp_addtype and V2 3-79
 - stored procedures cannot return V1 1-320
 - text and image columns V1 1-264, V3 7-35
 - triggers and V1 1-125
 - for user-defined datatypes V2 3-79
- Number (quantity of)
 - active dumps or loads V1 1-203, V1 1-218, V1 1-279, V1 1-289
 - and and or conditions allowed V1 1-409
 - arguments, in a where clause V1 1-410
 - arguments and placeholders V1 1-306
 - bytes in returned text V1 1-315
 - bytes per row V1 1-20, V1 1-105
 - clustered indexes V1 1-66
 - databases reported by
 - sp_countmetadada V2 3-184
 - databases Server can manage V1 1-56
 - device fragments V1 1-8, V1 1-56
 - different triggers V1 1-122
 - first-of-the-months V1 2-63
 - groups per user V2 3-146
 - having clause search arguments V1 1-246
 - indexes V2 3-184
 - logical reads (statistics io) V1 1-362
 - messages per constraint V2 3-122
 - midnights V1 2-63
 - named segments V1 1-56, V2 3-64
 - nesting levels V1 1-87
 - nesting levels, for triggers V1 1-126
 - nonclustered indexes V1 1-66, V1 1-70
 - open objects V2 3-184
 - parameters in a procedure V1 1-147
 - physical reads (statistics io) V1 1-362
 - placeholders in a format string V1 1-306
 - rows in count(*) V1 2-53
 - rows reported by rowcnt V1 2-130
 - scans (statistics io) V1 1-362
 - stored procedure parameters V1 1-82
 - Sundays V1 2-63
 - tables allowed in a query V1 1-340
 - tables per database V1 1-105
 - timestamp columns V3 7-17
 - updates V1 1-127
 - user-defined roles V1 1-89
- Number of characters

- in a column V3 7-24
 - date interpretation and V3 7-22
 - Number of columns
 - in an order by clause V1 1-298
 - per table V1 1-20, V1 1-105
 - in a view V1 1-132
 - Number of pages
 - allocated to table or index V1 2-118
 - in an extent V1 1-70, V1 1-105
 - reserved_pgs function V1 2-118
 - statistics io and V1 1-362
 - used by table and clustered index (total) V1 2-163
 - used by table or index V3 8-46
 - used_pgs function V1 2-163
 - written (statistics io) V1 1-362
 - Numbers
 - See also* IDs, user
 - asterisks (*) for overlength V1 2-144
 - converting strings of V3 7-27
 - database ID V1 2-69
 - datatype code V2 4-4
 - device V2 3-321
 - error return values (Server) V1 1-319
 - global variable unit V2 3-406
 - message V2 3-43, V2 3-91, V2 3-247, V2 3-293
 - object ID V1 2-99
 - ODBC datatype code V2 4-4
 - odd or even binary V3 7-29
 - placeholder (%nn!) V1 1-304
 - procid setting V1 1-360
 - random float V1 2-114
 - same name group procedure V1 1-76, V1 1-183, V1 1-224
 - select list V1 1-344
 - spid (server process ID) V2 3-535
 - statistics io V1 1-362
 - virtual device V1 1-160, V1 1-163, V1 1-169
 - weekday names and V1 1-358, V1 2-67, V2 3-35
 - Numeric data
 - row aggregates and V1 2-9
 - numeric datatype V3 7-10
 - range and storage size V3 7-2
 - Numeric expressions V1 xxi, V3 A-1
 - round function for V1 2-128
 - nvarchar datatype V3 7-24 to V3 7-25
 - spaces in V3 7-24
- O**
- Object Allocation Map (OAM) pages V1 2-163
 - dbcc indexalloc and V1 1-141
 - dbcc report on table V1 1-142
 - object_id system function V1 2-99
 - Object names, database
 - See also* Identifiers
 - checking with sp_checknames V2 3-148
 - checking with sp_checkreswords V2 3-153
 - as parameters V1 1-77
 - in stored procedures V1 1-86, V1 1-87
 - user-defined datatype names as V3 7-39
 - object_name system function V1 2-101
 - Object owners. *See* Database object owners
 - Object permissions
 - See also* Command permissions; Permissions
 - grant V1 1-232 to V1 1-244
 - grant all V1 1-241
 - Objects. *See* Database objects; Databases
 - ODBC. *See* Open Database Connectivity (ODBC) API datatypes
 - Official language name V2 3-36, V2 3-472
 - See also* Aliases; Languages, alternate
 - Offset position, readtext command V1 1-314
 - offsets option, set V1 1-359
 - of option, declare cursor V1 1-148
 - on keyword
 - alter database V1 1-6
 - alter table V1 1-17

- create database V1 1-54
- create index V1 1-68, V1 1-71
- create table V1 1-101, V1 1-103
- online database command V1 1-278, **V1 1-292 to V1 1-293**
 - bringing databases online V1 1-278
 - dump transaction and V1 1-286
 - load transaction and V1 1-286
 - upgrades and V1 1-289
- Open Client applications
 - connection security with V2 3-53
 - keywords V1 1-359
 - procid setting V1 1-360
 - set options for V1 1-359, V1 1-367
- open command **V1 1-294 to V1 1-295**
- Open Database Connectivity (ODBC)
 - API datatypes V2 4-4
- Opening cursors V1 1-294
- OpenVMS systems
 - contiguous option on V1 1-164
 - mirroring options V1 1-164
- Operating system commands V2 5-3
- Operators
 - arithmetic V3 A-3
 - bitwise V3 A-3 to V3 A-4
 - comparison V3 A-5
 - precedence V3 A-2
- Optimization
 - queries (sp_recompile) V2 3-443
- optimized report
 - dbcc indexalloc V1 1-140, V1 1-141
 - dbcc tablealloc V1 1-142
- Optimizer
 - join selectivity V1 1-363
- Options
 - See also* Configuration parameters
 - database V2 3-191 to V2 3-199
 - remote logins V2 3-448 to V2 3-450
 - remote servers V2 3-468 to V2 3-471
- @@options** global variable V1 1-372
- Order
 - See also* Indexes; Precedence; Sort order
- of arguments in translated strings V1 1-304
- ascending sort V1 1-296, V1 1-344
- of column list and insert data V1 1-261
- of columns (fixed- and variable-length) V1 1-300
- of creating indexes V1 1-70
- of date parts V1 1-358, V2 3-35, V3 7-21
- descending sort V1 1-296, V1 1-344
- error message arguments V1 1-304
- of evaluation V1 1-382
- of execution of operators in expressions V3 A-3
- of names in a group V1 1-256
- of null values V1 1-299
- of parameters in create procedure V1 1-224, V1 1-226
- reversing character expression V1 2-120
- for unbinding a rule V1 1-92
- weekday numeric V1 2-67
- order by** clause **V1 1-296 to V1 1-302**
 - compute by and V1 1-44, V1 1-298, V1 1-344
 - select and V1 1-344
- Order of commands V1 1-239, V1 1-327
- Original identity, resuming an (setuser command) V1 1-374
- or keyword
 - in expressions V3 A-9
 - number allowed in search conditions V1 1-409
 - where V1 1-408
- Other users, qualifying objects owned by V3 A-16
- Output
 - dbcc V1 1-144
 - packets, number of V2 3-406
 - zero-length string V1 1-306
- output option
 - create procedure V1 1-78, V1 1-224, V1 1-226

- execute V1 1-224
 - return parameter V1 1-224
 - sp_getmessage V2 3-293
 - Overflow errors
 - DB-Library V1 2-34, V1 2-150
 - set arithabort and V1 1-356
 - Overhead
 - data caches V2 3-136
 - triggers V1 1-123
 - Overlapping time ranges V2 3-77
 - Override. *See with override option*
 - Overwriting triggers V1 1-122, V1 1-192
 - Owners. *See Database object owners;*
Database Owners
 - Ownership
 - See also Permissions; setuser command*
 - of command and object
 - permissions V1 1-236
 - dump devices and V2 3-86
 - of objects being referenced V3 A-16
 - of rules V1 1-93
 - of stored procedures V1 1-88
 - of triggers V1 1-128
 - of views V1 1-136
- P**
- @@packet_errors* global variable
 - sp_monitor and V2 3-406
 - @@pack_received* global variable
 - sp_monitor and V2 3-406
 - @@pack_sent* global variable
 - sp_monitor and V2 3-406
 - Padding, data
 - blanks and V1 1-263, V3 7-24
 - image datatype V3 7-37
 - underscores in temporary table
 - names V3 A-12
 - with zeros V3 7-28
 - Page locks
 - types of V2 3-281, V2 3-371
 - Pages, control
 - syspartitions and V3 8-66
 - updating statistics on V1 1-396
 - Pages, data
 - See also Index pages; Table pages*
 - allocation of V1 2-118
 - chain of V1 1-19, V1 1-22 to V1 1-23,
V3 7-33
 - computing number of, with
 - sp_spaceused V2 3-499
 - extents and V1 1-71, V1 1-105
 - extents and dbcc tablealloc V1 1-142
 - extents reported by dbcc indexalloc V1
1-141
 - locks held on V2 3-281, V2 3-371
 - multibyte characters and V1 1-140
 - reserved_pgs system function V1 2-118
 - statistics io and V1 1-362
 - used for internal structures V1 2-118
 - used in a table or an index V1 2-163
 - used_pgs system function V1 2-163
 - Pages, global allocation map V3 8-44
 - Pages, index
 - number used in nonclustered V1
2-163
 - truncate table and V1 1-379
 - Pages, OAM (Object Allocation Map)
 - dbcc indexalloc report on V1 1-141
 - dbcc report on table V1 1-142
 - number of V1 2-163
 - Pages, overflow
 - descending scans and V1 1-301
 - Page splits V1 1-17, V1 1-67, V1 1-101
 - Pair, mirrored V1 1-174
 - Pair of columns. *See Common keys; Joins*
 - @@parallel_degree* global variable V1
1-372
 - set parallel_degree and V1 1-360
 - parallel_degree option, set command V1
1-359
 - parallel keyword, select command V1
1-341
 - Parameters, procedure
 - datatypes V1 1-77
 - defaults V1 1-77
 - execute and V1 1-224
 - naming V1 1-77

- not part of transactions V1 1-227
- ways to supply V1 1-224, V1 1-226, V2 3-11, V2 4-2
- Parentheses ()
 - See also Symbols section of this index*
 - in an expression V3 A-10
 - in SQL statements V1 xviii
 - in user-defined datatypes V2 3-79
- parseonly option, set V1 1-360
- Partial characters, reading V1 1-316
- partition clause, alter table command V1 1-19
- Partitioned tables
 - alter table V1 1-19
 - size of V1 2-110
- Partition statistics
 - updating with update partition statistics V1 1-396
 - updating with update statistics V1 1-394
- Passthrough mode
 - connect to command V1 1-49
 - sp_autoconnect system procedure V2 3-107
 - sp_passthru system procedure V2 3-412
 - sp_remotesql system procedure V2 3-451
- passwd keyword, alter role V1 1-10
- Passwords
 - adding to user-defined roles V1 1-11
 - changing for user-defined roles V1 1-12
 - date of last change V2 3-219
 - dropping from user-defined roles V1 1-11
 - encryption over network V2 3-469
 - roles and V1 1-10
 - setting with sp_addlogin V2 3-40
 - sp_password V2 3-415 to V2 3-417
 - sp_remotoption and V2 3-448
 - sp_serveroption and V2 3-469
 - trusted logins or verifying V2 3-448
 - user-defined roles and V1 1-89, V1 1-361
- Path name
 - DLL and extended stored procedures V1 1-78
 - dump device V2 3-85
 - hard-coded or logical device V1 1-162
 - mirror device V1 1-164
 - remote dump device V1 1-279
- patindex string function **V1 2-103**
- text/image function V3 7-37
- Pattern matching V3 A-16
 - See also String functions; Wildcard characters*
 - catalog stored procedure parameters V2 4-3
 - charindex string function V1 2-39
 - difference string function V1 2-72
 - patindex string function V1 2-104
- PC DB-Library. *See* DB-Library
- programs
- Percent sign (%)
 - error message placeholder V1 1-304
 - literal in error messages V1 1-306
 - modulo operator V3 A-3
 - wildcard character V3 A-18
- Performance
 - information V2 3-504
 - select into and V1 1-350
 - showplan and diagnostics V1 1-362
 - sort_resources and diagnostics V1 1-362
 - triggers and V1 1-123
 - writetext during dump database V1 1-416
- Period (.)
 - preceding milliseconds V1 2-66
 - separator for qualifier names V3 A-13
- Permissions
 - assigned by Database Owner V1 1-232
 - assigning V1 1-232
 - changing with setuser V1 1-374
 - command V1 1-237 to V1 1-239
 - creating and executing procedures V1 1-87
 - creating and using views V1 1-136
 - creating with create schema V1 1-95 to V1 1-96

