

## **Oracle® SQL Developer**

Oracle TimesTen In-Memory Database Support User's Guide

Release 4.0

**E39882-01**

June 2013

Beta Draft

Copyright © 2009, 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

This documentation is in preproduction status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

---

---

# Contents

<b>Preface</b> .....	vii
Audience .....	vii
Related documents .....	vii
Conventions .....	vii
Documentation Accessibility .....	viii
Technical support .....	viii
<b>What's New</b> .....	ix
New features for release of SQL Developer 4.0 .....	ix
New features for release of SQL Developer 3.1 .....	ix
<b>1 Connecting to a TimesTen Database</b>	
Setting environment variables for TimesTen and SQL Developer .....	1-1
Defining a TimesTen database named connection .....	1-2
DSN .....	1-3
Easy Client/Server .....	1-4
Advanced .....	1-6
Using an existing TimesTen database named connection .....	1-7
Disconnecting an existing TimesTen database connection .....	1-7
<b>2 Working with TimesTen Objects</b>	
Locating a TimesTen database object .....	2-1
Specifying and displaying LOB data types .....	2-2
Specifying the INLINE attribute for columns .....	2-3
Computing table size information .....	2-4
Using a REF CURSOR as an OUT parameter .....	2-6
Viewing the characteristics of a table .....	2-8
Viewing the columns .....	2-8
Viewing the indexes .....	2-9
Viewing the aging attributes .....	2-9
Viewing the table definition .....	2-9
Creating an index on a table .....	2-10
Creating a primary key constraint .....	2-12
Changing the attributes of primary key .....	2-13
Altering the aging properties of a table .....	2-14

Adding an aging policy to a table.....	2-14
Dropping an aging policy from a table.....	2-16
Changing the aging state of a table's aging policy .....	2-16
Changing the memory usage thresholds and LRU aging cycle .....	2-17
Changing the lifetime and cycle of a table's time-based aging policy.....	2-18
<b>Granting and revoking object privileges.....</b>	<b>2-18</b>
<b>Viewing the privileges granted on a database object.....</b>	<b>2-20</b>
<b>Loading data into tables.....</b>	<b>2-21</b>
<b>ttIsql commands in SQL worksheet .....</b>	<b>2-21</b>
 <b>3 Interfacing with the TimesTen Query Optimizer</b>	
Updating the table and column statistics used by the query optimizer.....	3-1
Exporting table statistics .....	3-4
Viewing the execution plan of a statement .....	3-6
 <b>4 Working with the TimesTen Index Advisor</b>	
Connection level capture .....	4-1
SQL worksheet.....	4-2
Including optimizer hints for Oracle BI server in a SQL worksheet .....	4-5
Saving index recommendations from a SQL worksheet.....	4-7
SQL workload script.....	4-8
Including optimizer hints for Oracle BI server for a SQL workload script .....	4-16
Database level capture.....	4-21
Saving index recommendations .....	4-28
Viewing the TimesTen index advisor capture state .....	4-29
 <b>5 Loading data from an Oracle database into a TimesTen database</b>	
Loading data from Oracle database tables .....	5-1
Adding an Oracle database schema .....	5-6
Removing an Oracle database schema.....	5-7
Applying a filter to Oracle database schema tables .....	5-7
Loading custom query results.....	5-9
Loading data from an Oracle database table into an existing TimesTen table .....	5-10
 <b>6 Prerequisites to Creating and Accessing a Cache Environment</b>	
Configuring the Oracle and TimesTen databases .....	6-1
Set the cache administration user name and password .....	6-1
Configure a cache grid.....	6-2
Start and stop the cache agent.....	6-2
 <b>7 Creating Cache Groups</b>	
Creating a cache group .....	7-1
Defining a cache group and cache table attributes .....	7-2
Defining the root table.....	7-4
Selecting the columns to cache.....	7-5

Specifying the table attributes.....	7-6
Specifying an aging policy.....	7-8
Verifying the root table definition.....	7-10
Defining the child tables .....	7-10
Selecting the columns to cache.....	7-11
Specifying the table attributes.....	7-12
Completing the definitions of the child tables.....	7-13
Verifying the cache group definition .....	7-13
Dropping a cache group .....	7-14
<b>8 Cache Group Operations</b>	
Loading or refreshing a cache group .....	8-1
Unloading a cache group .....	8-2
Flushing a cache group.....	8-3
<b>9 Cache Group and Cache Table Characteristics</b>	
Viewing the characteristics of a cache group .....	9-1
Viewing the cache tables .....	9-2
Viewing the cache group definition .....	9-3
Viewing the aging attributes .....	9-3
Altering the automatic refresh attributes of a cache group.....	9-4
<b>10 Interfacing with TimesTen Replication</b>	
Viewing the replication scheme .....	10-1
Viewing the details of the TimesTen subscriber databases .....	10-2
Viewing the store attributes of the replicated TimesTen databases .....	10-3
Starting the replication agent.....	10-3
<b>11 Working with TimesTen Performance Metrics</b>	
Creating snapshots.....	11-1
Dropping snapshots.....	11-3
Editing the TimesTen performance metrics configuration .....	11-6
Creating a TimesTen performance metrics report.....	11-7
Saving a report.....	11-9
<b>12 TimesTen Reports</b>	
Generating TimesTen Reports .....	12-1
About Your Database .....	12-1
All Objects .....	12-2
Cache Groups .....	12-2
Database Administration.....	12-2
Performance.....	12-3
PLSQL.....	12-3
Security .....	12-3

Table .....	12-3
-------------	------

## Index

---

---

# Preface

This guide is a supplement to the *Oracle SQL Developer User's Guide* for TimesTen-specific features.

Oracle SQL Developer is a graphical user interface (GUI) tool that gives database developers a convenient way to perform basic tasks. You can browse, create, edit and drop particular database objects; run SQL statements and scripts; manipulate and export data; and view and create reports.

You can connect to a target TimesTen database and perform operations on particular objects in the database.

## Audience

This guide is intended for those who use the Oracle SQL Developer tool with Oracle TimesTen In-Memory Database and Oracle In-Memory Database Cache. Application developers and system administrators can use SQL Developer to create and manage cache groups, and other TimesTen database objects.

## Related documents

TimesTen documentation is available on the product distribution media and on the Oracle Technology Network (OTN):

<http://www.oracle.com/technetwork/products/timesten/documentation/>

The SQL Developer software download is available on Oracle Technology Network (OTN) at:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at  
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit  
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit  
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Technical support

For information about obtaining technical support for TimesTen products, go to the following Web address:

<http://www.oracle.com/support/contact.html>



---

---

# What's New

This section summarizes the new features of TimesTen supported in Oracle SQL Developer Release 4.0. There are some features that are only supported in TimesTen Release 11.2.2.5 or later. If a feature is only supported in TimesTen Release 11.2.2.5 or later, it will be noted.

## New features for release of SQL Developer 4.0

- When you create a named connection to connect to a TimesTen database, the **User specified** connection type is now called **Advanced**.
- You can set the **Autocommit** option directly from the **New/Select Database Connection** dialog. This new autocommit option is set at the connection level. In previous releases the autocommit option is set at the running instance level.

Features supported in TimesTen Release 11.2.2.5 or later:

- You can capture `ttStats` snapshots and generate `ttStats` reports that compare two snapshots. A `ttStats` snapshot is a collection of performance metrics. TimesTen collects performance metrics from TimesTen system tables, system views, and built-in procedures. For more information, see ["Working with TimesTen Performance Metrics"](#) on page 11-1.
- There are new pre-defined Performance reports. Performance reports are reports that show statistics information for the TimesTen database and statistics snapshots of the database. These reports are available in the **Performance** category of the **TimesTen Reports**. For more information, see ["Performance"](#) on page 12-3.

Features supported in TimesTen Release 11.2.2.4 or later:

- SQL Developer supports the TimesTen Index Advisor which can evaluate a SQL workload and recommend indexes. The indexes that the TimesTen Index Advisor recommends can improve the performance for the following: joins, single table scans, and `ORDER BY` or `GROUP BY` operations. For more information, see ["Working with the TimesTen Index Advisor"](#) on page 4-1.
- You can load data using parallel threads from an Oracle database into a TimesTen database without creating a cache grid, cache group, or cache table. For more information, see ["Loading data from an Oracle database into a TimesTen database"](#) on page 5-1.

## New features for release of SQL Developer 3.1

Features supported in TimesTen Release 11.2.2 or later:

- You can define and use LOB data types. This includes defining columns with LOB data types, adding columns with LOB data types, displaying LOB data types, and defining LOB data types in IN, OUT, and IN OUT parameters in PL/SQL editors. For more information, see ["Specifying and displaying LOB data types"](#) on page 2-2.
- You can compute table size information. For more information on computing table size information, see ["Computing table size information"](#) on page 2-4.
- For table reports, there are reports for **Table Sizes** and **Column** reports will display LOB data types if applicable. For information on table reports, see ["Table"](#) on page 12-3.

Features supported in TimesTen Release 11.2.1.4.0 or later:

- You can specify `INLINE` when you define columns. For more information, see ["Specifying the `INLINE` attribute for columns"](#) on page 2-3.
- You can test your PL/SQL function, procedure, or package using a `REF CURSOR` as an `OUT` parameter. For more information, see ["Using a `REF CURSOR` as an `OUT` parameter"](#) on page 2-6.
- You can load data into tables. For more information, see ["Loading data into tables"](#) on page 2-21.
- There are reports for **Cache Groups**. For more information, see ["Cache Groups"](#) on page 12-2.

---

# Connecting to a TimesTen Database

You can connect to a TimesTen database after you install and start SQL Developer.

This chapter focuses on the steps necessary to connect to a TimesTen database in SQL Developer. The topics include:

- [Setting environment variables for TimesTen and SQL Developer](#)
- [Defining a TimesTen database named connection](#)
- [Using an existing TimesTen database named connection](#)
- [Disconnecting an existing TimesTen database connection](#)

For information about installing and starting SQL Developer, see *Oracle SQL Developer User's Guide*.

## Setting environment variables for TimesTen and SQL Developer

To access a TimesTen database from SQL Developer, install SQL Developer on the same system as the TimesTen Data Manager or TimesTen Client. If you use SQL Developer to access local databases, install SQL Developer on the same system as the TimesTen Data Manager. If you use SQL Developer to access remote databases, install SQL Developer on the same system as the TimesTen Client.

Before you start SQL Developer, execute the TimesTen `install_dir/bin/ttenv.sh` or TimesTen `install_dir/bin/ttenv.csh` shell script on Linux systems, or the TimesTen `install_dir\bin\ttenv.bat` batch file on Microsoft Windows systems. The shell script or batch file adds or modifies the environment variables:

- CLASSPATH includes the TimesTen `install_dir/lib/ttjdbc7.jar` file on Linux systems or the TimesTen `install_dir\lib\ttjdbc7.jar` file on Microsoft Windows systems. SQL Developer 4.0 requires that you have installed Java Development Kit (JDK) 7.

To determine the third party JDBC drivers SQL Developer is using, from the **Tools** menu, select **Preferences**. In the **Preferences** dialog, click the + to the left of the **Database** node. Select **Third Party JDBC Drivers**. View the path name of all third party JDBC drivers including the TimesTen JDBC drivers.

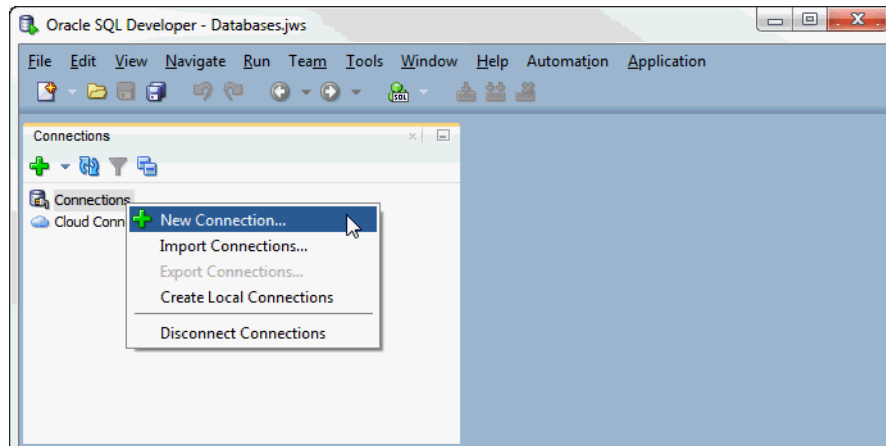
- LD\_LIBRARY\_PATH includes the TimesTen `install_dir/lib` directory on Linux systems.
- PATH includes the TimesTen `install_dir/bin` directory on Linux systems or the TimesTen `install_dir\bin` directory on Microsoft Windows systems.

## Defining a TimesTen database named connection

To create a named connection to a TimesTen database, make sure you are on the main SQL Developer page.

1. Right-click the **Connections** node and select **New Connection**.

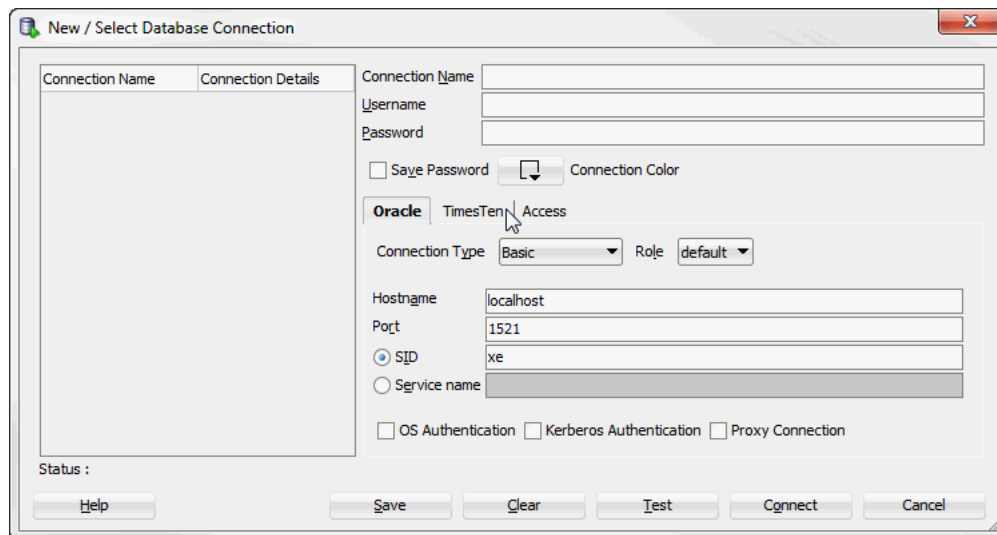
**Figure 1–1 Creating a new connection**



The **New/Select Database Connection** dialog appears.

2. Select the **TimesTen** tab.

**Figure 1–2 New database connection**

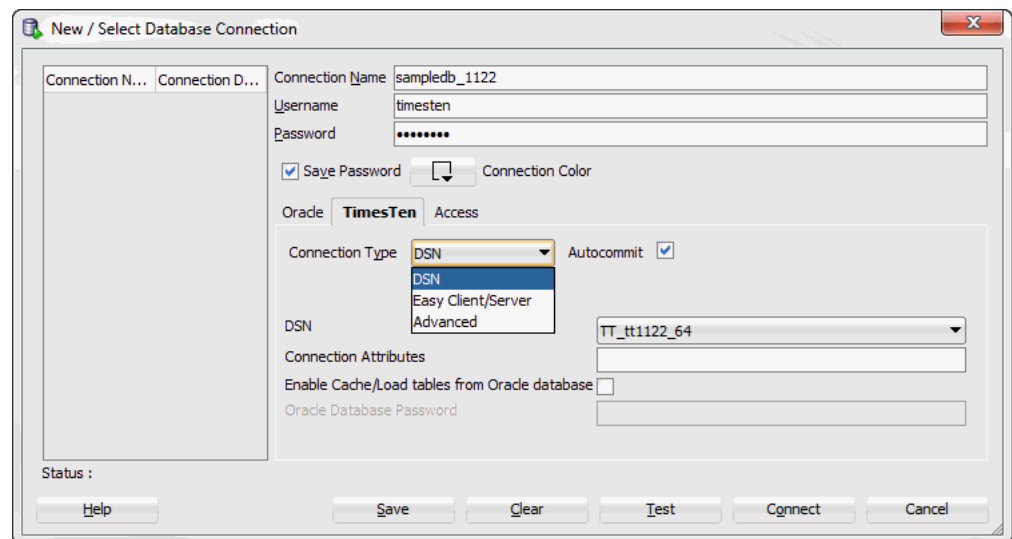


You are ready to enter information for the connection.

3. In the **Connection Name** field, enter the connection name.
4. In the **Username** field, enter the TimesTen user. If you use the TimesTen database to cache data from an Oracle database, enter the TimesTen cache manager user name. If you use the TimesTen database to load data from an Oracle database, enter the TimesTen user name that has **SELECT** privileges on the Oracle database tables that you want to load.

5. In the **Password** field, enter the password for the TimesTen user.  
To save the password, choose **Save Password**. If you save the password, you will not be prompted for the password on subsequent connections to the TimesTen database.
6. To allow TimesTen to issue an implicit commit after DML statements, select the **Autocommit** check box. By default, the **Autocommit** check box is selected.
7. Select a connection type from the **Connection Type** drop-down list. The following types of connections are available:
  - [DSN](#)
  - [Easy Client/Server](#)
  - [Advanced](#)

**Figure 1–3 New connection type**



You are now ready to define your new connection.

## DSN

The DSN connection type allows you to connect to databases by specifying a DSN. The DSN must exist in your `.ODBC.INI` on UNIX and the ODBC Data Source Administrator on Windows. Before defining a DSN connection in SQL Developer, make sure you have completed the steps up to step 7 from ["Defining a TimesTen database named connection"](#) on page 1-2.

---

**Note:** For more information about defining DSNs, see "Specifying Data Source Names to identify TimesTen databases" in the *Oracle TimesTen In-Memory Database Operations Guide*.

---

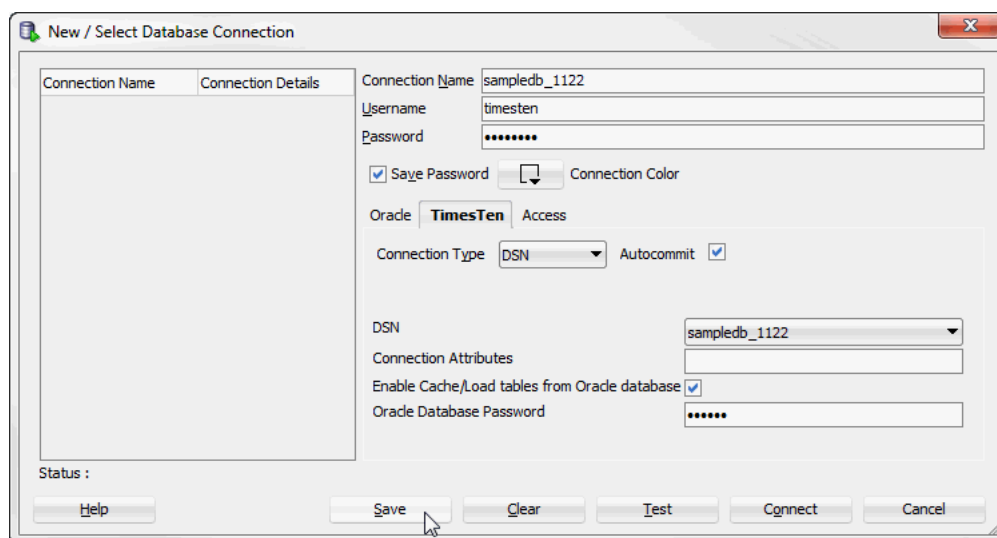
1. Select a predefined DSN that references the TimesTen database from the **DSN** drop-down list. You can select either a Data Manager DSN to establish a direct connection or a client DSN to establish a client/server connection.
2. In the **Connection Attributes** field, enter the attributes for the DSN. The **Connection Attributes** field is not required. Each attribute setting must be

separated by a semicolon. If you do not specify any connection attributes, TimesTen uses the connection attributes specified in the DSN.

For more information about connection attributes, see "Connection Attributes" in the *Oracle TimesTen In-Memory Database Reference*.

3. To cache data from an Oracle database or load data from an Oracle database into a TimesTen database, select the **Enable Cache/Load tables from Oracle database** check box.
4. In the **Oracle Database Password** field, enter the Oracle database password for the Oracle database user. The Oracle database user must be the same user as the TimesTen user you specified in the **Username** field. This text field is only available if you selected the **Enable Cache/Load tables from Oracle database** check box.
5. Click **Save** to save the connection name under the **Connections** node.

**Figure 1–4 Save DSN connection**



Your connection is saved. You can establish connections to the database using the settings in the named connection without having to specify the values in the fields of the **New/Select Database Connection** dialog box again.

6. Click **Connect**.

SQL Developer is connected to your TimesTen database.

For more information about creating and managing database connections in SQL Developer, see the *Oracle SQL Developer User's Guide*.

## Easy Client/Server

The easy client/server connection type allows you to connect to local and remote databases without having to define a DSN. Before defining an easy client/server connection in SQL Developer, make sure you have completed the steps up to step 7 from "[Defining a TimesTen database named connection](#)" on page 1-2.

1. In the **Server Host Name** field, enter the name or the IP address of the TimesTen server host.
2. In the **Server DSN** field, enter the DSN of the TimesTen database that is located on the TimesTen server host.

3. In the **Server Port** field, enter the TCP port of the TimesTen server. You can use the `ttStatus` utility to find the TCP port of your TimesTen server.

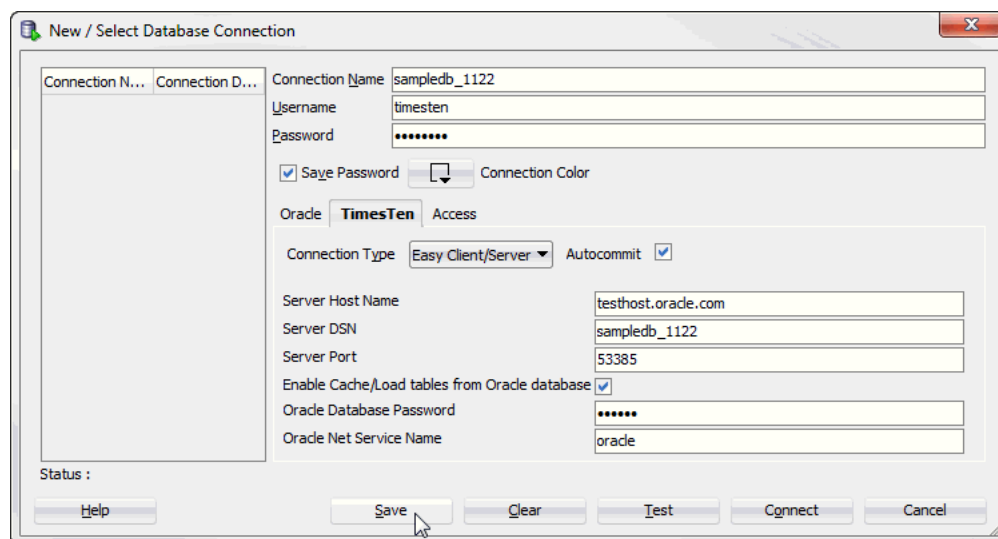
In the following example, the server port number is 53396. This example is part of a sample output from the `ttStatus` utility.

```
TimesTen server pid 24379 started on port 53396
```

For more information about the `ttStatus` utility, see "ttStatus" in the *Oracle TimesTen In-Memory Database Reference*.

4. To cache data from an Oracle database or load data from an Oracle database into a TimesTen database, select the **Enable Cache/Load tables from Oracle database** check box.
5. In the **Oracle Database Password** field, enter the Oracle database password for the Oracle database user. The Oracle database user must be the same user as the TimesTen user you specified in the **Username** field. This text field is only available if you selected the **Enable Cache/Load tables from Oracle database** check box.
6. In the **Oracle Net Service Name** field, enter the Oracle database net service name for the Oracle database. This text field is only available if you selected the **Enable Cache/Load tables from Oracle database** check box.
7. Click **Save** to save the connection name under the **Connections** node.

**Figure 1–5 Save easy client/server connection**



Your connection is saved. You can establish connections to the database using the settings in the named connection without having to specify the values in the fields of the **New/Select Database Connection** dialog box again.

8. Click **Connect**.

SQL Developer is connected to your TimesTen database.

For more information about creating and managing database connections in SQL Developer, see the *Oracle SQL Developer User's Guide*.

## Advanced

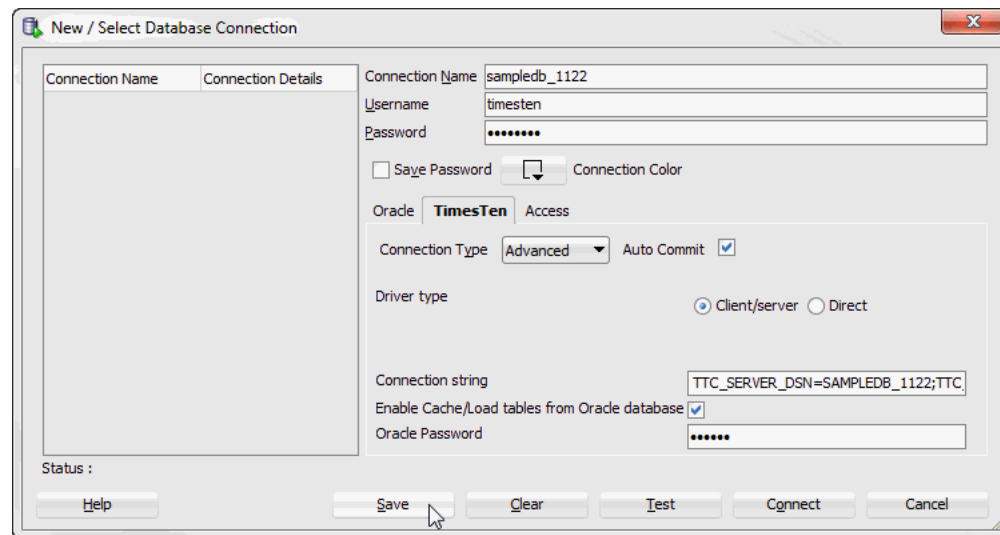
The advanced connection type allows you to define a connection string to connect to your TimesTen database. Before defining an advanced connection in SQL Developer, make sure you have completed the steps up to step 7 from ["Defining a TimesTen database named connection"](#) on page 1-2.

1. Select a driver type for the TimesTen database from the **Driver Type** option. Select either **Client/server** to establish a client/server connection or **Direct** to establish a direct connection.
2. In the **Connection string** field, enter a connection string to define the DSN and set attributes for the DSN. If you choose Client/Server as the Driver Type specify the server host, server port, and DSN. Each attribute setting must be separated by a semicolon. Do not delimit the connection string with a pair of double quotation marks.

For information about the format and contents of a connection string, see "Connecting to a database using a connection string" in the *Oracle TimesTen In-Memory Database Operations Guide*.

3. To cache data from an Oracle database or load data from an Oracle database into a TimesTen database, select the **Enable Cache/Load tables from Oracle database** check box.
4. In the **Oracle Database Password** field, enter the Oracle database password for the Oracle database user. The Oracle database user must be the same user as the TimesTen user you specified in the **Username** field. This text field is only available if you selected the **Enable Cache/Load tables from Oracle database** check box.
5. Click **Save** to save the connection name under the **Connections** node.

**Figure 1–6 Save advanced connection**



Your connection is saved. You can establish connections to the database using the settings in the named connection without having to specify the values in the fields of the **New/Select Database Connection** dialog box again.

6. Click **Connect**.

SQL Developer is connected to your TimesTen database.

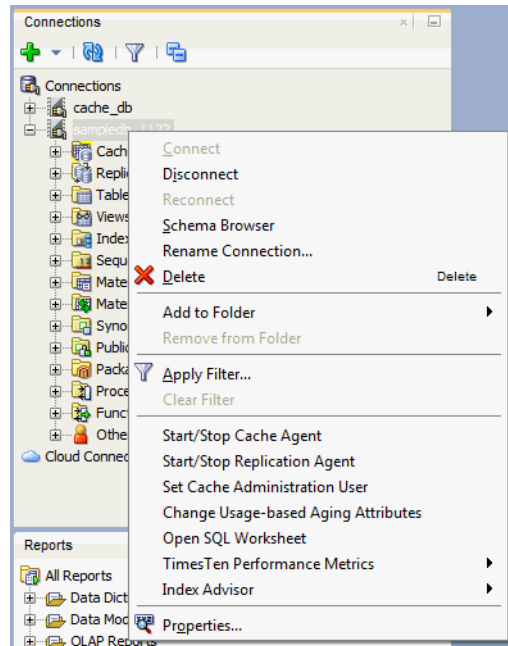


For more information about creating and managing database connections in SQL Developer, see the *Oracle SQL Developer User's Guide*.

## Using an existing TimesTen database named connection

If a named connection for the TimesTen database exists under the **Connections** node, then right-click the node of the connection name and select **Connect**, or click the **+** to the left of the connection name's node.

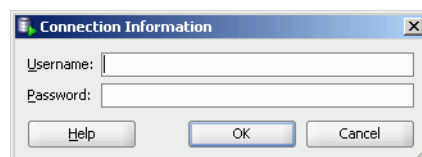
**Figure 1–7** Viewing a TimesTen connection



If the **Connection Information** dialog appears, in the **Username** field, enter the TimesTen user name. In the **Password** field, enter the password for the user.

Click **OK**.

**Figure 1–8** Connection Information dialog



After you connect to a TimesTen database, you can browse existing objects or create new objects such as tables, indexes, views, sequences, synonyms, PL/SQL packages, procedures and functions and so on. For information about creating database objects, see the *Oracle SQL Developer User's Guide*.

## Disconnecting an existing TimesTen database connection

To disconnect an existing connection from a TimesTen database, choose the **Connections** tab. Right-click the node of the connection name and select **Disconnect**.

For more information, see [Figure 1–7, "Viewing a TimesTen connection"](#).



---

## Working with TimesTen Objects

This chapter focuses on working with TimesTen objects in SQL Developer.

Topics include:

- Locating a TimesTen database object
- Specifying and displaying LOB data types
- Specifying the `INLINE` attribute for columns
- Computing table size information
- Using a `REF CURSOR` as an `OUT` parameter
- Viewing the characteristics of a table
- Creating an index on a table
- Creating a primary key constraint
- Changing the attributes of primary key
- Altering the aging properties of a table
- Granting and revoking object privileges
- Viewing the privileges granted on a database object
- Loading data into tables
- `ttIsql` commands in SQL worksheet

### Locating a TimesTen database object

If you are unable to find a database object, right-click the node corresponding to the object type and select **Refresh** to refresh the list of objects of that type.

For example, to find a cache group, right-click the **Cache Groups** node and select **Refresh** to refresh the list of cache groups.

You can also check if the object appears under the **Other Users** node. Click the + to the left of the **Other Users** node to expand the node. Then click the + to the left of any of the users' nodes to expand that user's node.

Click the + to the left of the node corresponding to the object type to view the list of objects of that type owned by the user.

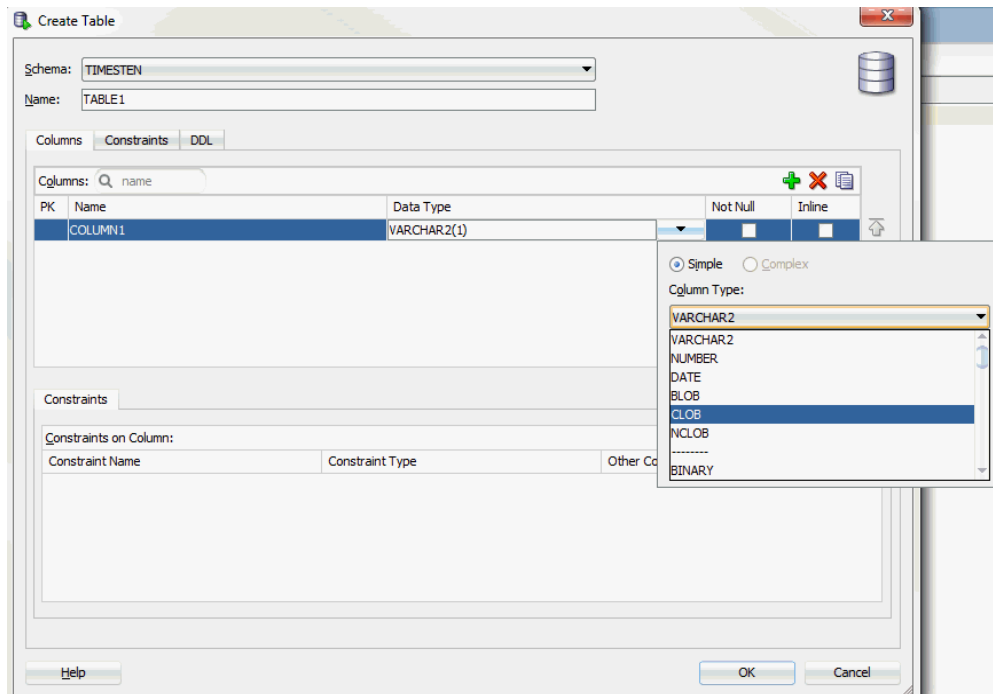
For example, to find a table, click the + to the left of the user's **Tables** node to view the list of tables owned by that user.

## Specifying and displaying LOB data types

If you are using TimesTen release 11.2.2 or later, then you can specify and display LOB data types in your table definitions.

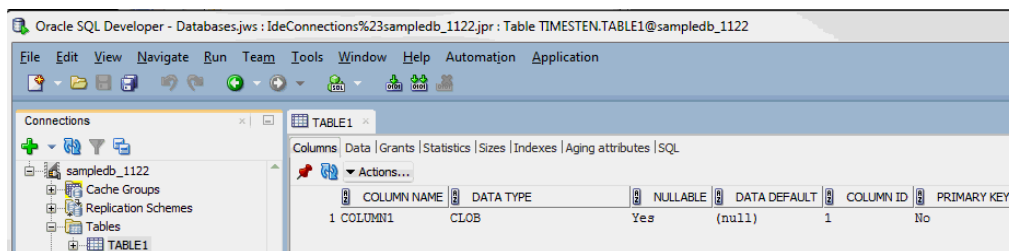
If you wish to create a table with a LOB data type, then in the **Create Table** dialog, expand the **Type** column header. You see CLOB, NCLOB, and BLOB as valid data types.

**Figure 2–1 Specifying LOB data types in the Create Table dialog**

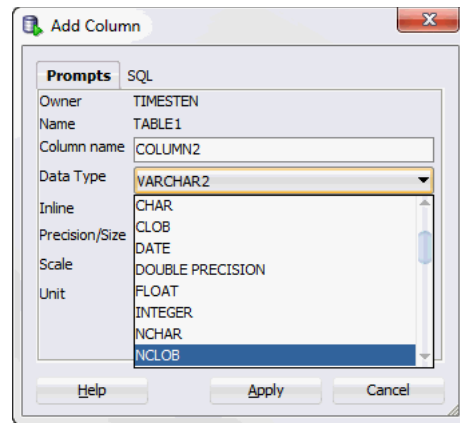


To view the data type for your columns, select the table and then choose the **Columns** tab. You see the LOB data type for your column.

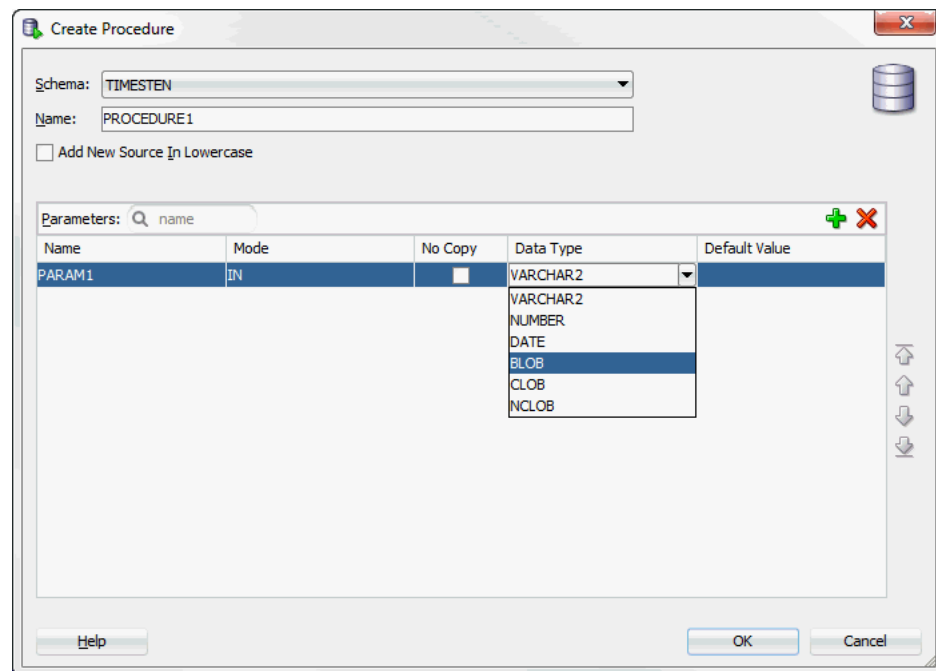
**Figure 2–2 Displaying Column data types**



If you wish to add a column and specify a LOB data type, in the **Add Column** dialog, expand **Data Type**. You see CLOB, NCLOB, and BLOB among the possible choices.