- displaying user's V2 3-218
- dump devices and V2 3-86
- for creating triggers V1 1-127, V1 1-240, V1 1-327
- grant V1 1-232 to V1 1-244
- granting V2 3-347
- information on V2 3-346
- new Database Owner V2 3-143
- new database user V2 3-390
- object V1 1-238
- "public" group V1 1-237 to V1 1-239
- revoke command V1 1-322 to V1 1-329
- revoking V2 3-347
- sp_column_privileges information on V2 4-5 to V2 4-7
- sysprotects table V3 8-71
- system procedures V2 3-10
- system tables V3 8-3
- system tables entries for V3 8-71
- Physical database consistency. *See* dbcc (Database Consistency Checker)
- Physical datatypes V2 3-79
- Physical device name V2 3-85
- Physical reads (statistics io) V1 1-362
- physname option
 - disk init V1 1-160
 - disk init in OpenVMS V1 1-162
 - disk reinit V1 1-169
- pi mathematical function **V1 2-106**
- Placeholders
 - error message percent sign (%) V2 3-44
 - print message V1 1-304
- Plan
 - create procedure and V1 1-78
 - object V3 8-67
 - set showplan on and V1 1-362
 - set sort_resources on and V1 1-362
 - sp_showplan output V2 3-492
- Platform-independent conversion
 - hexadecimal strings to integer values V1 2-78
 - integer values to hexadecimal strings V1 2-84
- Plus (+)
 - arithmetic operator V3 A-3
 - in integer data V3 7-9
 - null values and V3 A-5
 - string concatenation operator V3 A-5
- Pointers
 - null for uninitialized *text* or *image* column V1 2-154
 - text and image page V1 2-154
 - text* or *image* column V1 1-314, V3 7-34, V3 7-38
- Pointers, device. *See* Segments
- Pools, memory
 - configuring V2 3-425
 - defaults V2 3-130
- Pound sign (#) temporary table name prefix V1 1-98
- Pound sterling sign (£)
 - in identifiers V3 A-11
 - in money datatypes V3 7-15
- power mathematical function **V1 2-107**
- Precedence
 - binding defaults to columns and datatypes V2 3-115
 - of lower and higher datatypes V3 A-10
 - of operators in expressions V3 A-2
 - order-sensitive commands and V1 1-239, V1 1-327
 - resource limits V2 3-58
 - rule binding V1 1-93, V2 3-126
 - of user-defined return values V1 1-320
- Preceding blanks. *See* Blanks; Spaces, character
- Precision, datatype
 - approximate numeric types V3 7-14
 - exact numeric types V3 7-10
 - money types V3 7-15
 - sp_help report on V2 3-300
 - user-defined datatypes V2 3-79
- Preference, uppercase letter sort order V1 1-299
- Prefetch

- disabling V2 3-139
- enabling V2 3-139
- prefetch keyword
 - delete V1 1-155
 - select V1 1-341
 - set V1 1-360
 - update V1 1-386
- prepare transaction command **V1 1-303**
- primary key constraint
 - alter table V1 1-15
 - create table V1 1-100
- Primary keys V1 1-110
 - sp_dropkey procedure V2 3-240
 - sp_foreignkey and V2 3-288
 - sp_helpkey and V2 3-332
 - sp_primarykey definition of V2 3-432
 - syskeys table V3 8-48
 - updating V1 1-121
- primary option, disk unmirror V1 1-174
- print command **V1 1-304** to **V1 1-307**
 - local variables and V1 1-147
 - using raiserror or V1 1-306
- Printing user-defined messages V1 1-304 to V1 1-307
- Priority
 - sp_setpsexe V2 3-478
- Privileges. *See* Permissions
- "probe" login account V2 3-463, V3 8-56
- Probe Process, Two-Phase Commit V2 3-463, V3 8-56
- Procedure groups V1 1-183, V1 1-224
- Procedure plan, create procedure and V1 1-78
- Procedures. *See* Stored procedures; System procedures
- Processes (Server tasks)
 - See also* Servers
 - checking locks on V2 3-280 to V2 3-282, V2 3-370 to V2 3-373
 - ID number V1 1-270, V2 3-535
 - infected V2 3-537
 - infected, waitfor errexit V1 1-402
 - killing V1 1-270 to V1 1-272
 - sp_showplan display of V2 3-492 to V2 3-494
 - sp_who report on V1 1-270, V2 3-535 to V2 3-537
 - sysprocesses table V3 8-68
 - system tables entries for V3 8-68
 - processexit keyword, waitfor V1 1-401
 - process_limit_action option, set V1 1-360
 - Process logical name. *See* Logical device name
 - procid option, set V1 1-360
 - proc_role system function **V1 2-108**
 - Promotion, lock V2 3-474
 - Prompts, sp_volchanged V2 3-530 to V2 3-534
 - Protection system
 - command and object permissions V1 1-236
 - groups V2 3-33
 - hierarchy of roles, groups and users V1 1-243
 - locking logins V2 3-374
 - stored procedures V1 1-87
 - user-defined roles V1 1-89
 - proxy option, set V1 1-360
 - granting V1 1-233
 - revoking V1 1-323
 - ptn_data_pgs system function **V1 2-110**
 - "public" group V1 1-243, V1 1-328, V3 8-89
 - See also* Groups
 - grant and V1 1-234
 - information report V2 3-326
 - permissions V1 1-237 to V1 1-239
 - revoke and V1 1-324
 - sp_addgroup and V2 3-33
 - sp_adduser and V2 3-88
 - sp_changegroup and V2 3-146
 - sp_dropgroup and V2 3-238
 - sp_helpgroup report on V2 3-326
 - public keyword
 - grant V1 1-234
 - revoke V1 1-324
- Punctuation

- characters allowed in identifiers V3 A-11
 - enclosing in quotation marks V2 3-11, V2 4-2
 - in user-defined datatypes V2 3-79
- Q**
- qq. *See* quarter date part
 - Qualifier names V3 A-13, V3 A-16
 - quarter date part V1 2-20, V1 2-66
 - Queries
 - compilation and optimization V2 3-443
 - compilation without execution V1 1-359, V1 1-360
 - execution settings V1 1-354 to V1 1-373
 - keywords list V1 1-359
 - sp_tables and V2 4-38
 - syntax check (set parseonly) V1 1-360
 - trigger firing by V1 1-121
 - union V1 1-381 to V1 1-384
 - views and V1 1-132
 - with/without group by and having V1 1-248
 - Query analysis
 - set noexec V1 1-359
 - set statistics io V1 1-362
 - set statistics time V1 1-362
 - Query plans
 - recompiling with sp_recompile V2 3-443
 - set showplan on and V1 1-362
 - Query processing
 - limiting with sp_add_resource_limit V2 3-55
 - modes V2 3-437 to V2 3-439
 - set options for V1 1-354
 - Question marks (??)
 - for partial characters V1 1-316
 - Quotation marks (" ")
 - comparison operators and V3 A-6
 - for empty strings V3 A-9, V3 A-10
 - enclosing constant values V1 2-23
 - enclosing *datetime* values V3 7-19
 - enclosing parameter values V2 3-11, V2 4-2
 - enclosing reserved words V2 3-155
 - in expressions V3 A-10
 - literal specification of V1 1-409, V3 A-10
 - single, and quoted_identifier V2 3-162
 - quoted_identifier option, set V1 1-360
 - Quoted identifiers
 - testing V2 3-155
 - using V2 3-154, V2 3-161 to V2 3-162
- R**
- Radians, conversion to degrees V1 2-71
 - radians mathematical function **V1 2-112**
 - raiserror command **V1 1-308 to V1 1-313**
 - compared to print V1 1-312
 - local variables and V1 1-147
 - using print or V1 1-306
 - rand mathematical function **V1 2-114**
 - Range
 - See also* Numbers; Size
 - datediff results V1 2-62
 - of date part values V1 2-20, V1 2-66
 - errors in mathematical functions V1 2-21
 - of money values allowed V3 7-15
 - of recognized dates V3 7-19
 - set rowcount V1 1-361
 - specifying for resource limits V2 3-55
 - wildcard character specification of V3 A-19, V3 A-20
 - Range queries
 - and end keyword V3 A-7
 - between start keyword V3 A-7
 - Read-only cursors V1 1-151
 - read only database option
 - setting with sp_dboption V2 3-196
 - setting with sp_setsuspect_granularity V2 3-481
 - readonly option, sp_serveroption V2 3-468

- readtext command **V1 1-314 to V1 1-316**
 - text data initialization requirement V3 7-36
- real datatype **V3 7-14**
- Rebuilding
 - automatic, of nonclustered index V1 1-70
 - indexes V1 1-141
 - system tables V1 1-141, V1 1-142
- Recompilation
 - create procedure with recompile option V1 1-78, V1 1-82
 - execute with recompile option V1 1-224
 - stored procedures V1 1-83, V2 3-443 to V2 3-444
- reconfigure command **V1 1-317**
- Records, audit V2 3-18
- Recovery
 - data caches and V2 3-132
 - displaying mode V2 3-481
 - dump transaction and V1 1-217
 - forcing suspect pages online with *sp_forceonline_db* V2 3-283
 - forcing suspect pages online with *sp_forceonline_page* V2 3-285
 - listing offline pages V2 3-368
 - listing suspect databases V2 3-367
 - setting mode V2 3-481
 - setting threshold V2 3-485
 - to specified time in transaction log V1 1-287
 - time and checkpoint V1 1-35
- Recovery of *master* database V1 1-202
 - after using create database V1 1-56
 - after using disk init V1 1-162
- Re-creating
 - indexes V1 1-141
 - procedures V1 1-85
 - tables V1 1-189
- Recursions, limited V1 1-127
- Reference information
 - catalog stored procedures V2 4-1
 - datatypes V3 7-1
 - dbcc stored procedures V2 6-1
 - reserved words V3 B-1
 - system extended stored procedures V2 5-1
 - system procedures V2 3-1 to V2 3-13
 - system tables V3 8-1
 - Transact-SQL commands V1 1-1 to V1 1-5
 - Transact-SQL functions V1 2-1
- references constraint
 - alter table V1 1-17
 - create table V1 1-102
- Referencing, object. *See* Dependencies, database object
- Referential integrity
 - triggers for V1 1-118 to V1 1-128
- Referential integrity constraints V1 1-112
 - binding user messages to V2 3-122
 - create table and V1 1-108
 - cross-database V1 1-114, V1 1-190
 - renaming V2 3-454 to V2 3-457
 - sysconstraints* table V3 8-34
 - sysobjects* table V3 8-63 to V3 8-64
 - sysreferences* table V3 8-73
- Regulations
 - for finding objects V2 3-207, V2 3-301
 - sort order ties V1 1-299 to V1 1-300
- reindex option, dbcc V1 1-141
 - after *sp_indsuspect* V2 3-365
- Reinitializing, disk reinit and V1 1-169 to V1 1-170
- Relational expressions V3 A-2
 - See also* Comparison operators
- Remapping database objects V2 3-445 to V2 3-447
- Remirroring. *See* Disk mirroring
- Remote logins
 - See also* Logins; Users
 - dropping V2 3-252 to V2 3-253
 - information on V2 3-340
 - sp_remotoption* for V2 3-448 to V2 3-450
 - sysremotelogins* table V2 3-51 to V2 3-54, V3 8-75