**Figure 2–3 Adding column and specifying a LOB data type**

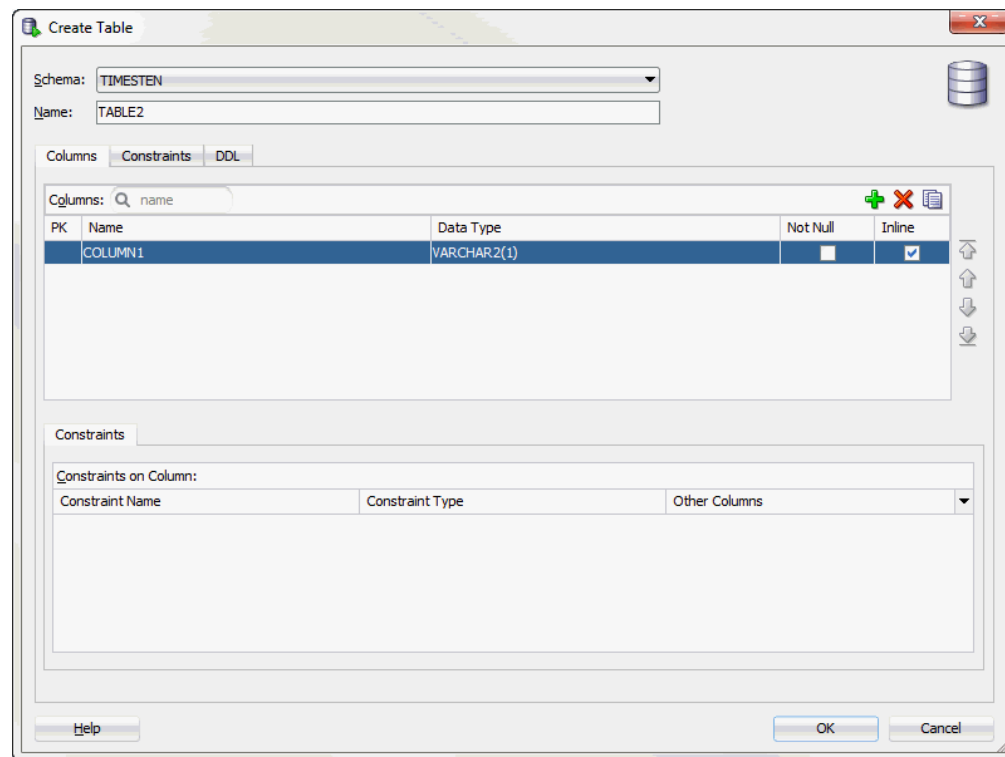
For PL/SQL objects, you can specify LOB data types as `IN`, `OUT`, and `IN OUT` parameters. In the PL/SQL edit dialogs, for parameter types, choose `CLOB`, `NCLOB`, or `BLOB`.

**Figure 2–4 Choosing a parameter type**

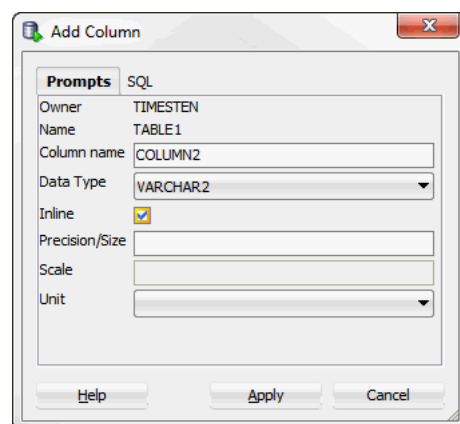
## Specifying the INLINE attribute for columns

You can specify the `INLINE` attribute for columns of type `VARCHAR2`, `NVARCHAR2`, and `VARBINARY`.

In the **Create Table** dialog, locate the column header named **Inline**. Click in the check box to define the column with the `INLINE` attribute.

**Figure 2–5 Viewing the Inline attribute in the Create Table dialog**

You can also specify the `INLINE` attribute when adding a column to a table. Right-click on the table, select **Column**, then select **Add**. The **Add Column** dialog appears. For columns of type `VARCHAR2`, `NVARCHAR2`, and `VARBINARY`, click in the **Inline** check box to add the column with the `INLINE` attribute.

**Figure 2–6 Viewing the Inline attribute in the Add Column dialog**

## Computing table size information

If you are using TimesTen Release 11.2.2 or later, you can compute table size information.

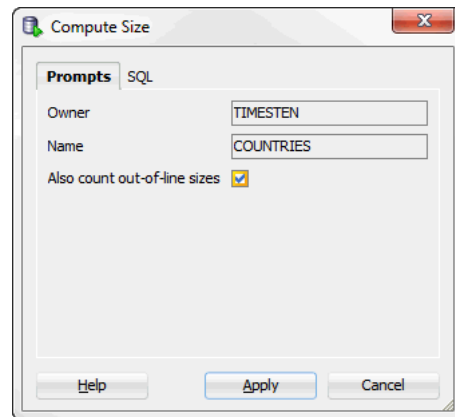
To view table size information, you must first compute the table size:

1. Choose **+** to the left of the **Tables** node to view the list of tables.

2. Right-click the name of the table to compute table size information.
3. Select **Table**, then select **Compute Size**.

The **Compute Size** dialog appears. The owner and name of the table are displayed.

**Figure 2–7 Compute Size dialog**



4. Click in the check box if you wish to compute out-of-line sizes.
5. Choose **Apply**.
6. Choose **OK**.

A **Confirmation** dialog appears.

The table size statistics for the table are computed.

To view the SQL for computing the table size, choose the **SQL** tab in the **Compute Size** dialog. You see that a TimesTen built-in procedure called `ttComputeTabSizes` is executed. After this built-in is executed, you can review the table size information for your table. Note that this table size information is on a per table basis.

To compute table sizes for all tables in your database including materialized views, system tables, and tables that are part of cache groups, use the **SQL Worksheet** and execute the command: `Call ttComputeTabSizes (NULL,0);` or to include out-of-line data: `Call ttComputeTabSizes (NULL,1);`.

After you compute the table size for one or more tables, you can view the table size information:

1. Choose **+** to the left of the **Tables** node to view the list of tables.
2. Choose the table you wish to review table size information.
3. Choose the **Sizes** tab located among the tabs that display the attributes of the table.

Table size information is displayed. Specifically, values for `INLINE BYTES`, `OUT OF LINE BYTES`, and `METADATA BYTES` are shown.

**Figure 2–8 Displaying table size information**

Columns   Data   Grants   Statistics   <b>Sizes</b>   Indexes   Aging attributes   SQL	
<div> </div> <div>Actions...</div>	
Name	Value
1 INLINE BYTES	25632
2 OUT OF LINE BYTES (null)	
3 METADATA BYTES	1288

You can choose the **Actions** menu to generate or regenerate the table size. If you select the **Actions** menu, then **Table**, then select **Compute Size**. The **Compute Size** dialog appears allowing you to compute the table size information.

Choose **Refresh** to refresh the displayed table size data.

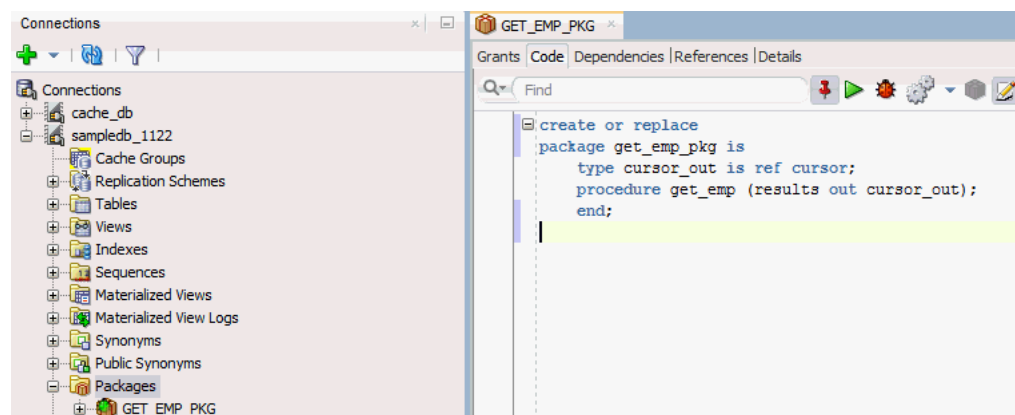
You can view table size information for all tables that you have computed table sizes. To view such information, select **TimesTen Reports**, then **Table**, then select **Table Sizes**. For more information on TimesTen reports, see ["Generating TimesTen Reports"](#) on page 12-1.

For more information on the `ttComputeTabSizes` built-in procedure, see `"ttComputeTabSizes"` in *Oracle TimesTen In-Memory Database Reference*.

## Using a REF CURSOR as an OUT parameter

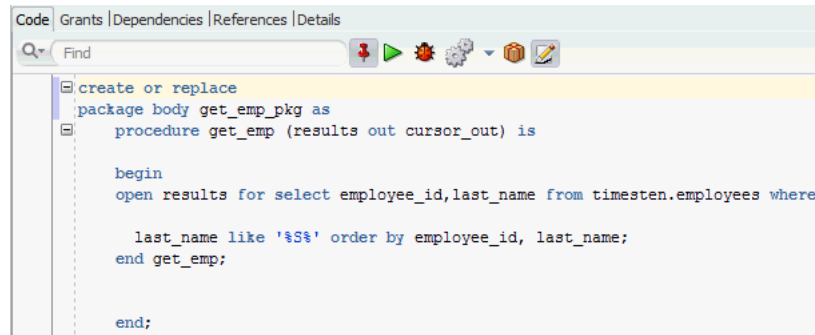
You can test a PL/SQL function, procedure, or package by defining a REF CURSOR as an OUT parameter in your PL/SQL function, procedure, or package. After you define a REF CURSOR, compile and run your PL/SQL function, procedure, or package. The **Run** dialog appears and when you choose **OK**, the details of the run are displayed and the output from the execution of the function, procedure, or package is displayed in the **Output Variables** tab.

In the following example, create a package called `get_emp_pkg`. The `get_emp_pkg` package defines a REF CURSOR as an OUT parameter and defines a procedure that uses the REF CURSOR as an OUT parameter.

**Figure 2–9 get\_emp\_pkg definition**

Next create a package body that defines the procedure `get_emp`. The procedure `get_emp` opens the REF CURSOR variable and performs a query on the employees table.



**Figure 2–10** *get\_emp\_pkg package body definition*


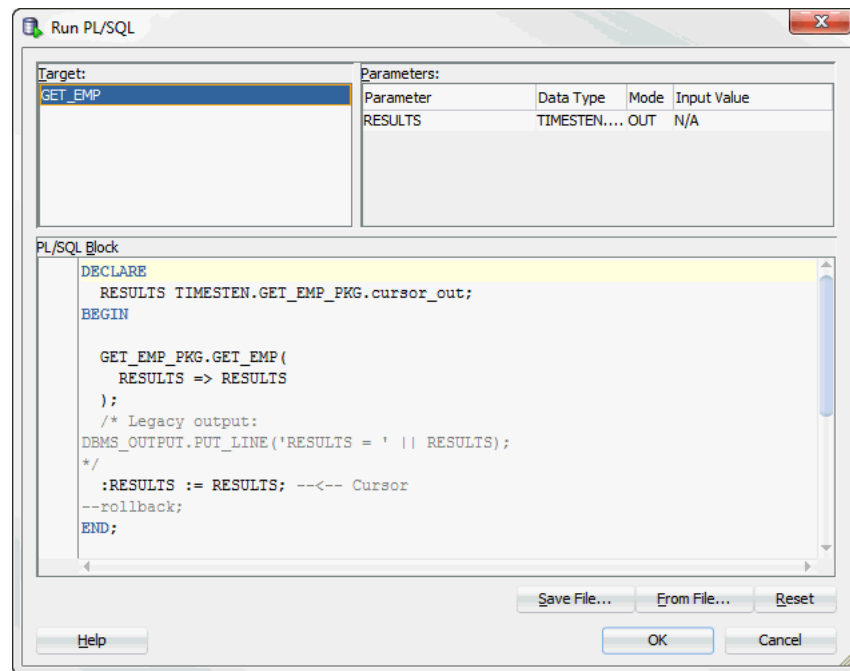
```

create or replace
package body get_emp_pkg as
  procedure get_emp (results out cursor_out) is
  begin
    open results for select employee_id,last_name from timesten.employees where
      last_name like '%S%' order by employee_id, last_name;
  end get_emp;

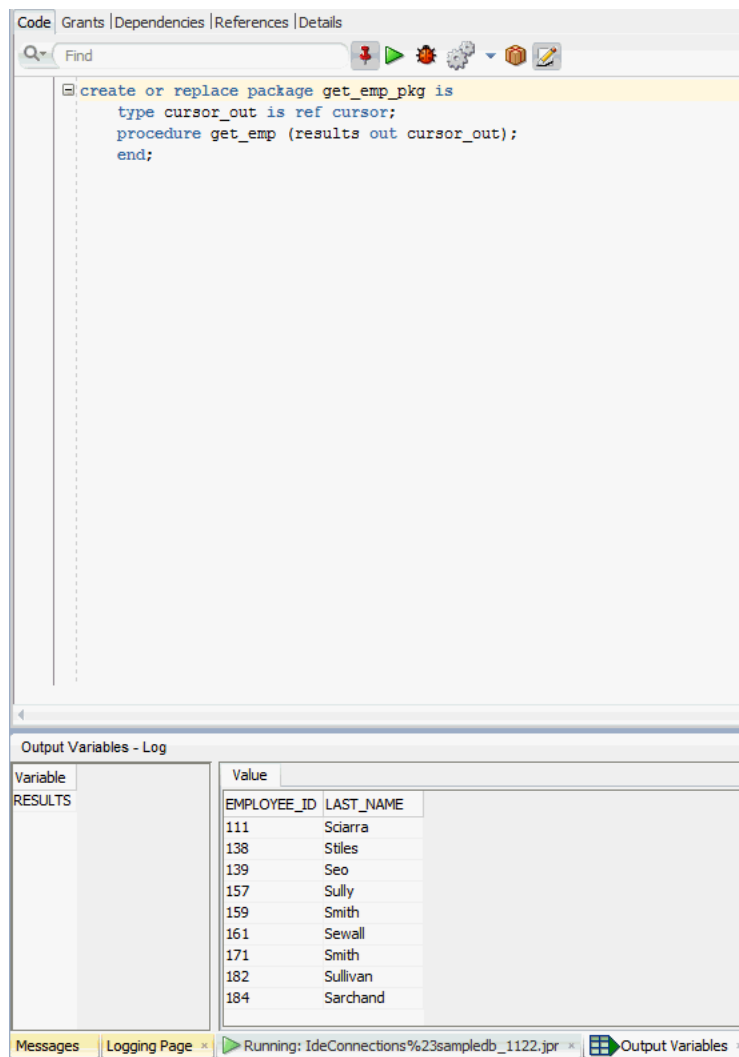
end;

```

Compile the package and package body. Right-click on the `get_emp_pkg` and choose **Run**. The **Run PL/SQL** dialog displays allowing you to run the test wrapper. Choose **OK** to run the test wrapper.

**Figure 2–11** *Viewing the Run PL/SQL dialog*

1. **IdeConnections:** Displays the status of your run including any errors encountered.
2. **Output Variables:** Shows the output from the execution of the function, procedure, or package.

**Figure 2–12 Viewing the Output Variables tab**

## Viewing the characteristics of a table

After you have created a regular table, a global temporary table or a cache table, you can view the characteristics of the table itself such as its columns, indexes and aging policy.

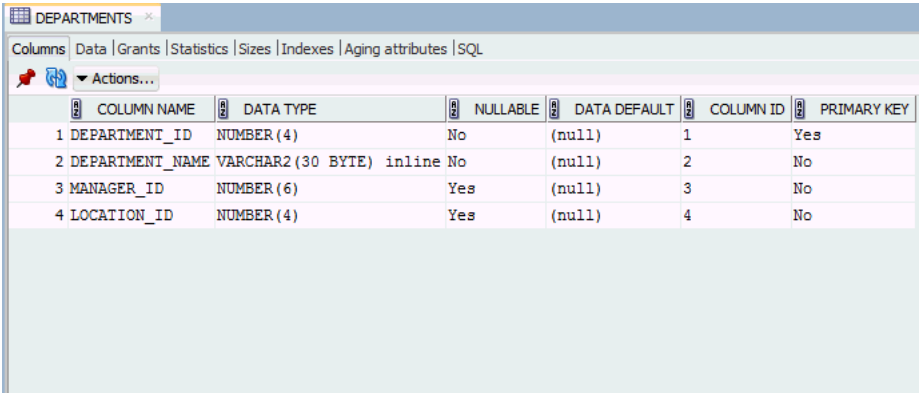
Click the + to the left of the **Tables** node to view the list of tables. Click the name of the table to view.

If you are unable to find the table that you are looking for, see ["Locating a TimesTen database object"](#) on page 2-1.

## Viewing the columns

The **Columns** tab, located within the table's tab, shows information about the columns of a table. The name of the column, data type, nullability property and default value is displayed.

To view the list of columns in a table, locate the **Tables** node. Click the + to the left of the node that contains the name of the table.

**Figure 2–13 Viewing the columns**


	COLUMN NAME	DATA TYPE	NULLABLE	DATA DEFAULT	COLUMN ID	PRIMARY KEY
1	DEPARTMENT_ID	NUMBER(4)	No	(null)	1	Yes
2	DEPARTMENT_NAME	VARCHAR2(30 BYTE) inline	No	(null)	2	No
3	MANAGER_ID	NUMBER(6)	Yes	(null)	3	No
4	LOCATION_ID	NUMBER(4)	Yes	(null)	4	No

## Viewing the indexes

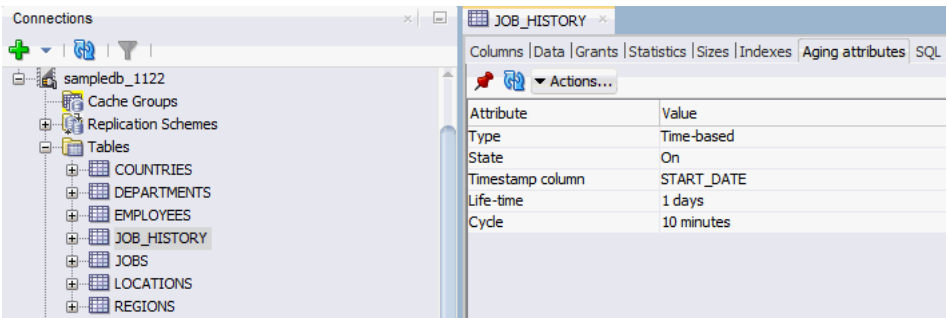
The **Indexes** tab, located within the table tab, shows information about the index for a table. The name of the index, type, uniqueness property, and the number and name of the columns are displayed.

## Viewing the aging attributes

The **Aging attributes** tab, located within the table tab, shows information about the aging attributes for a table. The aging policy type, aging cycle, and aging state are displayed.

For tables that have an LRU aging policy defined, the TimesTen database memory usage thresholds are displayed. For information about how to change these thresholds and the LRU aging cycle, see ["Specifying an aging policy"](#) on page 7-8.

For tables that have a time-based aging policy defined, the name of the non-nullable DATE or TIMESTAMP column used to store the timestamp value (indicating when each row was added or most recently updated) and the length of time non-updated data is not deleted from the table is displayed.

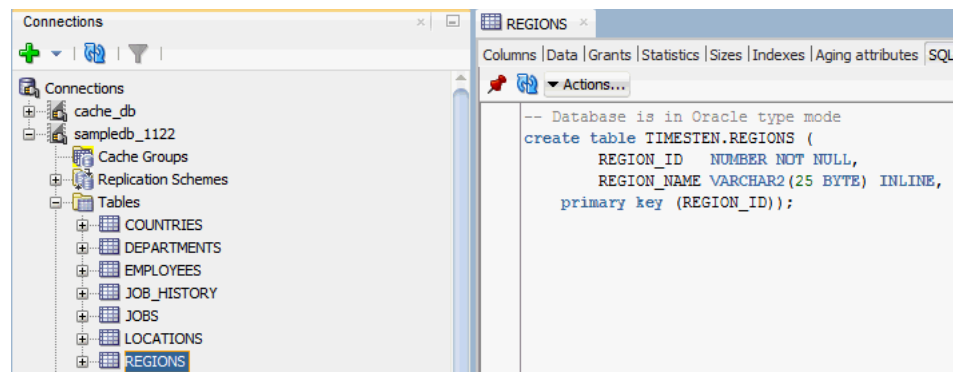
**Figure 2–14 Viewing the aging attributes for a table**


Attribute	Value
Type	Time-based
State	On
Timestamp column	START_DATE
Life-time	1 days
Cycle	10 minutes

## Viewing the table definition

The **SQL** tab, located within the table tab, shows the CREATE TABLE statement that was used to create the table for a regular table or a global temporary table.

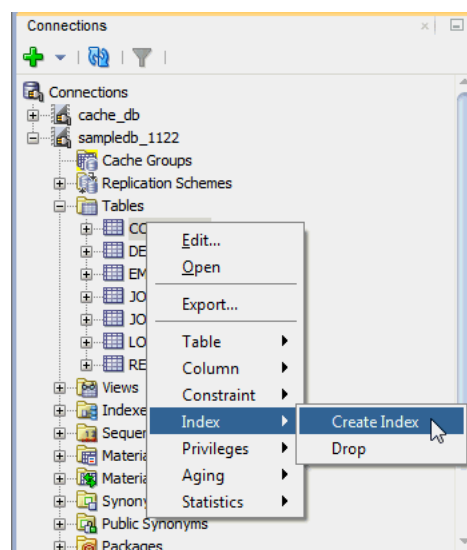
Since a cache table is created when its accompanying cache group is created, you can view a cache table's definition by viewing its cache group definition. See ["Viewing the cache group definition"](#) on page 9-3.

**Figure 2–15 Viewing the SQL for the table definition**

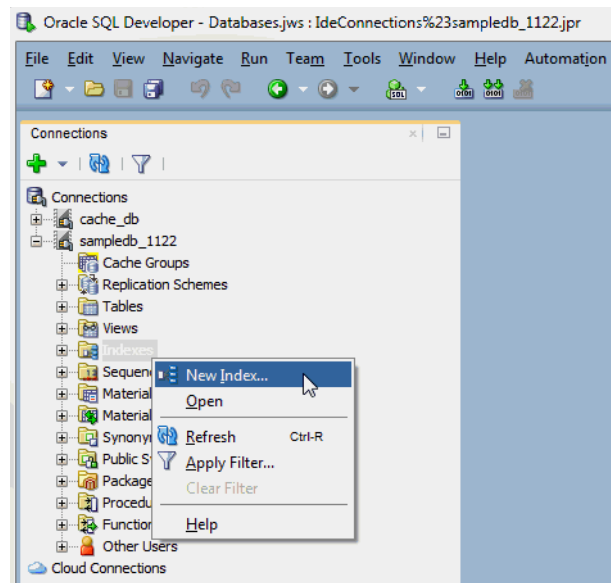
## Creating an index on a table

You can create a range index or a bitmap index on a regular table or on a cache table.

To create an index, click the + to the left of the **Tables** node to view the list of tables. Then right-click the name of the table to create an index on and select **Index**, then select **Create Index**.

**Figure 2–16 Creating an index from Tables node**

You can also right-click the **Indexes** node and select **New Index**.

**Figure 2–17** *Creating a new index from Indexes node*

In the **Properties** tab of the **Create INDEX** dialog, locate the **Table** drop-down menu. From this menu, select the table for the index.

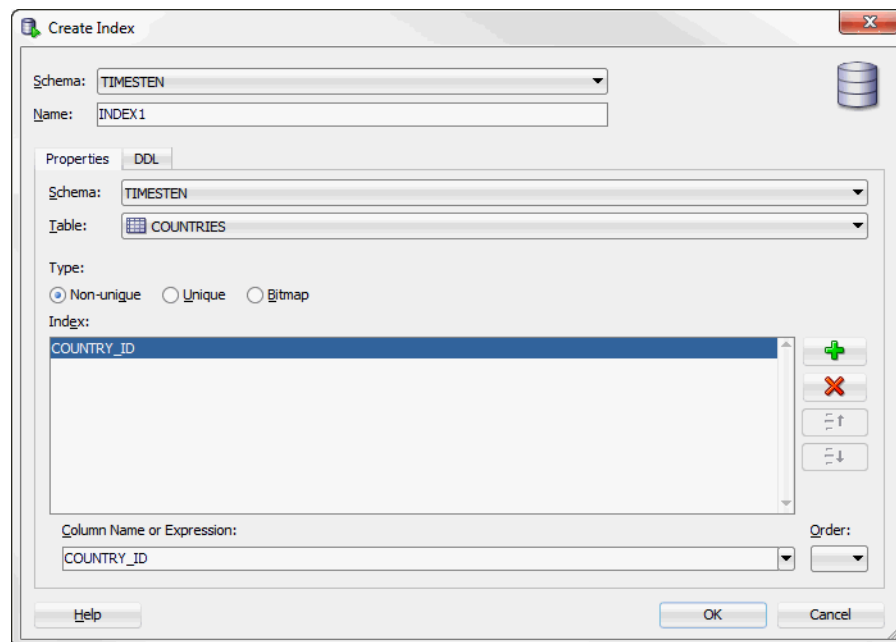
For **Type**:

- To create a non-unique range index, choose **Non-unique**.
- To create a unique range index, choose **Unique**.
- To create a bitmap index, choose **Bitmap**.

For information about the different types of indexes supported in TimesTen, see "CREATE INDEX" in the *Oracle TimesTen In-Memory Database SQL Reference*.

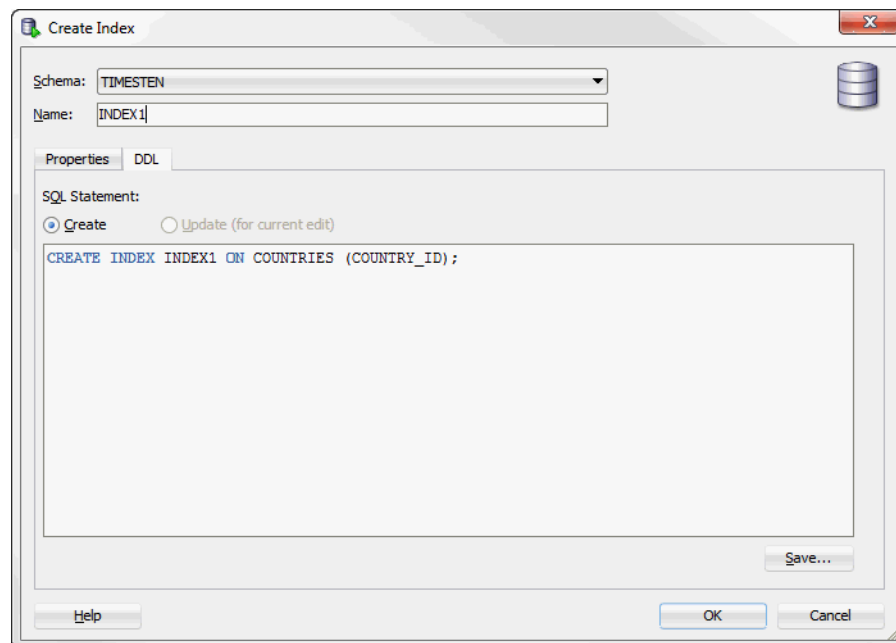
In the **Index** section, specify the columns for the index. You can select the columns from the **Column Name or Expression** drop-down select list. For each indexed column, from the **Order** select list, you can specify whether the column is to be sorted in ascending or descending order. The default sort order is ascending.

To create a composite index, click **+** to add columns to the index definition. Click **X** to remove columns from the index definition.

**Figure 2–18** Create index dialog

In the **DDL** tab of the **Create Index** dialog, you can view the **CREATE INDEX** statement used to create the index.

Click **OK** to create the index.

**Figure 2–19** Viewing the DDL for create index

## Creating a primary key constraint

You can create a primary key constraint on a table that does not have a primary key defined. Follow these steps:

1. Click the + to the left of the **Tables** node to view the list of tables. Then right-click the name of the table to create a primary key constraint.
2. Select **Constraint**, then select **Add Primary Key**.

The **Add primary key** dialog appears.

3. In the **key index type** field choose either **Use range** to create a primary key using a range index or **Use hash** to create a primary key using a hash index.

If you choose **Use hash**, you see the text **with number of** followed by a drop-down list of either **row** or **pages**. Select either **row** or **pages** and in the box after the **colon (:**), enter the number of rows or pages. The default selection is **row**.

For more information on hash index sizing, see "Column Definition" in *Oracle TimesTen In-Memory Database SQL Reference*.

4. In the **Constraint Name** field, enter the constraint name. The name cannot exceed 30 characters in length.
5. In the **Column** field(s), you see a drop-down list of column names. Choose the column name(s) to use for the primary key.
6. Click **Apply**.

The **Confirm** dialog box appears indicating that the primary key constraint has been added.

7. Click **OK**.

The primary key constraint has been added successfully.

To view the SQL statement used to create the primary key constraint, choose the **SQL** tab of the **Add primary key** dialog box.

For information about creating a primary key constraint, see "CREATE TABLE" in the *Oracle TimesTen In-Memory Database SQL Reference* or "ALTER TABLE" in the *Oracle TimesTen In-Memory Database SQL Reference*.

## Changing the attributes of primary key

You can change the attributes of a primary key constraint after you have defined a primary key on a table. Follow these steps:

1. Click the + to the left of the **Tables** node to view the list of tables. Then right-click the name of the table on which to change the primary key attributes.
2. Select **Constraint**, then select **Change Primary Key Parameters**.

The **Change parameters of a primary key** dialog appears.

3. In the **key index type** field choose either **Use range** to change the primary key constraint from using a hash index to using a range index or **Use hash** to change the primary key constraint from using a range index to using a hash index.

If you choose **Use hash**, you see the text **with number of** followed by a drop-down list of either **row** or **pages**. Select either **row** or **pages** and in the box after the **colon (:**), enter the number of rows or pages. The default selection is **row**.

For more information on hash index sizing, see "Column Definition" in *Oracle TimesTen In-Memory Database SQL Reference*.

4. Click **Apply**.

The **Confirm** dialog appears indicating that the index(es) for the primary key constraint has been changed.

5. Click **OK**.

The index(es) for the primary key constraint has been changed successfully.

To view the SQL statement used to change the attributes of the primary key constraint, choose the **SQL** tab of the **Change parameters of a primary key** dialog.

For information about changing the attributes of a primary key constraint, see "ALTER TABLE" in the *Oracle TimesTen In-Memory Database SQL Reference*.

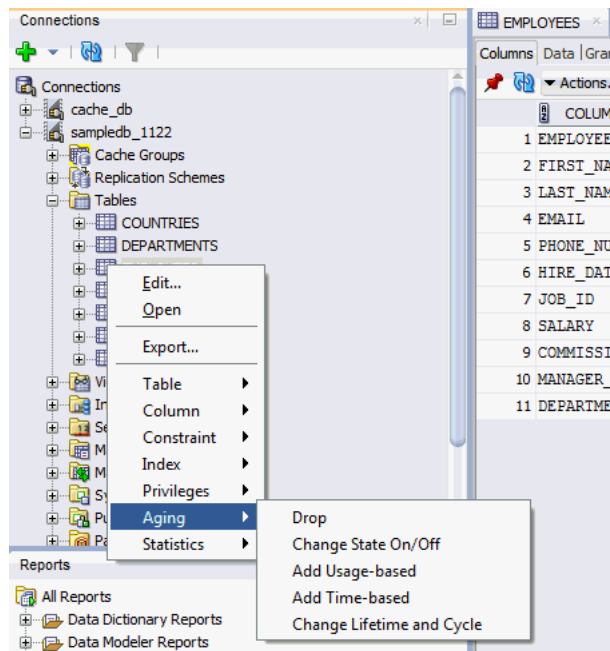
## Altering the aging properties of a table

You can add an aging policy to or drop an aging policy from a regular table or a cache table. You can also change the aging state for a table's existing aging policy, or change the lifetime and cycle for a table's existing time-based aging policy.

### Adding an aging policy to a table

To add an aging policy to a regular table or a cache table, under the **Tables** node, right-click the name of the table to add an aging policy to and select **Aging**, then select **Add Usage-based** to add an LRU aging policy. To add a time-based aging policy select **Aging**, then select **Add Time-based**. An aging policy can be added to a cache table only if it is the root table of a cache group.

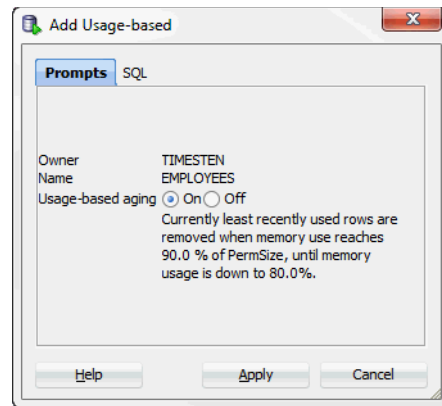
**Figure 2–20 Aging attributes**



If you are unable to find the table that you are looking for, see "[Locating a TimesTen database object](#)" on page 2-1.

To add an LRU aging policy to the table, in the **Prompts** tab of the **Add Usage-based** dialog box, specify an aging state by choosing either **On** or **Off** in the **Usage-based aging** field. The default aging state is on. Click **Apply** to add the LRU aging policy to the table.



**Figure 2–21 Adding usage-based aging**

The TimesTen database memory usage thresholds determine when data starts and stops being deleted from the table. The default memory usage threshold that determines when data starts being deleted from the table is 90%. The default memory usage threshold that determines when data stops being deleted from the table is 80%. The default LRU aging cycle is 1 minute. For information about how to change these thresholds and the LRU aging cycle., see ["Specifying an aging policy"](#) on page 7-8.

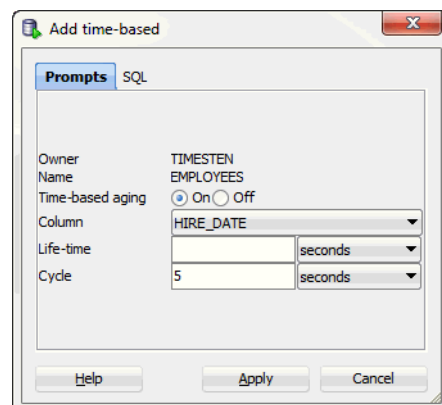
An LRU aging policy can be added to any regular table, and only to cache tables in an AWT, SWT or user managed cache group that does not have automatic refresh defined.

To add a time-based aging policy to the table, in the **Prompts** tab of the **Add time-based** dialog box, select the column name from **Column** to store the timestamp value indicating when each row was added or most recently updated.

In the **Life-time** field, indicate the length of time in which data that has not been updated is to be kept in the table by specifying a numeric value followed by a unit of minutes, hours or days in the drop-down menu.

In the **Cycle** field, indicate the frequency at which data is to be aged out of the table by specifying a numeric value followed by a unit of minutes, hours or days. The default time-based aging cycle is 5 minutes.

Specify an aging state by choosing either **On** or **Off** in the **Time-based aging** field. The default aging state is on. Then click **Apply** to add the time-based aging policy to the table.

**Figure 2–22 Adding time-based aging**

A time-based aging policy can only be added to a table that contains a non-nullable DATE or TIMESTAMP column.

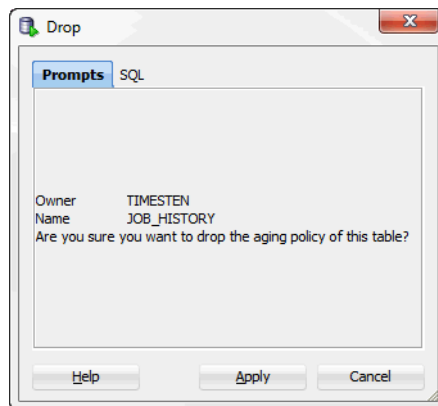
## Dropping an aging policy from a table

To drop an existing aging policy from a regular table or a cache table, under the **Tables** node, right-click the name of the table to drop an aging policy from and select **Aging**, then select **Drop**. See [Figure 2–20, "Aging attributes"](#).

If you are unable to find the table that you are looking for, see ["Locating a TimesTen database object"](#) on page 2-1.

Click **Apply** to drop the aging policy from the table.

**Figure 2–23 Drop aging policy dialog**

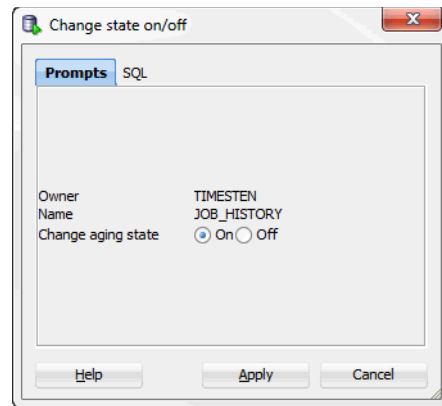


## Changing the aging state of a table's aging policy

To change the aging state of a regular table's or cache table's existing aging policy, under the **Tables** node, right-click the name of the table to change the aging state of and select **Aging**, then select **Change State On/Off**. See [Figure 2–20, "Aging attributes"](#).

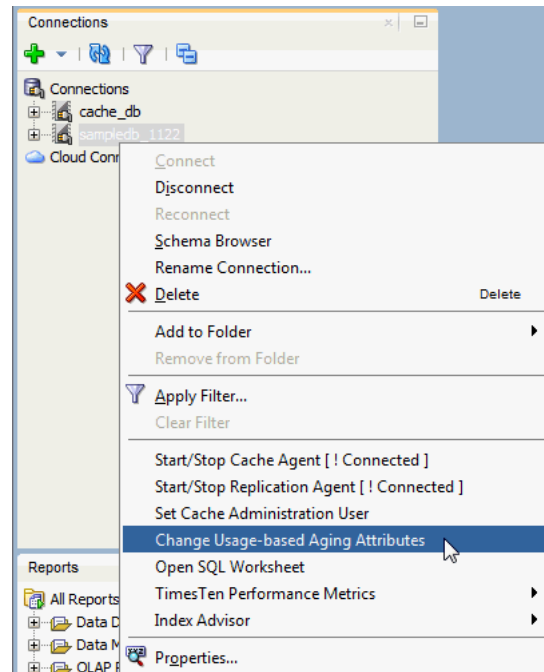
If you are unable to find the table that you are looking for, see ["Locating a TimesTen database object"](#) on page 2-1.

In the **Prompts** tab of the **Change state on/off** dialog, change the aging state by selecting either **On** or **Off** in the **Change aging state** field. Click **Apply** to change the aging state of the table.

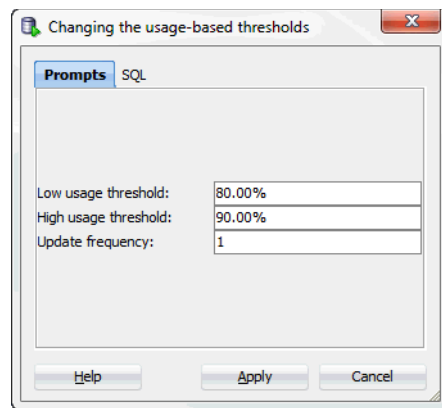
**Figure 2–24** *Changing aging state*

## Changing the memory usage thresholds and LRU aging cycle

To change the memory usage thresholds and the LRU aging cycle, right-click the node of the connection name for the TimesTen database and choose **Change Usage-based Aging Attributes**.

**Figure 2–25** *Change usage-based aging attributes*

In the **Prompts** tab of the **Changing the usage-based thresholds** dialog box, specify the threshold that determines when data starts being deleted from the tables in the **High usage threshold** field. Specify the threshold that determines when data stops being deleted from the tables in the **Low usage threshold** field. Specify the LRU aging cycle in the **Update Frequency** field. Then click **Apply** to change the memory usage thresholds and the LRU aging cycle.

**Figure 2–26 Usage-based aging thresholds dialog**

The new settings apply to all tables that have an LRU aging policy defined.

## Changing the lifetime and cycle of a table's time-based aging policy

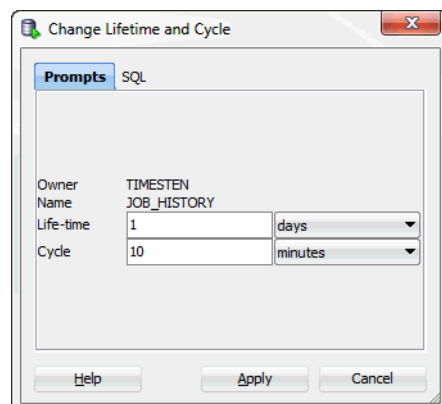
To change the lifetime and cycle of a regular table's or cache table's existing time-based aging policy, under the **Tables** node, right-click the name of the table to change the lifetime and cycle of and select **Aging**, then select **Change Lifetime and Cycle**. See [Figure 2–20, "Aging attributes"](#).

If you are unable to find the table that you are looking for, see ["Locating a TimesTen database object"](#) on page 2-1 for information.

In the **Life-time** field within the **Prompts** tab of the **Change Lifetime and Cycle** dialog box, change the length of time in which data that has not been updated is to be kept in the table by specifying a numeric value followed by a unit of minutes, hours or days.

In the **Cycle** field, change the frequency at which data is to be aged out of the table by specifying a numeric value followed by a unit of minutes, hours or days.

Then click **Apply** to change the lifetime and cycle for the table.

**Figure 2–27 Changing lifetime and cycle**

## Granting and revoking object privileges

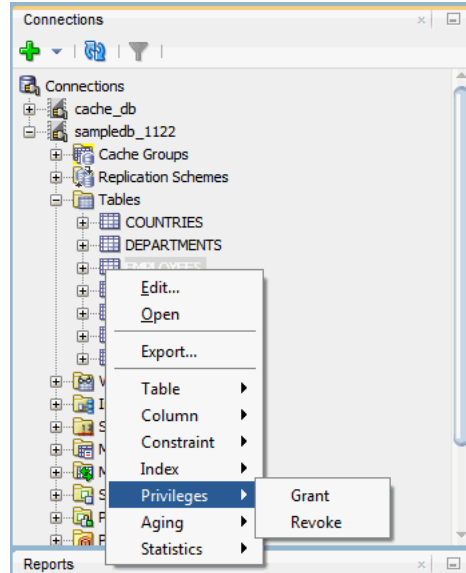
You can grant privileges to and revoke privileges from the following database objects:

- Regular and cache tables

- Regular and materialized views
- Sequences
- PL/SQL packages, procedures and functions

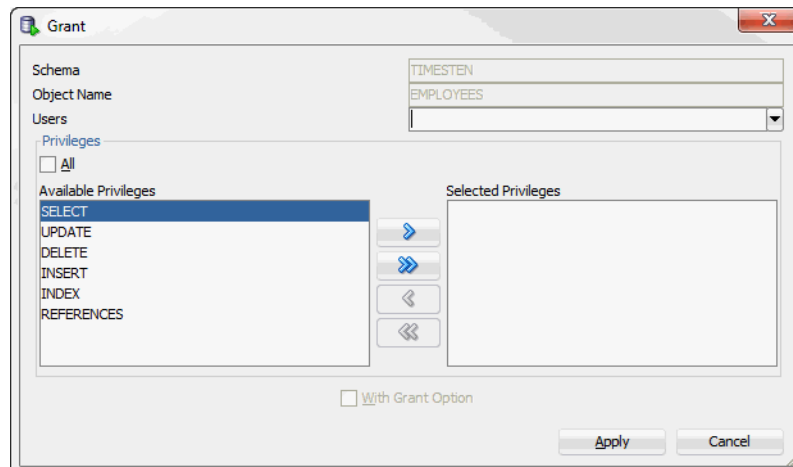
To grant privileges on an object, right-click the name of the object and select **Privileges**, then select **Grant**.

**Figure 2–28 Granting and revoking privileges**



In the **Users** drop-down menu of the **Grant** dialog box, select the user to grant object privileges to the object. In the **Privileges** section, select the **All** check box or click >> to grant all available object privileges on the object to the selected user. Otherwise, select the individual privileges from the **Available Privileges** list for the privileges to grant to the selected user and then click > to move those privileges into the **Selected Privileges** list. To select multiple privileges, press and hold the CTRL key, and click the desired privileges. Click **Apply** to grant the selected object privileges on the object to the selected user.

**Figure 2–29 Grant privileges dialog**



For a particular object type, only the available object privileges are shown. For example, DELETE, INDEX, INSERT, REFERENCES, SELECT and UPDATE privileges can be granted on a table. However, only INDEX, REFERENCES and SELECT privileges can be granted on a materialized view.

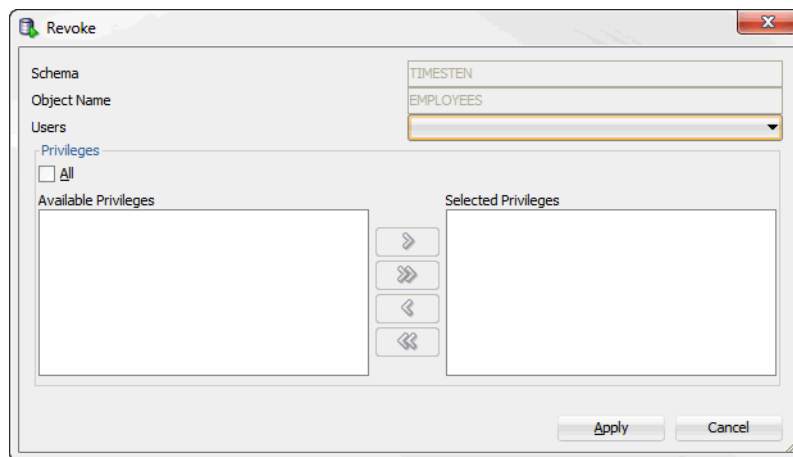
Only users with the ADMIN system privilege or the owner of an object can grant object privileges on the object.

For more information about the set of privileges available to each type of object, see "Object privileges" in the *Oracle TimesTen In-Memory Database SQL Reference*.

To revoke privileges from an object, right-click the name of the object and select **Privileges**, then select **Revoke**. See [Figure 2–28, "Granting and revoking privileges"](#).

In the **Users** drop-down menu of the **Revoke** dialog box, select the user to revoke object privileges from the object. In the **Privileges** section, select the **All** check box or click >> to revoke all granted object privileges on the object from the selected user. Otherwise, select the individual privileges from the **Available Privileges** list for the privileges to revoke from the selected user and then click > to move those privileges into the **Selected Privileges** list. To select multiple privileges, press and hold the CTRL key, and click the desired privileges. Click **Apply** to revoke the selected object privileges on the object from the selected user.

**Figure 2–30 Revoke privileges dialog**



The **Users** drop-down menu shows only users that have object privileges on the object. For a particular user, only the object privileges that they have been granted are shown.

Only users with the ADMIN system privilege or the owner of an object can revoke object privileges from the object.

## Viewing the privileges granted on a database object

You can view the object privileges granted to all users on a particular object such as a regular table or a cache table, a regular view or a materialized view, a sequence, or a PL/SQL package, procedure or function.

For example, to view the privileges granted on a regular table, click the + to the left of the user's **Tables** node to view the list of tables owned by that user and then click the name of the desired table.

The **Grants** tab, located within the table tab, shows the users who have privileges on the table, what privileges they have been granted, and the user who granted the privileges.

## Loading data into tables

You can load data into your table using **Import Data**. Before you can load your data, you must export the data into a file. To export the data, right-click on the name of the table and choose **Export**. After successfully exporting your data, you can import the data.

1. Choose **+** to the left of the **Tables** node to view the list of tables.
2. Right-click on the table name.
3. Select **Table**, then select **Import Data**.

An Open dialog appears. Locate the data file to import. This file is the file you previously exported.

4. Choose **Open**.

The data is loaded into the table.

## ttlsql commands in SQL worksheet

The following *ttlsql* commands are supported in the SQL Worksheet:

- `autocommit`
- `desc`
- `dssize`
- `version`

For more information, see "ttlsql" in the *Oracle TimesTen In-Memory Database Reference*.





---

## Interfacing with the TimesTen Query Optimizer

This chapter shows you how to interact with the TimesTen query optimizer in SQL Developer. Topics include:

- [Updating the table and column statistics used by the query optimizer](#)
- [Exporting table statistics](#)
- [Viewing the execution plan of a statement](#)

### Updating the table and column statistics used by the query optimizer

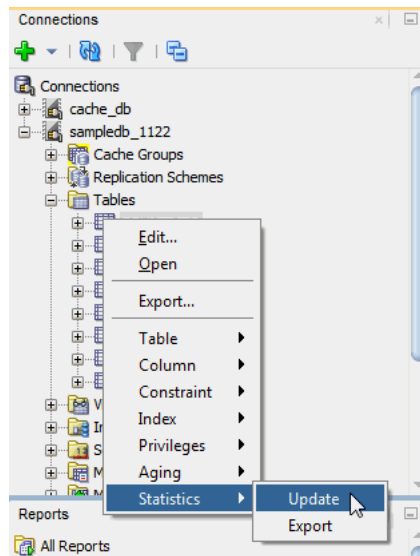
The query optimizer uses statistics stored in TimesTen database system tables to determine the optimal execution plan for a statement. TimesTen stores table level statistics in the `SYS.TBL_STATS` table. Column level statistics, such as the minimum and maximum value, and the number of unique values in a column, are stored in the `SYS.COL_STATS` table.

TimesTen does not compute table and column statistics as updates occur on regular and cache tables. Instead, TimesTen updates statistics when an explicit request is made.

You can update statistics for all tables owned by a user or for a specific table owned by an user.

To update statistics for all tables, right-click the **Tables** node and select **Statistics**, then select **Update**.

To update statistics for a specific table, click the + to the left of the **Tables** node. Right-click the name of the table and select **Statistics**, then select **Update**.

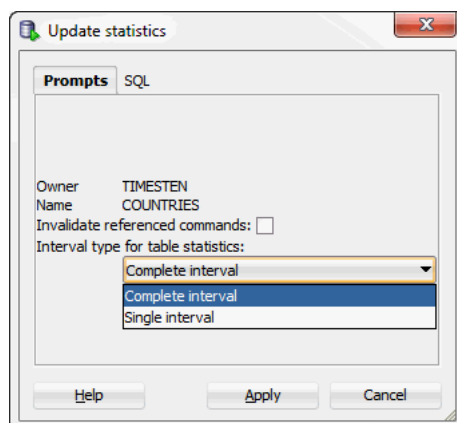
**Figure 3–1 Update statistics for a table**

If you are unable to find the table, see ["Viewing the characteristics of a table"](#) on page 2-8.

In the **Prompts** tab of the **Update statistics** dialog, choose **Invalidate referenced commands** to invalidate the execution plans of statements that reference the table for which statistics are updated. When you invalidate the execution plan of a statement, TimesTen recompiles or reprepares that statement upon its next execution.

For **Interval type for table statistics**, choose **Complete interval** to divide the rows of the table into two or more intervals and compute statistics on each interval, or **Single interval** to compute statistics on the entire set of rows as a single interval. You must define a range index on the table to compute complete interval statistics.

Click **Apply**.

**Figure 3–2 Update statistics dialog**

To view the optimizer statistics of a table, click the name of the table.

**Figure 3–3 Statistics for employees table**

Name	Value
1 NUMTUPS	107
2 LASTSTATSUPDATE	Wed May 22 10:53:18 2013

TBLNAME	COLNAME	STATS
1 EMPLOYEES	EMPLOYEE_ID	(6, 0,
2 EMPLOYEES	FIRST_NAME	(1, 0,
3 EMPLOYEES	LAST_NAME	(6, 0,
4 EMPLOYEES	EMAIL	(6, 0,
5 EMPLOYEES	PHONE_NUMBER	(1, 0,
6 EMPLOYEES	HIRE_DATE	(6, 0,
7 EMPLOYEES	JOB_ID	(4, 0,
8 EMPLOYEES	SALARY	(6, 0,
9 EMPLOYEES	COMMISSION_PCT	(2, 0,

If you are unable to find the table that you are looking for, see "[Locating a TimesTen database object](#)" on page 2-1.

In the **Statistics** tab of the table pane, the top section shows the table statistics such as:

- The number of rows in the table
- A timestamp indicating when statistics were most recently updated for the table

The bottom section shows the column statistics. For each column, the following information is displayed:

- The name of the column
- The number of intervals the data is divided into to compute statistics
- Total number of NULL values
- Total number of non-NULL unique values
- Total number of rows in each interval

For each interval of each column, SQL Developer displays the following:

- The number of unique values other than the most frequently occurring value
- The number of rows that contain a value other than the most frequently occurring value
- The number of rows that contain the most frequently occurring value
- The minimum value
- The maximum value
- The most frequently occurring value

For more information about query optimizer statistics, see "Statistics" in the *Oracle In-Memory Database Cache Introduction*, "When optimization occurs" in the *Oracle TimesTen In-Memory Database Operations Guide* or "ttOptUpdateStats" in the *Oracle TimesTen In-Memory Database Reference*.

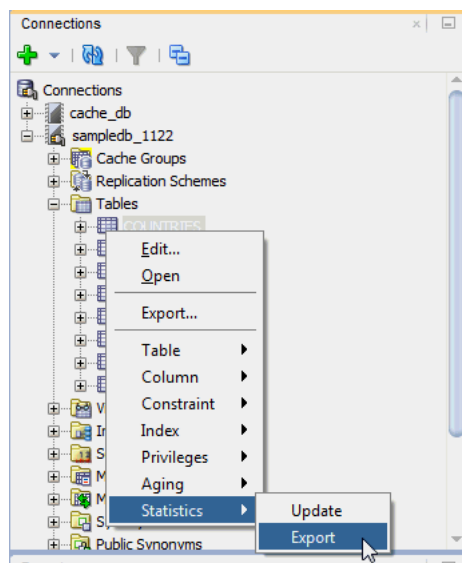
## Exporting table statistics

You can generate a set of SQL statements that allow you to restore the table statistics to the current state. You can generate SQL statements for a single table or for user tables that the connection user has permission to access. After you export your table statistics to a .sql file, you can import your table statistics by opening the .sql file in SQL Developer and running the script file.

To export the current TimesTen table statistics, make sure you are on the main SQL Developer page.

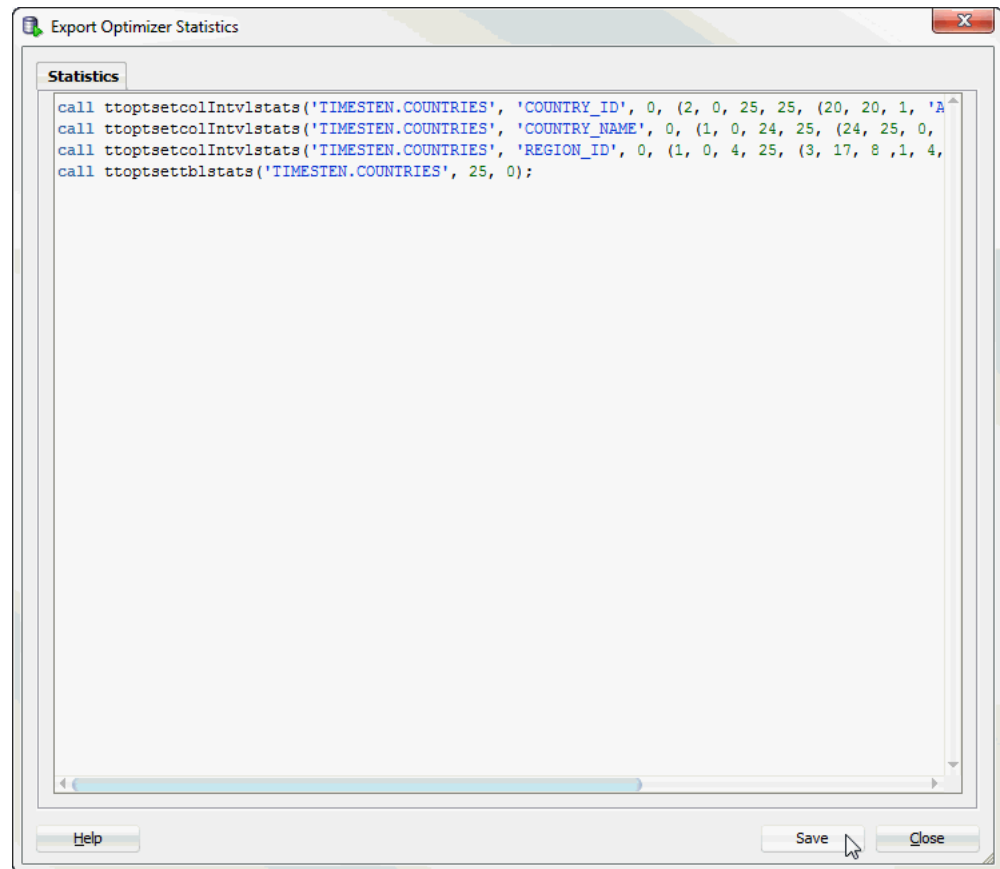
1. To save table statistics for a single table, right-click the name of the table and select **Statistics**, then select **Export**. To save table statistics for all user tables, right-click **Tables** and select **Statistics**, then select **Export**.

**Figure 3–4 Exporting statistics**



The **Export Optimizer Statistics** dialog displays. Locate the **Save** button at the bottom right of the dialog.

2. Click **Save**.

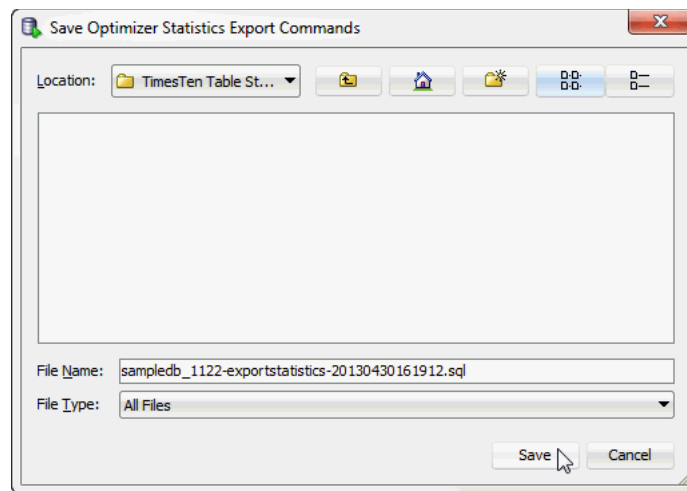
**Figure 3–5 Click save**

The **Save Optimizer Statistics Export Commands** dialog displays.

3. Select the directory where you want to save the current TimesTen table statistics.
4. In the **File Name** field, define the name of your current TimesTen table statistics.

The default filename of the current TimesTen table statistics is *connection\_name-exportstatistics-YYYYMMDDMISS.sql*, where *connection\_name* is the name of the connection. *YYYYMMDDHHMISS* is a timestamp of when the recommendations were made, where *YYYY* is the year, *MM* the month, *DD* the day, *HH* the hour, *MI* the minutes, and *SS* the seconds.

5. Click **Save**.

**Figure 3–6 Save current TimesTen table statistics**

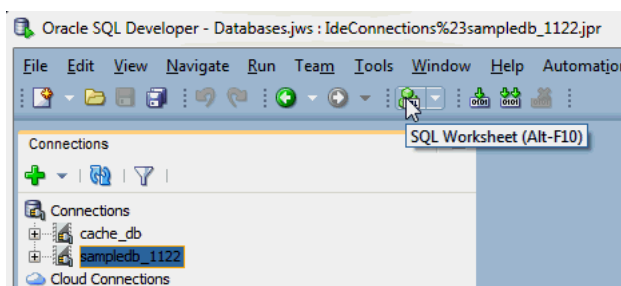
TimesTen saves your current TimesTen table statistics.

## Viewing the execution plan of a statement

A statement must be prepared or compiled before it is executed. When the statement is prepared or compiled, the query optimizer generates an execution plan for the statement.

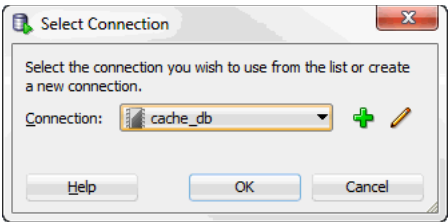
To view the execution plan of a statement, create the statement in the SQL Worksheet. If you do not have a worksheet open in your connection, right-click the node of the connection name and select **Open SQL Worksheet**. See [Figure 1–7, "Viewing a TimesTen connection"](#).

If you do not have a connection open, select **SQL Worksheet** to establish a connection and then open a worksheet for that connection.

**Figure 3–7 Selecting the SQL Worksheet**

The **Select Connection** dialog appears. In the **Connection** field, select the name of the connection. Click **OK**.

Figure 3–8 Select Connection dialog

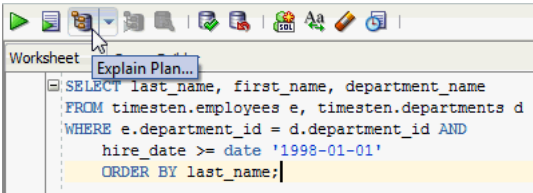


A connection to the TimesTen database is established if a connection associated with the selected connection name does not exist.

Before composing the statement in **SQL Worksheet**, make sure appropriate indexes have been created on the tables that are being referenced in the statement, and that table and column statistics have been updated. For more information about updating statistics, see ["Updating the table and column statistics used by the query optimizer"](#) on page 3-1.

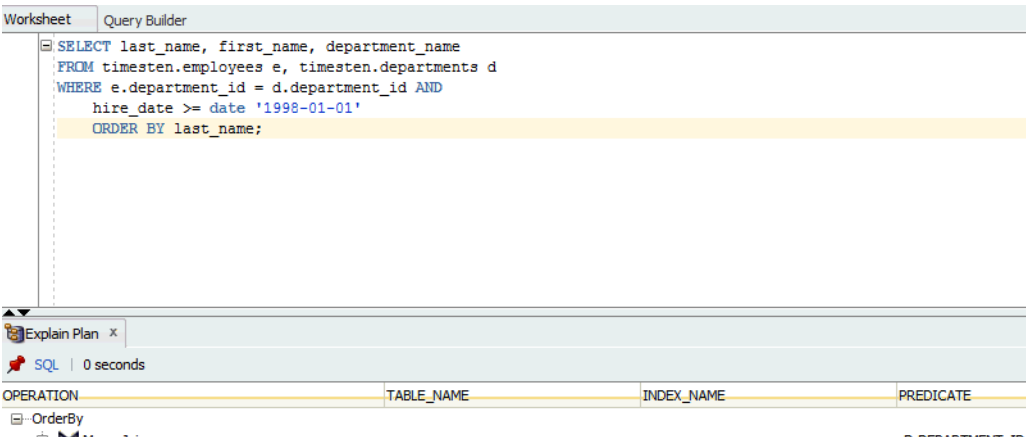
Compose the statement in the **SQL Worksheet**. Then choose **Explain Plan** to display the execution plan for the statement.

Figure 3–9 Viewing explain plan



The execution plan is shown in the **Explain Plan** pane and is displayed in a hierarchical format in reverse chronological order such that the operation at the bottom of the tree is processed first and the operation at the top of the tree is processed last. For each operation, the plan shows the table or index the operation is performed on, or if applicable, the expressions of the statement's predicate the operation is processing. Types of operations that can appear in the execution plan include table or index scan (with lock granularity shown), join, group and sort.

Figure 3–10 Viewing the execution plan



For more information about interpreting an execution plan, see "Optimizer plan" in the *Oracle In-Memory Database Cache Introduction* or "Viewing SQL query plans" in the *Oracle TimesTen In-Memory Database Operations Guide*.

For more information about query performance tuning, see "Viewing SQL query plans" in the *Oracle TimesTen In-Memory Database Operations Guide*.



---

## Working with the TimesTen Index Advisor

SQL Developer supports the TimesTen Index Advisor which can evaluate a SQL workload and recommend indexes. The indexes that the TimesTen Index Advisor recommends can improve the performance for the following: joins, single table scans, and `ORDER BY` or `GROUP BY` operations. The TimesTen Index Advisor can collect index information for the current connection or for the entire database.

---

**Note:** SQL Developer uses procedures from the TimesTen Index Advisor to recommend indexes. For more information about the TimesTen Index Advisor, see "Using the Index Advisor to recommend indexes" in the *Oracle TimesTen In-Memory Database Operations Guide*.

---

Before you use the TimesTen Index Advisor:

- Make sure you are using Oracle TimesTen In-Memory Database Release 11.2.2.4 (or later).

This chapter provides information about working with the TimesTen index advisor.

Topics include:

- [Connection level capture](#)
- [Database level capture](#)
- [Saving index recommendations](#)
- [Viewing the TimesTen index advisor capture state](#)

### Connection level capture

Before attempting to capture data for the TimesTen Index Advisor, make sure your table statistics have been updated in the past 24 hours. Up-to-date table statistics provide the most up-to-date statistics for data collection and allow statements to be prepared with the optimized query plan. For more information about updating table statistics, see "[Updating the table and column statistics used by the query optimizer](#)" on page 3-1.

There are two ways to capture data for the TimesTen Index Advisor at the connection level.

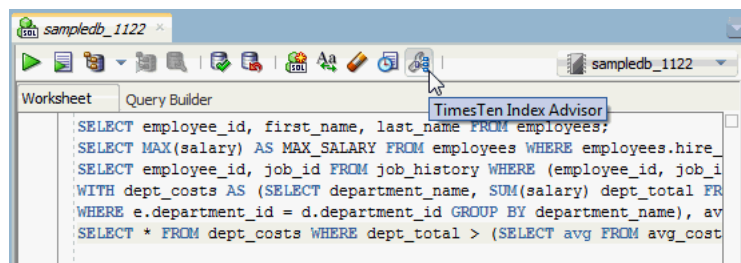
- [SQL worksheet](#)
- [SQL workload script](#)

## SQL worksheet

You can capture data for the TimesTen Index Advisor at the connection level from the SQL Developer SQL worksheet. Make sure your SQL workload is in the SQL Developer SQL worksheet.

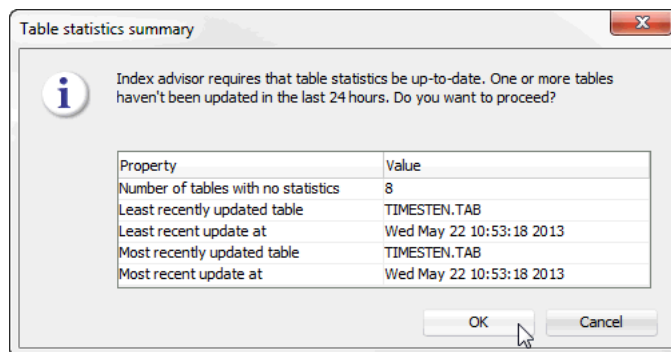
1. Click the **TimesTen Index Advisor** button from the SQL worksheet menu bar.

**Figure 4–1 TimesTen Index Advisor from SQL worksheet**



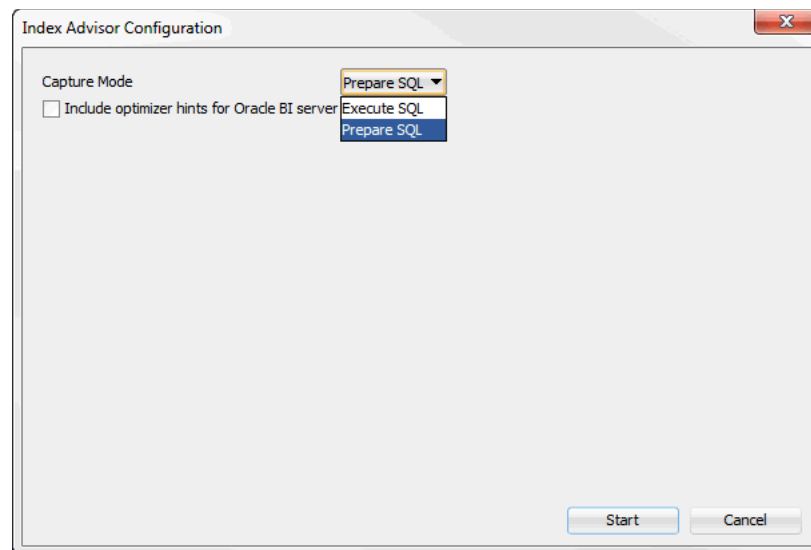
- If your table statistics are outdated, an information dialog displays information about the outdated tables. TimesTen recommends that you update your table statistics.
  - If your table statistics are up to date, the **Index Advisor Configuration** dialog displays.
2. To update your table statistics, click **Cancel** and follow the steps in ["Updating the table and column statistics used by the query optimizer"](#) on page 3-1. To proceed without updating your table statistics, click **OK**.

**Figure 4–2 Outdated table statistics**



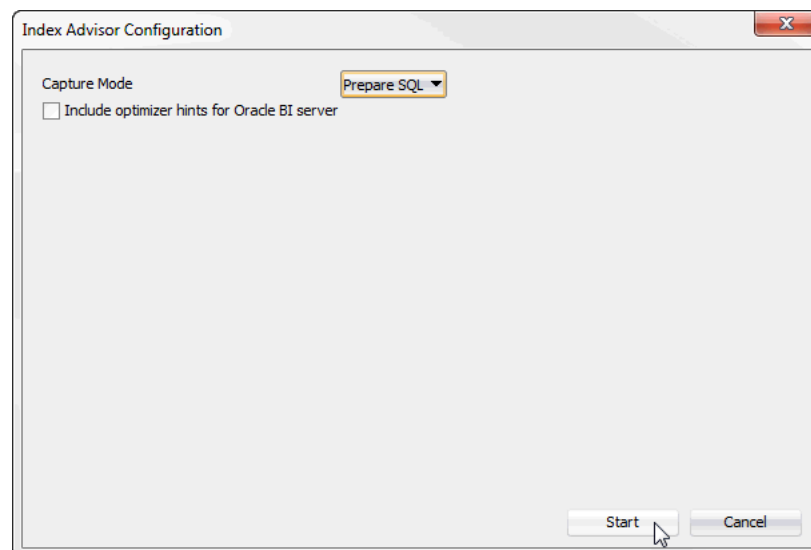
The **Index Advisor Configuration** dialog displays.

3. Select the desired capture mode from the **Capture mode** drop-down list:
  - **Prepare SQL**: TimesTen prepares but does not execute the SQL commands from the SQL worksheet. The TimesTen Index Advisor makes recommendations using computed statistics and query plan analysis. This is the default capture mode.
  - **Execute SQL**: TimesTen executes the SQL commands from the SQL worksheet. The TimesTen Index Advisor makes recommendations using the actual execution of the SQL commands. This capture mode may take longer than the Prepare SQL mode because TimesTen has to complete SQL execution.

**Figure 4–3 Index Advisor configuration**

You can now choose to include optimizer hints for Oracle BI server.

4. The **Include optimizer hints for Oracle BI server** checkbox enables you to use optimizer hints that direct the TimesTen query optimizer to generate a specific execution plan. These optimizer hints are recommended queries generated by the Oracle BI server. For more information on using optimizer hints, see ["Including optimizer hints for Oracle BI server in a SQL worksheet"](#) on page 4-5. If you do not want to use optimizer hints, click **Start**.

**Figure 4–4 Start collecting data for the TimesTen index advisor without optimizer hints**

The **Index Advisor Configuration** dialog closes. A **TimesTen index advisor** pane displays at the bottom of the SQL worksheet. See [Figure 4–5](#).

5. In the **Selector** column, select the index recommendations you want to create.  
You can also click on the column header to sort the table based on the column. A description of each column follows:

- **Selector**

The check box you select to determine the index recommendations that you want to create.

- **Index Recommendation**

The CREATE INDEX statement that the TimesTen index advisor recommends.

- **Affected Statement Count**

The number of statements that benefit from the recommendation.

- **Created**

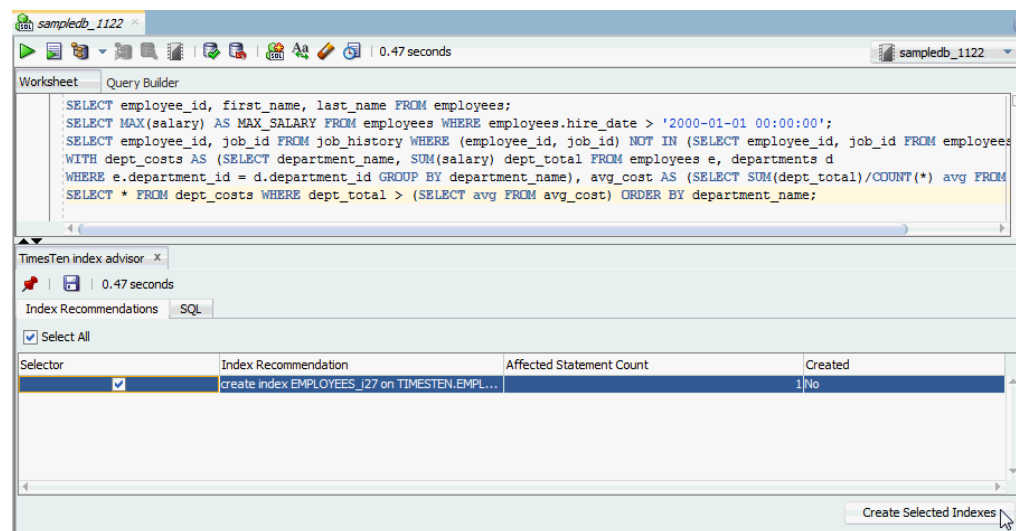
Shows if you have already created the index based on the recommendation by the TimesTen index advisor. This value can be Yes or No. If this value is Yes, TimesTen created the index successfully. If this value is No, TimesTen is in the process of creating the index or is unable to create the index.

To select all index recommendations, click the **Select all** check box.

To review the SQL commands that are being evaluated by the TimesTen index advisor, click the **SQL** tab. The **SQL** tab is located at the top right of the **TimesTen index advisor** pane.

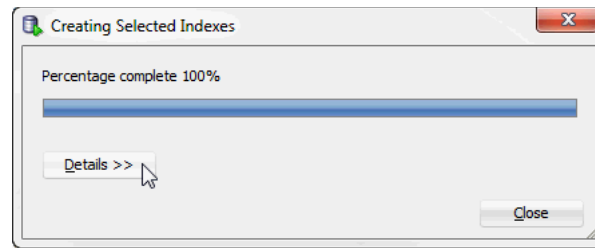
6. Once you select the index recommendations that you want to create, click the **Create Selected Indexes** button.

**Figure 4–5 Create index recommendations**



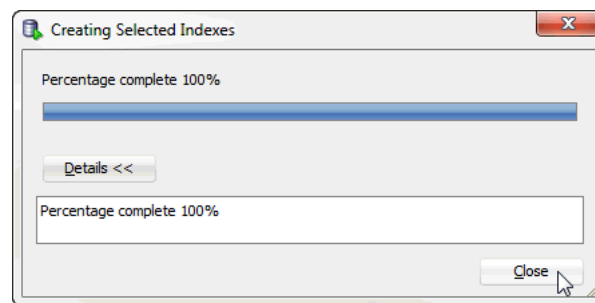
The **Create Selected Indexes** dialog displays. Locate the **Details >>** button.

7. Click **Details >>**.

**Figure 4–6 Create selected indexes progress**

A details pane expands that shows progress information about creating the selected indexes. Locate the **Close** button at the bottom of the dialog.

8. Click **Close**.

**Figure 4–7 Click close**

The **Create Selected Indexes** dialog closes. TimesTen creates the indexes. To save the index recommendations, see ["Saving index recommendations from a SQL worksheet"](#) on page 4-7.

### Including optimizer hints for Oracle BI server in a SQL worksheet

Use transaction level optimizer hints to direct the TimesTen query optimizer to generate a specific execution plan.

---

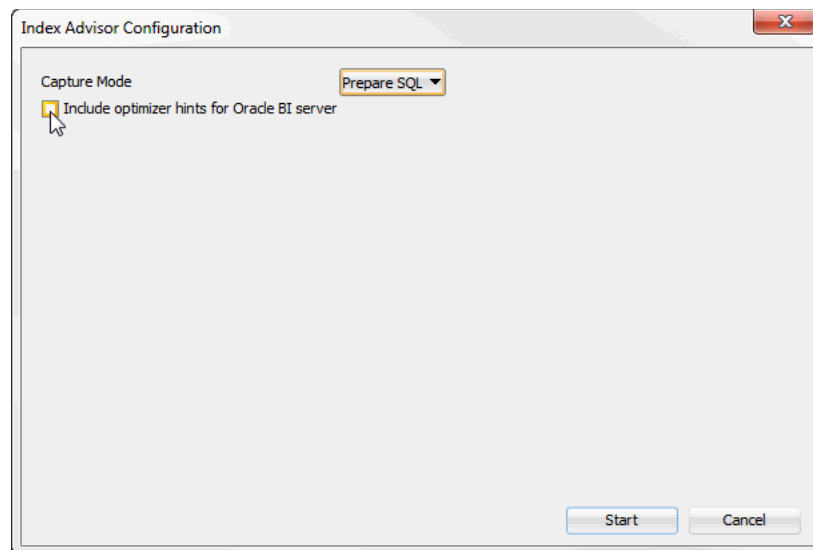
**Note:** For more information about optimizer hints, see "Understanding hints" in the *Oracle TimesTen In-Memory Database SQL Reference*.