- system tables entries for V3 8-75
- trusted or untrusted mode V2 3-448
- Remote procedure calls V1 1-349
 - execute and V1 1-227
 - rollback and V1 1-331
 - sp_password V2 3-416
 - sysremotelogins table and V3 8-75
 - sys.servers table and V3 8-80
- Remote servers V1 1-349
 - See also* Servers
 - changing names of V2 3-158, V2 3-161
 - dropping logins V2 3-252
 - information on V2 3-353
 - information on logins of V2 3-340
 - names of V2 3-65
 - passwords on V2 3-416
 - sp_remotoption and V2 3-448 to V2 3-450
 - sys.servers table V3 8-80
 - system tables entries for V3 8-80
- Remote users. *See* Remote logins
- remove option, disk unmirror V1 1-174
- Removing. *See* Dropping
- Renaming V2 3-454 to V2 3-457
 - See also* sp_rename system procedure
 - a database V2 3-458 to V2 3-461
 - identity of object owner V1 1-236
 - stored procedures V1 1-82
 - triggers V1 1-123
 - views V1 1-132
 - warnings about V2 3-455, V2 3-459
- Repairing a damaged database V1 1-140
- Repeated execution. *See* while loop
- replace keyword, alter table V1 1-18
- Replacing user-defined messages V2 3-43
- replicate string function V1 2-116
- Reporting from dbccdb database
 - allocation statistics V2 6-30
 - comprehensive information V2 6-26
 - configuration information V2 6-7, V2 6-23, V2 6-26
 - fault information V2 6-17, V2 6-23
 - full details V2 6-26
- I/O statistics V2 6-17
- Reports
 - sp_who V1 1-270, V2 3-535 to V2 3-537
 - types of dbcc V1 1-142
- Reserved columns V3 8-3
- reserved_pgs system function V1 2-118
- Reserved return status values V1 1-319
- Reserved words V3 B-1 to V3 B-5
 - See also* Keywords
 - catalog stored procedures and V2 4-2
 - database object identifiers and V3 A-11
 - as identifiers V2 3-150 to V2 3-163
 - SQL92 V3 B-3
 - system procedures and V2 3-11
 - Transact-SQL V3 B-1 to V3 B-2
- Resource limits
 - creating V2 3-55
 - dropping V2 3-254
 - information about V2 3-342
 - modifying V2 3-392
 - sysresourcelimits table V3 8-76
 - types of V2 3-55
- Restarting while loops V1 1-52
- Restarts, Server
 - after using disk refit V1 1-168
 - before using create database V1 1-55
 - using dataserver utility V1 1-166, V1 1-172
- Restoring
 - See also* Recovery
 - a damaged master database V1 1-168, V1 1-169
 - database with load database V1 1-273 to V1 1-281
- Results
 - See also* Output
 - of aggregate operations V1 1-248
 - cursor result set V1 1-151, V1 1-228
 - order by and sorting V1 1-296 to V1 1-302
 - of row aggregate operations V1 2-9
- retaindays option
 - dump database V1 1-197

- dump transaction** V1 1-211
- retain option, disk unmirror** V1 1-174
- Retrieving**
 - error message text V1 1-304, V2 3-293
 - similar-sounding words or names V1 2-139
- return command** V1 1-318 to V1 1-321
- Return parameters**
 - output keyword V1 1-78, V1 1-224
- Return status**
 - catalog stored procedures V2 4-2
 - sp_checkreswords V2 3-153
 - stored procedure V1 1-223, V1 1-318
 - system procedures V2 3-10
- reverse string function** V1 2-120
- revoke command** V1 1-322 to V1 1-329
 - object and command permissions V1 1-237
 - "public" group and V1 1-324
 - sysprotects table V3 8-71
- revoke option, sp_role** V2 3-466
- Revoking**
 - create trigger permission V1 1-127, V1 1-240, V1 1-327
 - role privileges using with override V1 1-185
- Right-justification of str function** V1 2-144
- right string function** V1 2-122
- role_contain system function** V1 2-123
- Role hierarchies**
 - displaying with sp_activeroles V2 3-14
 - displaying with sp_displayroles V2 3-221
 - role_contain and V1 2-123
- role_id system function** V1 2-125
- role_name system function** V1 2-127
- role option**
 - grant V1 1-234
 - revoke V1 1-324
 - set command V1 1-361
- Roles**
 - adding passwords V1 1-10
 - checking with proc_role V1 2-108
 - creating (user-defined) V1 1-89
 - displaying with sp_activeroles V2 3-14
 - granting V1 1-242
 - permissions and V1 1-243
 - showing system with show_role V1 2-133
 - stored procedure permissions and V1 1-242
 - sysroles table V3 8-77
 - sysserverroles table V3 8-82
 - system, in sysloginroles table V3 8-55
 - turning on and off with set role V1 1-361
- Roles, system**
 - revoking V1 1-324
- Roles, user-defined**
 - limitations V1 1-89
 - mutual exclusivity and V1 2-97
 - revoking V1 1-324
 - turning on and off V1 1-361
- rollback command** V1 1-330 to V1 1-332
 - begin transaction and V1 1-27
 - commit and V1 1-39
 - triggers and V1 1-124, V1 1-126
- rollback transaction command.** *See* rollback command
- rollback trigger command** V1 1-124, V1 1-333 to V1 1-334
- rollback work command.** *See* rollback command
- Rolling back processes**
 - checkpoint and V1 1-35
 - parameter values and V1 1-227
- Rounding** V1 2-128
 - approximate numeric datatypes V3 7-14
 - datetime values V1 2-15
 - money values V1 2-14, V3 7-15
 - str string function and V1 2-143
- round mathematical function** V1 2-128
- Row aggregates** V1 2-9
 - compute and V1 1-40, V1 2-8
 - difference from aggregate functions V1 2-10

- rowcnt system function **V1 2-130**
 - @@rowcount* global variable V1 1-372
 - cursors and V1 1-230
 - set nocount and V1 1-372
 - triggers and V1 1-125
 - rowcount option, set V1 1-361
 - Rows, table
 - See also* select command
 - aggregate functions applied to V1 1-248
 - comparison order of V1 1-300
 - computing number of, with *sp_spaceused* V2 3-499
 - create index and duplication of V1 1-65, V1 1-68
 - deleting with truncate table V1 1-379
 - detail and summary results V1 2-9
 - displaying command-affected V1 1-359
 - grouping V1 1-245
 - insert V1 1-262
 - limiting how many returned V2 3-56
 - number of V1 2-130
 - row aggregates and V1 2-9
 - rowcount setting V1 1-361
 - scalar aggregates applied to V1 1-248
 - update V1 1-385
 - ways to group V1 1-248
 - rpc security model A option, *sp_serveroption* V2 3-468
 - rtrim string function **V1 2-132**
 - Rules
 - See also* Database objects
 - binding V1 1-93, V2 3-125 to V2 3-128
 - changing names of V2 3-157
 - checking name with *sp_checkreswords* V2 3-153
 - column definition conflict with V1 1-93
 - creating new V1 1-91 to V1 1-94
 - default violation of V1 1-60
 - displaying source text of V2 3-357
 - dropping user-defined V1 1-187
 - insert and V1 1-263
 - naming user-created V1 1-91, V2 3-125
 - remapping V2 3-445 to V2 3-447
 - renaming V2 3-454 to V2 3-457
 - system tables and V2 3-126
 - system tables entries for V3 8-31, V3 8-63 to V3 8-64, V3 8-67
 - unbinding V2 3-524 to V2 3-526
 - Running a procedure with execute V1 1-223 to V1 1-227
- ## S
- Savepoints
 - See also* Checkpoint process
 - rollback and V1 1-330
 - setting using save transaction V1 1-336
 - save transaction command **V1 1-335 to V1 1-337**
 - Scalar aggregates
 - group by and V1 1-248
 - nesting vector aggregates within V1 2-7
 - Scale, datatype V3 7-10
 - decimal* V3 7-6
 - IDENTITY columns V3 7-10
 - loss during datatype conversion V3 7-8
 - numeric* V3 7-6
 - in user-defined datatypes V2 3-79
 - @@scan_parallel_degree* global variable V1 1-372
 - set *scan_parallel_degree* and V1 1-361
 - scan_parallel_degree* option, set V1 1-361
 - Scans, cursor V1 1-151
 - Scans, number of (statistics io) V1 1-362
 - Schemas **V1 1-95 to V1 1-96**
 - permissions V1 1-96
 - Scope of cursors V1 1-149
 - Scope of resource limits
 - changes to active time ranges and V2 3-77
 - information on V2 3-343
 - specifying V2 3-57

- Search conditions
 - datetime* data V3 7-22
 - group by** and **having** query V1 1-246, V1 1-250
 - select** V1 1-342
 - where** clause V1 1-404 to V1 1-410
 - secondary option, **disk unmirror** V1 1-174
 - second date part V1 2-20, V1 2-66
 - Seconds, **datediff** results in V1 2-62
- Security
 - See also* Permissions
 - command and object permissions V1 1-236
 - views and V1 1-132
- Security functions V1 2-22
- Seed values
 - rand** function V1 2-114
 - set identity_insert** and V1 1-358
 - segmap* column, *sysusages* table V3 8-87
 - segment* column, *syssegments* table V3 8-79
- Segments
 - See also* Database devices; Log segment; Space allocation
 - adding V2 3-62 to V2 3-64
 - changing names of V2 3-158, V2 3-161
 - checking names with
 - sp_checkreswords** V2 3-154
 - clustered indexes on V1 1-71
 - creating indexes on V1 1-17, V1 1-68, V1 1-71, V1 1-101
 - dbcc checktable** report on V1 1-139
 - dbcc indexalloc** report on V1 1-140
 - dropping V2 3-259 to V2 3-261
 - extending V2 3-63, V2 3-277
 - information on V2 3-350
 - last device reference for V2 3-261
 - mapping V2 3-63
 - mapping to a new device V1 1-9
 - monitoring remaining space V2 3-69 to V2 3-74, V2 3-400 to V2 3-404
 - names of V1 1-17, V1 1-101, V1 1-103, V2 3-63
 - number of named V1 1-56, V2 3-64
 - placing objects on V1 1-68
 - separation of table and index V1 1-70, V1 1-107
 - sp_helpthreshold** report on V2 3-359
 - syssegments* table V3 8-79
 - system tables entries for V3 8-79
 - select** command V1 1-338 to V1 1-353
 - aggregates and V1 2-6
 - create procedure** and V1 1-82
 - create view** and V1 1-130
 - for browse** V1 2-159
 - group by** and **having** clauses V1 1-245
 - insert** and V1 1-264
 - local variables and V1 1-147
 - restrictions in standard SQL V1 2-7
 - size of *text* data to be returned with V1 1-363
 - in Transact-SQL compared to standard SQL V1 2-7
 - triggers and V1 1-122
 - union** operation with V1 1-381
 - variables and V1 1-146
 - select into/bulkcopy/pllsort** database option
 - select into** and V1 1-350
 - transaction log dumping and V1 1-214
 - select into** command V1 1-340 to V1 1-350
 - column changes V1 1-20
 - not allowed with **compute** V1 1-45, V1 1-344, V1 2-11
 - Select list V1 1-309 to V1 1-310, V1 1-339 to V1 1-340
 - order by** and V1 1-344
 - union** statements V1 1-382
 - select** option, **create view** V1 1-130
 - self_recursion** option, **set** V1 1-127, V1 1-361
- Sentence order and numbered placeholders V1 1-304
- Separation, physical
 - of table and index segments V1 1-70, V1 1-107
 - of transaction log device V1 1-166, V1 1-172

- Sequence. *See* order by clause; Sort order
- Sequence tree, object V3 8-67
- serial option, disk mirror V1 1-164
- Server aliases V2 3-65
- Server information options. *See* Information (Server)
- Server process ID number. *See* Processes (Server tasks)
- Servers
- See also* Processes (Server tasks); Remote servers
 - adding V2 3-65 to V2 3-68
 - attribute names V2 4-21 to V2 4-23
 - capacity for databases V1 1-56
 - dropping V2 3-262 to V2 3-263
 - information on remote logins V2 3-340
 - local V2 3-65
 - monitoring activity of V2 3-405
 - names of V2 3-65
 - options, changing with
 - sp_serveroption V2 3-468 to V2 3-471
 - remote V2 3-353
 - sp_server_info information on V2 4-21 to V2 4-23
 - upgrading and sp_checknames V2 3-148
 - upgrading and sp_checkreswords V2 3-153
- Server user name and ID
- suser_id function V1 2-151
 - suser_name function for V1 2-152
- session authorization option, set V1 1-362
- revoking V1 1-233, V1 1-323
- set command **V1 1-354 to V1 1-373**
- See also individual set options*
 - default settings V1 1-367
 - inside a stored procedure V1 1-86
 - inside a trigger V1 1-123
 - roles and V1 1-361
 - sp_setlangalias and language option V2 3-472
 - within update V1 1-386
- setuser command **V1 1-374 to V1 1-375**
- user impersonation using V1 1-236
- 7-bit terminal, sp_helpsort output V2 3-355
- Severity levels, error
- user-defined messages V1 1-311
- shared keyword
- select V1 1-342
- Shared locks V2 3-281, V2 3-371
- showplan option, set V1 1-362
- show_role system function **V1 2-133**
- show_sec_services security function **V1 2-135**
- shutdown command **V1 1-376 to V1 1-378**
- side option, disk unmirror V1 1-174
- sign mathematical function **V1 2-136**
- Similar-sounding words. *See* soundex string function
- Single-byte character sets
- char datatype for V3 7-24
- Single quotes. *See* Quotation marks
- single user database option
- setting with sp_dboption V2 3-197
- Single-user mode V2 3-197
- sp_renamedb and V2 3-459
- sin mathematical function **V1 2-138**
- Size
- See also* Length; Number (quantity of); Range; Size limit; Space allocation
 - column V1 2-43
 - columns in table V1 1-20
 - compiled stored procedure V1 1-83
 - composite index V1 1-66
 - database device V1 1-161
 - database extension V1 1-6
 - estimation of a compiled stored procedure V1 1-83
 - floor mathematical function V1 2-76
 - identifiers (length) V3 A-11
 - image data to be returned with writetext V1 1-415
 - image datatype V2 3-499, V3 7-33
 - initialized database device V1 1-163

- log device V1 1-161, V1 1-163, V2 3-378
- model* database V1 1-161
- new database V1 1-54
- of pi V1 2-106
- readtext data V1 1-314, V1 1-315
- recompiled stored procedures V1 1-83
- row V1 1-20, V1 1-105
- set textsize function V1 1-363
- tables V1 1-105
- text* data to be returned with select V1 1-363
- text* data to be returned with writetext V1 1-415
- text* datatype V3 7-33
- text* storage V2 3-499
- transaction log device V1 1-57, V1 1-163
- Size limit
 - approximate numeric datatypes V3 7-14
 - binary* datatype V3 7-28
 - char* columns V3 7-24
 - columns allowed per table V1 1-105
 - datatypes V3 7-2 to V3 7-3
 - datetime* datatype V3 7-19
 - double precision* datatype V3 7-14
 - exact numeric datatypes V3 7-9
 - fixed-length columns V3 7-24
 - float* datatype V3 7-14
 - image* datatype V3 7-28
 - integer value smallest or largest V1 2-76
 - money datatypes V3 7-15
 - nchar* columns V3 7-24
 - nvarchar* columns V3 7-25
 - order by results V1 1-298
 - print command V1 1-306
 - real* datatype V3 7-14
 - smalldatetime* datatype V3 7-19
 - tables per database V1 1-105
 - varbinary* datatype V3 7-28
 - varchar* columns V3 7-24
 - size of auto identity column configuration parameter V2 3-194, V2 3-197
 - size option
 - disk init V1 1-160
 - disk reinit V1 1-169
 - skip_ncindex option, dbcc V1 1-139
 - Slash (/)
 - division operator V3 A-3
 - smalldatetime* datatype V3 7-19 to V3 7-23
 - date functions and V1 2-66
 - smallint* datatype V3 7-9
 - smallmoney* datatype V3 7-15, V3 7-19
 - Sort operations (order by)
 - sorting plan for V1 1-362
 - Sort order
 - See also* Order
 - ascending V1 1-296
 - changing, and sp_indsuspect system procedure V2 3-365
 - choices and effects V1 1-298
 - comparison operators and V3 A-5
 - descending V1 1-296
 - group by and having and V1 1-256
 - groups of names V1 1-256
 - information about V2 3-355
 - order by and V1 1-299
 - rebuilding indexes after changing V1 1-141
 - syscharsets* system table V3 8-27
 - sort_resources option, set V1 1-362
 - soundex string function V1 2-139
 - Source text
 - checking for existence of V2 3-164
 - displaying V2 3-357
 - hiding V2 3-363
 - Space
 - See also* Size; Space allocation
 - adding to database V1 1-6 to V1 1-9
 - for a clustered index V1 1-16, V1 1-67, V1 1-71, V1 1-101
 - clustered indexes and
 - max_rows_per_page V1 1-17, V1 1-68

- database storage V1 1-16, V1 1-67, V1 1-71, V1 1-101
- dbcc checktable reporting free V1 1-139
- estimating table and index size V2 3-273 to V2 3-276
- extents V1 1-70, V1 1-105
- extents for indexes V1 1-141
- freeing with truncate table V1 1-379
- for index pages V1 1-16, V1 1-66 to V1 1-67, V1 1-100
- max_rows_per_page and V1 1-17, V1 1-67, V1 1-101
- monitoring remaining with
 - sp_modifythreshold V2 3-400 to V2 3-404
- new database V1 1-54
- for recompiled stored procedures V1 1-83
- retrieving inactive log V1 1-209
- running out of V1 1-209
- sp_spaceused procedure V2 3-498 to V2 3-500
- for stored procedures V1 1-82
- unused V2 3-500
- used on the log segment V1 1-139, V1 1-209
- Space allocation
 - See also Database devices; Segments
 - dbcc commands for checking V1 1-138 to V1 1-140
 - future V2 3-418 to V2 3-420
 - log device V1 1-57, V2 3-378
 - pages V1 1-141
 - sp_placeobject procedure V2 3-418 to V2 3-420
 - system tables entries for V3 8-87
 - sysusages table V3 8-87
 - table V1 1-105, V1 1-138
- Spaces, character
 - See also Blanks
 - in character datatypes V3 7-24 to V3 7-27
 - empty strings (" ") or (' ') as V3 A-9, V3 A-10
 - inserted in text strings V1 2-141
 - like *datetime* values and V3 7-23
 - not allowed in identifiers V3 A-11
 - update of V1 1-390
- space string function V1 2-141
- sp_activeroles system procedure V2 3-14
- sp_addalias system procedure V2 3-16 to V2 3-17
- sp_addauditrecord system procedure V2 3-18 to V2 3-20
- sp_addauditable system procedure V2 3-21
- sp_addengine system procedure V2 3-24
- sp_addexclass system procedure V2 3-26
- sp_addextendedproc system procedure V2 3-28 to V2 3-29
- sp_addexternlogin system procedure V2 3-30 to V2 3-32
- sp_addgroup system procedure V2 3-33 to V2 3-34
- sp_addlanguage system procedure V2 3-35 to V2 3-39
- sp_addlogin system procedure V2 3-40 to V2 3-42
- sp_addmessage system procedure V2 3-43 to V2 3-45
- sp_addobjectdef system procedure V2 3-46 to V2 3-50
- sp_addremotelogin system procedure V2 3-51 to V2 3-54
- sp_add_resource_limit system procedure V2 3-55 to V2 3-61
- sp_addsegment system procedure V2 3-62 to V2 3-64
 - in mixed data and log databases V2 3-63
- sp_addserver system procedure V2 3-65 to V2 3-68
- sp_addthreshold system procedure V2 3-69 to V2 3-74
- sp_add_time_range system procedure V2 3-75 to V2 3-78
- sp_addtype system procedure V2 3-79 to V2 3-84

- sp_addumpdevice** system procedure V2 3-85 to V2 3-87
- sp_adduser** system procedure V2 3-88 to V2 3-90
- sp_altermessage** system procedure V2 3-91 to V2 3-92
- sp_auditdisplay** system procedure V2 3-211 to V2 3-215
- sp_audit** system procedure V2 3-93
- sp_autoconnect** system procedure V2 3-107 to V2 3-108
- sp_bindcache** system procedure V2 3-109 to V2 3-113
- sp_bindefault** system procedure V2 3-114 to V2 3-117
- create default and V1 1-59, V2 3-115
- user-defined datatypes and V3 7-39
- sp_bindexclass** system procedure V2 3-118
- sp_bindmsg** system procedure V2 3-122 to V2 3-124
- sp_bindrule** system procedure V2 3-125 to V2 3-128
- create rule and V1 1-92
- user-defined datatypes and V3 7-39
- sp_cacheconfig** system procedure V2 3-129 to V2 3-138
- sp_cachestrategy** system procedure V2 3-139 to V2 3-142
- sp_changedbowner** system procedure V2 3-143 to V2 3-145
- sp_changegroup** system procedure V2 3-146 to V2 3-147
- sp_dropgroup and V2 3-238
- sp_checknames** system procedure V2 3-148 to V2 3-149
- sp_checkreswords** system procedure V2 3-150 to V2 3-163
- return status V2 3-153
- sp_checksourc** system procedure V2 3-164
- sp_chgattribute** system procedure V2 3-167 to V2 3-169
- sp_clearpsex** system procedure V2 3-170
- sp_clearstats** system procedure V2 3-172 to V2 3-173
- sp_column_privileges** catalog stored procedure V2 4-5 to V2 4-7
- sp_columns** catalog stored procedure V2 4-9 to V2 4-11
- datatype code numbers V2 4-4
- and sp_datatype_info V2 4-14
- sp_commonkey** system procedure V2 3-174 to V2 3-176
- sp_configure** system procedure V2 3-177 to V2 3-183
- setting display levels for V2 3-216
- sp_countmetadata** system procedure V2 3-184
- sp_cursorinfo** system procedure V2 3-187 to V2 3-190
- sp_databases** catalog stored procedure V2 4-12
- sp_datatype_info** catalog stored procedure V2 4-14 to V2 4-15
- sp_dbcc_alterws** stored procedure V2 6-4 to V2 6-6
- sp_dbcc_configreport** stored procedure V2 6-7 to V2 6-8
- sp_dbcc_createws** stored procedure V2 6-9 to V2 6-12
- sp_dbcc_deletedb** stored procedure V2 6-13 to V2 6-14
- sp_dbcc_deletehistory** stored procedure V2 6-15 to V2 6-16
- sp_dbcc_differentialreport** stored procedure V2 6-17 to V2 6-19
- sp_dbcc_evaluatedb** stored procedure V2 6-20 to V2 6-22
- sp_dbcc_faultreport** stored procedure V2 6-23 to V2 6-25
- sp_dbcc_fullreport** stored procedure V2 6-26 to V2 6-27
- sp_dbcc_plandb** system procedure V2 3-421 to V2 3-424
- sp_dbcc_runcheck** stored procedure V2 6-28 to V2 6-29