---

Before using transaction level optimizer hints, make sure you have completed the steps up to step 4 from ["SQL worksheet"](#) on page 4-2. When you use optimizer hints, make sure to not use any DDL statements or enable AUTOCOMMIT mode in your SQL workload.

To use optimizer hints for Oracle BI server, follow these steps:

1. Select the **Include optimizer hints for Oracle BI server** check box.

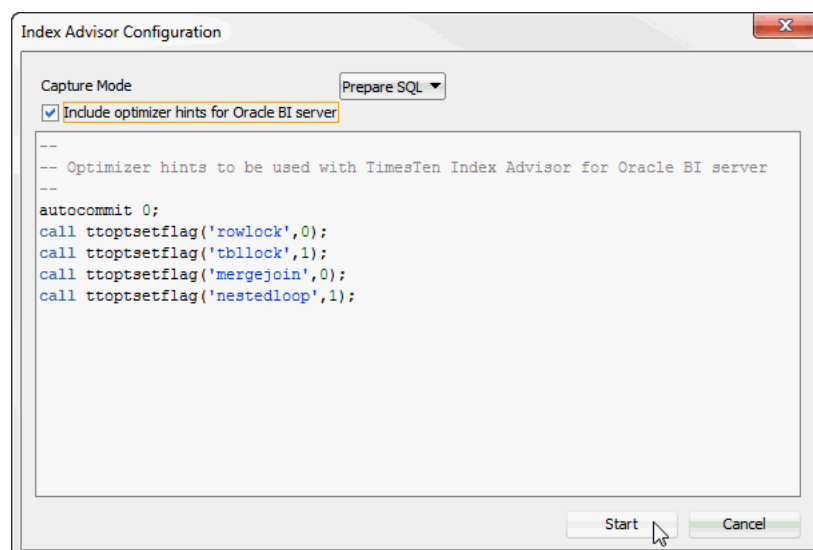
**Figure 4–8 Including optimizer hints for Oracle BI server**

A field with the optimizer hints for Oracle BI server displays in the **Index Advisor Configuration** dialog. You cannot edit the optimizer hints. By default, the optimizer hints for Oracle BI server are as follows:

```
--
-- Optimizer hints to be used with TimesTen Index Advisor for Oracle BI server
--
autocommit 0;
call ttoptsetflag('rowlock',0);
call ttoptsetflag('tbllock',1);
call ttoptsetflag('mergejoin',0);
call ttoptsetflag('nestedloop',1);
```

Locate the **Start** button at the bottom right of the dialog.

2. Click **Start**.

**Figure 4–9 Start collecting data for the TimesTen index advisor with optimizer hints**

The **Index Advisor Configuration** dialog closes. A **TimesTen index advisor** pane displays at the bottom of the SQL worksheet. The **AUTO COMMIT** mode of your transaction is returned to the **AUTO COMMIT** mode that your transaction had prior to running the TimesTen index advisor.

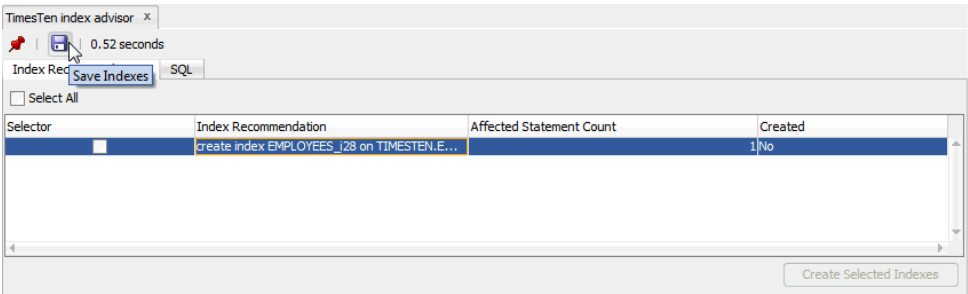
For more information on the **TimesTen index advisor** pane and how to create indexes, see "In the **Selector** column, select the index recommendations you want to create." on page 4-3 from "SQL worksheet" on page 4-2.

**Saving index recommendations from a SQL worksheet**

To save index recommendations for future reference, follow these steps:

- 1. Click **Save Indexes**. TimesTen saves all index recommendations. TimesTen does not save individual index recommendations.

**Figure 4–10 Save Indexes**

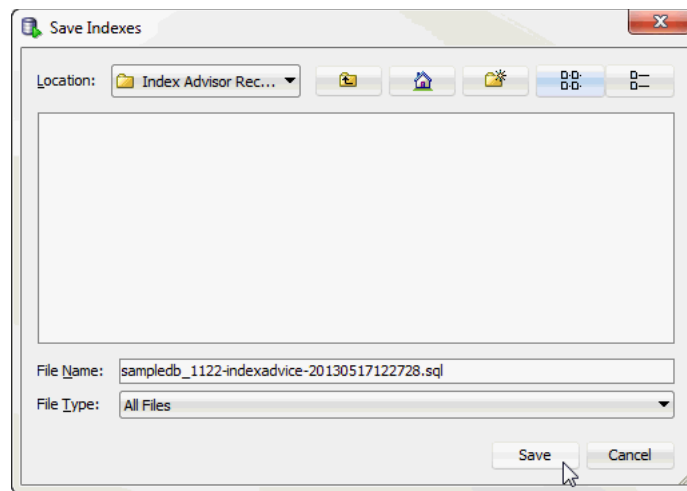


The **Save Indexes** dialog displays.

- 2. Select the directory where you want to save the TimesTen index advisor recommendations.
- 3. In the **File Name** field, define the file name of your TimesTen index advisor recommendations.

The default filename of the TimesTen index advisor recommendations is *connection\_name-indexadvice-YYYYMMDDHHMISS.sql*, where *connection\_name* is the name of the connection. *YYYYMMDDHHMISS* is a timestamp of when the recommendations were made, where *YYYY* is the year, *MM* the month, *DD* the day, *HH* the hour, *MI* the minutes, and *SS* the seconds.

- 4. Click **Save**.

**Figure 4–11 Save index recommendations**

TimesTen saves the index recommendations.

The saved file header contains the following information:

- The timestamp of when TimesTen generated the report.
- The version of SQL Developer.
- The version of the TimesTen database.
- The name of the TimesTen connection.

For example:

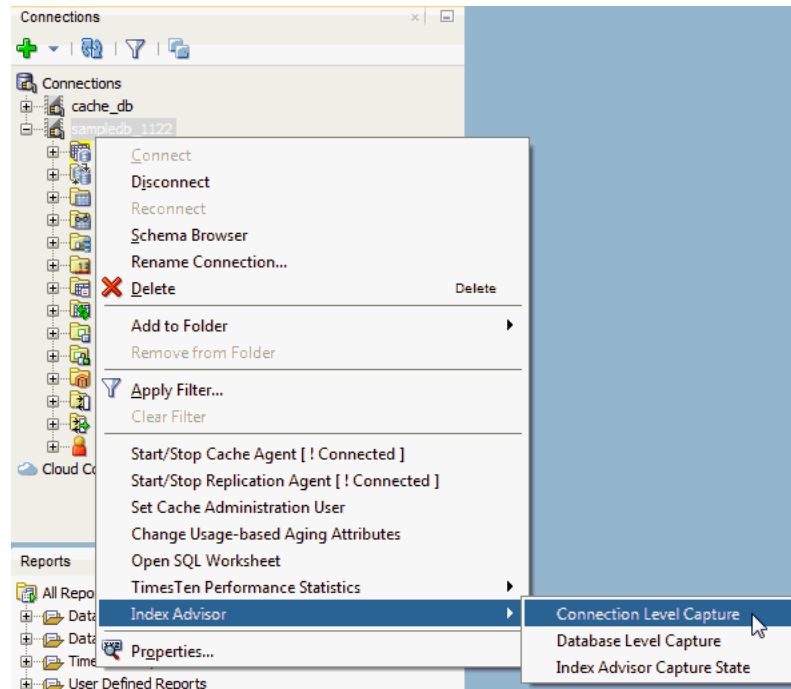
```
-- This file was generated by SQL Developer at 2013-04-29 11:08:45
-- SQL Developer version 4.0.0.11.51
-- Database version: 11.02.02.0005 Oracle TimesTen IMDB version 11.2.2.5.0
-- Connection name: sampledb_1122
```

## SQL workload script

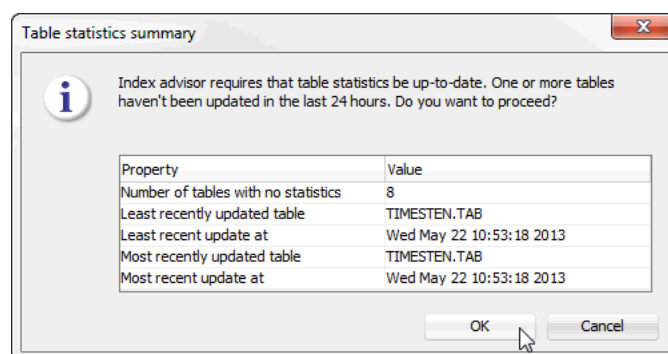
You can capture data for the TimesTen Index Advisor at the connection level using the SQL queries from a SQL workload script file. Make sure you are on the main SQL Developer page and that your TimesTen connection is expanded.

1. Right-click the name of the database and select **Index Advisor**, then select **Connection Level Capture**.



**Figure 4–12 Collecting data at the connection level**

- If your table statistics are outdated, an information dialog displays information about the outdated tables. TimesTen recommends that you update your table statistics.
  - If your table statistics are up to date, the **Index Advisor Configuration** dialog displays.
2. To update your table statistics, click **Cancel** and follow the steps in ["Updating the table and column statistics used by the query optimizer"](#) on page 3-1. To proceed without updating your table statistics, click **OK**.

**Figure 4–13 Outdated table statistics**

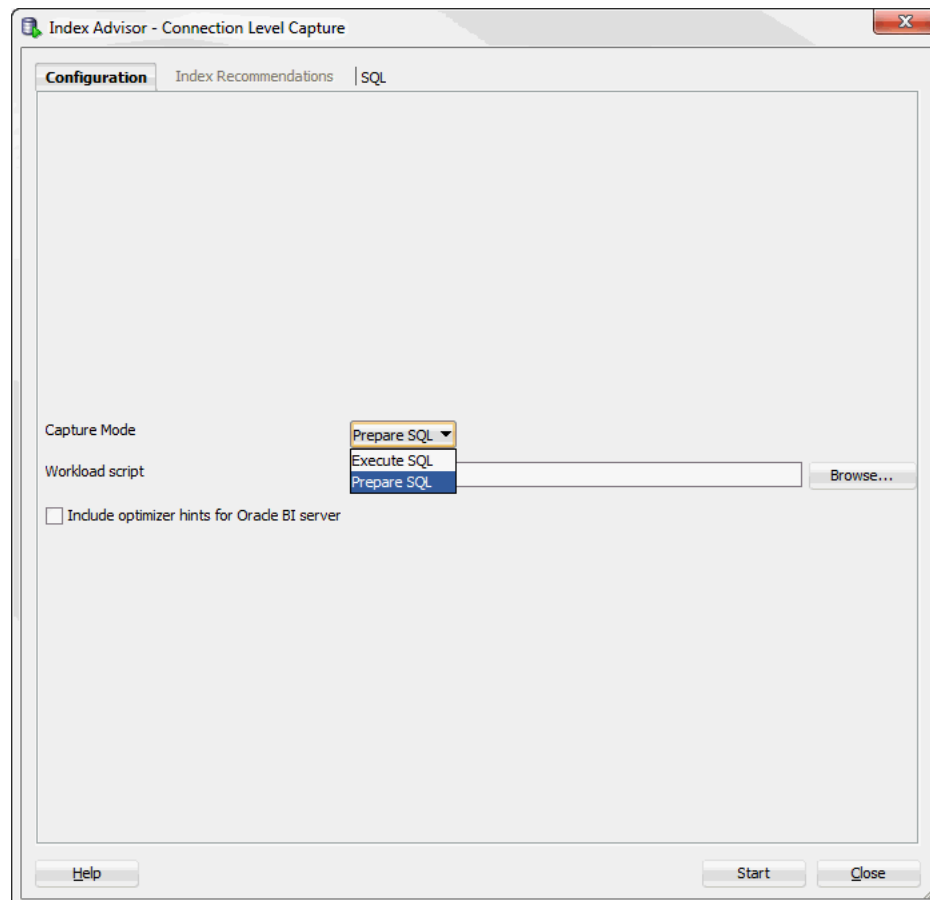
The **Index Advisor - Connection Level Capture** dialog displays.

3. In the **Configuration** tab of the **Index Advisor - Connect Level Capture** dialog, select the desired capture mode from the **Capture mode** drop-down list:
  - **Prepare SQL:** TimesTen prepares but does not execute the SQL commands from the workload script. The TimesTen Index Advisor makes

recommendations using computed statistics and query plan analysis. This is the default capture mode.

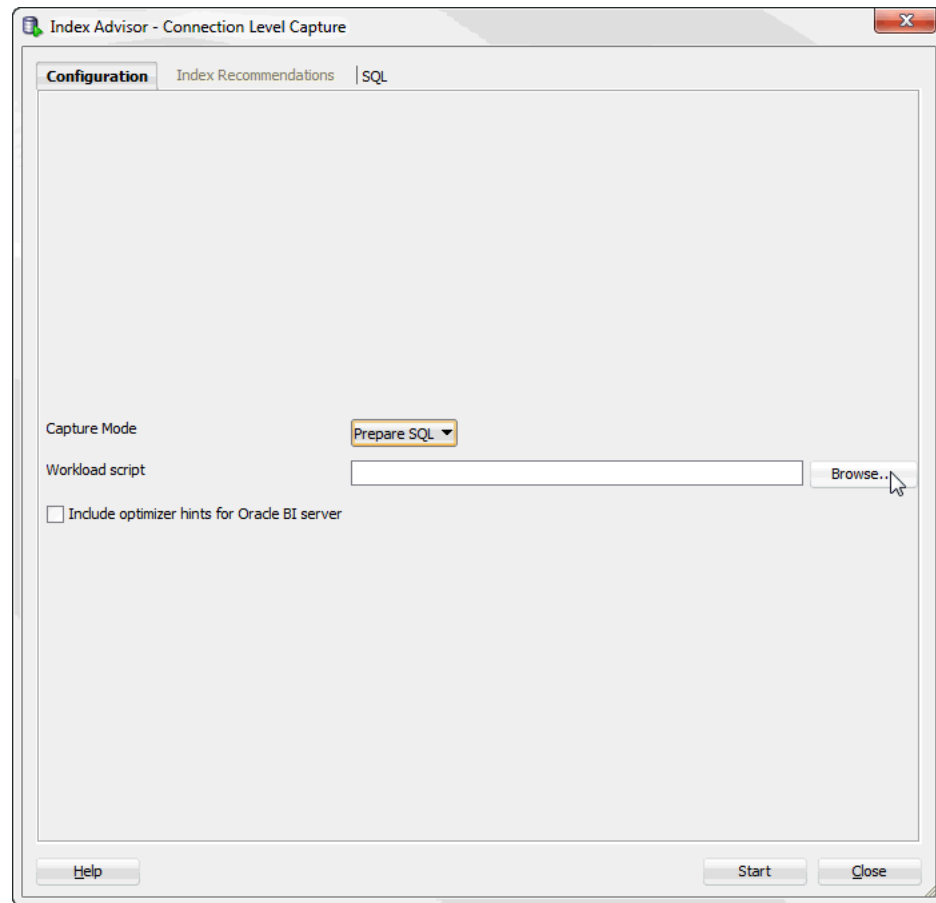
- **Execute SQL:** TimesTen executes SQL commands from the workload script. The TimesTen Index Advisor makes recommendations using the actual execution of the workload script. This capture mode may take longer than the Prepare SQL mode because TimesTen has to complete SQL execution.

**Figure 4–14 Index advisor - Connection Level Capture**



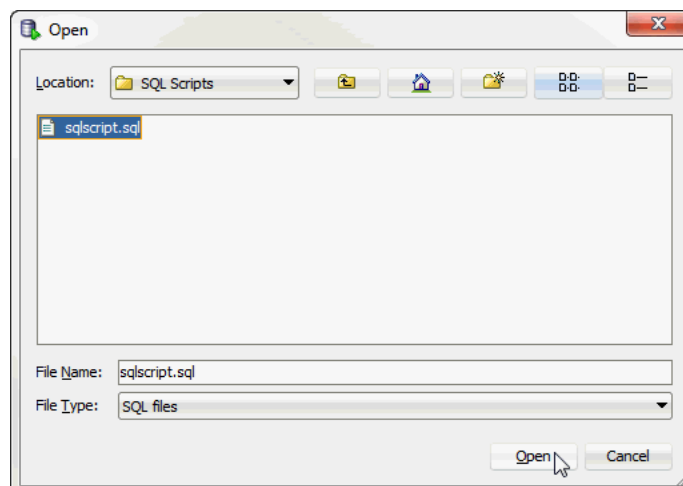
You are ready to specify your SQL workload script. SQL Developer uses the `ttIsql` utility to execute the SQL workload script. Locate the **Browse** button that is to the right of the **Workload script** field.

4. Click **Browse**.

**Figure 4–15 Click Browse**

The **Open** dialog displays.

5. Select the directory where you saved your SQL workload script.
6. Select the SQL workload script for which you want to generate index recommendations.
7. Click **Open**.

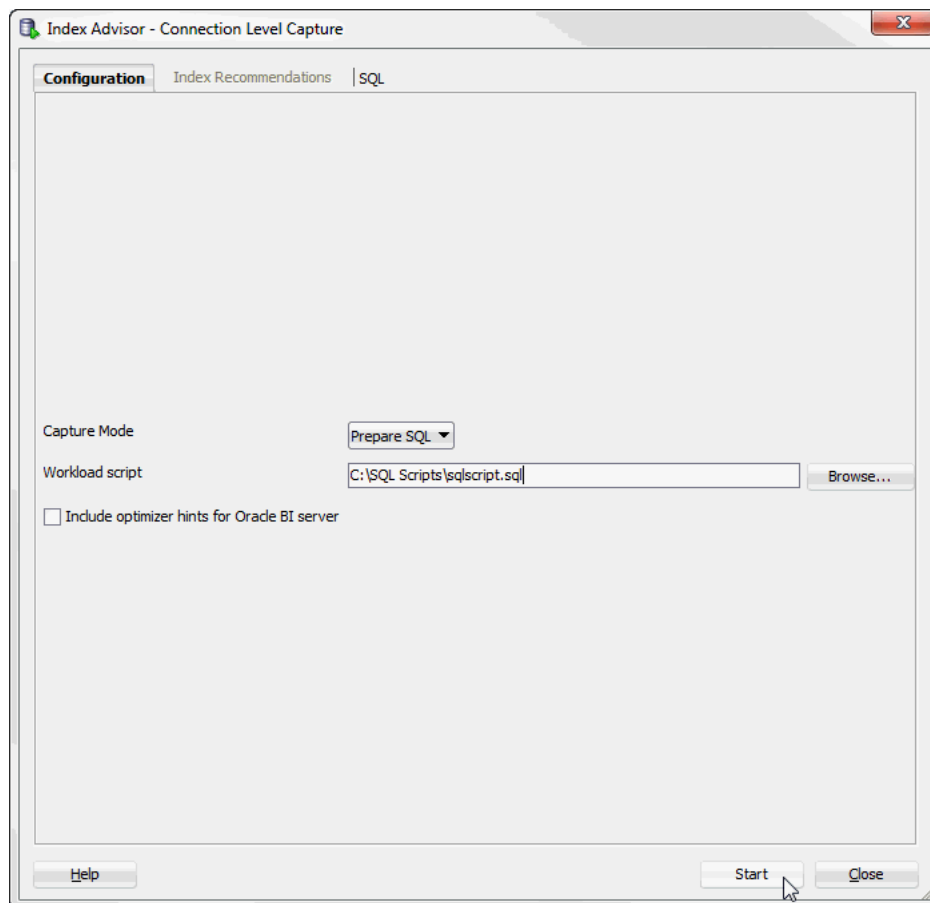
**Figure 4–16 SQL workload script open dialog**

The TimesTen index advisor is now ready to collect data for your SQL workload. Locate the **Start** button.

To review the SQL commands that are being evaluated by the TimesTen index advisor, click the **SQL** tab. The **SQL** tab is located at the top of the **Index Advisor - Connection Level Capture** dialog.

8. The **Include optimizer hints for Oracle BI server** checkbox enables you to use optimizer hints that direct the TimesTen query optimizer to generate a specific execution plan. These optimizer hints are recommended queries generated by the Oracle BI server. For more information on using optimizer hints, see ["Including optimizer hints for Oracle BI server for a SQL workload script"](#) on page 4-16. If you do not want to use optimizer hints, click **Start**.

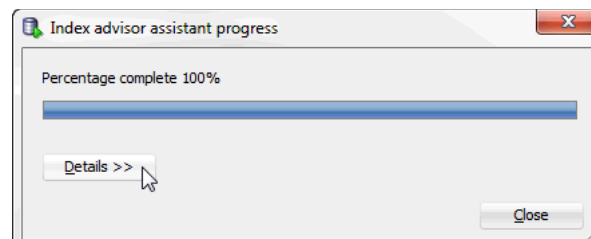
**Figure 4–17 Start collecting data for the TimesTen index advisor**



The **Index advisor assistant progress** dialog displays. Locate the **Details >>** button.

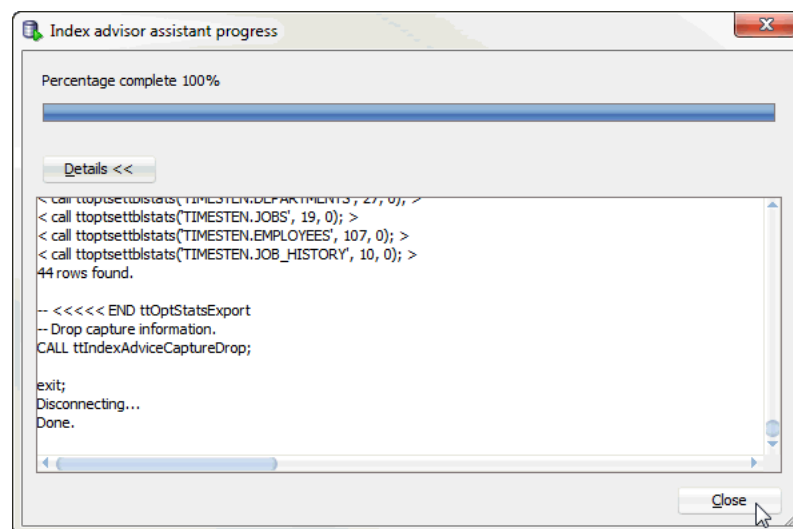
If the TimesTen index advisor does not have any recommendations, the **Index Recommendation Feedback** dialog displays. Close this dialog to return to the main SQL developer page.

9. Click **Details >>**.

**Figure 4–18 Index advisor assistant progress**

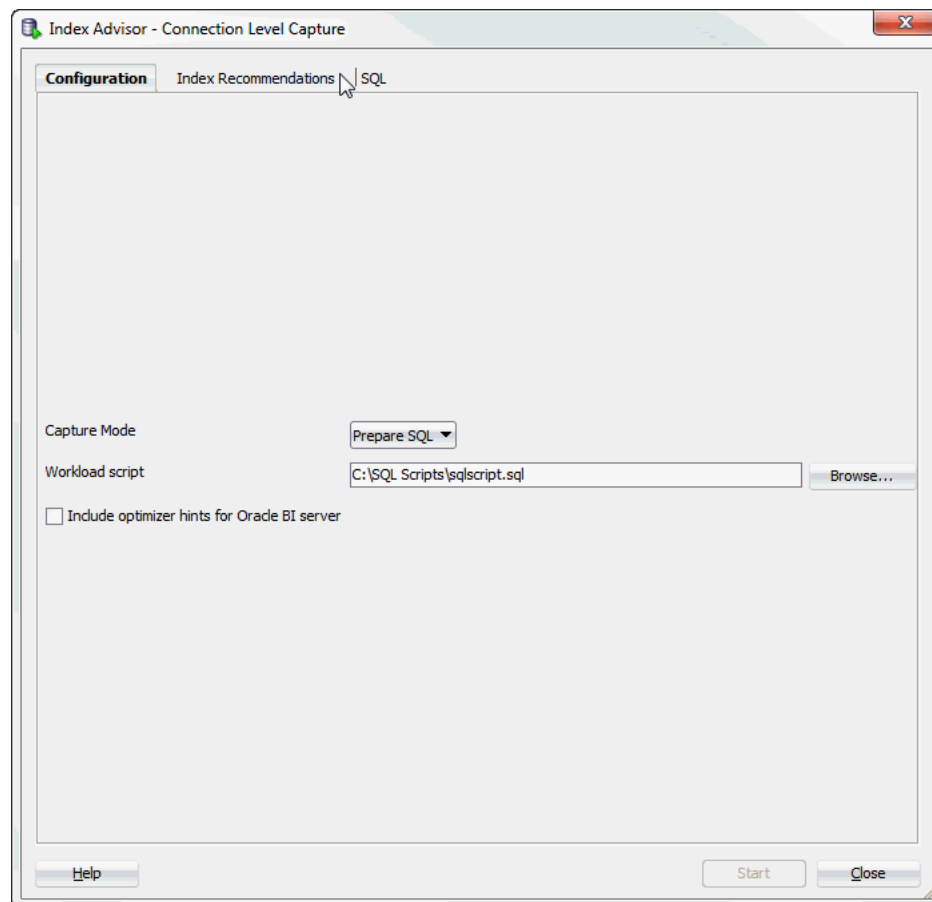
A details pane expands that shows progress information about the index advisor connection level capture. Locate the **Close** button at the bottom of the dialog.

10. Click **Close**.

**Figure 4–19 Click Close**

The **Index advisor assistant progress** dialog closes. Locate the **Index Recommendations** tab.

11. Click the **Index Recommendations** tab.

**Figure 4–20 Index recommendations for connection level capture**

The **Index Recommendations** tab of the **Index Advisor - Connection Level Capture** dialog displays.

12. In the **Selector** column, select the index recommendations that you want to create.

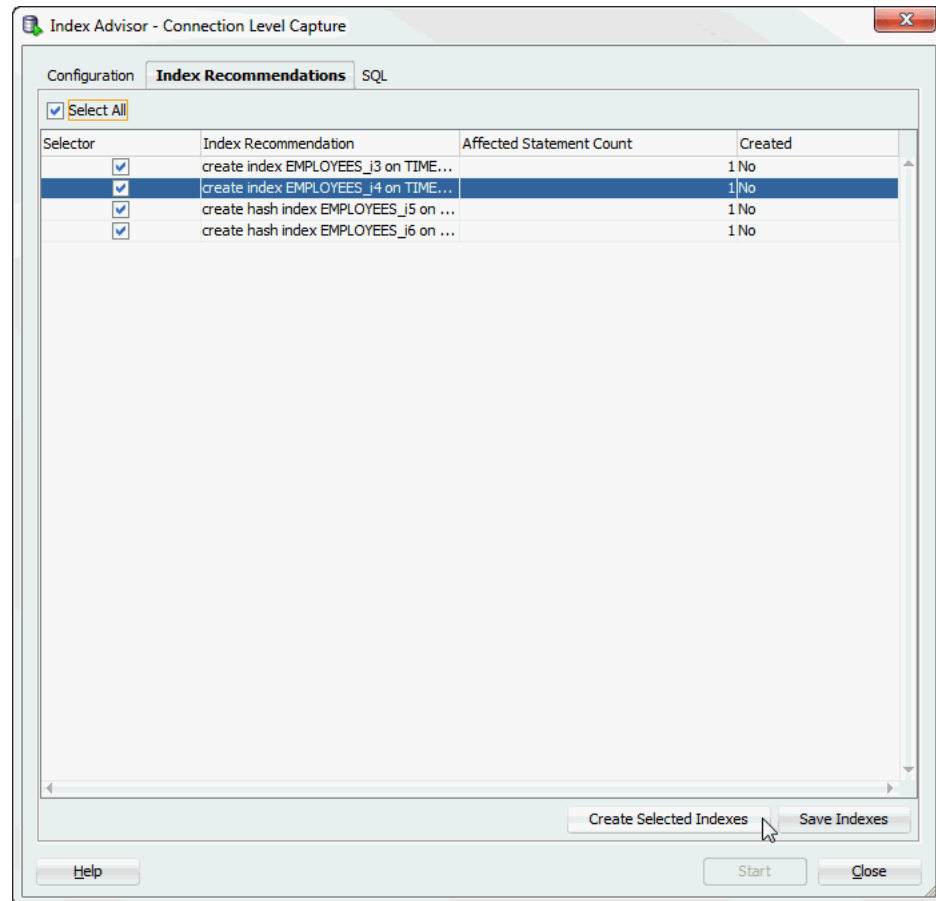
You can also click on the column header to sort the table based on the column. A description of each column follows:

- **Selector**  
The check box you select to determine the index recommendations that you want to create.
- **Index Recommendation**  
The `CREATE INDEX` statement that the TimesTen index advisor recommends.
- **Affected Statement Count**  
The number of statements that benefit from the recommendation.
- **Created**  
Shows if you have already created the index based on the recommendation by the TimesTen index advisor. This value can be Yes or No. If this value is Yes, TimesTen created the index successfully. If this value is No, TimesTen is in the process of creating the index or is unable to create the index.

To select all index recommendations, click the **Select all** check box.

13. Once you select the index recommendations that you want to apply, click the **Create Selected Indexes** button.

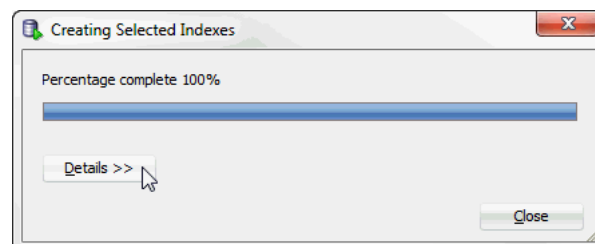
**Figure 4–21 Create index recommendations**



The **Creating Selected Indexes** dialog displays. Locate the **Details >>** button.

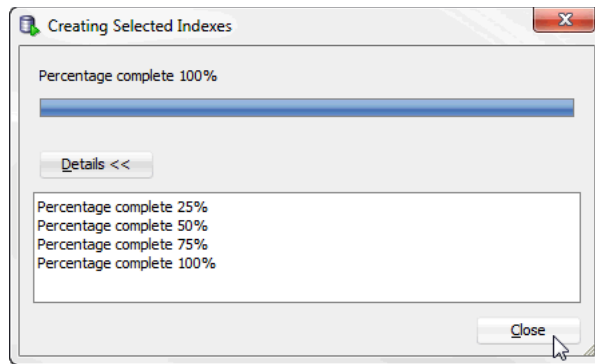
14. Click **Details >>**.

**Figure 4–22 Create Selected Indexes**



A details pane expands that shows progress information about creating the selected indexes. Locate the **Close** button at the bottom of the dialog.

15. Click **Close**.

**Figure 4–23 Click close**

The **Creating Selected Indexes** dialog closes. The indexes are created. To save the index recommendations, see ["Saving index recommendations"](#) on page 4-28.

### Including optimizer hints for Oracle BI server for a SQL workload script

Use transaction level optimizer hints to direct the TimesTen query optimizer to generate a specific execution plan.

---

**Note:** For more information about transaction level optimizer hints, see "Understanding hints" in the *Oracle TimesTen In-Memory Database SQL Reference*.

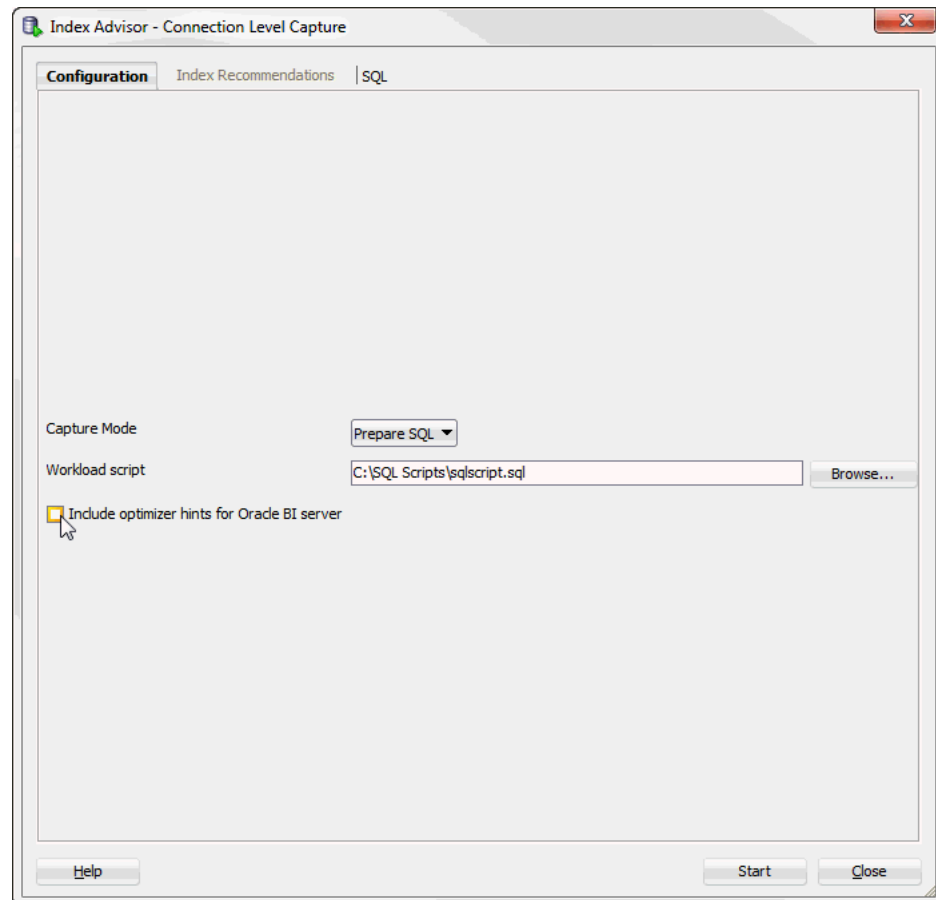
---

Before using transaction level optimizer hints, make sure you have completed the steps up to step 8 from ["SQL workload script"](#) on page 4-8. When you use optimizer hints, make sure to not use any DDL statements or enable `AUTOCOMMIT` mode in your SQL workload.

To use optimizer hints for Oracle BI server, follow these steps:

1. Select the **Include optimizer hints for Oracle BI server** check box.



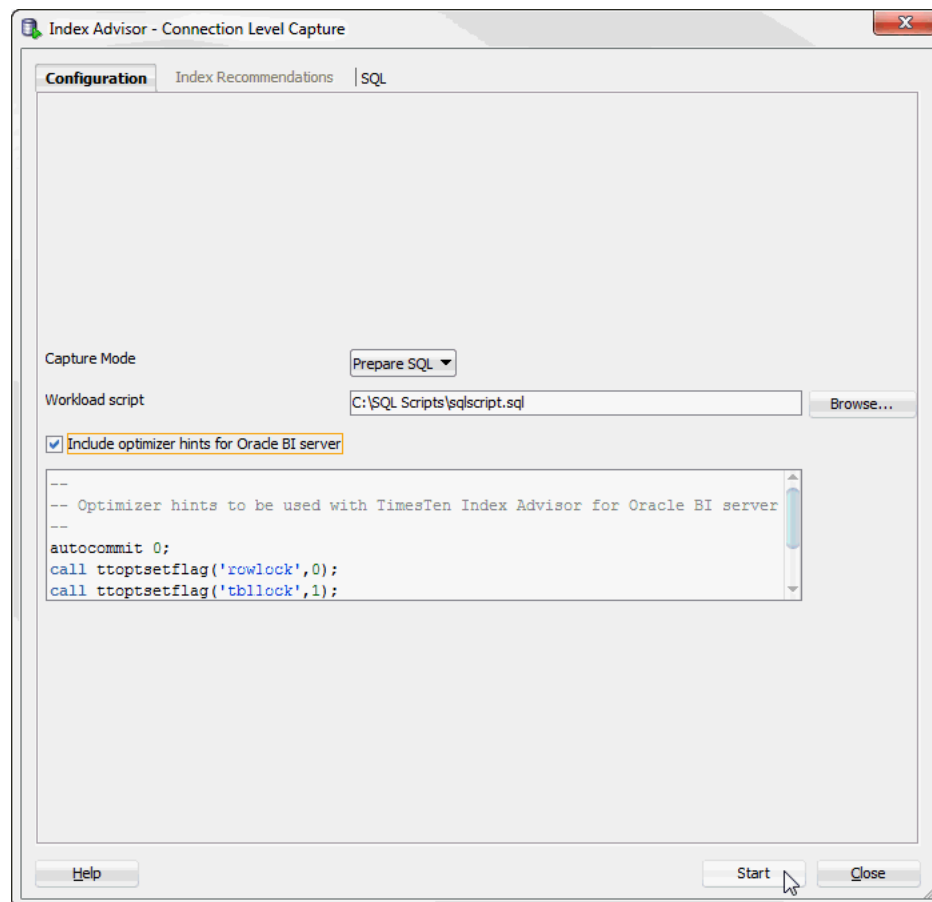
**Figure 4–24 Including optimizer hints for Oracle BI server**

A field with the optimizer hints for Oracle BI server displays in the **Index Advisor - Connection Level Capture** dialog. These optimizer hints are not editable. By default, the optimizer hints for Oracle BI server are as follows:

```
--
-- Optimizer hints to be used with TimesTen Index Advisor for Oracle BI server
--
autocommit 0;
call ttoptsetflag('rowlock',0);
call ttoptsetflag('tbllock',1);
call ttoptsetflag('mergejoin',0);
call ttoptsetflag('nestedloop',1);
```

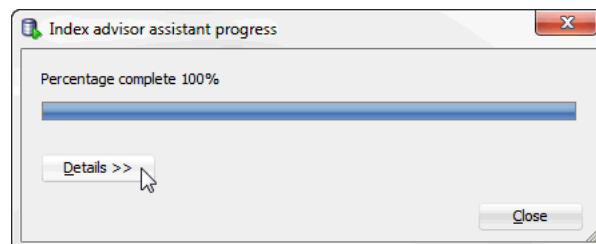
Locate the **Start** button at the bottom right of the dialog.