- sp_dbcc_statisticsreport** stored procedure **V2 6-30 to V2 6-33**
sp_dbcc_summaryreport stored procedure **V2 6-34 to V2 6-36**
sp_dbcc_updateconfig stored procedure **V2 6-37 to V2 6-41**
sp_dboption system procedure **V2 3-191 to V2 3-199**
 checkpoints and **V1 1-35**
sp_dbremap system procedure **V2 3-200 to V2 3-201**
sp_defaultloc system procedure **V2 3-202 to V2 3-205**
sp_depends system procedure **V1 1-106, V2 3-206 to V2 3-208**
sp_diskdefault system procedure **V2 3-209 to V2 3-210**
sp_displaylevel system procedure **V2 3-216 to V2 3-217**
sp_displaylogin system procedure **V2 3-218 to V2 3-220**
sp_displayroles system procedure **V2 3-221**
sp_dropalias system procedure **V2 3-223 to V2 3-224**
sp_dropdevice system procedure **V2 3-225 to V2 3-226**
sp_dropengine system procedure **V2 3-227**
sp_dropexclass system procedure **V2 3-229**
sp_dropextendedproc system procedure **V2 3-231 to V2 3-232**
sp_dropexternlogin system procedure (Component Integration Services only) **V2 3-233 to V2 3-234**
sp_dropgllockpromote system procedure **V2 3-235 to V2 3-237**
sp_dropgroup system procedure **V2 3-238 to V2 3-239**
See also **sp_changegroup**
sp_dropkey system procedure **V2 3-240 to V2 3-242**
sp_droplanguage system procedure **V2 3-243 to V2 3-244**
sp_droplogin system procedure **V2 3-245 to V2 3-246**
sp_dropmessage system procedure **V2 3-247 to V2 3-248**
sp_dropobjectdef system procedure (Component Integration Services only) **V2 3-249 to V2 3-251**
sp_droptellogin system procedure **V2 3-252 to V2 3-253**
sp_drop_resource_limit system procedure **V2 3-254 to V2 3-258**
sp_dropsegment system procedure **V2 3-259 to V2 3-261**
sp_placeobject and **V2 3-260**
sp_dropserver system procedure **V2 3-262 to V2 3-263**
sp_droptreshold system procedure **V2 3-264 to V2 3-265**
sp_drop_time_range system procedure **V2 3-266 to V2 3-267**
sp_droptype system procedure **V2 3-268 to V2 3-269**
sp_dropuser system procedure **V2 3-270 to V2 3-272**
 Speed (Server)
binary and *varbinary* datatype access **V3 7-28**
 create database for load **V1 1-56**
 create index with sorted_data **V1 1-68**
 dump transaction compared to dump database **V1 1-217**
 execute **V1 1-227**
 truncate table compared to delete **V1 1-379**
 writetext compared to dbwritetext and dbmoretext **V1 1-416**
sp_estspace system procedure **V2 3-273 to V2 3-276**
sp_extendsegment system procedure **V2 3-277 to V2 3-279**
sp_familylock system procedure **V2 3-280 to V2 3-282**
sp_fkeys catalog stored procedure **V2 4-16 to V2 4-18**

- sp_forceonline_db** system procedure V2 3-283 to V2 3-284
- sp_forceonline_page** system procedure V2 3-285 to V2 3-287
- sp_foreignkey** system procedure V2 3-288 to V2 3-290
- sp_freedll** system procedure V2 3-291 to V2 3-292
- sp_getmessage** system procedure V2 3-293 to V2 3-294
- sp_grantlogin** system procedure (Windows NT only) V2 3-295
- sp_helppartition** system procedure V2 3-303
- sp_helpcache** system procedure V2 3-306 to V2 3-307
- sp_helpconfig** system procedure V2 3-308 to V2 3-311
- sp_helpconstraint** system procedure V2 3-312 to V2 3-316
- sp_helpdb** system procedure V2 3-317 to V2 3-319
- sp_helpdevice** system procedure V2 3-320 to V2 3-321
- sp_helpextendedproc** system procedure V2 3-322 to V2 3-323
- sp_helpexternlogin** system procedure (Component Integration Services only) V2 3-324
- sp_helpgroup** system procedure V2 3-326 to V2 3-327
- sp_helpindex** system procedure V2 3-328 to V2 3-329
- sp_helpjoins** system procedure V2 3-330 to V2 3-331
- sp_helpkey** system procedure V2 3-332 to V2 3-334
- sp_helplanguage** system procedure V2 3-335 to V2 3-336
- sp_helplog** system procedure V2 3-337
- sp_helpobjectdef** system procedure (Component Integration Services only) V2 3-338
- sp_helpremotelogin** system procedure V2 3-340 to V2 3-341
- sp_help_resource_limit** system procedure V2 3-342 to V2 3-345
- sp_helprotect** system procedure V2 3-346 to V2 3-349
- sp_helpsegment** system procedure V2 3-350 to V2 3-352
- sp_helpserver** system procedure V2 3-353 to V2 3-354
- sp_helpsort** system procedure V2 3-355 to V2 3-356
- sp_help** system procedure V2 3-298 to V2 3-302, V3 7-40
- sp_helptext** system procedure V2 3-357 to V2 3-358
- sp_helpthreshold** system procedure V2 3-359 to V2 3-360
- sp_helpuser** system procedure V2 3-361 to V2 3-362
- sp_hidetext** system procedure V2 3-363
- spid** number V3 8-68
- See also* Processes (Server tasks)
- sp_who** output V2 3-536
- in *sysaudits* table V3 8-10
- in *syslocks* table V3 8-53
- in *syslogshold* V3 8-59
- sp_indsuspect** system procedure V2 3-365 to V2 3-366
- sp_listsuspect_db** system procedure V2 3-367
- sp_listsuspect_page** system procedure V2 3-368 to V2 3-369
- sp_locklogin** system procedure V2 3-374 to V2 3-376
- sp_lock** system procedure V2 3-370 to V2 3-373
- sp_logdevice** system procedure V2 3-377 to V2 3-380
- log on extension to create database and V2 3-377
- sp_loginconfig** system procedure (Windows NT only) V2 3-381 to V2 3-382
- sp_logininfo** system procedure (Windows NT only) V2 3-383 to V2 3-384

sp_logiosize system procedure **V2 3-385**
sp_modifylogin system procedure **V2 3-389 to V2 3-391**
sp_modify_resource_limit system procedure **V2 3-392 to V2 3-395**
sp_modifythreshold system procedure **V2 3-400 to V2 3-404**
sp_modify_time_range system procedure **V2 3-396 to V2 3-399**
sp_monitorconfig system procedure **V2 3-408 to V2 3-411**
sp_monitor system procedure **V2 3-405 to V2 3-407**
sp_passthru system procedure **V2 3-412 to V2 3-414**
sp_password system procedure **V2 3-415 to V2 3-417**
sp_pkeys catalog stored procedure **V2 4-19 to V2 4-20**
sp_placeobject system procedure **V2 3-418 to V2 3-420**
sp_poolconfig system procedure **V2 3-425 to V2 3-431**
sp_primarykey system procedure **V2 3-432 to V2 3-433**
 sp_foreignkey and **V2 3-288**
sp_processmail system procedure **V2 3-434 to V2 3-436**
sp_procmode system procedure **V2 3-437 to V2 3-439**
sp_procxmode system procedure **V2 3-440 to V2 3-442**
sp_recompile system procedure **V2 3-443 to V2 3-444**
sp_remap system procedure **V2 3-445 to V2 3-447**
sp_remotoption system procedure **V2 3-448 to V2 3-450**
sp_remotesql system procedure **V2 3-451 to V2 3-453**
sp_renamedb system procedure **V2 3-158, V2 3-458 to V2 3-461**
sp_rename system procedure **V2 3-454 to V2 3-457**
sp_reportstats system procedure **V2 3-462 to V2 3-463**
sp_revokelogin system procedure (Windows NT only) **V2 3-464**
sp_role system procedure **V2 3-466 to V2 3-467**
sp_server_info catalog stored procedure **V2 4-21 to V2 4-23**
 sp_tables and **V2 4-39**
sp_serveroption system procedure **V2 3-468 to V2 3-471**
sp_setlangalias system procedure **V2 3-472 to V2 3-473**
sp_setpglockpromote system procedure **V2 3-474 to V2 3-477**
sp_setpsex system procedure **V2 3-478**
sp_setsuspect_granularity system procedure **V2 3-481 to V2 3-484**
sp_setsuspect_threshold system procedure **V2 3-485 to V2 3-487**
sp_showcontrolinfo system procedure **V2 3-488**
sp_showexeclass system procedure **V2 3-490**
sp_showplan system procedure **V2 3-492**
sp_showpsex system procedure **V2 3-496**
sp_spaceused system procedure **V2 3-498 to V2 3-500**
sp_special_columns catalog stored procedure **V2 4-25 to V2 4-27**
sp_sproc_columns catalog stored procedure **V2 4-28 to V2 4-29**
 datatype code numbers **V2 4-4**
sp_statistics catalog stored procedure **V2 4-30 to V2 4-32**
sp_stored_procedures catalog stored procedure **V2 4-33 to V2 4-34**
 sp_server_info information **V2 4-23**
sp_syntax system procedure **V2 3-501 to V2 3-503**
sp_sysmon system procedure **V2 3-504 to V2 3-507**
sp_table_privileges catalog stored procedure **V2 4-35**

- sp_tables catalog stored procedure **V2 4-38 to V2 4-39**
 - sp_server_info information V2 4-23
- spt_committab table V2 3-13
- spt_datatype_info_ext table V2 4-3
- spt_datatype_info table V2 4-3
- sp_thresholdaction system procedure **V2 3-508 to V2 3-510**
 - threshold procedure V2 3-70, V2 3-401
- spt_monitor table V2 3-13
- spt_server_info table V2 4-3
- spt_values table V2 3-13
- sp_unbindcache_all system procedure **V2 3-514 to V2 3-515**
- sp_unbindcache system procedure **V2 3-511 to V2 3-513**
- sp_unbinddefault system procedure V1 1-179, **V2 3-516 to V2 3-518**
- sp_unbindexclass system procedure **V2 3-519**
- sp_unbindmsg system procedure **V2 3-522 to V2 3-523**
- sp_unbindrule system procedure **V2 3-524 to V2 3-526**
 - create rule and V1 1-92
 - drop rule and V1 1-187
- sp_volchanged system procedure **V2 3-527 to V2 3-534**
 - messages V2 3-530 to V2 3-534
- sp_who system procedure **V2 3-535 to V2 3-537**
- SQL. *See* Transact-SQL
- SQL standards
 - aggregate functions and V1 2-7
 - concatenation and V3 A-5
 - set options for V1 1-372
 - set session authorization and V1 1-362
 - SQL pattern matching V2 4-3
 - user-defined datatypes and V2 3-80
- SQLSTATE codes V3 C-1 to V3 C-7
 - exceptions V3 C-1 to V3 C-7
- @sqlstatus global variable
 - fetch and V1 1-229
- sqrt mathematical function **V1 2-142**
- Square brackets []
 - caret wildcard character [^] and V3 A-18, V3 A-20
 - in SQL statements V1 xviii
 - wildcard specifier V3 A-18
- Square root mathematical function V1 2-142
- ss. *See* second date part
- Starting days of named time ranges V2 3-75
- Starting Servers
 - disk mirroring of master device and V1 1-166
 - disk remirroring of master device and V1 1-172
- Starting times of named time ranges V2 3-75
- startserver utility command
 - See also* *Utility Programs* manual
 - disk mirror and V1 1-166
 - disk remirror and V1 1-172
- Statements
 - create trigger V1 1-119
 - in create procedure V1 1-78
- Statistics
 - returned by global variables V2 3-405
 - sp_clearstats procedure V2 3-172
 - sp_monitor V2 3-405
 - sp_reportstats V2 3-462 to V2 3-463
 - update statistics V1 1-398
- statistics io option, set V1 1-362
- statistics subquerycache option, set V1 1-362
- statistics time option, set V1 1-362
- Status
 - database device V2 3-209
 - stored procedures execution V1 1-227
- status bits in sysdevices V3 8-41
- Stopping
 - Servers V1 1-376
- Stopping a procedure. *See* return command
- Storage management