2. Click **Start**.

**Figure 4–25 Start collecting data for the TimesTen index advisor with optimizer hints**

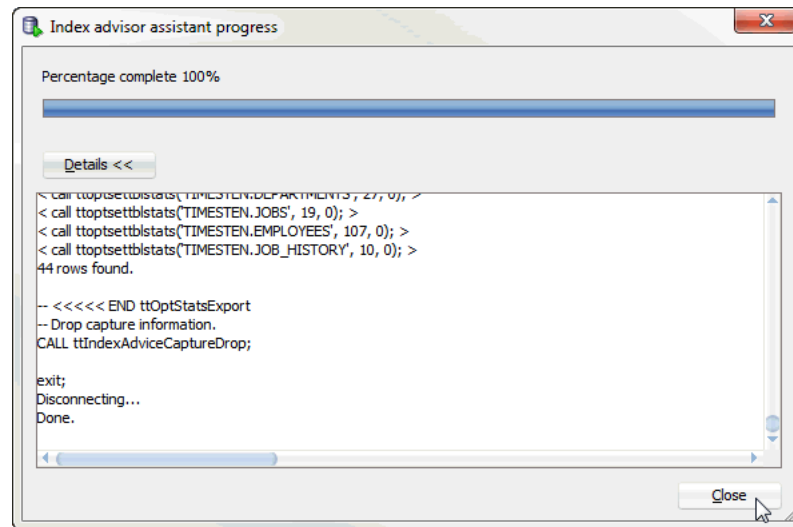
The **Index advisor assistant progress** dialog displays. Locate the **Details >>** button.

3. Click **Details >>**.

**Figure 4–26 Index advisor assistant progress**

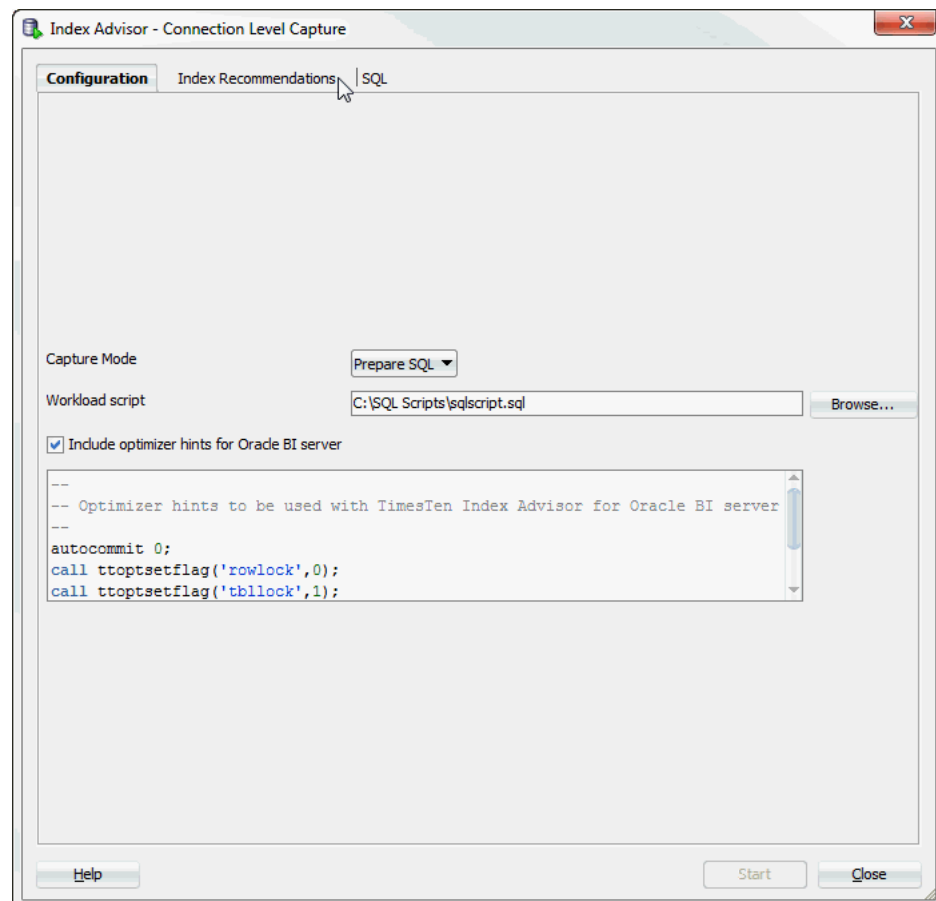
A details pane expands that shows progress information about the index advisor connection level capture. Locate the **Close** button at the bottom of the dialog.

4. Click **Close**.

**Figure 4-27 Click Close**

The **Index advisor assistant progress** dialog closes. Locate the **Index Recommendations** tab.

5. Click the **Index Recommendations** tab.

**Figure 4-28 Index recommendations for connection level capture**

The **Index Recommendations** tab of the **Index Advisor - Connection Level Capture** dialog displays.

6. In the **Selector** column, select the index recommendations that you want to create.

You can also click on the column header to sort the table based on the column. A description of each column follows:

- **Selector**

The check box you select to determine the index recommendations that you want to create.

- **Index Recommendation**

The `CREATE INDEX` statement that the TimesTen index advisor recommends.

- **Affected Statement Count**

The number of statements that benefit from the recommendation.

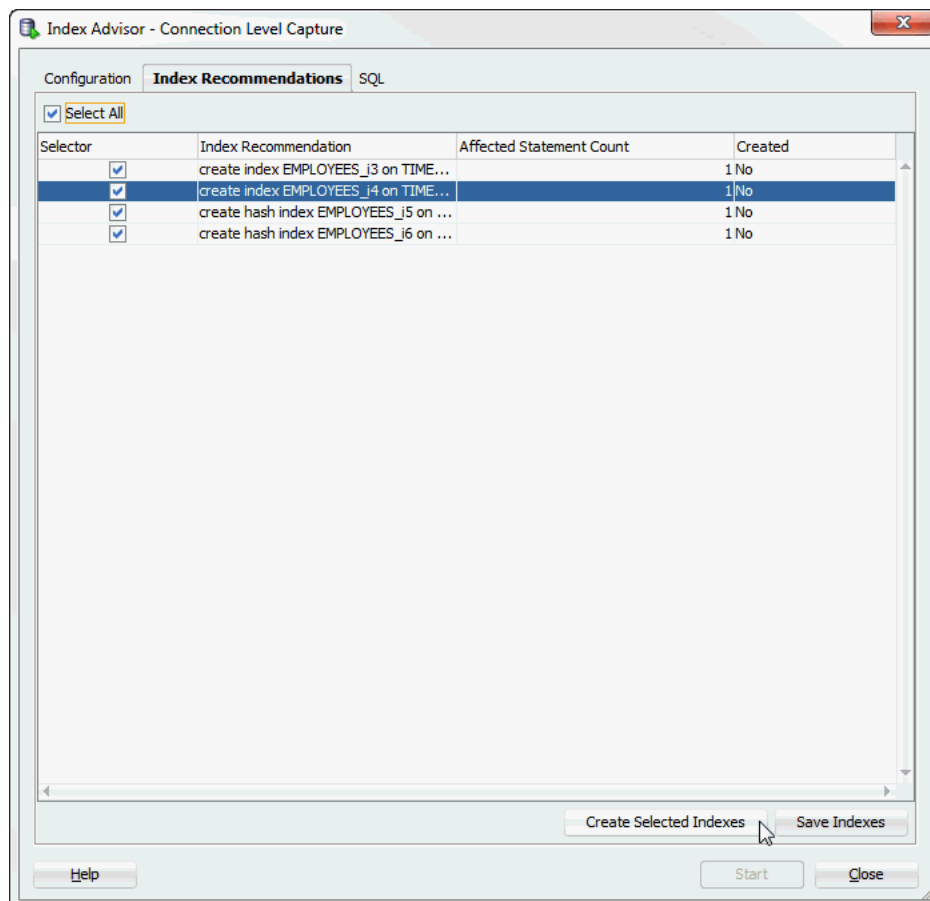
- **Created**

Shows if you have already created the index recommendation. This value can be Yes or No.

To select all index recommendations, click the **Select all** check box.

7. Once you select the index recommendations that you want to apply, click the **Create Selected Indexes** button.

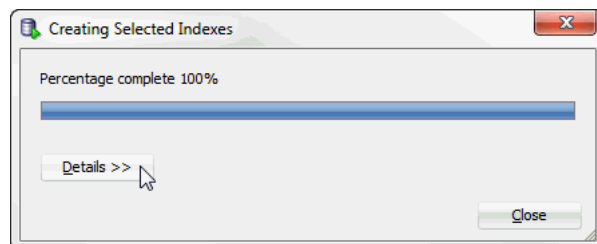
**Figure 4–29 Create index recommendations**



The **Creating Selected Indexes** dialog displays. Locate the **Details >>** button.

8. Click **Details >>**.

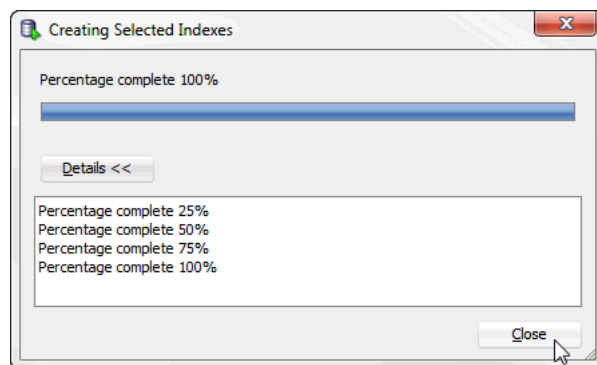
**Figure 4–30 Creating Selected Indexes**



A details pane expands that shows progress information about creating the selected indexes. Locate the **Close** button at the bottom of the dialog.

9. Click **Close**.

**Figure 4–31 Click close**



The **Creating Selected Indexes** dialog closes. The indexes are created. The AUTOCOMMIT mode of your transaction is returned to the AUTOCOMMIT mode that your transaction had prior to running the TimesTen index advisor. To save the index recommendations, see ["Saving index recommendations"](#) on page 4-28.

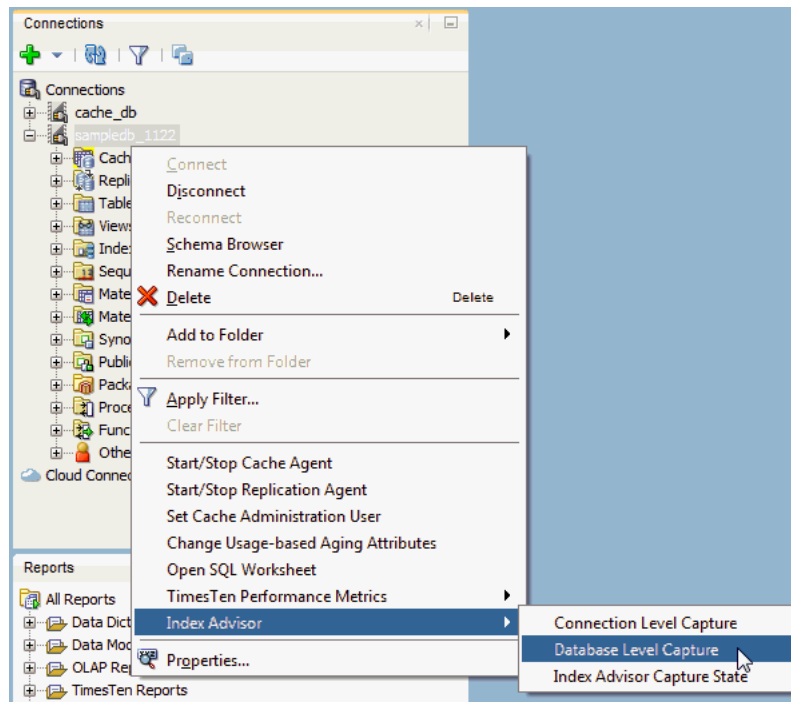
## Database level capture

Before attempting to capture data for the TimesTen Index Advisor, make sure your table statistics have been updated in the past 24 hours. Up-to-date table statistics provide the most up-to-date statistics for data collection and allow statements to be prepared with the optimized query plan. For more information about updating table statistics, see ["Updating the table and column statistics used by the query optimizer"](#) on page 3-1.

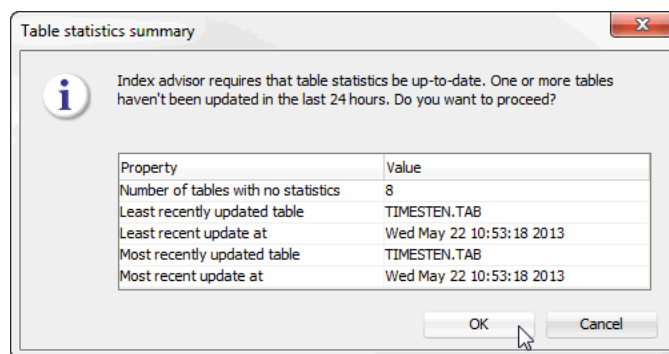
Also, make sure that you have granted ADMIN privileges to the TimesTen connection user. For more information on the GRANT SQL statement, see "GRANT" in the *Oracle TimesTen In-Memory Database SQL Reference*.

To collect data for the TimesTen index advisor at the database level, make sure you are on the main SQL Developer page and that your TimesTen connection is expanded.

1. Right-click the name of the database and select **Index Advisor**, then select **Database Level Capture**.

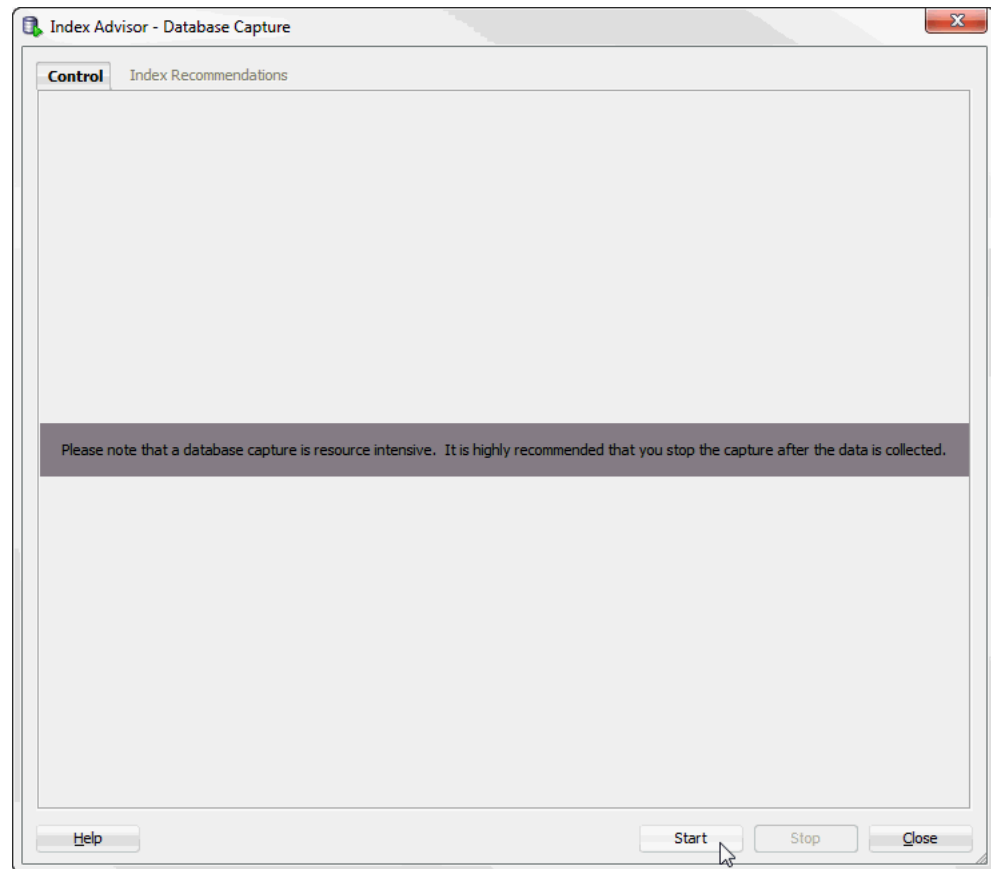
**Figure 4–32 Collecting data at the database level**

- If your table statistics are outdated, an information dialog displays information about the outdated tables. It is recommended that your table statistics are up to date.
  - If your table statistics are up to date, the **Index Advisor - Connection Level Capture** dialog displays.
2. To update your table statistics, click **Cancel** and follow the steps in "[Updating the table and column statistics used by the query optimizer](#)" on page 3-1. To proceed without updating your table statistics, click **OK**.

**Figure 4–33 Outdated table statistics**

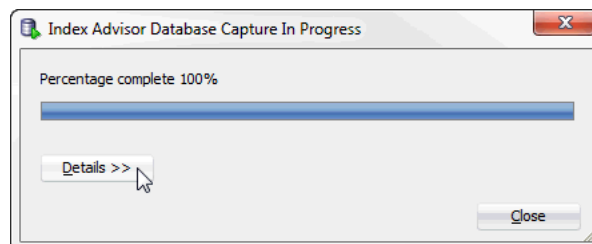
The **Index Advisor - Database Capture** dialog displays. Locate the **Start** button at the bottom of the dialog.

3. In the **Control** tab of the **Index Advisor - Connect Level Capture** dialog, click **Start**.

**Figure 4–34 Index Advisor - Database Capture**

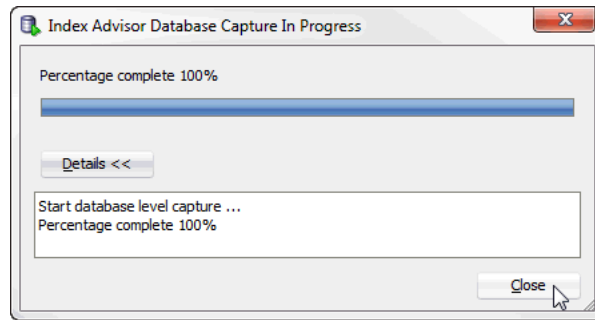
The **Index Advisor Database Capture In Progress** dialog displays. Locate the **Details >>** button.

4. Click **Details >>**.

**Figure 4–35 Index advisor database capture progress**

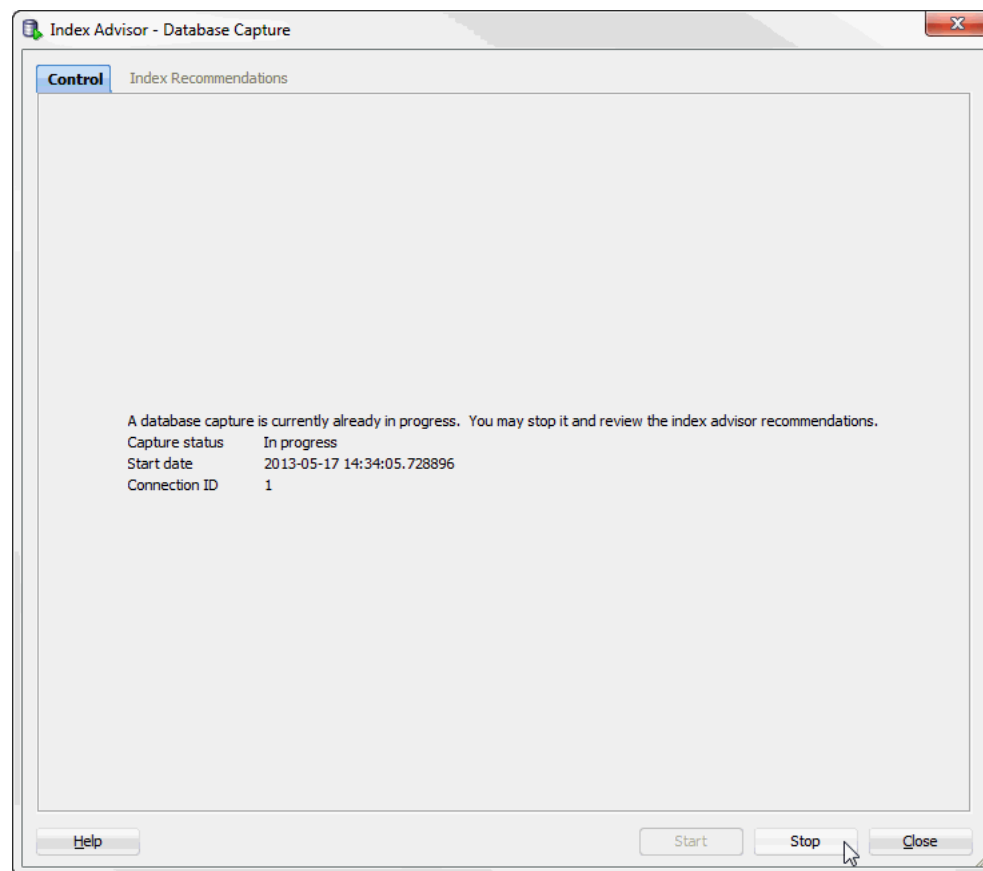
A details pane expands that shows progress information about the index advisor database level capture. Locate the **Close** button at the bottom of the dialog.

5. Click **Close**.

**Figure 4–36 Click Close**

The **Index Advisor Database Capture In Progress** dialog closes.

6. Once you have captured your desired SQL workload, click **Stop**.

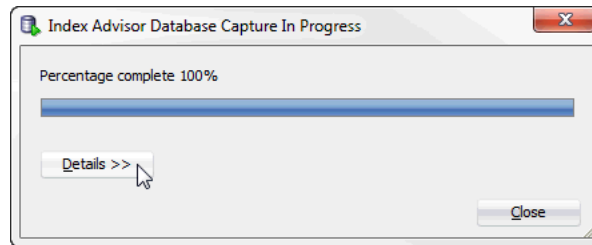
**Figure 4–37 Stop database capture**

If the TimesTen index advisor has recommendations, the **Index Advisor Database Capture In Progress** dialog displays.

If the TimesTen index advisor does not have any recommendations, the **Index Recommendation Feedback** dialog displays. Close this dialog to return to the main SQL developer page.

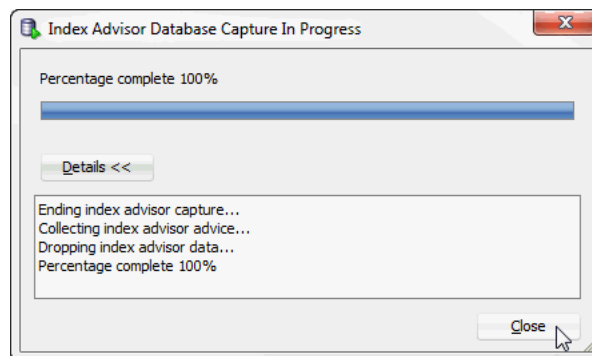
7. Click **Details >>**.



**Figure 4–38 Index advisor database capture progress**

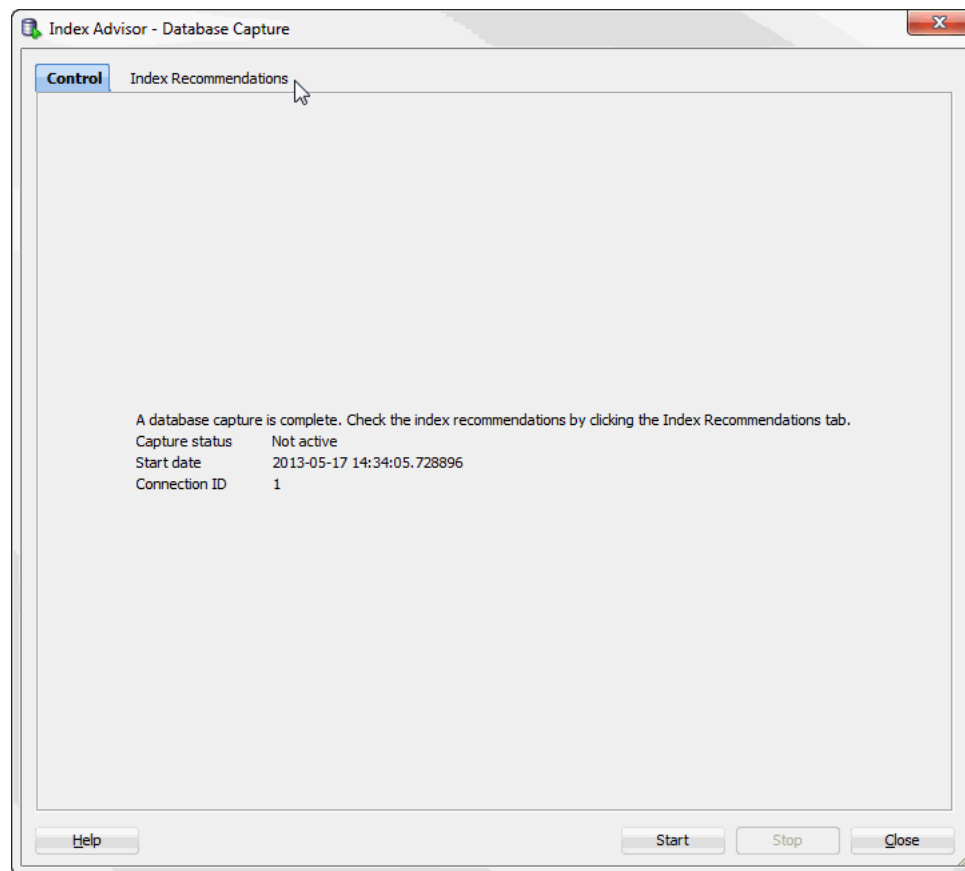
A details pane expands that shows progress information about the index advisor database capture. Locate the **Close** button at the bottom of the dialog.

8. Click **Close**.

**Figure 4–39 Click Close**

The **Index Advisor Database Capture In Progress** dialog closes. The Index Advisor Database Capture dialog shows information about the completed database capture. Locate the **Index Recommendations** tab at the top of the dialog.

9. Click the **Index Recommendations** tab.

**Figure 4–40 Index recommendations for database capture**

The **Index Recommendations** tab of the **Index Advisor - Database Capture** dialog displays.

10. In the **Selector** column, select the index recommendations that you want to create.

You can also click on the column header to sort the table based on the column. A description of each column follows:

- **Selector**

The check box you select to determine the index recommendations that you want to create.

- **Index Recommendation**

The `CREATE INDEX` statement that the TimesTen index advisor recommends.

- **Affected Statement Count**

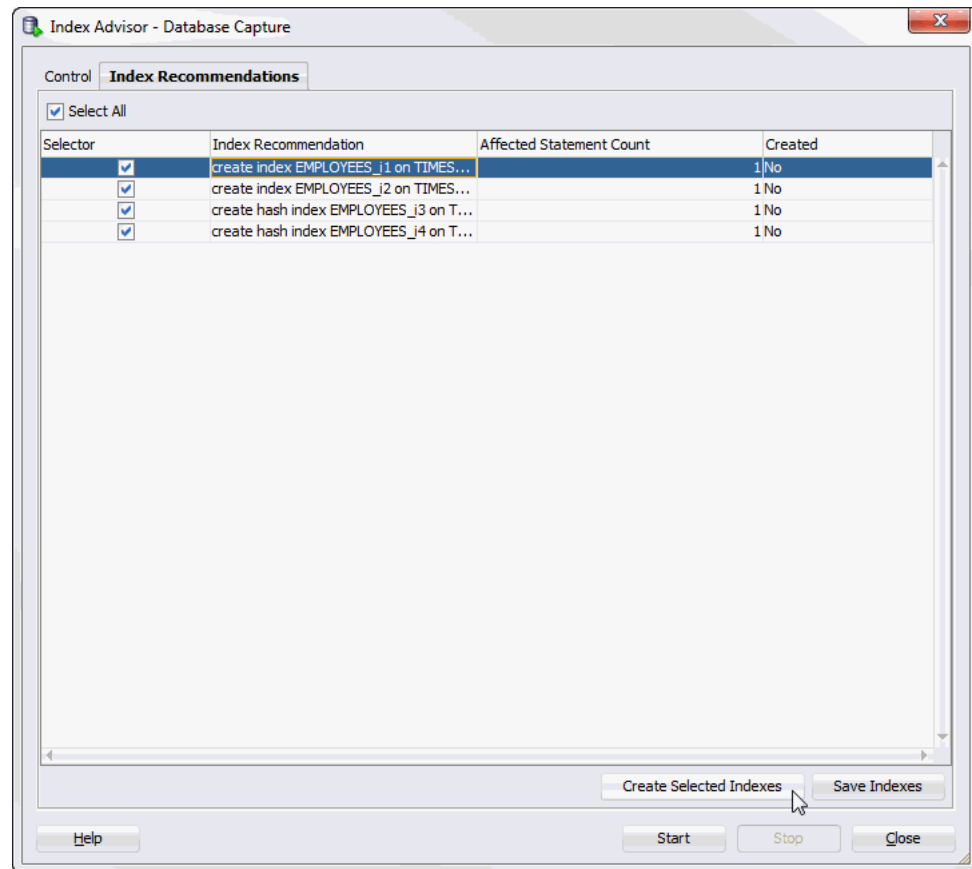
The number of statements that benefit from the recommendation.

- **Created**

Shows if you have already created the index recommendation. This value can be Yes or No.

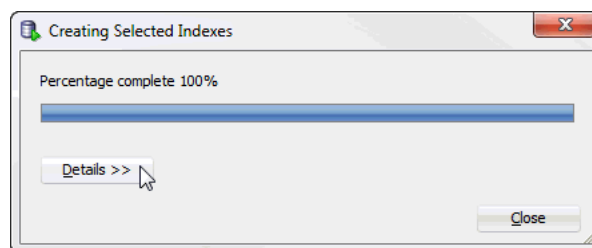
To select all index recommendations, click the **Select all** check box.

11. Once you select the index recommendations that you want to apply, click the **Create Selected Indexes** button.

**Figure 4–41 Create index recommendations**

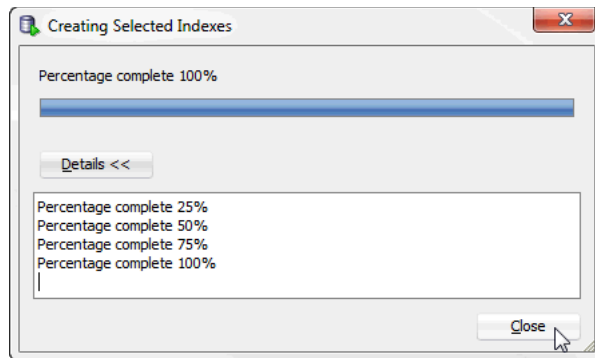
The **Creating Selected Indexes** dialog displays. Locate the **Details >>** button.

12. Click **Details >>**.

**Figure 4–42 Creating Selected Indexes**

A details pane expands that shows progress information about creating the selected indexes. Locate the **Close** button at the bottom of the dialog.

13. Click **Close**.

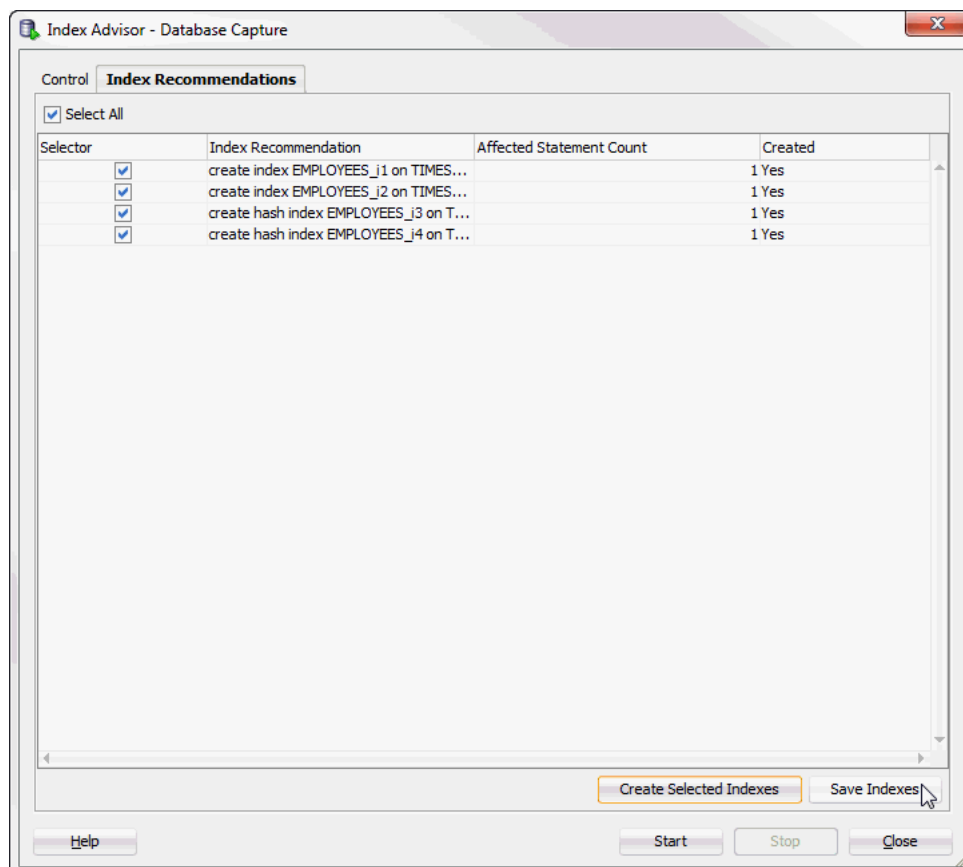
**Figure 4–43 Click close**

The **Creating Selected Indexes** dialog closes. The indexes are created. To save the index recommendations, see ["Saving index recommendations"](#) on page 4-28.

## Saving index recommendations

These steps describe how to save index recommendations for future reference. These steps are valid for ["Database level capture"](#) on page 4-21 and ["SQL workload script"](#) on page 4-8 for a connection level capture.

1. Click **Save Indexes**. By default, all index recommendations are saved. You cannot save individual index recommendations.

**Figure 4–44 Save Indexes**

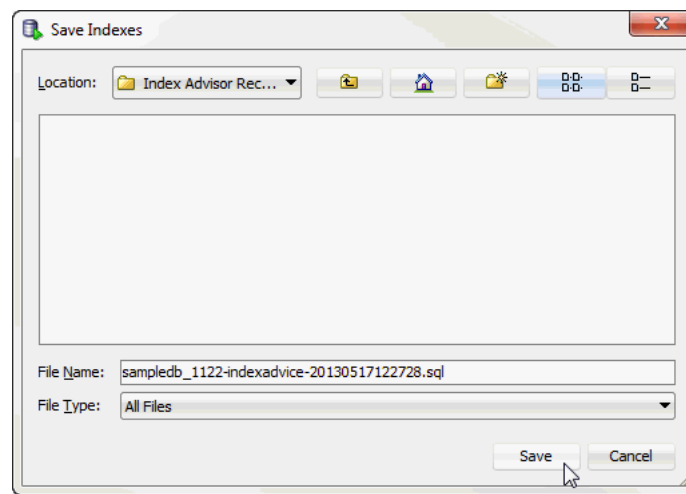
The **Save Indexes** dialog displays.

2. Select the directory where you want to save the TimesTen index advisor recommendations.
3. In the **File Name** field, define the file name of your TimesTen index advisor recommendations.

The default filename of the TimesTen index advisor recommendations is *connection\_name-indexadvice-YYYYMMDDHHMISS.sql*, where *connection\_name* is the name of the connection. *YYYYMMDDHHMISS* is a timestamp of when the recommendations were made, where *YYYY* is the year, *MM* the month, *DD* the day, *HH* the hour, *MI* the minutes, and *SS* the seconds.

4. Click **Save**.

**Figure 4–45 Save index recommendations**



Your index recommendations are saved.

The saved file contains a header with following information:

- The timestamp of when the report was generated.
- The version of SQL Developer.
- The version of the TimesTen database.
- The name of the TimesTen connection.

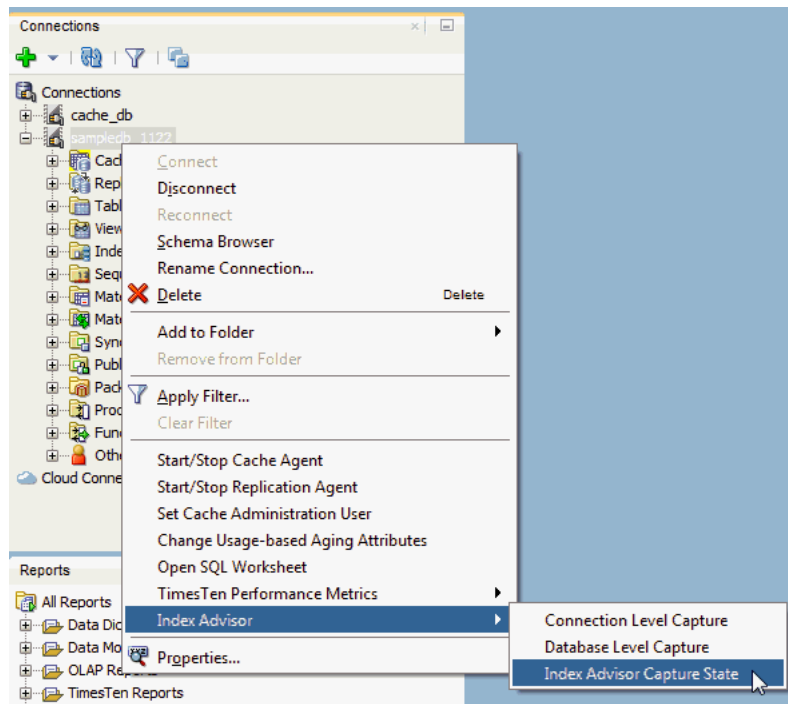
For example:

```
-- This file was generated by SQL Developer at 2013-04-29 11:08:45
-- SQL Developer version 4.0.0.11.51
-- Database version: 11.02.02.0005 Oracle TimesTen IMDB version 11.2.2.5.0
-- Connection name: sampledb_1122
```

## Viewing the TimesTen index advisor capture state

The TimesTen index advisor capture state enables you to view the status of connection and database level captures. To view the TimesTen index advisor capture state for a database, make sure you are on the main SQL Developer page.