- text* and *image* data V3 7-35
- Stored procedures
 - See also* Database objects; System procedures
 - alter table and V1 1-20
 - cache binding and V2 3-111, V2 3-512
 - catalog V2 4-1 to V2 4-39
 - changing transaction modes with
 - sp_procxmode* V2 3-440 to V2 3-442
 - creating V1 1-76 to V1 1-88
 - for *dbccdb* database V2 6-1
 - displaying query processing modes
 - with *sp_procqmode* V2 3-437 to V2 3-439
 - dropping V1 1-76, V1 1-183 to V1 1-184
 - dropping groups V1 1-183
 - executing V1 1-223 to V1 1-227
 - grouping V1 1-76, V1 1-224
 - ID numbers V1 1-360
 - naming V1 1-76
 - nesting V1 1-82, V1 1-227
 - object dependencies and V2 3-206 to V2 3-208, V3 8-40
 - parseonly not used with V1 1-360
 - permissions granted V1 1-233
 - permissions revoked V1 1-323
 - procid* option V1 1-360
 - remapping V2 3-445 to V2 3-447
 - renamed database and V2 3-459
 - renaming V1 1-82, V2 3-454 to V2 3-457
 - return status V1 1-84 to V1 1-85, V1 1-223, V1 1-227, V1 1-318
 - set commands in V1 1-354
 - sp_checkreswords* and V2 3-154
 - sp_recompile* and V2 3-443 to V2 3-444
 - sp_sproc_columns* information on V2 4-28 to V2 4-29
 - sp_stored_procedures* information on V2 4-33 to V2 4-34
 - storage maximums V1 1-82
 - system tables entries for V3 8-31, V3 8-63 to V3 8-64, V3 8-67
- Stored procedure triggers. *See* Triggers
- String functions V1 2-22 to V1 2-23
 - See also text* datatype
- string_truncation* option, set V1 1-363
 - insert and V1 1-264
 - update and V1 1-390
- Strings
 - concatenating V3 A-5
 - print message V1 1-304
 - truncating V1 1-263, V1 1-390
- stripe on option
 - dump database V1 1-197
 - dump transaction V1 1-211
 - load database V1 1-274
 - load transaction V1 1-283
- str string function **V1 2-143**
- Structure
 - See also* Order
 - clustered and nonclustered index V1 1-65 to V1 1-66
 - configuration V3 8-35
- stuff string function V1 2-145
- Style values, date representation V1 2-47
- Subgroups, summary values for V1 1-44
- Subqueries
 - any keyword and V3 A-6
 - in expressions V3 A-6
 - order by and V1 1-298
 - union prohibited in V1 1-384
- substring string function V1 2-147
- Subtraction operator (-) V3 A-3
- suid* (server user ID)
 - sysalternates* table listing V3 8-5
 - syslogins* table listing V3 8-56
- sum aggregate function **V1 2-150**
- Summary values
 - generation with *compute* V1 1-44
- Sundays, number value V1 2-63
- user_id* system function V1 2-151
- user_name* system function V1 2-152
- Suspect databases, listing V2 3-367
- Suspect indexes. *See* *reindex* option, *dbcc*
- Suspect pages

- bringing online V2 3-283 to V2 3-284, V2 3-285 to V2 3-287
- isolating on recovery V2 3-481 to V2 3-484, V2 3-485 to V2 3-487
- listing V2 3-368
- syb_identity keyword
 - select and V1 1-351
- sybsecurity database
 - dropping V1 1-178
 - system tables in V3 8-3
- sybsyntax database V2 3-502
- sybsystemprocs database
 - permissions and V2 3-10
- Symbols
 - See also Wildcard characters; *Symbols section of this index*
 - arithmetic operator V3 A-3
 - comparison operator V3 A-5
 - in identifier names V3 A-11
 - matching character strings V3 A-18
 - money V3 A-11
 - in SQL statements V1 xviii
 - wildcards V3 A-18
- Synonyms
 - chars and characters, patindex V1 2-103
 - chars for characters, readtext V1 1-314
 - datatype V3 7-1
 - out for output V1 1-78, V1 1-224
 - tran, transaction, and work, commit
 - command V1 1-38
 - tran, transaction, and work, rollback
 - command V1 1-330
- Syntax
 - catalog stored procedures V2 4-2 to V2 4-3
 - checking for reserved words V2 3-153
 - check using set parseonly V1 1-360
 - display procedure (sp_syntax) V2 3-501 to V2 3-503
- Syntax conventions, Transact-SQL V1 xviii
- sysalternates table V3 8-5
 - aliases V2 3-16
 - sp_dropalias and V2 3-223
 - sysusers table and V2 3-16
- sysattributes table V3 8-6 to V3 8-8
- sysauditoptions table V3 8-9
- sysaudits_01...sysaudits_08 tables V3 8-10 to V3 8-26
- syscharsets table V3 8-27 to V3 8-28
- syscolumns table V1 1-139, V3 7-31, V3 8-29 to V3 8-30
- syscomments table V3 8-31
 - default definitions in V1 1-60
 - procedure definitions in V1 1-86
 - rule definitions in V1 1-93
 - source text in V2 3-357
 - trigger definitions in V1 1-123, V1 1-135
- sysconfigures table V3 8-33
 - database size parameter V1 1-55
- sysconstraints table V3 8-34
 - sp_bindmsg and V2 3-122
- syscurconfigs table V3 8-35
- sysdatabases table V2 4-12, V3 8-37 to V3 8-39
- sysdepends table V3 8-40
- sysdevices table V2 3-209, V2 3-320, V3 8-41 to V3 8-42
 - disk init and V1 1-162
 - mirror names in V1 1-174
- sysengines table V3 8-43
- sysgams table V3 8-44
- sysindexes table V3 8-45 to V3 8-47
 - composite indexes and V1 1-73
 - name column in V3 7-35
- syskeys table V3 8-48 to V3 8-49
 - sp_dropkey and V2 3-240
 - sp_foreignkey and V2 3-288
 - sp_primarykey and V2 3-432
- syslanguages table V2 3-335, V3 8-50 to V3 8-51
 - sp_droplanguage and V2 3-243
- syslocks table V3 8-53 to V3 8-54
- sysloginroles table V3 8-55
- syslogins table V3 8-56 to V3 8-57
 - sp_modifylogin and V2 3-391
- syslogshold table V3 8-59

- syslogs** table V2 3-377, **V3 8-58**
See also Recovery; Transaction logs
 danger of changing the V3 8-4
 infinite loop if changes to V3 8-58
 put on a separate device V1 1-166, V1 1-172, V2 3-377
 running dbcc checktable on V1 1-139
- sysmessages** table **V3 8-61**
 error message text V2 3-293
 raiserror and V1 1-308
- sysmonitors** table **V3 8-62**
- sysobjects** table **V3 8-63** to **V3 8-64**
 trigger IDs and V1 1-123
- syspartitions** table **V3 8-66**
- sysprocedures** table **V3 8-67**
 trigger execution plans in V1 1-122
- sysprocesses** table **V3 8-68** to **V3 8-70**
- sysprotects** table **V3 8-71** to **V3 8-72**
 grant/revoke statements and V1 1-240, V1 1-327
 sp_changegroup and V1 1-243
- sysreferences** table **V3 8-73** to **V3 8-74**
- sysremotelogins** table V2 3-51 to V2 3-54, V2 3-262, V2 3-340, **V3 8-75**
 sp_dropremotelogin and V2 3-252
- sysresourcelimits** table **V3 8-76**
 applicable limits for a login
 session V2 3-58
 sp_help_resource_limit and V2 3-344
- sysroles** table **V3 8-77**
- syssecmechs** table **V3 8-78**
- syssegments** table V2 3-261, **V3 8-79**
- sys.servers** table **V3 8-80**
 Backup Server and V1 1-203, V1 1-218
 load database and V1 1-279
 sp_addserver and V2 3-65
 sp_helpremotelogin and V2 3-341
 sp_helpserver and V2 3-353
- sys.srvroles** table **V3 8-82**
 role_id system function and V1 2-125
- System activities
 setting query-processing options
 for V1 1-354 to V1 1-373
 shutdown V1 1-376
- System databases
 dumping V1 1-202
- System datatypes. *See* Datatypes
- System extended stored procedures V2 5-1 to V2 5-24
 list of V2 5-1
- System functions V1 2-23 to V1 2-25
- System logical name. *See* Logical device name
- System messages, language setting
 for V1 1-359
See also Error messages; Messages
- System procedures
See also create procedure command;
individual procedure names
 catalog stored V2 4-1 to V2 4-39
 changing names of V2 3-157
 create procedure and V1 1-76 to V1 1-88
 displaying source text of V2 3-357
 displaying syntax of V2 3-501 to V2 3-503
 dropping user-defined V1 1-183 to V1 1-184
 extended stored procedures V2 5-1 to V2 5-24
 help reports V2 3-298 to V2 3-362
 list of V2 3-1 to V2 3-10
 permissions V2 3-10
 return status V2 3-10
 updating and V3 8-4
 using V2 3-10
- System procedures results. *See* Information (Server)
- System procedure tables V2 3-13
 catalog stored procedures and V2 4-3
- System roles
 displaying with sp_activeroles V2 3-14
 revoking V1 1-324
 show_role and V1 2-133
 stored procedures and V1 1-242
 sysloginroles table V3 8-55
 sys.srvroles table V3 8-82
- system segment
 alter database V1 1-9

- dropping V2 3-260
- mapping V2 3-63
- System tables **V3 8-1 to V3 8-90**
 - See also* Tables; *individual table names*
 - affected by drop table V1 1-189
 - affected by drop view V1 1-194
 - allow updates to system tables parameter and V3 8-4
 - binding to caches V2 3-110
 - changes dangerous to V3 8-4
 - dbcc checkcatalog and V1 1-139
 - default definitions in V1 1-60
 - defaults and V2 3-115
 - descriptions of individual V3 8-5 to V3 8-90
 - direct updates dangerous to V2 3-159
 - direct updates to V3 8-4
 - fixing allocation errors found in V1 1-141, V1 1-142
 - keys for V3 8-48 to V3 8-49
 - master database V3 8-2 to V3 8-3
 - permissions on V3 8-3
 - rebuilding of V1 1-141, V1 1-142
 - rule information in V1 1-92
 - rules and V2 3-126
 - space allocation V2 3-418
 - sysname datatype V3 7-32
 - triggers and V1 1-122, V3 8-4
 - updating V2 3-1, V3 8-4
- sys thresholds* table V2 3-265, **V3 8-83**
 - sp_helpthreshold* and V2 3-359
- system ranges* table **V3 8-84**
 - ID number storage in V2 3-76
 - range name storage in V2 3-55
- system types* table V2 3-268, **V3 8-85 to V3 8-86**
- sysusages* table **V3 8-87**
- sysusermessages* table **V3 8-88**
 - error message text V2 3-293
 - raiserror and V1 1-308
 - sp_dropmessage* and V2 3-247
- sysusers* table **V3 8-89 to V3 8-90**
 - sysalternates* table and V2 3-16, V3 8-5

T

- tablealloc option, dbcc V1 1-141
- table count option, set V1 1-363
- Table locks
 - types of V2 3-281, V2 3-371
- Table pages
 - See also* Pages, data
 - allocation with dbcc tablealloc V1 1-141
 - system functions V1 2-163
- Tables
 - allowed in a from clause V1 1-340
 - binding to data caches V2 3-109
 - changing V1 1-13 to V1 1-24
 - changing names of V2 3-156
 - checking name with
 - sp_checkreswords* V2 3-153
 - column information V2 4-9 to V2 4-11
 - column permission information from
 - sp_column_privileges* V2 4-6 to V2 4-7
 - common key between V2 3-174 to V2 3-176
 - creating duplicate V1 1-350
 - creating new V1 1-97 to V1 1-117, V1 1-340
 - creating with create schema V1 1-95 to V1 1-96
 - dbcc checkdb and V1 1-139
 - dividing, with group by and having clauses V1 1-245 to V1 1-257
 - dropping V1 1-189 to V1 1-191
 - dropping keys between V2 3-240
 - estimating space for V2 3-273
 - identifying V3 A-13
 - index location V1 1-181, V1 1-398
 - joined common key V2 3-174 to V2 3-176
 - lock promotion thresholds for V2 3-475
 - locks held on V2 3-281, V2 3-371
 - migration to a clustered index V1 1-70, V1 1-107
 - names as qualifiers V3 A-13
 - with no data V1 1-350