1. Right-click the name of the database and select **Index Advisor**, then select **Index Advisor Capture State**.

**Figure 4–46 Viewing the TimesTen index advisor capture state**

The **Index Advisor Capture State** dialog displays and you see the current status of connection and database level captures.

You can also click on the column header to sort the table based on the column. A description of each column follows:

- **Connection ID**  
The connection ID of the connection that initiated the TimesTen index advisor capture.
- **Capture Level**  
The capture level of the TimesTen index advisor capture. This value can be Connection or Database.
- **Capture Mode**  
The capture mode of the TimesTen index advisor capture. This value can be Execute SQL or Prepare SQL.
- **Capture State**  
The capture state of the TimesTen index advisor capture. This value can be In Progress or Completed.
- **Prepared Count**  
The number of prepared SQL statements during the capture period.
- **Execute Count**  
The number of executed SQL statements during the capture period.
- **Start Time**

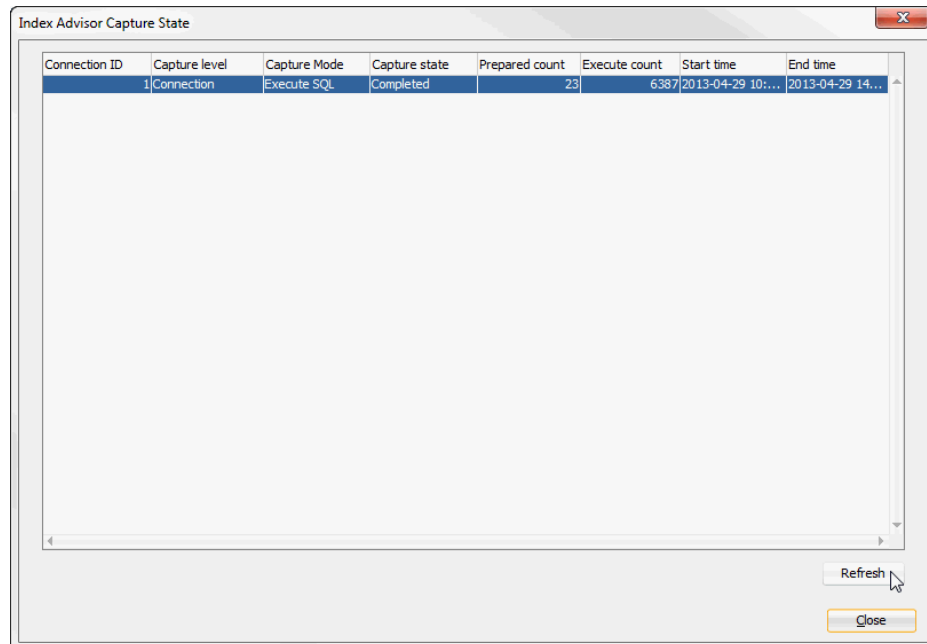
The timestamp of when the index advisor capture began. The timestamp format is *YYYY-MM-DD HH:MI:SS.FF*, where *YYYY* is the year, *MM* the month, *DD* the day, *HH* the hour, *MI* the minutes, *SS* the seconds, and *FF* the milliseconds.

- **End Time**

The timestamp of when the index advisor capture ended. The timestamp format is *YYYY-MM-DD HH:MI:SS.FF*, where *YYYY* is the year, *MM* the month, *DD* the day, *HH* the hour, *MI* the minutes, *SS* the seconds, and *FF* the milliseconds. This value is empty if the **Capture State** is in In Progress.

2. Click **Refresh** to update the table with the most current information of the TimesTen index advisor captures.

**Figure 4–47 Click refresh**



The **Index Advisor Capture State** dialog is refreshed with the most recent capture state information.





---

## Loading data from an Oracle database into a TimesTen database

SQL Developer allows you to load data using parallel threads from an Oracle database into a TimesTen database without creating a cache grid, cache group, or cache table.

---

**Note:** SQL Developer uses functions from the `ttLoadFromOracle` built-in procedure to load data from an Oracle database into a TimesTen database. For more information about the `ttLoadFromOracle` utility, see "ttLoadFromOracle" in the *Oracle TimesTen In-Memory Database Reference*.

---

This chapter provides information about loading Oracle database data into a TimesTen table.

Topics include:

- [Loading data from Oracle database tables](#)

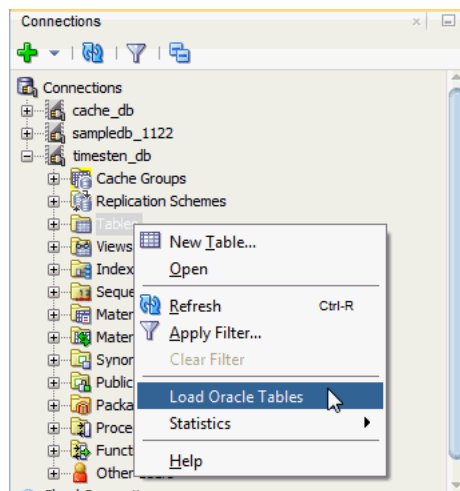
### Loading data from Oracle database tables

Before attempting to load data from an Oracle database into a TimesTen database:

- Make sure you are using Oracle TimesTen In-Memory Database Release 11.2.2.4 (or later).
- Make sure you have specified the Oracle connection information (Oracle Database Password and Oracle Net Service Name) for your TimesTen SQL developer connection. For more information on defining your TimesTen database connection, see "[Defining a TimesTen database named connection](#)" on page 1-2.

To load data from an Oracle database into a TimesTen database, make sure you are on the main SQL Developer page and that your TimesTen connection is expanded.

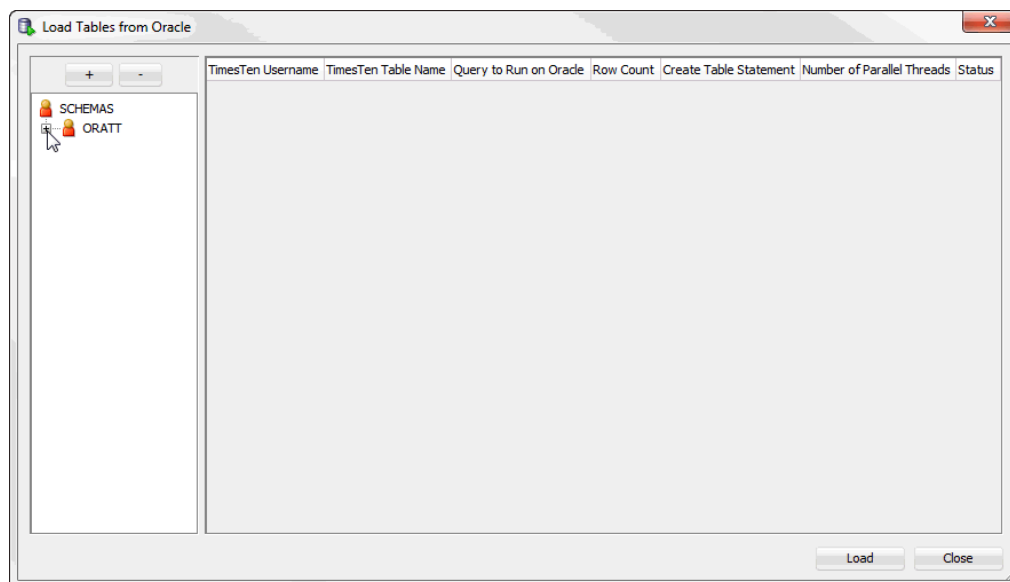
1. Right-click the **Tables** option and select **Load Oracle Tables**.

**Figure 5–1 Loading data from Oracle database tables**

The **Load Tables from Oracle** dialog displays. Locate your Oracle database schema name from the navigation tree on the left of the **Load Tables from Oracle** dialog.

2. Click the + located next to your Oracle database schema name to expand the Oracle database tables list.

To add another Oracle database schema, see ["Adding an Oracle database schema"](#) on page 5-6. To remove an Oracle database schema, see ["Removing an Oracle database schema"](#) on page 5-7. To load custom query results for an Oracle database schema, see ["Loading custom query results"](#) on page 5-9.

**Figure 5–2 Expand the Oracle database schema name**

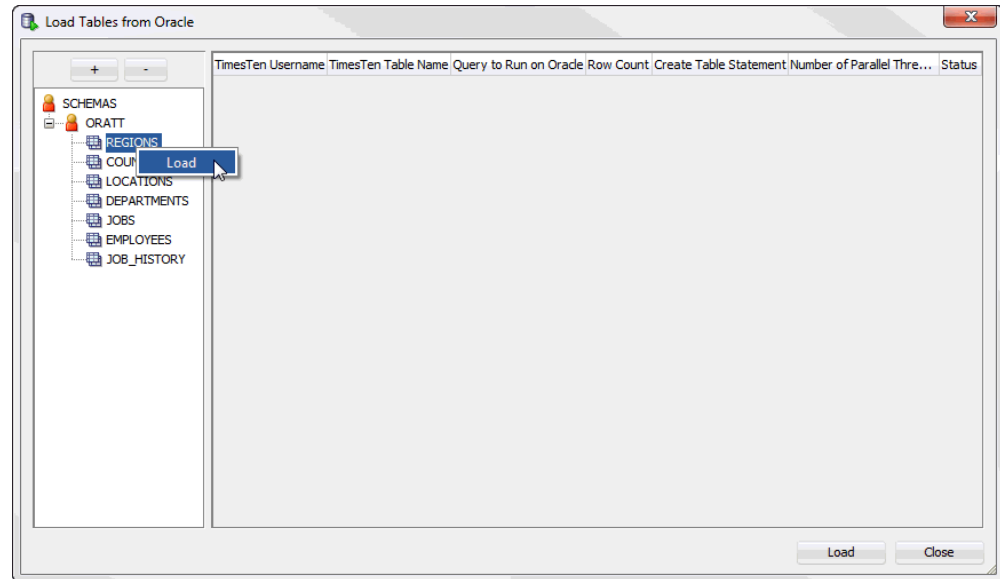
Your Oracle database schema expands showing a list of available Oracle database tables. Locate the Oracle database tables you want to load into the TimesTen database.

3. Right-click the name of the Oracle database table that you want to load into the TimesTen database and select **Load**. You can repeat this step, including for the same Oracle database table, to load multiple Oracle database tables.

If you want to load all of the Oracle database tables, right-click the Oracle database schema name and select **Load All**. The TimesTen Username dialog displays. Enter the TimesTen user for which you want to create the tables and click **OK**.

To apply a filter to the Oracle database schema tables, see ["Applying a filter to Oracle database schema tables"](#) on page 5-7.

**Figure 5–3 Load an Oracle database table**



Your selected tables are added to the right pane and are displayed in a table format.

4. Review the selected Oracle database tables and edit any necessary fields. You can edit a field by double clicking it. A description of each column follows:
  - **TimesTen Username**  
The owner of the TimesTen table that you are creating. By default, this is the TimesTen connection user. You can edit this column. An error message is displayed if the specified table owner does not exist or the TimesTen connection user does not have privileges to access the specified table owner.
  - **TimesTen Table Name**  
The name of the TimesTen table that you are creating. By default, this is the same table name as the Oracle database table. You can edit this column.
  - **Query to Run on Oracle**  
The SQL query executed on the Oracle database to generate the desired result set. By default, this is a `SELECT * FROM oracledb_tbl` query, where `oracledb_tbl` is the Oracle database table from which you are loading data. You can edit this column. For more information on valid Oracle Database SQL queries, see *Oracle Database SQL Language Reference*.
  - **Row Count**

The number of rows that TimesTen loads from the Oracle database table. By default, this value is empty.

To load **Row Count** for a specific Oracle database table, right-click on the **Row Count** field for your table, then select **Count**.

To load **Row Count** for all the selected Oracle database tables, right-click on a **Row Count** field of any table, then select **Count All**. A confirmation dialog displays. Click **Yes** to continue with the row count of all of your previously selected Oracle database tables.

- **Create Table Statement**

The SQL statement that TimesTen uses to create the table to load your Oracle database data. You cannot edit this column. Review the `CREATE TABLE` statement by double clicking on the **Create Table Statement** field.

If a TimesTen table with the same name already exists in the TimesTen database, the data from the Oracle database table is appended to the existing TimesTen table. TimesTen does not load table constraints from the Oracle database table. An error dialog displays if a TimesTen table with the same name but different table structure already exists in the TimesTen database.

- **Number of Parallel Threads**

The number of parallel threads that will be used to load the Oracle database table into the TimesTen database. By default, the value is four. You can edit this column. An error dialog displays if you attempt to use a number of parallel threads that is less than two.

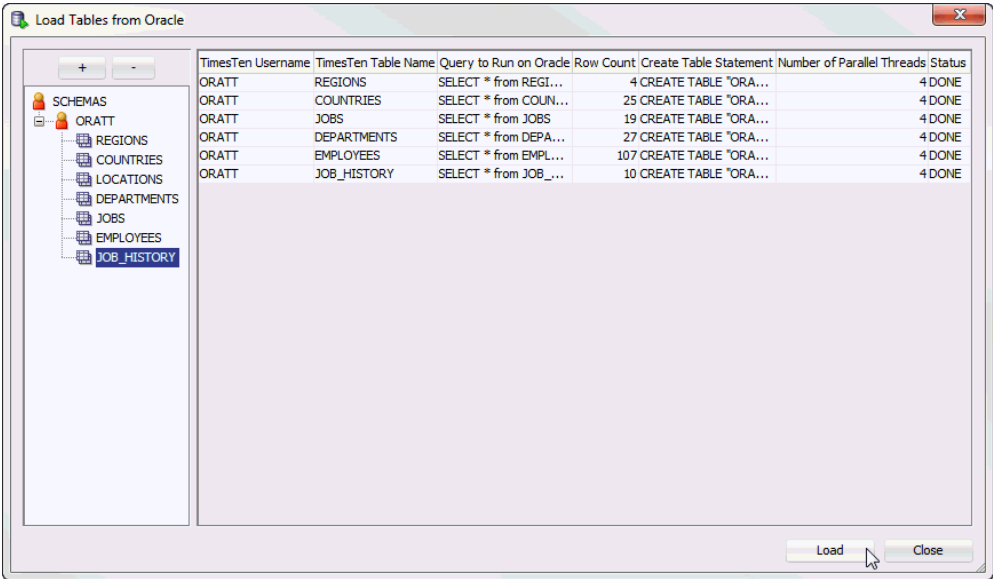
- **Status**

The status of the data loading operation from the Oracle database to the TimesTen database. This value is empty before starting the load operation. Once you start the load tables from the Oracle database operation, this value can be `IN PROGRESS`, `DONE`, or `FAILED`.

Locate the **Load** button in the bottom right of the **Load Tables from Oracle** dialog.

5. Once you edit all of your desired fields, click the **Load** button.

Figure 5–4 Click Load



A progress dialog displays. Verify that each field of the **Status** column of the **Load Tables from Oracle** dialog is marked as DONE.

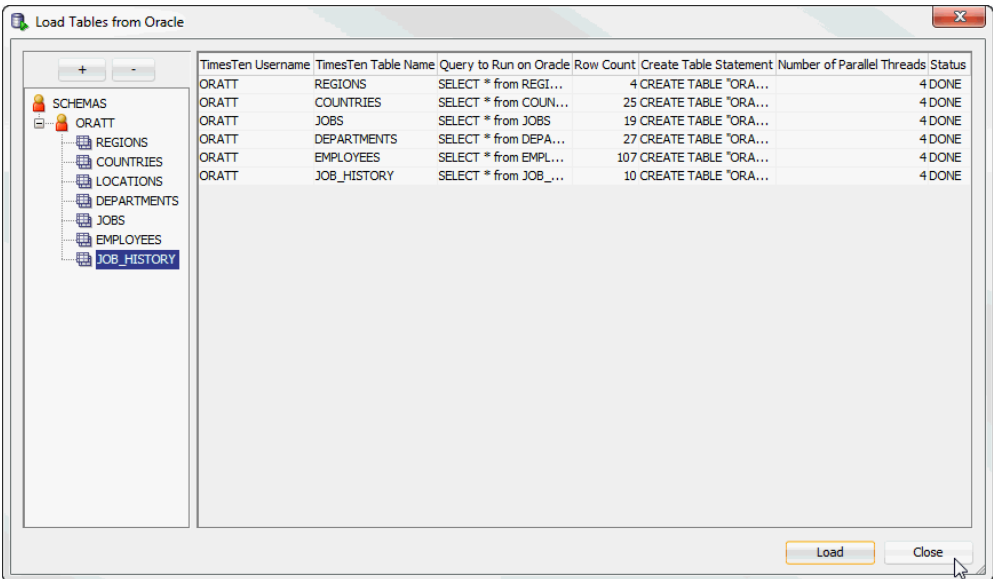
If any **Status** field is marked as FAILED, make sure that all the data you entered is correct.

If all of the **Status** fields are marked as DONE, your TimesTen tables are created and populated with the Oracle database table data.

Locate the **Close** button in the bottom right of the **Load Tables from Oracle** dialog.

- 6. Click Close.

Figure 5–5 Click Close



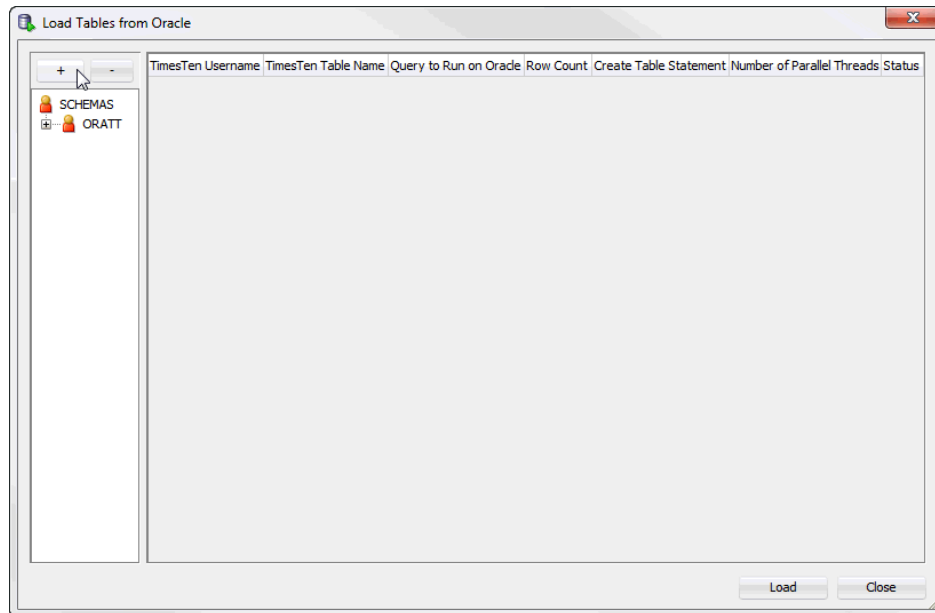
You have successfully loaded data from the Oracle database to the TimesTen database.

## Adding an Oracle database schema

To add tables from another Oracle database schema, follow these steps. Before following these steps, make sure you have completed the steps up to step 1 from ["Loading data from Oracle database tables"](#) on page 5-1.

1. Click the + located above the Oracle database schema navigation tree.

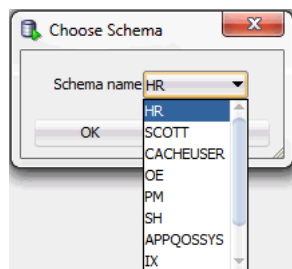
**Figure 5–6 Add an Oracle database schema**



The **Choose Schema** dialog displays. Only Oracle database schemas that the connection user has access to display.

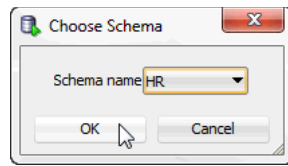
2. Select the Oracle database schema from the **Schema name** drop-down list.

**Figure 5–7 Choose an Oracle database schema**



You are now ready to add the Oracle database schema.

3. Click **Ok**.

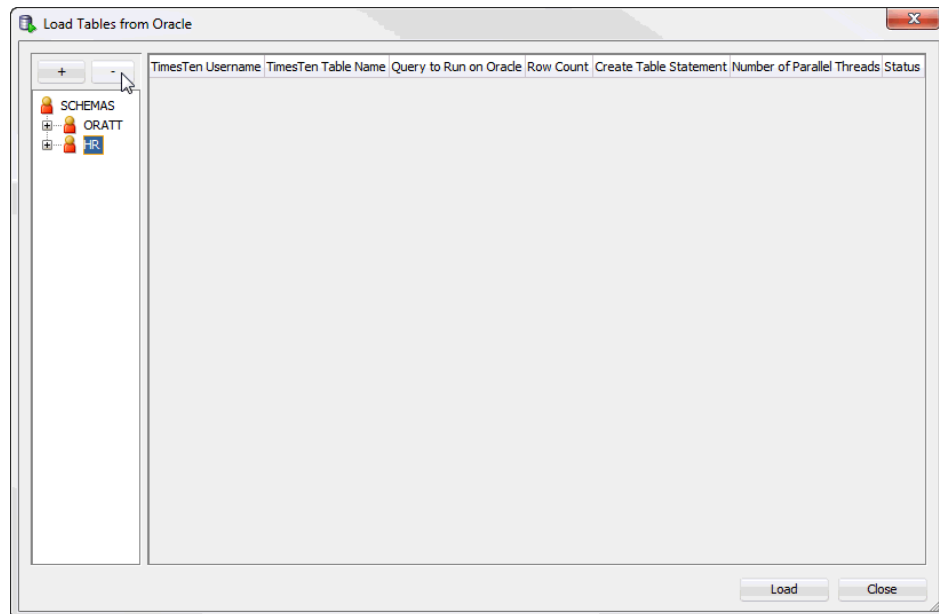
**Figure 5–8 Click Ok**

You have successfully added the Oracle database schema. To load data from the tables of the Oracle database schema, see step 2 from ["Loading data from Oracle database tables"](#) on page 5-1.

## Removing an Oracle database schema

To remove an Oracle database schema, follow these steps. This only removes Oracle database schemas from being available for the Load from Oracle database feature. You cannot remove the Oracle database schema of the connection user. Before following these steps, make sure you have completed the steps up to step 1 from ["Loading data from Oracle database tables"](#) on page 5-1.

1. Select the Oracle database schema that you want to remove in the schema navigation tree.
2. Click the - located above the Oracle database schema navigation tree.

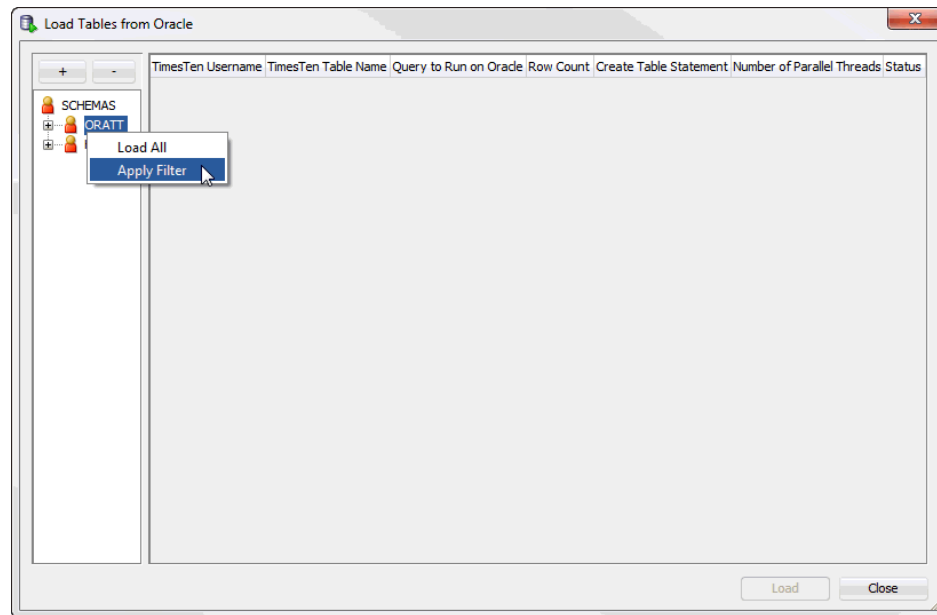
**Figure 5–9 Remove the Oracle database schema**

You have successfully removed the Oracle database schema.

## Applying a filter to Oracle database schema tables

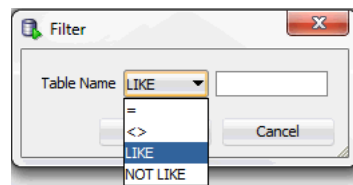
To apply a filter to the Oracle database schema tables, follow these steps. Before following these steps, make sure you have completed the steps up to step 1 from ["Loading data from Oracle database tables"](#) on page 5-1.

1. Right-click the name of the Oracle database schema that you want to apply a filter to and select **Apply Filter**.

**Figure 5–10 Apply Filter**

The **Filter** dialog displays.

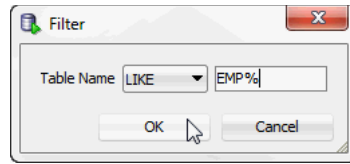
2. Select the filter condition from the **Table Name** drop-down list:
  - **=**: Returns results that are equal to a specified pattern.
  - **<>**: Returns results that are not equal to a specified pattern.
  - **LIKE**: Returns results that match a specified pattern.
  - **NOT LIKE**: Returns results that do not match a specified pattern.

**Figure 5–11 Filter condition**

Locate the text field located next to the **Table Name** drop-down list.

3. Specify a pattern for the filter condition in the text field located next to the **Table Name** drop-down list. You can use a wildcard for your filter:
  - **\_**: A substitute for exactly one character. For example `Sm_th` shows all tables that contain five characters, begin with `Sm`, and end with `th`.
  - **%**: A substitute for zero or more characters. For example `EMP%` shows all tables that begin with the letters `EMP`.
4. Click **OK**.



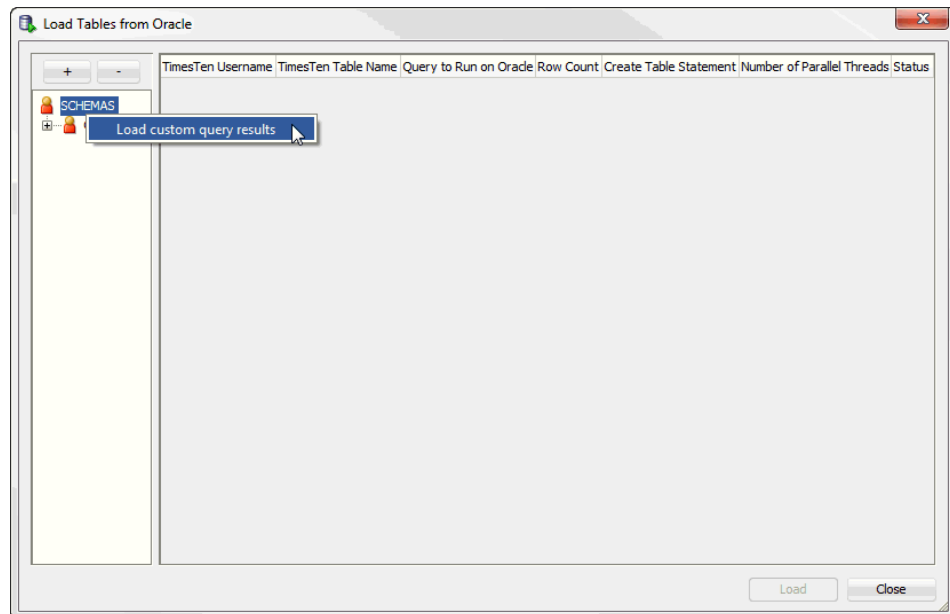
**Figure 5–12 Click OK**

You have successfully applied a filter to the Oracle database schema tables. To load data from the tables of the Oracle database schema, see step 2 from ["Loading data from Oracle database tables"](#) on page 5-1.

## Loading custom query results

To load custom query results for an Oracle database schema, follow these steps. Before following these steps, make sure you have completed the steps up to step 1 from ["Loading data from Oracle database tables"](#) on page 5-1.

1. Right-click the **Schemas** option and select **Load custom query results**.

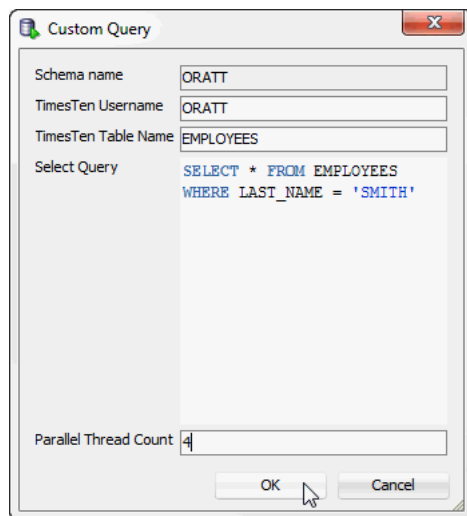
**Figure 5–13 Load custom query results**

The **Custom Query** dialog displays. You are now required to enter the information for your custom Oracle query. The first property is the **Schema name**. You cannot edit this property and the default value is the connection user.

2. In the **TimesTen Username** text field, enter the owner of the TimesTen table. Make sure that the table owner you specify exists in TimesTen and that the TimesTen connection user has privileges to create a table or insert into an existing table owned by the specified table owner.
3. In the **TimesTen Table Name** text field, enter the name of the TimesTen table.
4. In the **Select Query** text field, enter the SQL query that you want to execute on the Oracle database to generate the desired result set. For more information on valid Oracle Database SQL queries, see *Oracle Database SQL Language Reference*.

5. In the **Parallel Thread Count** text field, enter the number of parallel threads that will be used to load the Oracle database table into the TimesTen database. TimesTen recommends a value of four.
6. Click **OK**.

**Figure 5–14 Load the custom query**



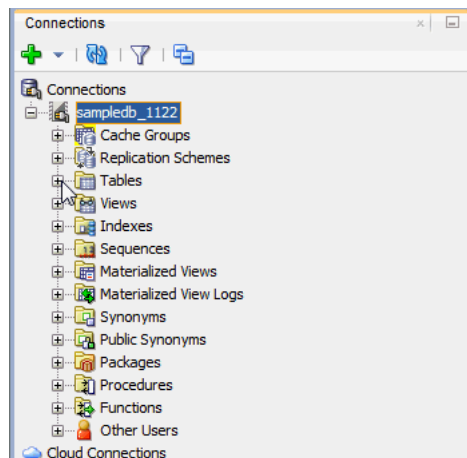
You have successfully loaded a custom Oracle database query. Your custom query is loaded into the right pane of the Load Tables from Oracle dialog. For more information on how to load Oracle database data from your custom query into a TimesTen database, see step 4 from ["Loading data from Oracle database tables"](#) on page 5-1.

## Loading data from an Oracle database table into an existing TimesTen table

To load data from an Oracle database table into an existing TimesTen table, follow these steps. Make sure you are on the main SQL Developer page and that your TimesTen connection is expanded.

1. Click the + located next to **Tables** to expand the TimesTen tables list.

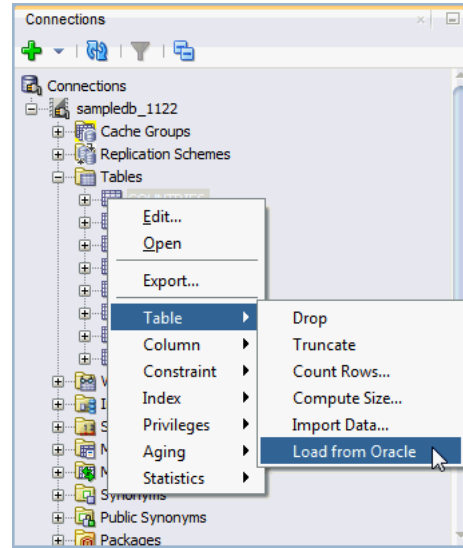
**Figure 5–15 Expand Tables**



The TimesTen tables list expands.

2. Right-click the name of the table and select the **Table** option, then select the **Load from Oracle** option.

**Figure 5-16 Load data from an Oracle database table**



The **Load Tables from Oracle** dialog displays. The Schema name, TimesTen Username, and TimesTen Table Name are auto-filled and you cannot edit these fields. Prepare to enter the SQL query and the Parallel Thread Count.

3. In the **Select Query** field, enter the SQL query to execute on the Oracle database. The result of the SQL query is loaded into your TimesTen table. For more information on valid Oracle Database SQL queries, see *Oracle Database SQL Language Reference*.
4. In the **Parallel Thread Count** field, enter the number of parallel threads that will be used to load the Oracle database table into the TimesTen database. An error dialog displays if you attempt to use a number of parallel threads that is less than two.
5. Click **Load**.

**Figure 5–17 Load Tables from Oracle**

Load Tables from Oracle

Schema name: TIMESTEN

TimesTen Username: TIMESTEN

TimesTen Table Name: COUNTRIES

Select Query: `SELECT * FROM ORATT.COUNTRIES`

Parallel Thread Count: 4

Load Close

A progress dialog displays. Once TimesTen loads the data from the Oracle database table, the progress dialog closes. If you execute the query on a small Oracle database table, the progress dialog may close almost immediately.

---

## Prerequisites to Creating and Accessing a Cache Environment

After you create the TimesTen database, and before you cache Oracle data in the TimesTen database, you must perform the tasks:

- [Configuring the Oracle and TimesTen databases](#)
- [Set the cache administration user name and password](#)
- [Configure a cache grid](#)
- [Start and stop the cache agent](#)

### Configuring the Oracle and TimesTen databases

Before you cache Oracle data in a TimesTen database, you must perform certain tasks either outside of SQL Developer using command-line interfaces such as SQL\*Plus and the `ttIsql` utility, or within SQL Developer using the **SQL Worksheet**.

The tasks include:

- Creating the timesten user, a schema user and the cache administration user in the Oracle database
- Granting privileges required to perform cache operations to the Oracle users
- Creating the cache manager user and the cache table users in the TimesTen database
- Granting cache privileges to the TimesTen users

For information about creating Oracle and TimesTen users, see "Create the Oracle users" and "Create the TimesTen users" in the *Oracle In-Memory Database Cache User's Guide*.

For information on configuring Oracle and TimesTen databases, see "Configuring the Oracle database to cache data in TimesTen" and "Configuring a TimesTen database to cache Oracle data" in the *Oracle In-Memory Database Cache User's Guide*.

### Set the cache administration user name and password

You must create a cache administration user in the Oracle database. This user creates, owns and maintains Oracle objects that store information used to manage a specific cache grid. This user also enforces predefined behaviors of cache groups.

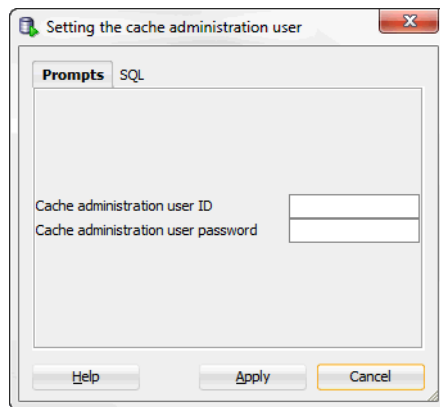
After you create the cache administration user in the Oracle database, you must set the name and password for the user in the TimesTen database.

To set the cache administration user name and password, right-click the node of the connection name for the TimesTen database and select **Set Cache Administration User**. See [Figure 1–7, "Viewing a TimesTen connection"](#).

The **Setting the cache administration user** dialog appears. For the **Prompts** tab, in the **Cache administration user ID** field, enter the cache administration user name. In the **Cache administration user password** field, enter the Oracle password for the cache administration user.

Click **Apply**.

**Figure 6–1** *Setting the cache administration user dialog*



You must set the cache administration user name and password for each TimesTen database that caches Oracle. If you drop and re-create the TimesTen database or drop and re-create the cache administration user in the Oracle database, then the cache administration user name and password must be reset in the TimesTen database.

For more information about the cache administration user, see "Set the cache administration user name and password" in the *Oracle In-Memory Database Cache User's Guide*.

## Configure a cache grid

Before you can create a cache group in a TimesTen database, you must perform certain tasks either outside of SQL Developer using a command-line interface such as the `ttIsql` utility, or within SQL Developer using **SQL Worksheet**. See "Configuring a cache grid" in the *Oracle In-Memory Database Cache User's Guide* for information regarding the following tasks:

- Create a cache grid
- Associate the TimesTen database with the cache grid

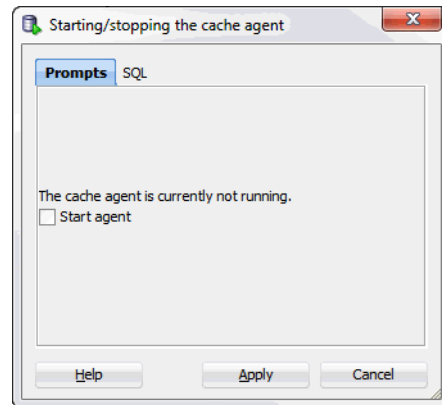
## Start and stop the cache agent

After you set the cache administration user name and password, create a cache grid and associate the TimesTen database with the cache grid, you must start the TimesTen cache agent process. The cache agent processes cache operations such as loading a cache group and automatic refresh.

To start the cache agent, right-click the node of the connection name for the TimesTen database and select **Start/Stop Cache Agent**. See [Figure 1–7, "Viewing a TimesTen connection"](#).

In the **Starting/stopping the cache agent** dialog box, select **Start agent** within the **Prompts** tab. Then click **Apply** to start the cache agent.

**Figure 6–2 Starting/stopping the cache agent**



To stop a running cache agent process, right-click the node of the connection name for the TimesTen database and select **Start/Stop Cache Agent** in an identical fashion as was done when starting the cache agent.

In the **Starting/stopping the cache agent** dialog box, select **Stop agent** within the **Prompts** tab. Then click **Apply** to stop the cache agent. See [Figure 6–2, "Starting/stopping the cache agent"](#).

For more information about the cache agent process, see "Managing the cache agent" in the *Oracle In-Memory Database Cache User's Guide*.





---

## Creating Cache Groups

A cache group defines the set of Oracle data to cache in a TimesTen database. When you create a cache group, cache tables are created in the TimesTen database that correspond to the Oracle tables being cached.

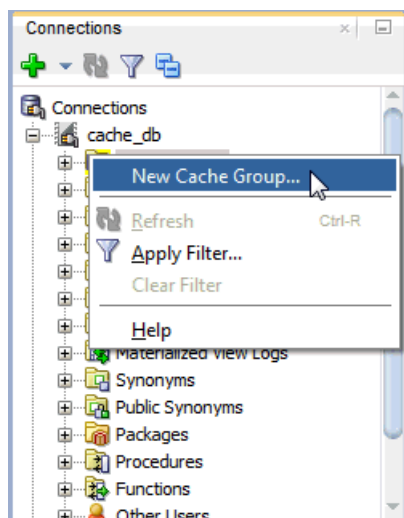
This chapter includes topics:

- [Creating a cache group](#) including:
  - [Defining a cache group and cache table attributes](#)
  - [Defining the root table](#)
  - [Defining the child tables](#)
- [Dropping a cache group](#)

### Creating a cache group

To create a cache group, right-click the **Cache Groups** node under the node of the connection name for the TimesTen database and select **New Cache Group**.

**Figure 7–1** *Creating a new cache group*



In the **Create cache group** dialog, specify a name for the cache group in the **Name** field within the **Cache Group Attributes** tab. For **Type**, choose the desired cache group type (read-only, asynchronous writethrough, synchronous writethrough or user managed).

Cache groups are categorized as either explicitly loaded or dynamic. If you are creating a dynamic cache group, then choose **Dynamic** in the **Type** section.

In an explicitly loaded cache group, data is loaded manually into its cache tables from the corresponding Oracle tables using a load or refresh operation, or automatically using an automatic refresh operation. In a dynamic cache group, data is loaded into its cache tables on demand from the corresponding Oracle tables using a dynamic load operation, or manually using a load operation.

A read-only, asynchronous writethrough (AWT) or synchronous writethrough (SWT) cache group can be defined as a dynamic cache group. A user managed cache group can be defined as a dynamic cache group if it does not contain all of the following characteristics:

- Automatic refresh is defined
- At least one of its cache tables are updatable
- Committed updates on one or more of its updatable cache tables are propagated to the corresponding Oracle tables

For more information about explicitly loaded and dynamic cache groups, see "Loading data into a cache group: Explicitly loaded and dynamic cache groups" and "Dynamic cache groups" in the *Oracle In-Memory Database Cache User's Guide*.

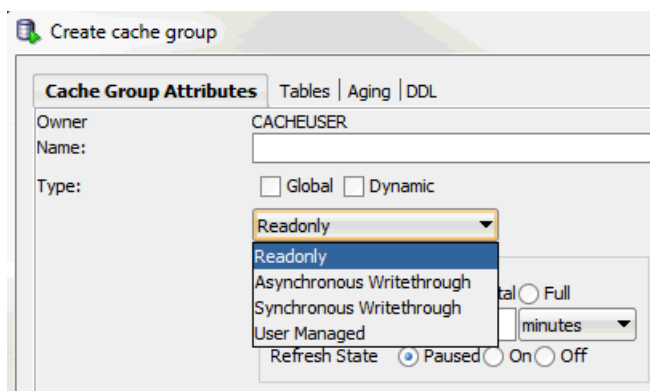
Cache groups are classified as either local or global. If you are creating a global cache group, then select the **Global** in the **Type** section.

In a local cache group, data in its cache tables are not shared across TimesTen databases even if the databases are members of the same cache grid. In a global cache group, data in its cache tables are shared across TimesTen databases that are members of the same cache grid.

Only an AWT cache group can be defined as global cache group.

For more information about local and global cache groups, see "Sharing data across a cache grid: Local and global cache groups" in the *Oracle In-Memory Database Cache User's Guide*.

**Figure 7–2 Defining cache group type**



## Defining a cache group and cache table attributes

Within the **Cache Group Attributes** tab of the **Create cache group** dialog, if you are creating a user managed cache group and you select **Autorefresh** within the **Table Attributes** section to enable automatic refresh when either **All tables readonly** or

**Propagate all tables** is selected, or you are creating a read-only cache group, then for the following automatic refresh characteristics:

- Set the automatic refresh mode in the **Mode** field by choosing either **Full** or **Incremental**. The default automatic refresh mode is Incremental.
- Set the automatic refresh interval in the **Refresh Interval** field by specifying a numeric value followed by a unit of time in minutes, seconds or milliseconds. The default automatic refresh interval is 5 minutes.
- Set the automatic refresh state in the **Refresh State** field by choosing either **Paused**, **On** or **Off**. The default automatic refresh state is Paused.

**Figure 7–3 Autorefresh attributes**

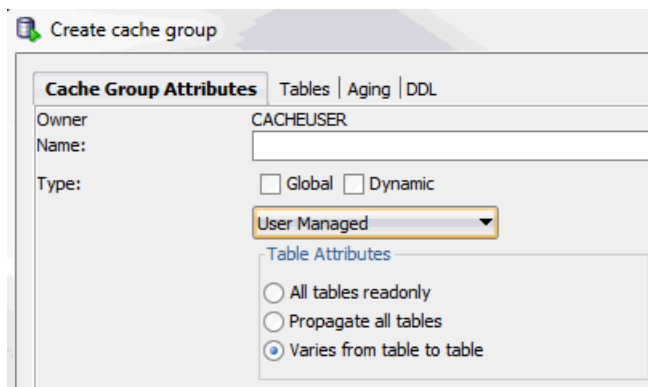
The screenshot shows the 'Create cache group' dialog box with the 'Cache Group Attributes' tab selected. The 'Owner' field is set to 'CACHEUSER'. The 'Name' field is empty. The 'Type' section has three options: 'Global' (unchecked), 'Dynamic' (unchecked), and 'Readonly' (selected). Below this, the 'Autorefresh Attributes' section is expanded, showing three settings: 'Mode' with 'Incremental' selected (radio button), 'Refresh Interval' set to '5 minutes' (text box and dropdown), and 'Refresh State' with 'Paused' selected (radio button).

For more information about automatic refresh of cache groups, see "AUTOREFRESH cache group attribute" in the *Oracle In-Memory Database Cache User's Guide*.

For more information about read-only cache groups, see "Read-only cache group" in the *Oracle In-Memory Database Cache User's Guide*.

If you are creating a user managed cache group, then from the **Table Attributes** section in the **Cache Group Attributes** tab within the **Create cache group** dialog, choose one:

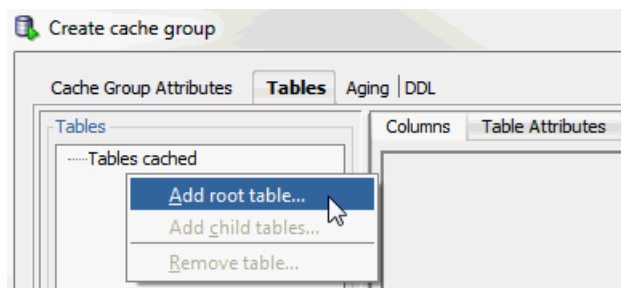
- **All tables readonly:** All cache tables are read-only
- **Propagate all tables:** All cache tables are updatable and committed updates on the cache tables are propagated to the corresponding Oracle tables
- **Varies from table to table:** Some cache tables are read-only and other cache tables are updatable where committed updates on the cache tables may or may not be propagated to the corresponding Oracle tables

**Figure 7–4 User managed cache group attributes**

For more information about user managed cache groups, see "User managed cache group" in the *Oracle In-Memory Database Cache User's Guide*.

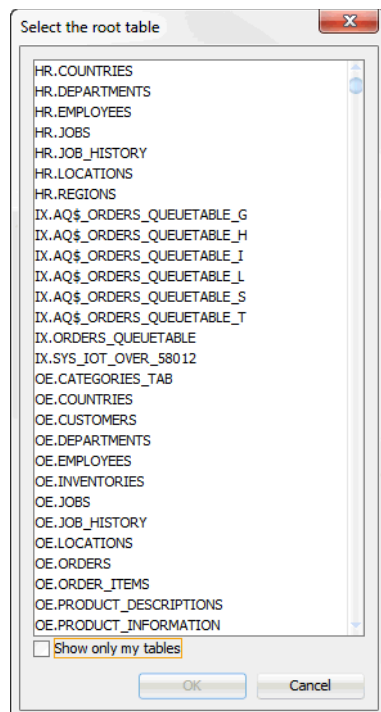
## Defining the root table

Click the **Tables** tab in the **Create cache group** dialog. Right-click in the **Tables** panel and select **Add root table**.

**Figure 7–5 Adding the root table**

The **Select the root table** dialog appears. The list of Oracle tables and Oracle synonyms that you can choose to designate as the root table for your cache group is displayed. Oracle tables that have primary keys and Oracle tables that have unique non-null indexes are available for selection.

Select the table and click **OK**.

**Figure 7-6** List of tables and synonyms for root table

To show only Oracle tables owned by the current connection user, choose **Show only my tables**.

For more information on caching Oracle synonyms, see "Caching Oracle synonyms" in the *Oracle In-Memory Database Cache User's Guide*.

### Selecting the columns to cache

In the **Columns** tab within the **Tables** tab of the **Create cache group** dialog, select the check boxes under the **Inline** field to specify the `INLINE` attribute for the column. You can specify the `INLINE` attribute for columns of type `VARCHAR2`, `NVARCHAR2`, and `VARBINARY`.

Select the check boxes under the **Allow null values** field for each cached column to accept null values. By default, the nullability of a cache table's column is identical to the nullability of the corresponding Oracle table's column.

Select the check boxes under the **Cached** field for each column to cache. By default, all columns of the table are selected and are cached. The Oracle table's unique identifying columns (primary key, unique index or unique constraint) must be included in the TimesTen cache table and constitutes the cache table's primary key.

To select all columns or deselect all columns, except for the Oracle table's unique identifying columns, to accept null values or to cache, select or deselect the check boxes that appear to the left of the **Allow null values** and **Cached** fields, respectively.

For more information about the characteristics of cache tables and cache groups, see "Cache groups and cache tables" in the *Oracle In-Memory Database Cache User's Guide*.

Some Oracle data types have multiple compatible TimesTen data types. For columns in the Oracle table whose data type can map multiple TimesTen data types, a menu appears under the **Data type in TimesTen** field for those columns. Select the desired data type for the column in the cache table. The default compatible TimesTen data type for the Oracle data type appears in italics.

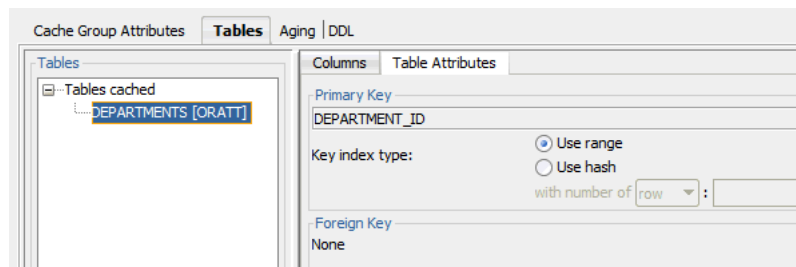
For more information about compatibility between Oracle and TimesTen data types, see "Mappings between Oracle and TimesTen data types" in the *Oracle In-Memory Database Cache User's Guide*.

For updatable cache tables, you can specify a default column value under the **Default Value** field for each cached column to define a default value. By default, the default value of a cache table's column is identical to the default value of the corresponding Oracle table's column.

### Specifying the table attributes

By default, TimesTen creates a range index on the cache table's primary key columns. To create a hash index instead of a range index on the primary key columns, in the **Primary Key** section of the **Table Attributes** tab within the **Tables** tab of **Create cache group**, choose **Use hash**. You must size the hash index by specifying either the maximum number of rows the underlying cache table is anticipated to have or the number of pages to allocate to the index in the **with number of** menu and the adjacent field after the colon (:) which requires a numeric value. The default selection is row.

**Figure 7–7 Primary key attributes**



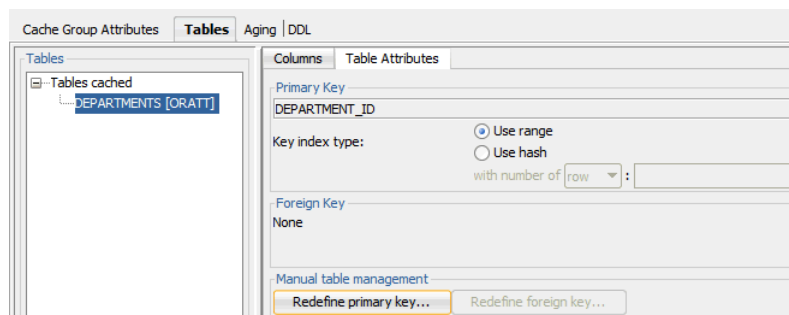
For more information about the index that is automatically created on a cache table's primary key columns, see "CREATE CACHE GROUP" in the *Oracle TimesTen In-Memory Database SQL Reference*.

For more information about properly sizing a hash index, see "CREATE TABLE" in the *Oracle TimesTen In-Memory Database SQL Reference*.

By default, the primary key columns of a cache table are identical to the primary key columns of the corresponding Oracle table. You can define a different column or set of columns as the primary key for the cache table.

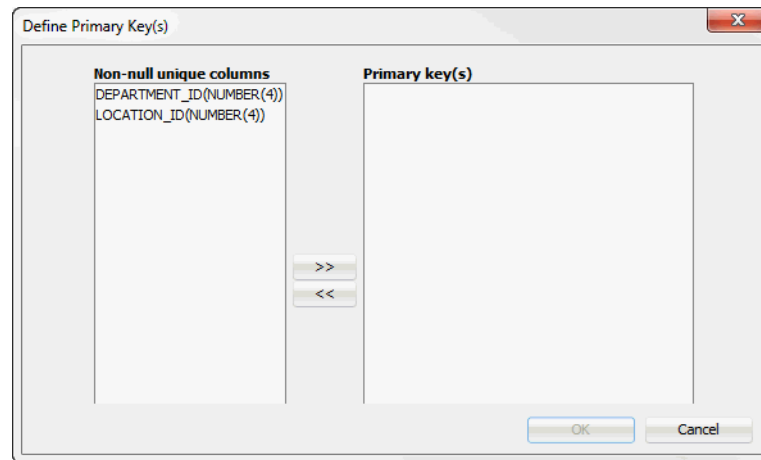
In the **Manual table management** section of the **Table Attributes** tab within the **Tables** tab of the **Create cache group** dialog, click **Redefine primary key**.

**Figure 7–8 Redefining a primary key**



In the **Define Primary Key(s)** dialog, the non-nullable unique columns of the corresponding Oracle table is shown in the **Non-null unique columns** list. Select the columns for the primary key of the cache table and then choose **>>** to move those columns into the **Primary key(s)** list. The order of the columns in a composite primary key is significant if a range index is to be created on the columns. Click **OK** to define the alternate primary key for the cache table.

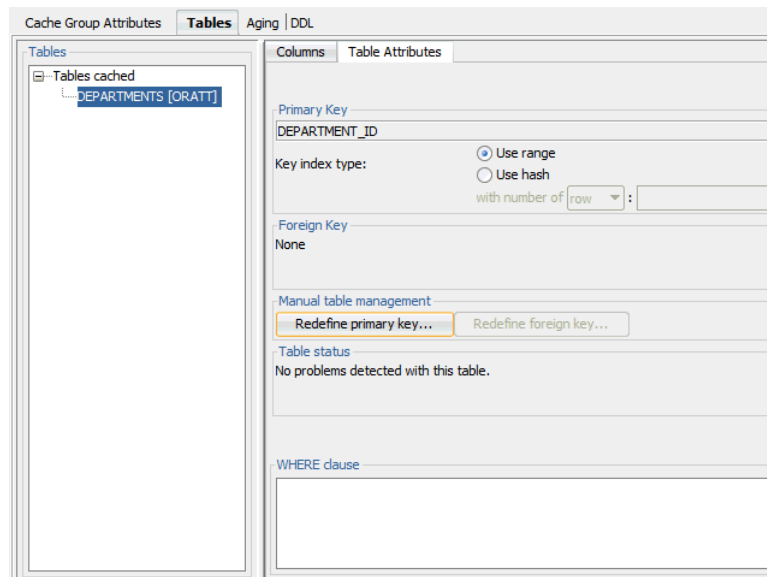
**Figure 7–9** Defining a new primary key



If the primary key of the cache table consists of a different set of columns than the primary key of the corresponding Oracle table, then the tables may become unsynchronized when committed updates are made on the cache table or the corresponding Oracle table. Unique constraint violations can occur when:

- Committed updates on the cache table are propagated to the corresponding Oracle table
- Committed updates on the Oracle table are refreshed to the corresponding cache table

You can specify an optional **WHERE** clause for the cache table in the **WHERE clause** field of the **Table Attributes** tab within the **Tables** tab of **Create cache group**. A **WHERE** clause is only allowed for table definitions in a read-only cache group or a user managed cache group that has automatic refresh defined.

**Figure 7–10 Specifying a where clause**

For more information about specifying a WHERE clause in the table definition of a cache group, see "Using a WHERE clause" in the *Oracle In-Memory Database Cache User's Guide*.

If the cache table is in a user managed cache group that has selected the **Varies from table to table** in the **Table attributes** section within the **Cache Group Attributes** tab of the **Create cache group** dialog, then you must specify the table attribute for the cache table.

In the **Propagation Type** section of the **Table Attributes** tab within the **Tables** tab of **Create cache group**, choose:

- **Readonly** to define the cache table as read-only
- **Propagate** to define the cache table as updatable with updates on the table to be propagated to the corresponding Oracle table
- **Neither** to define the cache table as updatable with updates on the table not propagated to the corresponding Oracle table

By default, cache tables in a user managed cache group are updatable with updates on the table not propagated to the corresponding Oracle table.

### Specifying an aging policy

You can specify an aging policy for the cache table in the **Aging** tab of the **Create cache group** dialog. An aging policy is explicitly defined on the root table of a cache group and it implicitly applies to all the child tables.

Choose:

- **Usage-based** to define a least recently used (LRU) aging policy
- **Time-based** to define a time-based aging policy
- **No aging** to not define an aging policy

An LRU aging policy can be defined for tables in an explicitly loaded AWT, SWT or user managed cache group that does not have automatic refresh defined. The default aging state is set to off for global explicitly loaded cache groups. For dynamic cache



groups that do not have automatic refresh defined, the default aging state is set to on and the type of aging is set to usage-based.

The TimesTen database memory usage thresholds that determine when data starts and stops being deleted from tables that are subject to LRU aging are shown when the **Usage-based** is selected. The default usage threshold that determines when data starts being deleted from the tables is 90%. The default usage threshold that determines when data stops being deleted from the tables is 80%. The default LRU aging cycle is 1 minute.

For information about how to change the memory usage thresholds and LRU aging cycle, see ["Changing the memory usage thresholds and LRU aging cycle"](#) on page 2-17.

**Figure 7–11 Specifying usage-based aging**

The screenshot shows the 'Aging' tab of the 'Create cache group' dialog. The 'Usage-based' radio button is selected and highlighted with a yellow box. Below it, the 'On' radio button is selected. The 'Current usage threshold' section explains that rows are removed when memory use reaches 90.0% of PermSize and are kept until it drops to 80.0%. The 'Time-based' and 'No Aging' options are unselected.

For more information about defining an LRU aging policy on cache tables, see "LRU aging" in the *Oracle In-Memory Database Cache User's Guide*.

A time-based aging policy can be defined on a cache table that contains a non-nullable DATE or TIMESTAMP column. Columns in the table of these data types appear in the **Column** within the **Aging** tab of the **Create cache group** dialog when **Time-based** is selected. Select the column to store the timestamp value indicating when each row was inserted or most recently updated.

In the **Life-time** field, indicate the length of time in which data that has not been updated is to be kept in the table by specifying a numeric value followed by a unit of time in minutes, hours or days from the drop-down menu list.

In the **Cycle** field, indicate the frequency in which data is to be aged out of the table by specifying a numeric value followed by a unit of time in minutes, hours or days in the drop-down menu list. The default time-based aging cycle is 5 minutes.

Specify an aging state by choosing **on** or **off**.

**Figure 7–12 Specifying time-based aging**

The screenshot shows the 'Aging' tab of the 'Create cache group' dialog. The 'Time-based' radio button is selected and highlighted with a yellow box. The 'Column' dropdown menu shows 'No timestamp columns found'. The 'Life-time' field has a numeric value and a 'minutes' unit dropdown. The 'Cycle' field has a numeric value of 5 and a 'minutes' unit dropdown. The 'On' radio button is selected. The 'Usage-based' and 'No Aging' options are unselected.

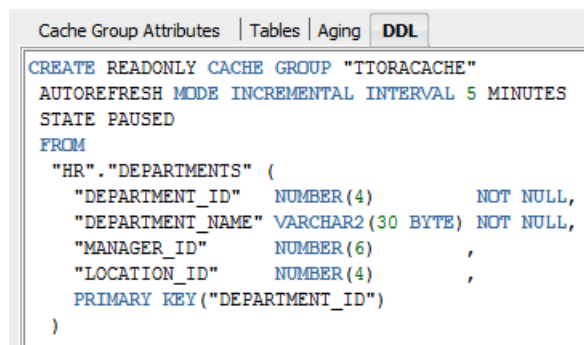
For more information about defining a time-based aging policy on cache tables, see "Time-based aging" in the *Oracle In-Memory Database Cache User's Guide*.

### Verifying the root table definition

After you have defined the tables and columns to cache, the type of index to create on the primary key columns, the table attributes including an optional `WHERE` clause, and an optional aging policy for the root table, you can view the `CREATE CACHE GROUP` statement used to create the cache group and its cache table in the **DDL** tab of the **Create cache group** dialog.

Click **Apply** to create the cache group and its cache table if you are not going to add any child tables to the cache group.

**Figure 7–13 Verifying the cache group definition**



For information about the syntax and semantics of the `CREATE CACHE GROUP` statement, see "CREATE CACHE GROUP" in the *Oracle TimesTen In-Memory Database SQL Reference*.

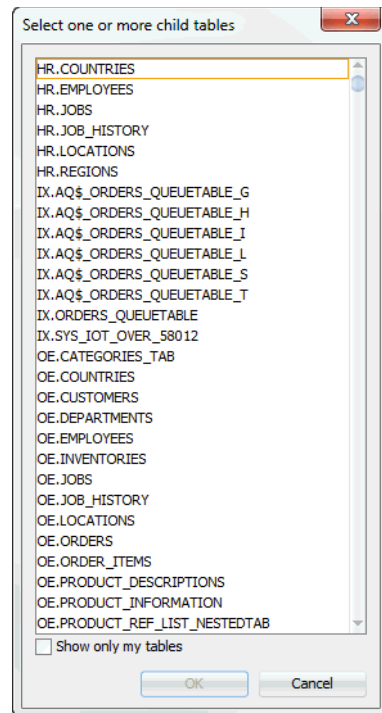
## Defining the child tables

Within the **Tables** tab of the **Create cache group** dialog, right-click in the **Tables** panel and select **Add child tables**.

The **Select one or more child tables** dialog appears. This dialog displays the list of Oracle tables and Oracle synonyms that you can choose to designate as the child table(s) for your cache group. Oracle tables that have primary keys and Oracle tables that have unique non-null indexes are available for selection.

Select the tables to designate as the child tables for your cache group and click **OK**.

For more information on caching Oracle synonyms, see "Caching Oracle synonyms" in the *Oracle In-Memory Database Cache User's Guide*.

**Figure 7-14** List of table and synonyms for child tables

To select multiple child tables, press and hold the CTRL key, and click the desired tables. The child tables appear under their appropriate parent table. You can also drag and drop a particular child table so that it appears under the desired parent table.

To show only Oracle tables owned by the current connection user, choose **Show only my tables**.

### Selecting the columns to cache

Click a child table and in the **Columns** tab within the **Tables** tab of the **Create cache group** dialog, select the check boxes under the **Allow null values** field for each cached column to accept null values. By default, the nullability of a cache table's column is identical to the nullability of the corresponding Oracle table's column.

Select the check boxes under the **Cached** field for each column to cache. By default, all columns of the table are selected and are cached. The Oracle table's unique identifying columns must be included in the TimesTen cache table and constitute the cache table's primary key. The Oracle table's foreign key columns that reference its parent table must also be included in the cache table.

To select all columns or deselect all columns, except for the Oracle table's unique identifying columns and the foreign key columns that reference its parent table, to accept null values or to cache, select or deselect the check boxes that appear to the left of the **Allow null values** and **Cache** fields.

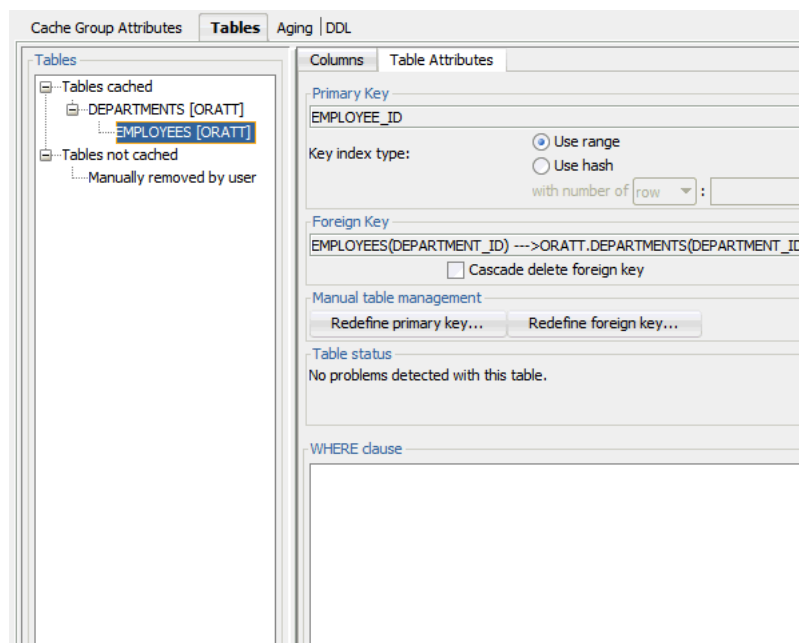
For columns in the Oracle table whose data type can map to multiple TimesTen data types, a drop-down select list appears under the **Data type in TimesTen** field for those columns. Select the desired data type for the column in the child cache table.

For updatable cache tables, you can specify a default column value under the **Default Value** field for each cached column in the child cache table that defines a default value.

## Specifying the table attributes

In the **Table Attributes** tab within the **Tables** tab of the **Create cache group** dialog, select whether to create a range index or a hash index on the child cache table's primary key columns in a similar fashion as was specified for the root table. You can define a different column or set of columns as the primary key for the cache table than the columns of the corresponding Oracle table in a similar fashion as was specified for the root table. You can also specify an optional **WHERE** clause for the cache table in a similar fashion as was specified for the root table. A **WHERE** clause is only allowed for table definitions in a read-only cache group or a user managed cache group that has automatic refresh defined.

**Figure 7–15 Specifying attributes for child table**

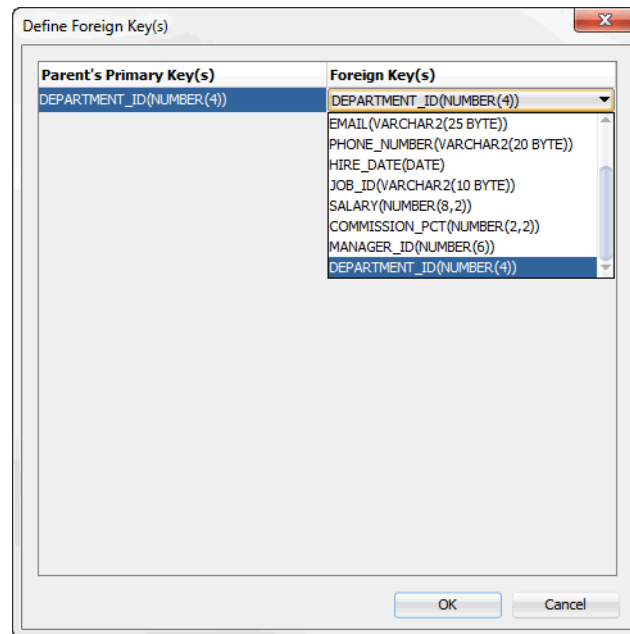


If you would like the foreign key columns of a child table to have cascade delete enabled such that when rows containing referenced key values are deleted from its parent table, rows in the child table with dependent foreign key values are automatically deleted, then select **Cascade delete foreign key**.

By default, the foreign key columns of a child cache table are identical to the foreign key columns of the corresponding Oracle table. You can define a different column or set of columns as the foreign key for the cache table.

In the **Manual table management** section of the **Table Attributes** tab within the **Tables** tab of the **Create cache group** dialog, select **Redefine foreign key**. See [Figure 7–15, "Specifying attributes for child table"](#).

In the **Define Foreign Key(s)** dialog, the columns in the child table whose data types are compatible with the primary key columns of the parent table are shown in the **Foreign Key(s)** drop-down select list. Select the columns for the foreign key of the cache table. Click **OK** to define the alternate foreign key for the cache table.

**Figure 7–16 Defining foreign keys**

For more information about the hierarchy and relationship of cache tables in a multiple-table cache group, see "Multiple-table cache group" in the *Oracle In-Memory Database Cache User's Guide*.

If the child cache table is in a user managed cache group that has selected **Varies from table to table** in the **Table attributes** section within the **Cache Group Attributes** tab of the **Create cache group** dialog, then you must specify the table attribute for the cache table in a similar fashion as was specified for the root table. If the cache table is updatable, then specify whether updates on the table are to be propagated or not propagated to the corresponding Oracle table.

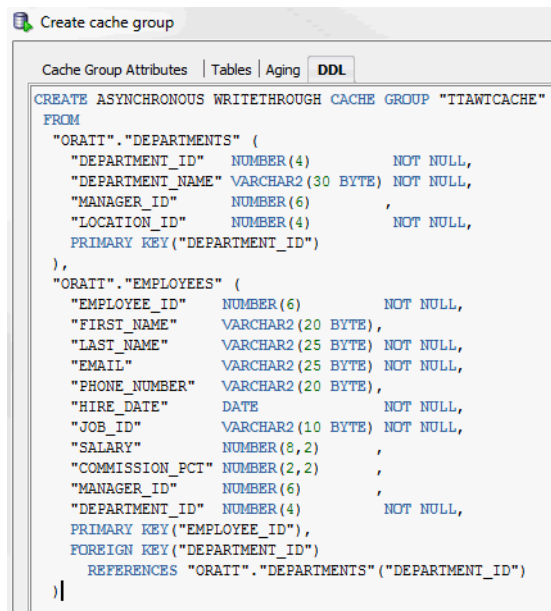
### Completing the definitions of the child tables

Define the columns to cache, the type of index to create on the primary key columns, the cascade delete setting for the foreign key columns, and the table attributes including an optional WHERE clause for any remaining child tables.

### Verifying the cache group definition

In the **DDL** tab of the **Create cache group** dialog, you can view the `CREATE CACHE GROUP` statement used to create the cache group and its cache tables.

Click **Apply** to create the cache group and its cache tables.

**Figure 7–17 Viewing the cache group definition**

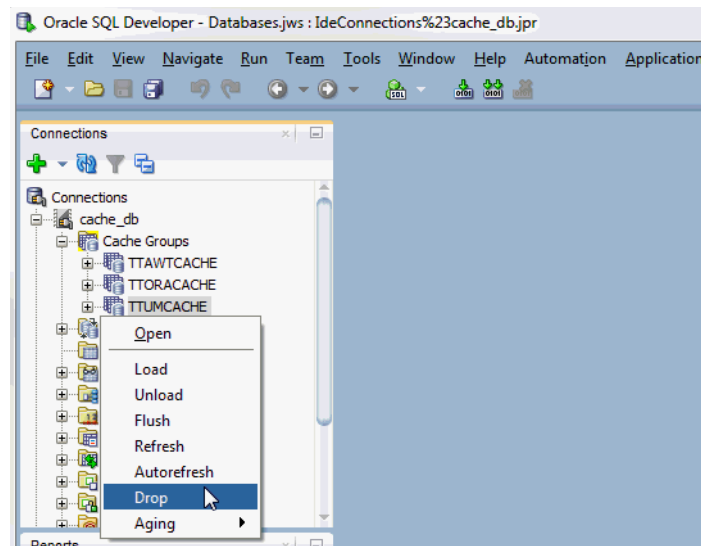
If you created an AWT cache group, then you must start the replication agent for committed updates on its cache tables to be asynchronously propagated to the corresponding Oracle tables. For information about how to start the replication agent, see ["Starting the replication agent"](#) on page 10-3.

If you created a global cache group, then you must attach the TimesTen database to the cache grid that it is associated with before you can perform operations on the cache group or on its cache tables. This task must be performed either outside of SQL Developer using a command-line interface such as the `ttIsql` utility, or within SQL Developer using **SQL Worksheet**. See "Attaching a TimesTen database to a cache grid" in the *Oracle In-Memory Database Cache User's Guide* for more information.

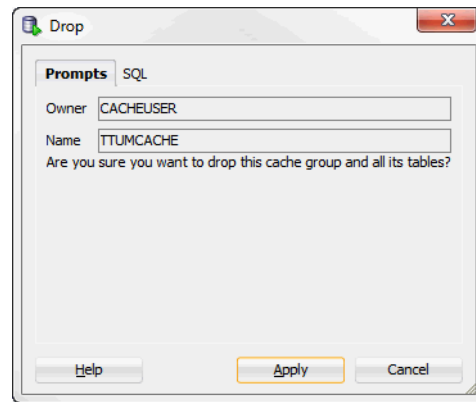
## Dropping a cache group

When you drop a cache group, its cache tables are also dropped.

Under the **Cache Groups** node within the node of the connection name for the TimesTen database, right-click the name of the cache group to drop and select **Drop**.

**Figure 7–18 Drop cache group**

Click **Apply** to drop the cache group.

**Figure 7–19 Drop cache group confirmation**

For more information about dropping a cache group, see "Dropping a cache group" in the *Oracle In-Memory Database Cache User's Guide*.





---

## Cache Group Operations

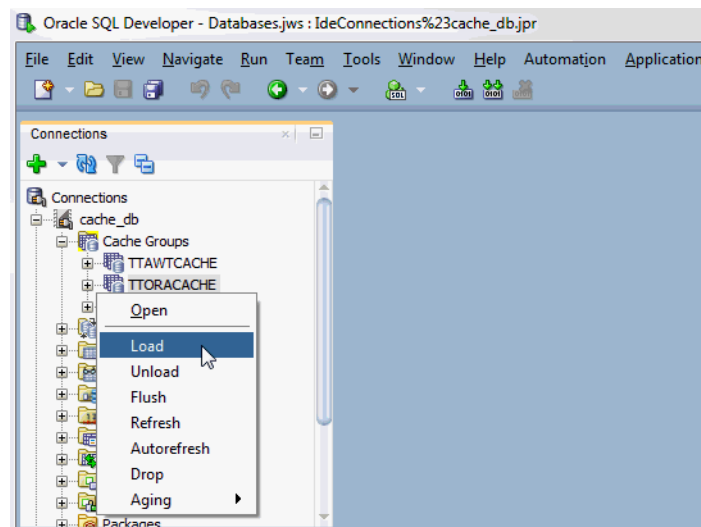
This chapter focuses on the operations you can perform on your cache group. Topics include:

- [Loading or refreshing a cache group](#)
- [Unloading a cache group](#)
- [Flushing a cache group](#)

### Loading or refreshing a cache group

Click the + to the left of the **Cache Groups** node to view the list of cache groups. Next right-click the name of the cache group to load or refresh. Select **Load** to load the cache group or **Refresh** to refresh the cache group.

**Figure 8–1** Cache group operations



The cache agent must be running to load or refresh a cache group. For information about how to start the cache agent, see ["Start and stop the cache agent"](#) on page 6-2.

For a read-only cache group or a user managed cache group that has automatic refresh defined, you must set the automatic refresh state to Paused before you can load or refresh the cache group. For information about changing the automatic refresh state of a cache group, see ["Altering the automatic refresh attributes of a cache group"](#) on page 9-4.

In the **Prompts** tab of the **Load** or **Refresh** dialog, specify a numeric value in the **Commit every n rows** field to indicate the frequency, based on the number of rows that are loaded or refreshed into the cache group, at which a commit is issued during the load or refresh operation. By default, a transaction is committed after every 256 rows are loaded or refreshed into the cache group.

In the **Number of threads to run in parallel** field within the **Prompts** tab of the **Load** or **Refresh** dialog box, specify the number of cache agent process threads to spawn and use for the load or refresh operation. Parallel processing of the load or refresh operation can potentially improve the operation's throughput. The numeric value specified in this field cannot exceed 10 and should not exceed the number of CPUs on the system that can be dedicated to the parallel load or refresh operation. The default value of 1 indicates that the load or refresh operation is processed serially. For more information about parallel loading or refreshing of cache groups, see "Improving the performance of loading or refreshing a large number of cache instances" in the *Oracle In-Memory Database Cache User's Guide*.