- number considered in joins V1 1-363
- Object Allocation Maps of V1 1-142
- object dependencies and V2 3-206 to V2 3-208, V3 8-40
- partitioning V1 1-19, V1 1-22 to V1 1-23
- permissions on V1 1-233
- permissions revoked V1 1-323
- primary keys on V2 3-432
- renaming V2 3-454 to V2 3-457
- single-group V1 1-249
- space used by V2 3-500
- sp_placeobject space allocation for V2 3-418 to V2 3-420
- sp_recompile and V2 3-443 to V2 3-444
- sp_table_privileges information on V2 4-35
- sp_tables V2 4-38
- with suspect indexes V2 3-365
- system procedure V2 3-13, V2 4-3
- system tables entries for V3 8-29, V3 8-63 to V3 8-64
- Transact-SQL extension effects and querying V1 1-250
- unbinding from data caches V2 3-511
- unpartitioning V1 1-19
- update statistics on V1 1-399
- worktables V1 2-6
- Tangents, mathematical functions for V1 2-153
- tan mathematical function V1 2-153
- Tape dump devices
 - adding V2 3-85 to V2 3-87
 - sp_volchanged messages and V2 3-531
 - sysdevices table V3 8-41
- Tape labels
 - listonly option to load database V1 1-275
 - listonly option to load transaction V1 1-284
- tape option, sp_addumpdevice V2 3-85
- tempdb database
 - See also Databases
 - adding objects to V1 1-107
- auto identity database option and V2 3-194
- sysobjects table and V1 1-106
- system tables entries and V3 8-63 to V3 8-64
- systypes table and V1 1-107
- unique auto_identity index database option and V2 3-198
- user-defined datatypes in V3 7-39
- Temporary names. See Alias, user
- Temporary tables
 - catalog stored procedures and V2 4-3
 - create procedure and V1 1-86
 - create table and V1 1-98, V1 1-106
 - identifier prefix (#) V1 1-98
 - indexing V1 1-70
 - naming V1 1-106, V3 A-12
 - sp_help and V2 3-301
 - system procedure V2 3-13
- Terminals
 - 7-bit, sp_helpsort output example V2 3-355
 - 8-bit, sp_helpsort output example V2 3-355
- Text
 - copying with defncopy V2 3-155
 - user-defined message V2 3-43
 - @@textcolid global variable V3 7-37
 - text datatype V3 7-33 to V3 7-38
 - convert command V3 7-37
 - converting V1 2-14
 - initializing with null values V3 7-34
 - initializing with update V1 1-390
 - length of data returned V1 1-349, V1 1-363
 - null values V3 7-35
 - order by not allowed V1 1-298
 - prohibited actions on V3 7-36
 - size of storage V2 3-499
 - storage on separate device V1 1-314
 - sysindexes table and V3 8-45
 - textsize setting V1 1-363
 - triggers and V1 1-122
 - union not allowed on V1 1-384

- @@textdbid* global variable V3 7-37
- Text functions V1 2-25
- @@textobjid* global variable V3 7-37
- Text page pointer V1 2-43
- Text pointer values V1 2-154
 - readtext and V1 1-314
- textptr function V1 1-314, V1 1-315, **V1 2-154**
- @@textptr* global variable V3 7-37
- @@textsize* global variable V1 1-373
 - readtext and V1 1-315
 - set textsize and V1 1-363, V3 7-37
- textsize option, set V1 1-363
- @@textts* global variable V3 7-37
- textvalid function **V1 2-156**
- then keyword. *See when...then conditions*
- @@thresh_hysteresis* global variable
 - threshold placement and V2 3-70
- Threshold procedures V2 3-70
 - creating V2 3-508
 - executing V2 3-72, V2 3-403
 - parameters passed to V2 3-71, V2 3-402
- Thresholds
 - adding V2 3-69 to V2 3-74
 - changing V2 3-400 to V2 3-404
 - crossing V2 3-70
 - database dumps and V1 1-202
 - disabling V2 3-72, V2 3-264, V2 3-403
 - hysteresis value V2 3-70, V2 3-401
 - information about V2 3-359
 - last-chance V1 2-88, V2 3-70, V2 3-73, V2 3-264, V2 3-401, V2 3-403
 - maximum number V2 3-71, V2 3-402
 - removing V2 3-264 to V2 3-265
 - space between V2 3-71
 - systhresholds* table V3 8-83
 - transaction log dumps and V1 1-217
- Ties, regulations for sort order V1 1-299 to V1 1-300
- Time interval
 - See also* Timing
 - automatic checkpoint V1 1-35
 - elapsed execution (statistics time) V1 1-362
 - estimating index creation V2 3-273
 - limiting V2 3-56
 - for running a trigger V1 1-123
 - since *sp_monitor* last run V2 3-405
 - waitfor V1 1-401
- time option, waitfor V1 1-401
- timeouts option, *sp_serveroption* V2 3-468
- Time ranges
 - adding V2 3-75
 - “at all times” V2 3-76, V2 3-266
 - changing active time ranges V2 3-77
 - creating V2 3-75
 - dropping V2 3-266
 - entire day V2 3-75
 - IDs for V2 3-76
 - modifying V2 3-396
 - overlapping V2 3-77
 - systimeranges* system table V3 8-84
- timestamp* datatype V3 7-17 to V3 7-18
 - automatic update of V3 7-17
 - browse mode and V1 2-158, V3 7-17
 - comparison using *tsequal* function V1 2-158
- Timestamps, order of transaction log dumps V1 1-278
- Time values
 - datatypes V3 7-19 to V3 7-23
- Timing
 - See also* Time interval
 - automatic checkpoint V1 1-35
- tinyint* datatype **V3 7-9**
- to option
 - dump database V1 1-196
 - dump transaction V1 1-210
 - revoke V1 1-327
- @@total_errors* global variable
 - sp_monitor* and V2 3-407
- @@total_read* global variable
 - sp_monitor* and V2 3-406
- Totals
 - compute command V1 1-298
- @@total_write* global variable

- sp_monitor and V2 3-406
- Trailing blanks. *See* Blanks
- @@tranchained* global variable V1 1-373
- transactional_rpc option, set V1 1-364
- Transaction canceling. *See* rollback command
- transaction isolation level option, set V1 1-363
- Transaction logs
 - See also* dump transaction command; syslogs table
 - backing up V1 1-195
 - data caches and V2 3-428
 - of deleted rows V1 1-157
 - dump database and V1 1-195
 - dumping V1 1-208
 - inactive space V1 1-209
 - insufficient space V1 1-216
 - loading V1 1-282 to V1 1-291
 - log I/O size and V2 3-428
 - master database V1 1-202, V1 1-215
 - placing on separate segment V1 1-216
 - purging V1 1-202
 - on a separate device V1 1-162, V1 1-166, V1 1-172, V1 1-214, V2 3-377 to V2 3-380
 - space, monitoring V1 1-217
 - space extension V1 1-9
 - syslogs table trunc log on chkpt V1 1-214
 - system tables entries for V3 8-63 to V3 8-64
 - thresholds and V2 3-264
 - writetext with log and V1 1-414
- Transactions
 - See also* Batch processing; rollback command; User-defined transactions
 - begin V1 1-27
 - chained V1 1-39
 - dump transaction command V1 1-208 to V1 1-222
 - ending with commit V1 1-38
 - fetch and V1 1-229
 - isolation levels V1 1-363
 - modes V2 3-440 to V2 3-442
 - parameters not part of V1 1-227
 - preparing V1 1-303
 - save transaction and V1 1-335 to V1 1-337
 - update iteration within given V1 1-390
- Transact-SQL
 - aggregate functions in V1 2-7
 - commands summary table V1 1-1 to V1 1-5
 - extensions V1 1-250
 - reserved words V2 3-153, V3 B-1 to V3 B-2
- Translation
 - of arguments V1 1-304
 - of integer arguments into binary numbers V3 A-4
 - of user-defined messages V2 3-44
- Triggers
 - See also* Database objects; Stored procedures
 - changing names of V2 3-157
 - checking name with
 - sp_checkreswords V2 3-153
 - creating V1 1-118 to V1 1-128, V1 1-240, V1 1-327
 - delete and V1 1-157
 - displaying source text of V2 3-357
 - dropping V1 1-192
 - enabling self-recursion V1 1-127
 - on *image* columns V1 1-122
 - insert and V1 1-264
 - nested V1 1-126 to V1 1-127
 - nested, and rollback trigger V1 1-333
 - @@nestlevel* and V1 1-126
 - object dependencies and V2 3-206 to V2 3-208, V3 8-40
 - parseonly not used with V1 1-360
 - recursion V1 1-127
 - remapping V2 3-445 to V2 3-447
 - renamed database and V2 3-459
 - renaming V1 1-123, V2 3-454 to V2 3-457
 - rollback in V1 1-124, V1 1-331

- rolling back V1 1-333
 - `@@rowcount` and V1 1-125
 - self-recursion V1 1-127
 - set commands in V1 1-354
 - `sp_recompile` and V2 3-443 to V2 3-444
 - stored procedures and V1 1-127
 - system tables and V1 1-122, V3 8-4
 - system tables entries for V3 8-31, V3 8-63 to V3 8-64, V3 8-67
 - on *text* columns V1 1-122
 - time interval V1 1-123
 - `truncate table` command and V1 1-379
 - update and V1 1-387
 - Trigger tables V1 1-124
 - Trigonometric functions V1 2-20, V1 2-20 to V1 2-153
 - True/false data, *bit* columns for V3 7-31
 - `true | false` clauses
 - `sp_dboption` V2 3-191
 - `sp_remotoption` V2 3-448
 - `true` option, `sp_changedbowner` V2 3-143
 - `truncate_only` option, `dump transaction` V1 1-209, V1 1-215
 - `truncate table` command **V1 1-379 to V1 1-380**
 - delete triggers and V1 1-124
 - faster than `delete` command V1 1-156
 - update statistics after V1 1-399
 - Truncation
 - `arithabort numeric_truncation` V3 7-7
 - binary datatypes V3 7-28
 - character string V3 7-24
 - datatypes with no length specified V1 1-77
 - `datediff` results V1 2-62
 - default values V1 1-60
 - insert and V1 1-263
 - log, prohibited on mixed device V1 1-54
 - set `string_rtruncation` and V1 1-363
 - spaces to a single space V1 1-390
 - str conversion and V1 2-144
 - temporary table names V3 A-12
 - transaction log V1 1-208
 - `trunc log on chkpt database` option V2 3-197
 - Trusted mode
 - remote logins and V2 3-53
 - trusted option, `sp_remotoption` V2 3-448
 - Truth tables
 - logical expressions V3 A-9 to V3 A-10
 - `tsequal` system function V1 2-158
 - Twenty-first century numbers V3 7-19
 - Two-digit year numbers V1 2-20, V1 2-66
 - Two-phase commit
 - probe process V2 3-463, V3 8-56
- ## U
- Unbinding
 - data caches V2 3-511 to V2 3-513
 - defaults V1 1-60, V1 1-179, V2 3-516 to V2 3-518
 - objects from caches V2 3-511 to V2 3-513
 - rules V1 1-187
 - Unconditional branching to a user-defined label V1 1-231
 - Underscore (`_`)
 - character string wildcard V3 A-18, V3 A-19
 - object identifier prefix V1 2-169, V3 A-11
 - in temporary table names V3 A-12
 - Undoing changes. *See* `rollback` command
 - union operator **V1 1-381 to V1 1-384**
 - maximum number of tables V1 1-382
 - restrictions on use V1 1-384
 - unique `auto_identity` index database
 - option V2 3-197
 - Unique constraints V1 1-110
 - unique keyword
 - `alter table` V1 1-15
 - `create index` V1 1-65
 - `create table` V1 1-100
 - Unique names as identifiers V3 A-12
 - unload option

- dump database V1 1-197
- dump transaction V1 1-211
- load database V1 1-274
- load transaction V1 1-283
- Unlocking login accounts V2 3-374
- Unmapping a segment from a
 - database V2 3-259 to V2 3-261
- Unmirroring devices. *See* Disk mirroring
- unpartition clause, alter table V1 1-19
- Unused space
 - sp_spaceused reporting of V2 3-500
- Updatable cursors V1 1-151
- update all statistics command V1 1-394
- update command **V1 1-385 to V1 1-393**
 - ignore_dup_key and V1 1-68
 - ignore_dup_row and V1 1-72
 - insert and V1 1-262
 - triggers and V1 1-122
 - triggers and if update V1 1-125
 - views and V1 1-133, V1 1-392
- update partition statistics command **V1 1-396 to V1 1-397**
- update statistics command **V1 1-398 to V1 1-399**
 - create index and V1 1-69
- Updating
 - See also* Changing; *timestamp* datatype
 - in browse mode V1 2-158
 - data in views V1 1-133
 - direct to system tables V3 8-4
 - “dirty” pages V1 1-35 to V1 1-36
 - ignore_dup_key and V1 1-68
 - prevention during browse mode V1 2-158
 - primary keys V1 1-121
 - system procedures and V3 8-4
 - system tables V3 8-4
 - trigger firing by V1 1-127
 - writetext V1 1-414
- Uppercase letter preference V1 1-299
 - See also* Case sensitivity; order by clause
- upper string function V1 2-161
- Usage statistics V2 3-462
- use command **V1 1-400**
- used_pgs system function V1 2-163
- us_english language V2 3-36, V3 8-50
 - weekdays setting V1 1-367, V1 2-67
- User context for operating system
 - commands (xp_cmdshell) V2 5-3
- User-created objects. *See* Database objects
- User-defined datatypes
 - See also* Datatypes
 - binding defaults to V2 3-114 to V2 3-117
 - binding rules to V2 3-125
 - changing names of V2 3-157
 - checking name with
 - sp_checkreswords V2 3-153
 - creating V2 3-79 to V2 3-84, V3 7-39
 - dropping V2 3-268 to V2 3-269, V3 7-39
 - hierarchy V2 3-81
 - naming V2 3-81
 - sysname as V3 7-32
 - timestamp as V3 7-17
 - unbinding defaults from V2 3-516 to V2 3-518
 - unbinding rules with sp_unbindrule V2 3-524 to V2 3-526
- User-defined event logging
 - (xp_logevent) V2 5-11
- User-defined messages V2 3-43 to V2 3-45
 - unbinding with sp_unbindmsg V2 3-522 to V2 3-523
- User-defined procedures
 - creating V1 1-76 to V1 1-88
 - creating ESPs with
 - sp_addextendedproc V2 3-28
 - executing V1 1-223 to V1 1-227
- User-defined roles
 - adding passwords to V1 1-10
 - conflicting V1 1-11
 - creating V1 1-89
 - displaying with sp_activeroles V2 3-14
 - mutual exclusivity and V1 2-97
 - revoking V1 1-324

- sysssrvroles table V3 8-82
 - system procedures and V1 1-242
 - turning on and off V1 1-361
 - User-defined transactions
 - See also Transactions
 - begin transaction V1 1-27
 - ending with commit V1 1-38
 - User errors. See Errors; Severity levels
 - User groups. See Groups; "public" group
 - User IDs
 - displaying V2 3-219
 - dropping with sp_droplogin and V2 3-245
 - user_id function for V1 2-165
 - valid_user function V1 2-171
 - user_id system function V1 2-165
 - user keyword
 - alter table V1 1-14
 - create table V1 1-99
 - system function V1 2-164
 - User names V1 2-167
 - See also Database object owners; Logins
 - changing V2 3-158
 - checking with sp_checkreswords V2 3-154
 - finding V1 2-152
 - user_name system function V1 2-167
 - User objects. See Database objects
 - User permissions. See Database Owners; Permissions
 - Users
 - accounting statistics V2 3-172, V2 3-462
 - adding V2 3-40 to V2 3-42, V2 3-88 to V2 3-90
 - change group for V2 3-146 to V2 3-147
 - changing names of V2 3-161, V2 3-389 to V2 3-391
 - dropping aliased V2 3-223 to V2 3-224
 - dropping from databases V2 3-270 to V2 3-272
 - dropping from Servers V2 3-245 to V2 3-246
 - dropping remote V2 3-262
 - guest V2 3-271
 - guest permissions V1 1-243
 - impersonating (setuser) V1 1-236
 - information on V2 3-218, V2 3-361
 - other object owner V3 A-14
 - password change for accounts V2 3-415 to V2 3-417
 - permissions of V2 3-346
 - remote V2 3-340
 - sp_who report on V2 3-535 to V2 3-537
 - syslogins table V3 8-56 to V3 8-57
 - system procedure permissions and V1 1-239, V2 3-10
 - system tables entries for V3 8-56 to V3 8-57, V3 8-89
 - sysusers table V2 3-16, V3 8-89
 - user system function V1 2-164
 - using bytes option, patindex string function V1 2-103, V1 2-104
 - using option, readtext V1 1-314, V1 1-316
 - Utility commands
 - See also Utility Programs manual
 - display syntax V2 3-501 to V2 3-503
- ## V
- valid_name system function V1 2-169
 - using after changing character sets V3 A-15
 - valid_user system function V1 2-171
 - Values
 - displaying with sp_server_info V2 4-21 to V2 4-23
 - IDENTITY columns V1 1-265
 - procedure parameter or argument V1 1-224
 - values option, insert V1 1-261
 - varbinary datatype V3 7-28 to V3 7-29
 - in timestamp columns V3 7-17
 - varchar datatype V3 7-24
 - datetime values conversion to V3 7-23
 - in expressions V3 A-10
 - spaces in V3 7-24

- spaces in and insert V1 1-263
 - Variable-length character. *See varchar datatype*
 - Variable-length columns
 - empty strings in V1 1-263
 - stored order of V1 1-300
 - Variables
 - assigning as part of a select list V1 1-340
 - in update statements V1 1-388
 - local V1 1-146 to V1 1-147
 - in print messages V1 1-305
 - return values and V1 1-226
 - vdevno option
 - disk init V1 1-160
 - disk reinit V1 1-169
 - Vector aggregates V1 2-7
 - group by and V1 1-248
 - nesting inside scalar aggregates V1 2-7
 - @@version* global variable V1 1-305
 - Views
 - See also* Database objects; Multitable views
 - allowed in a from clause V1 1-340
 - changes to underlying tables of V1 1-133
 - checking name with
 - sp_checkreswords V2 3-153
 - check option and V1 1-391 to V1 1-392
 - columns V2 4-9 to V2 4-11
 - common key between V2 3-174 to V2 3-176
 - creating V1 1-129 to V1 1-137
 - creating with create schema V1 1-95 to V1 1-96
 - displaying source text of V2 3-357
 - dropping V1 1-194
 - dropping keys between V2 3-240
 - inserting data through V1 1-267
 - names as qualifiers V3 A-13
 - object dependencies and V2 3-206 to V2 3-208, V3 8-40
 - permissions on V1 1-233, V1 1-237
 - permissions revoked V1 1-323
 - primary keys on V2 3-432
 - readtext and V1 1-316
 - remapping V2 3-445 to V2 3-447
 - renamed database and V2 3-459
 - renaming V1 1-132, V2 3-156, V2 3-454 to V2 3-457
 - system tables entries for V3 8-29, V3 8-31, V3 8-63 to V3 8-64, V3 8-67
 - update and V1 1-133, V1 1-391 to V1 1-393
 - updating restrictions V1 1-392
 - with check option V1 1-133, V1 1-267 to V1 1-268
 - Violation of domain or integrity rules V1 1-263
 - Virtual address V1 1-169
 - Virtual device number V1 1-160, V1 1-163, V1 1-169
 - Virtual page numbers V2 3-321
 - Virtual tables V3 8-3
 - Volume handling V2 3-527
 - Volume names, database dumps V1 1-204
 - vstart option
 - disk init V1 1-161
 - disk reinit V1 1-169
- ## W
- waitfor command V1 1-401 to V1 1-403
 - Waiting for shutdown V1 1-377
 - wait option, shutdown V1 1-376
 - Wash area
 - configuring V2 3-428
 - defaults V2 3-428
 - wash keyword, sp_poolconfig V2 3-425
 - week date part V1 2-20, V1 2-66
 - weekday date part V1 2-20, V1 2-66
 - Weekday date value
 - first V2 3-35
 - names and numbers V1 1-358, V1 2-67, V2 3-35
 - when...then conditions V1 1-31

- when keyword. *See* when...then conditions
 - where clause **V1 1-404 to V1 1-410**
 - aggregate functions not permitted in **V1 1-409**
 - delete **V1 1-154**
 - group by clause and **V1 1-250**
 - having and **V1 1-409**
 - null values in a **V3 A-8**
 - repeating a **V1 1-253**
 - where current of clause
 - delete **V1 1-155**
 - update **V1 1-386**
 - while keyword **V1 1-411 to V1 1-413**
 - continue and **V1 1-52**
 - exiting loop with break **V1 1-29**
 - loops **V1 1-411**
 - Wildcard characters **V3 A-16 to V3 A-22**
 - See also* patindex string function in a like match string **V3 A-18**
 - literal characters and **V3 A-20**
 - SQL standards pattern matching (\$ and _) **V2 4-3**
 - used as literal characters **V3 A-20**
 - with check option option
 - create view **V1 1-130**
 - views and **V1 1-135**
 - with consumers clause, create index **V1 1-68**
 - with grant option option, grant **V1 1-234**
 - with keyword
 - rollback trigger **V1 1-333**
 - set role command **V1 1-361**
 - with log option, writetext **V1 1-414**
 - with no_error option, set char_convert **V1 1-357**
 - with no_log option, dump transaction **V1 1-209**
 - with no_truncate option, dump transaction **V1 1-212**
 - with nowait option, shutdown **V1 1-376**
 - with override option **V1 1-185**
 - alter database **V1 1-7**
 - for load and **V1 1-54**
 - with recompile option
 - create procedure **V1 1-78**
 - execute **V1 1-224**
 - with truncate_only option, dump transaction **V1 1-209, V1 1-215**
 - with wait option, shutdown **V1 1-376**
 - wk. *See* week date part
 - Words, finding similar-sounding **V1 2-139**
 - Work session, set options for **V1 1-354 to V1 1-373**
 - Worktables
 - number of **V1 2-6**
 - Write operations
 - logging text or image **V1 1-414**
 - writes option, disk mirror **V1 1-164**
 - writetext command **V1 1-414 to V1 1-416**
 - text data initialization requirement **V3 7-36**
 - triggers and **V1 1-124**
- X**
- xp_cmdshell context configuration
 - parameter **V2 5-3**
 - xp_cmdshell system extended stored procedure **V2 5-3 to V2 5-4**
 - xp_deletemail system extended stored procedure **V2 5-5 to V2 5-6**
 - sp_processmail and **V2 3-435**
 - xp_enumgroups system extended stored procedure **V2 5-7**
 - xp_findnextmsg system extended stored procedure **V2 5-9**
 - sp_processmail and **V2 3-435**
 - xp_logevent system extended stored procedure **V2 5-11**
 - xp_readmail system extended stored procedure **V2 5-13**
 - sp_processmail and **V2 3-435**
 - xp_sendmail system extended stored procedure **V2 5-17**
 - sp_processmail and **V2 3-435**
 - XP Server **V2 5-2**
 - freeing memory from **V2 3-291 to V2 3-292**

xp_startmail system extended stored
procedure **V2 5-21**
xp_stopmail system extended stored
procedure **V2 5-23**

Y

year date part V1 2-20, V1 2-66
Year values, date style V1 2-48
Yen sign (¥)
 in identifiers V3 A-11
 in money datatypes V3 7-15
Yes/no data, *bit* columns for V3 7-31
yy. *See* year date part

Z

Zero-length string output V1 1-306
Zeros
 trailing, in binary datatypes V3 7-28
 to V3 7-29
Zero x (0x) V1 2-17, V3 7-28, V3 7-29