You can specify an optional `WHERE` clause for the load or refresh operation in the **Where clause** field within the **Prompts** tab of the **Load** or **Refresh** dialog. You can use a `WHERE` clause for a load or refresh operation on an AWT, SWT or user managed cache group that does not have automatic refresh defined. For more information about specifying a `WHERE` clause in a load or refresh operation, see "Loading and refreshing a cache group" in the *Oracle In-Memory Database Cache User's Guide*.

Click **Apply** to load or refresh the cache group.

For more information about loading or refreshing a cache group, see "Loading and refreshing a cache group" in the *Oracle In-Memory Database Cache User's Guide*.

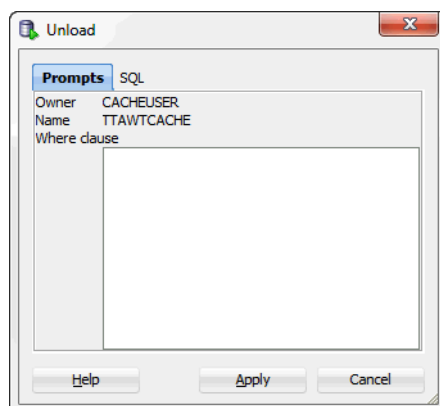
## Unloading a cache group

Click the **+** to the left of the **Cache Groups** node to view the list of cache groups. Next right-click the name of the cache group to unload. Select **Unload** to unload the cache group. See [Figure 8–1, "Cache group operations"](#).

In the **Where clause** field within the **Prompts** tab of the **Unload** dialog, specify an optional `WHERE` clause for the unload operation. You can use a `WHERE` clause for an unload operation on an AWT, SWT or user managed cache group that does not have automatic refresh defined.

Click **Apply** to unload the cache group.

**Figure 8–2 Unload/refresh with optional where clause**



For more information about unloading a cache group, see "Unloading a cache group" in the *Oracle In-Memory Database Cache User's Guide*.

## Flushing a cache group

Only user managed cache groups containing cache tables that are not all read-only, or not all updatable where updates on the tables are propagated to the corresponding Oracle tables can be flushed.

Click the **+** to the left of the **Cache Groups** node to view the list of cache groups. Next right-click the name of the cache group that you to flush. Select **Flush** to flush the cache group. See [Figure 8–1, "Cache group operations"](#).

In the **Where clause** field within the **Prompts** tab of the **Flush** dialog, specify an optional WHERE clause for the flush operation.

Click **Apply** to flush the cache group. See [Figure 8–2, "Unload/refresh with optional where clause"](#).

For more information about flushing a cache group, see "Flushing a user managed cache group" in the *Oracle In-Memory Database Cache User's Guide*.



---

## Cache Group and Cache Table Characteristics

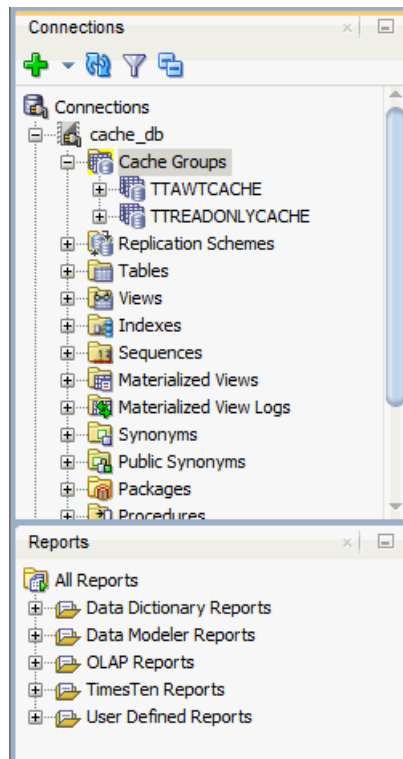
This chapter focuses on viewing cache table characteristics. After you create a cache group, you can view the characteristics of the cache group including the type of cache group, the automatic refresh attributes and the aging policy (if any). You can also alter the automatic refresh attributes.

Topics include:

- [Viewing the characteristics of a cache group](#) including:
  - [Viewing the cache tables](#)
  - [Viewing the cache group definition](#)
  - [Viewing the aging attributes](#)
- [Altering the automatic refresh attributes of a cache group](#)

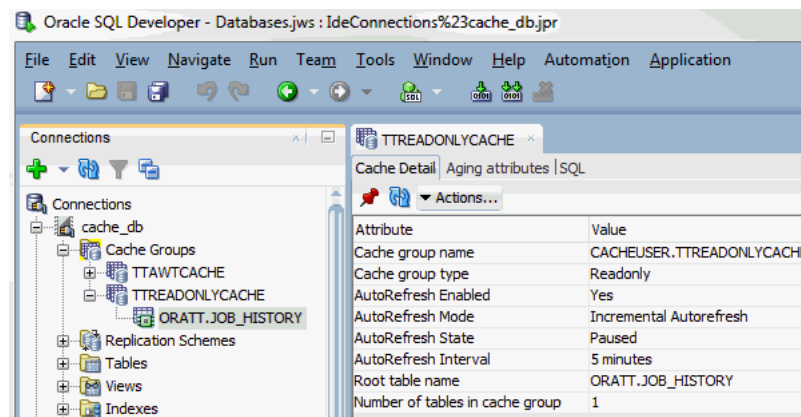
### Viewing the characteristics of a cache group

Click the + to the left of the **Cache Groups** node to view the list of cache groups. Then click the name of the cache group.

**Figure 9–1 List of cache groups**

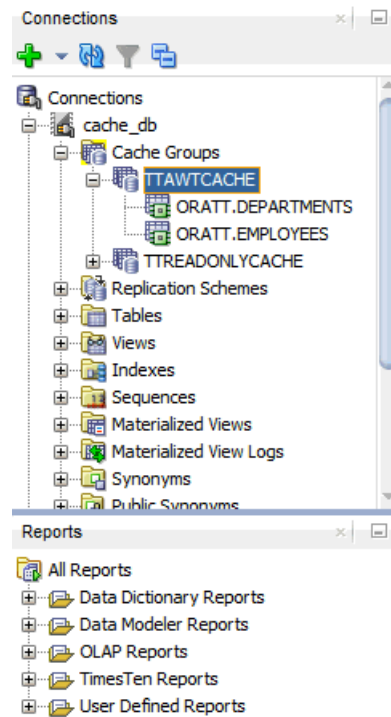
The **Cache Detail** tab shows information such as the cache group type, the name of the root table, and the number of tables in the cache group.

For read-only cache groups and user managed cache groups that have automatic refresh defined, the automatic refresh mode, interval and state is displayed.

**Figure 9–2 Viewing the cache group in detail**

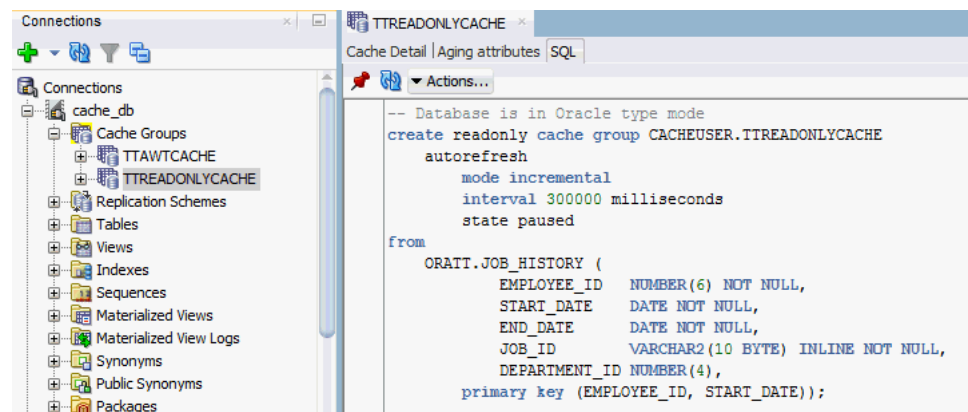
## Viewing the cache tables

To view the list of cache tables in a cache group, right-click the **Cache Groups** node, click the + to the left of the node that contains the name of the cache group.

**Figure 9–3 Viewing cache group tables**

## Viewing the cache group definition

To view the `CREATE CACHE GROUP` statement, select the **cache group** tab and then select the **SQL** tab.

**Figure 9–4 Viewing the cache group definition**

If you are unable to find the cache group, see ["Locating a TimesTen database object"](#) on page 2-1 for information.

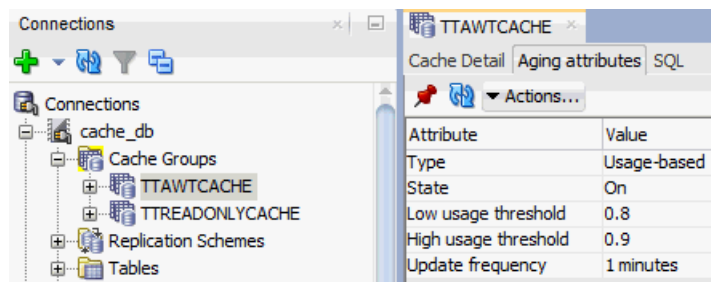
## Viewing the aging attributes

The **Aging attributes** tab, located within the cache group tab, shows information such as the aging policy type, aging cycle, and aging state for cache groups that have an aging policy defined.

For cache groups that have an LRU aging policy, the TimesTen database memory usage thresholds are displayed. For information on LRU aging and LRU thresholds, see ["Specifying an aging policy"](#) on page 7-8.

For cache groups that have a time-based aging policy, the name of the non-nullable DATE or TIMESTAMP column used to store the timestamp value (indicating when each row was added or most recently updated) and the length of time that non-updated data is not deleted is displayed.

**Figure 9–5 Viewing the aging attributes**



## Altering the automatic refresh attributes of a cache group

You can alter the automatic refresh attributes of a read-only cache group or a user managed cache group that has automatic refresh defined.

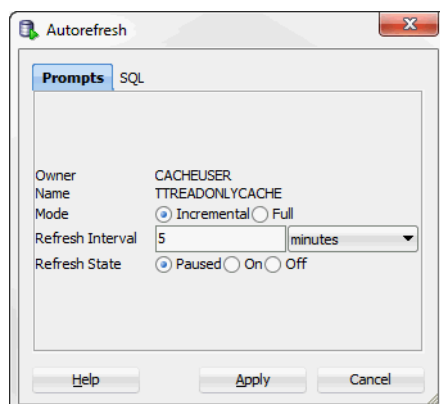
Under the **Cache Groups** node, right-click the name of the cache group and select **Autorefresh**. See [Figure 8–1, "Cache group operations"](#).

In the **Prompts** tab of the **Autorefresh** dialog, you can change any of the following automatic refresh attributes:

- To change the automatic refresh mode, in the **Mode** field, choose either **Incremental** or **Full**.
- To change the automatic refresh interval, in the **Refresh Interval** field, specify a numeric interval and units of time in minutes, seconds or milliseconds.
- To change the automatic refresh state, in the **Refresh State** field, choose **Paused**, **On** or **Off**.

Click **Apply**.

**Figure 9–6 Viewing the autorefresh cache options**





For more information about automatic refresh of cache groups, see "AUTOREFRESH cache group attribute" in the *Oracle In-Memory Database Cache User's Guide*.



---

## Interfacing with TimesTen Replication

A replication scheme defines the set of objects that are replicated between two or more TimesTen databases when committed updates are made on those objects. Replication can be configured at the table or database level in a unidirectional or bidirectional fashion with asynchronous or synchronous transmit.

This chapter focuses on viewing the replication scheme including subscriber databases and `STORE` attributes. The chapter also illustrates how to start the replication agent.

Topics include:

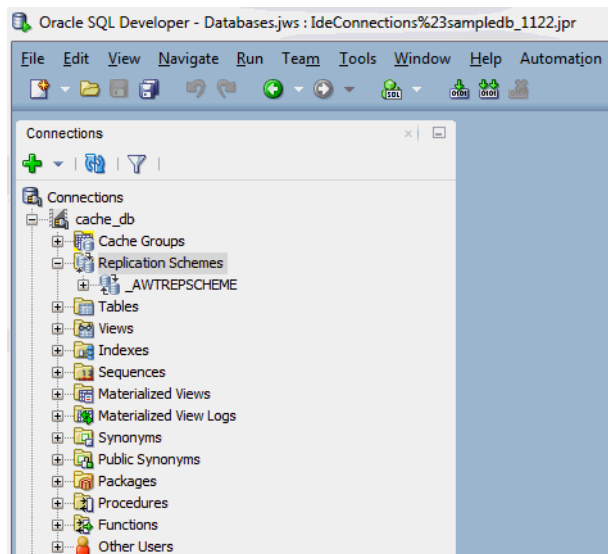
- [Viewing the replication scheme](#) including
  - [Viewing the details of the TimesTen subscriber databases](#)
  - [Viewing the store attributes of the replicated TimesTen databases](#)
- [Starting the replication agent](#)

### Viewing the replication scheme

To create a replication scheme, you must manually compose and then execute a `CREATE REPLICATION` or `CREATE ACTIVE STANDBY PAIR` statement either outside of SQL Developer using a command-line interface such as the `ttIsql` utility, or within SQL Developer using the **SQL Worksheet**. After you have created a replication scheme, you can view the characteristics of the scheme itself such the database name and host name of the participating TimesTen databases, and its replicated objects.

A replication scheme is automatically created when you create an AWT cache group. This replication scheme is owned by the `TTREP` user.

Click the **+** to the left of the **Replication Schemes** node to view the list of replication schemes. Click the name of the replication scheme to view.

**Figure 10–1 Viewing the replication scheme**

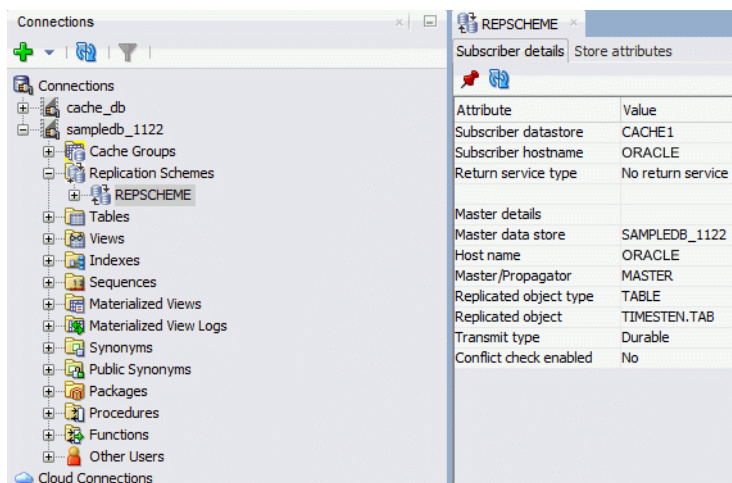
The name of the replication scheme for an active standby pair is `_ACTIVESTANDBY` and it can be found under the **Replication Schemes** node of any existing database user.

The name of the replication scheme that is automatically created to propagate updates on cache tables in AWT cache groups to the corresponding Oracle tables is `_AWTREPScheme`.

If you are unable to find the replication scheme that you are looking for, see ["Locating a TimesTen database object"](#) on page 2-1.

## Viewing the details of the TimesTen subscriber databases

The **Subscriber details** tab, located within the replication scheme tab, displays information such as the database name and host name of the master and subscriber TimesTen databases, the type and database name of the replicated objects, and the return service type.

**Figure 10–2 Viewing the subscriber**

## Viewing the store attributes of the replicated TimesTen databases

The **Store attributes** tab, located within the replication scheme tab, displays information such as the TimesTen database name, host name, replication agent port number, replication response timeout period, and transaction log fail threshold for all databases participating in the replication scheme.

For replication schemes that have a return service, the return service failure policy, acknowledgment timeout period, and re-enable policy is displayed.

**Figure 10–3** Viewing the store attributes

Attribute	Value
Store name	CACHE1
Host name	ORACLE
Replication port number	0
Replication peer timeout	120 seconds
Replication failure threshold	0 accumulated log files
Compress traffic	No
Return service failure policy	No policy
Return service failure threshold	0
Return service state when replication is stopped	Off
Durable commits on return receipt failure	Off
Resume return service latency	0 seconds
Default commit behavior for RETURN TWOSAFE transactions	No policy
Default timeout value for RETURN TWOSAFE transactions	10 seconds
Store name	SAMPLEDB_1122
Host name	ORACLE
Replication port number	0
Replication peer timeout	120 seconds
Replication failure threshold	0 accumulated log files
Compress traffic	No
Return service failure policy	No policy
Return service failure threshold	0
Return service state when replication is stopped	Off
Durable commits on return receipt failure	Off
Resume return service latency	0 seconds
Default commit behavior for RETURN TWOSAFE transactions	No policy
Default timeout value for RETURN TWOSAFE transactions	10 seconds

For more information about the creating a replication scheme, see "Defining Replication Schemes" in the *Oracle TimesTen In-Memory Database Replication Guide*.

## Starting the replication agent

The replication agent process transmits committed updates on replicated tables in a master TimesTen database to the corresponding replicated tables in a subscriber TimesTen database. The replication agent is also responsible for asynchronously propagating committed updates on cache tables in AWT cache groups to the corresponding Oracle tables. After you create a replication scheme or an AWT cache group, you must start the replication agent if it is not running.

To start the replication agent, right-click the node of the connection name for the TimesTen database and select **Start/Stop Replication Agent**. See [Figure 1–7, "Viewing a TimesTen connection"](#).

The **Starting/stopping the replication agent** dialog appears. In the **Prompts** tab, select **Start agent**. Click **Apply**.

If there are no replication schemes or AWT cache groups in your TimesTen connection, the replication agent is not started.

To stop a running replication agent process, right-click the node of the connection name for the TimesTen database and select **Start/Stop Replication Agent**. See [Figure 1–7, "Viewing a TimesTen connection"](#).

The **Starting/stopping the replication agent** dialog appears. In the **Prompts** tab, select **Stop agent**. Click **Apply**.

For more information about the replication agent process, see "Starting and stopping the replication agents" in the *Oracle TimesTen In-Memory Database Replication Guide* or "Managing the replication agent" in the *Oracle In-Memory Database Cache User's Guide*.

---

## Working with TimesTen Performance Metrics

---

TimesTen collects performance metrics from TimesTen system tables, system views, and built-in procedures. SQL Developer allows you to capture snapshots and generate reports that compare two snapshots. A snapshot is a collection of performance metrics.

---

**Note:** SQL Developer uses functions from the `TT_STATS` PL/SQL package to capture snapshots and generate reports that compare two snapshots. For more information about the `TT_STATS` PL/SQL package, see "TT\_STATS" in the *Oracle TimesTen In-Memory Database PL/SQL Packages Reference*.

---

Before attempting to use the TimesTen Performance Metrics:

- Make sure you are using Oracle TimesTen In-Memory Database Release 11.2.2.5 (or later).
- Make sure that the TimesTen connection user has execution privileges on the `TT_STATS` PL/SQL package.

This chapter provides information about working with TimesTen performance metrics.

Topics include:

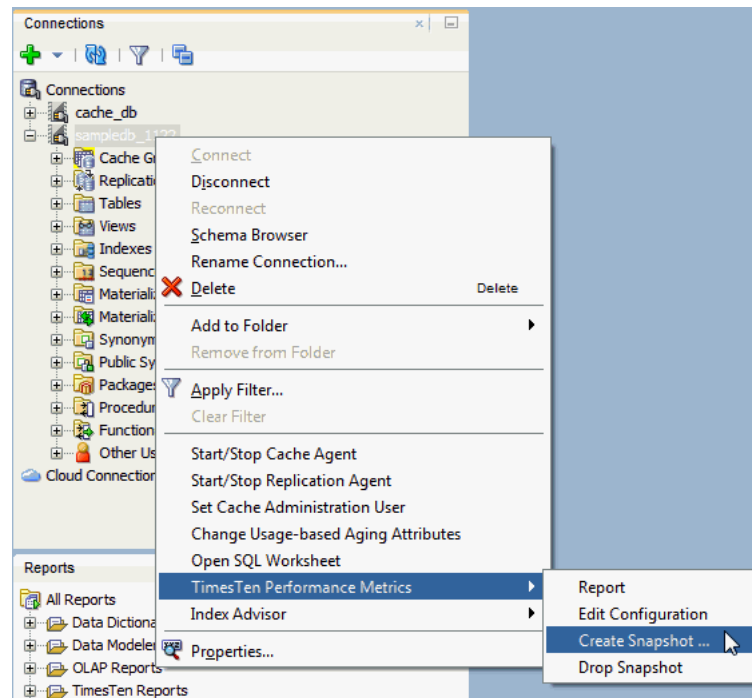
- [Creating snapshots](#)
- [Dropping snapshots](#)
- [Editing the TimesTen performance metrics configuration](#)
- [Creating a TimesTen performance metrics report](#)

### Creating snapshots

A snapshot is a collection of performance metrics from a database. You can create a report comparing two snapshots to analyze the performance of your database.

To create a new snapshot of a database, make sure you are on the main SQL Developer page and that you have expanded the TimesTen connection in the connections navigator.

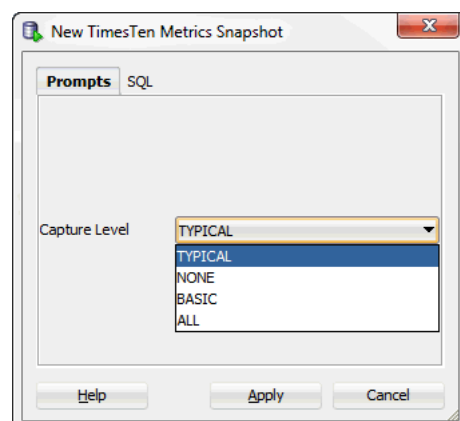
1. Right-click the name of the database and select **TimesTen Performance Metrics**, then select **Create Snapshot**.

**Figure 11–1 Creating a snapshot**

The New TimesTen Metrics Snapshot dialog displays.

2. In the **Prompts** tab of the **New TimesTen Metrics Snapshot** dialog, select the desired capture level from the **Capture Level** drop-down list:
  - **TYPICAL**: For a typical set of metrics. This level is appropriate for most purposes.
  - **NONE**: For metrics outside of SYS.SYSTEMSTATS only.
  - **BASIC**: For a minimal basic set of metrics.
  - **ALL**: For all available metrics.

Make sure to use the same capture level for any two snapshots to be used in a report. An error dialog displays if you attempt to generate a report that compares two snapshots with different capture levels.

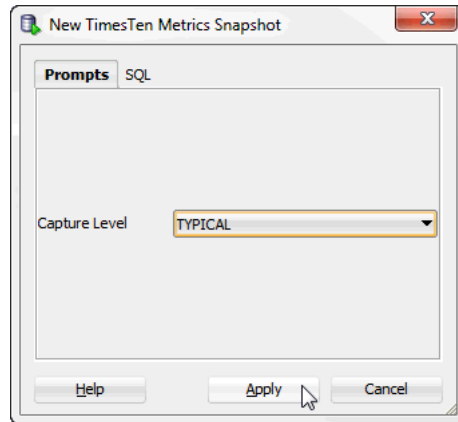
**Figure 11–2 New TimesTen Metrics Snapshot dialog**



You are now ready to create a new snapshot.

3. Click **Apply** to create the snapshot.

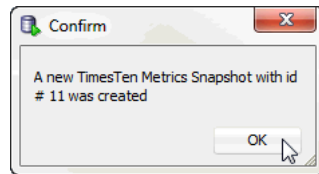
**Figure 11–3 Creating new snapshot**



A confirmation dialog shows the ID number of the snapshot. TimesTen generates the snapshot ID.

4. Click **OK**.

**Figure 11–4 Snapshot ID confirmation**



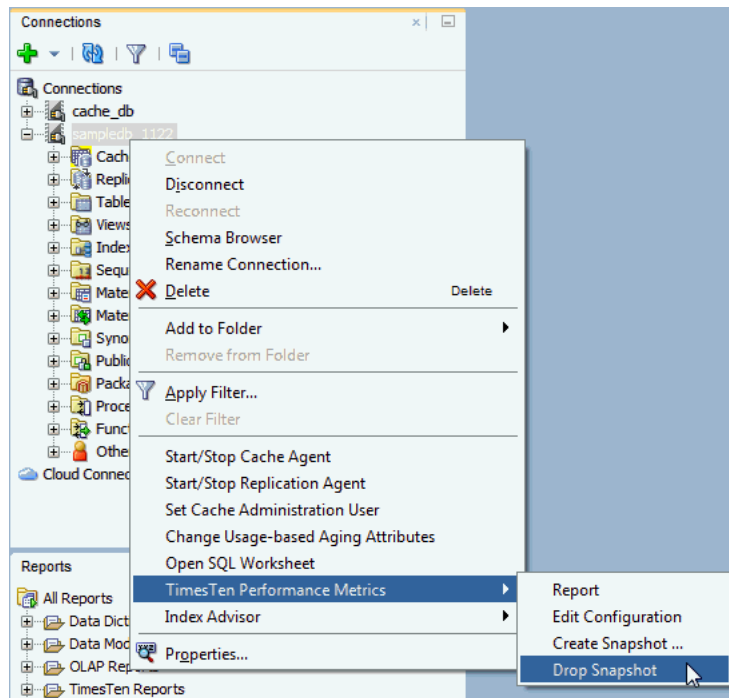
TimesTen creates the snapshot with the specified snapshot ID. The snapshot ID is useful for creating TimesTen metrics reports.

## Dropping snapshots

Consider dropping unneeded snapshots to free up permanent space. If you need to save more snapshots of your database, consider editing the TimesTen performance metrics configuration. See ["Editing the TimesTen performance metrics configuration"](#) on page 11-6.

To drop a new snapshot from a database, make sure you are on the main SQL Developer page.

1. Right-click the name of the database and select **TimesTen Performance Metrics**, then select **Drop Snapshot**.

**Figure 11–5 Dropping a snapshot**

2. By default, the **Prompts** tab of the **Drop TimesTen Metrics Snapshot(s)** dialog is selected. In the **Selector** column, select the snapshots you want to drop.

You can also click on the column header to sort the table based on the column. A description of each column follows:

- **Selector**

The check box you select to determine the snapshots that you want to drop.

- **Snapshot**

The snapshot ID of the snapshot.

- **Timestamp**

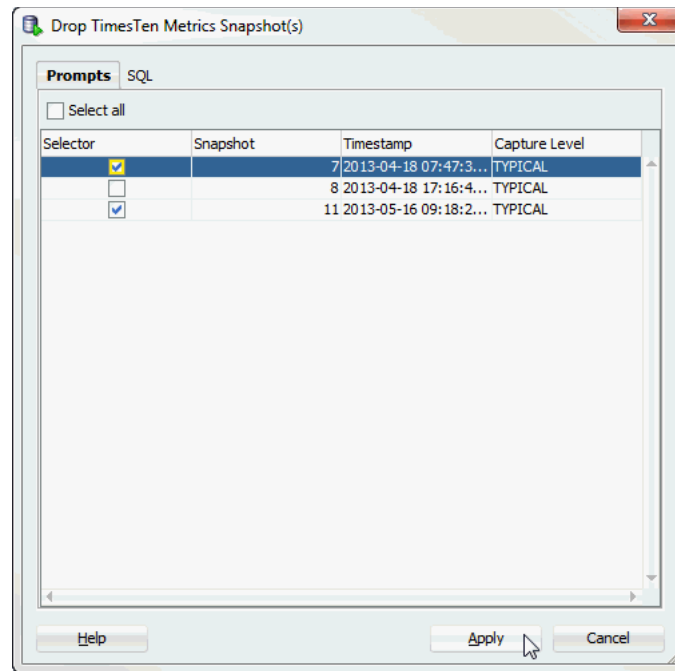
The timestamp of when the snapshot was captured.

- **Capture Level**

The capture level of the snapshot. This value can be TYPICAL, NONE, BASIC, or ALL.

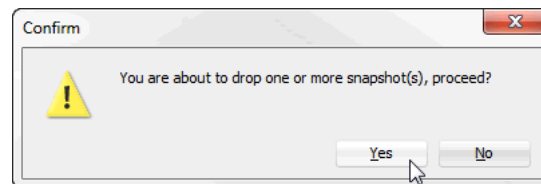
To select all snapshots, click the **Select all** check box.

3. Once you have selected the snapshots that you want to drop, click the **Apply** button.

**Figure 11–6** *Selecting snapshots that you want to drop*

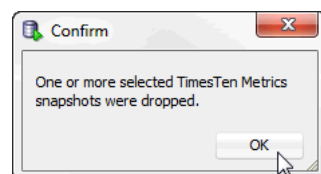
A Confirm dialog displays asking you if you want to proceed with dropping the selected snapshots.

4. Click **Yes**.

**Figure 11–7** *Confirmation dialog for dropping snapshots*

A confirmation dialog displays a message indicating that one or more TimesTen Metrics snapshots were dropped.

5. Click **OK**.

**Figure 11–8** *Dropped snapshots*

You successfully dropped the TimesTen metrics snapshots.

## Editing the TimesTen performance metrics configuration

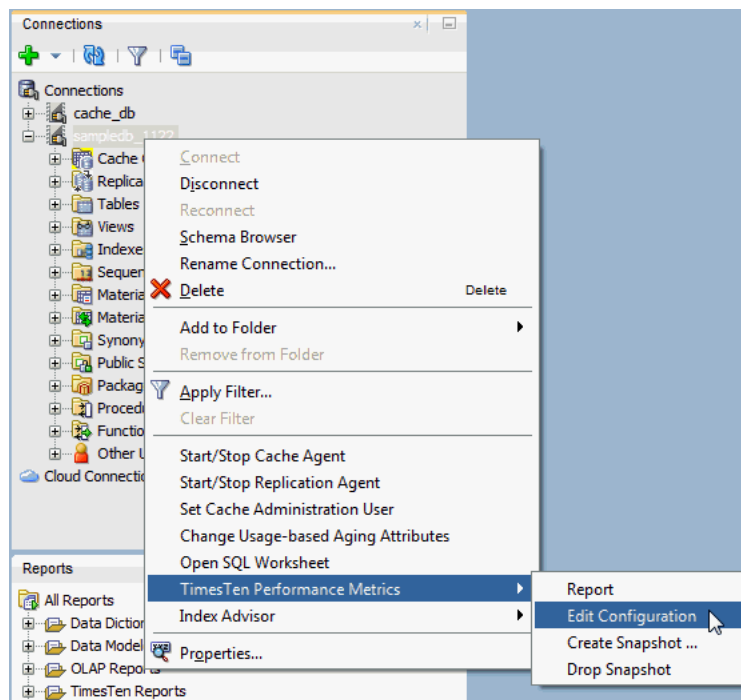
TimesTen performance metrics configuration allows you to modify the maximum snapshot count and the maximum snapshot retention size. These configuration settings are useful in preventing your database from running out of permanent space and in allowing you to save more snapshots.

Once the maximum snapshot count or the maximum snapshot retention size is reached, TimesTen removes the oldest snapshots to free up permanent space. By default, the maximum snapshot count is 256 and the maximum snapshot retention size is 256 MB.

To edit the TimesTen performance metrics configuration for a database, make sure you are on the main SQL Developer page.

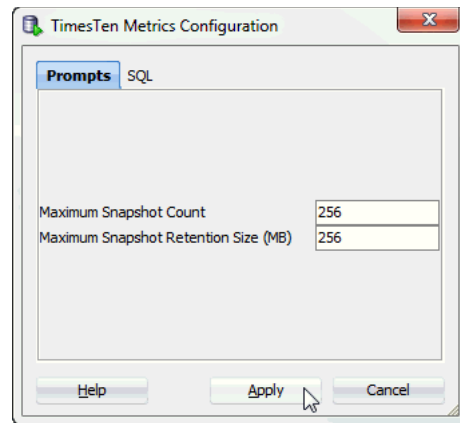
1. Right-click the name of the database and select **TimesTen Performance Metrics**, then select **Edit Configuration**.

**Figure 11–9** Editing the TimesTen performance metrics configuration



The **TimesTen Metrics Configuration** dialog displays and you see the current configuration for the TimesTen metrics.

2. The **Prompts** tab of the **TimesTen Metrics Configuration** dialog is selected. In the **Maximum Snapshot Count** field, specify a numeric value to indicate the maximum number of snapshots. By default, the maximum number of snapshots is 256.
3. In the **Maximum Snapshot Retention Size (MB)** field, specify the maximum total size of all stored snapshots, in megabytes. By default, the maximum snapshot retention size is 256 MB.
4. Click **Apply**.

**Figure 11–10 TimesTen metrics configuration dialog**

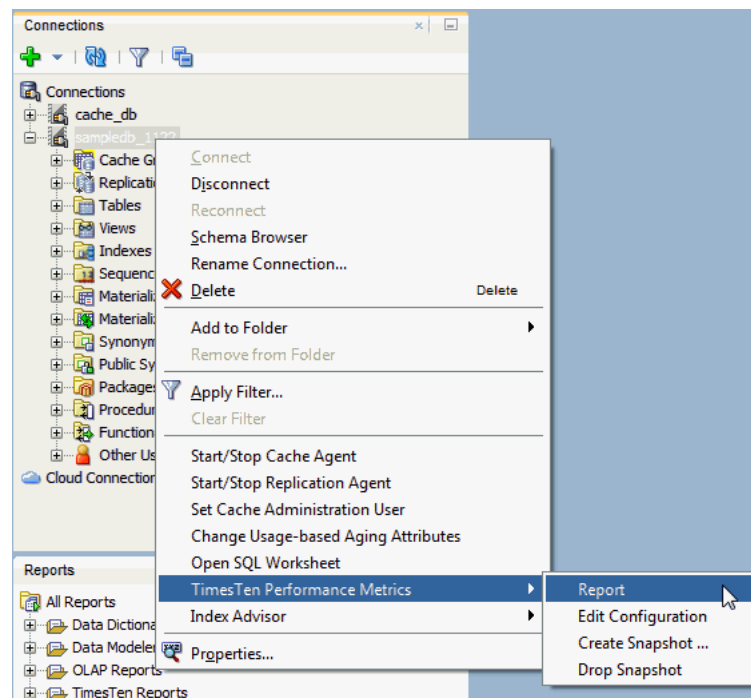
You successfully applied the new TimesTen Metrics configuration settings.

## Creating a TimesTen performance metrics report

TimesTen performance metrics reports use the data from two specified snapshots of TimesTen metrics to produce a report in HTML format. TimesTen metrics reports include information for each metric. For more information about the various metrics displayed in the report, see "Report examples" in the *Oracle TimesTen In-Memory Database Reference*.

To generate a new report for a database, make sure you are on the main SQL Developer page.

1. Right-click the name of the database and select **TimesTen Performance Metrics**, then select **Report**.

**Figure 11–11 Creating a report**

The **Create TimesTen Metrics Report** dialog displays.

2. By default, the **Report generator** tab of the **Create TimesTen Metrics Report** dialog is selected. In the **Selector** column, select two snapshots to compare.

You can also click on the column header to sort the table based on the column. A description of each column follows:

- **Selector**

The check box that determines the snapshots that are compared in the metrics report. Make sure to only select two snapshots with the same capture level. An error dialog displays if you attempt to generate a report that compares more than two snapshots.

- **Snapshot**

The snapshot ID of the snapshot.

- **Timestamp**

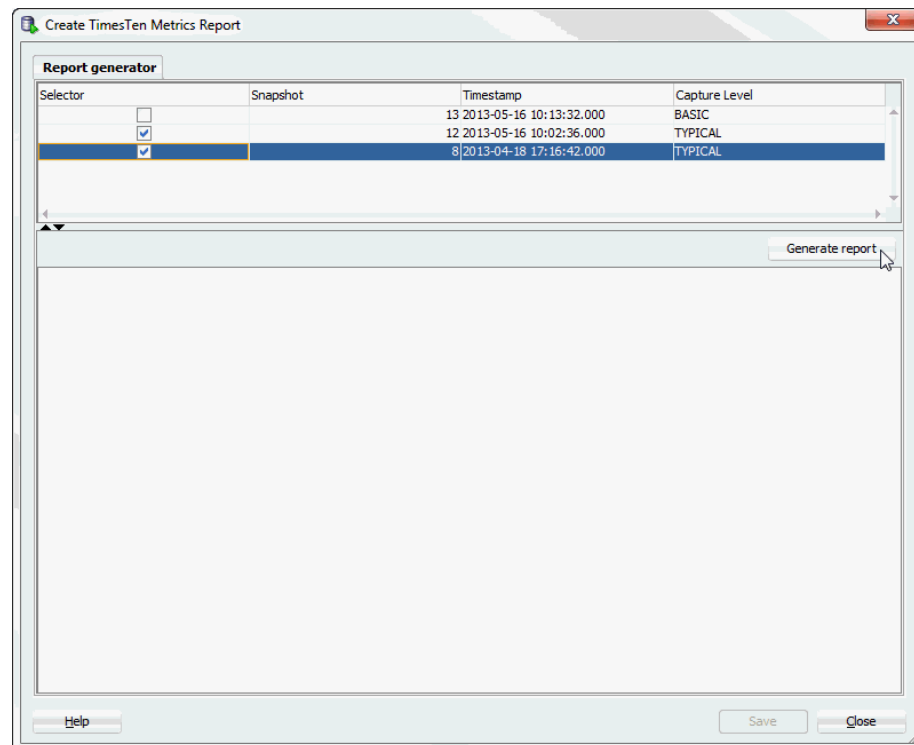
The timestamp of when the snapshot was captured.

- **Capture Level**

The capture level of the snapshot. This value can be TYPICAL, NONE, BASIC, or ALL.

3. Once you have selected two snapshots with the same capture level, click the **Generate report** button. An error dialog displays if you attempt to generate a report that compares snapshots with different capture levels.

**Figure 11–12 Create TimesTen metrics report**



TimesTen generates your TimesTen metrics report. You can view the generated TimesTen metrics report in the pane located below the **Generate report** button.

For more information about the various metrics displayed in the report, see "Report examples" in the *Oracle TimesTen In-Memory Database Reference*.

4. If you do not want to save the report, click **Close**. To save the report, see ["Saving a report"](#) on page 11-9.

## Saving a report

To save the report for future reference, follow these steps:

1. Click **Save**.

The Save dialog displays.

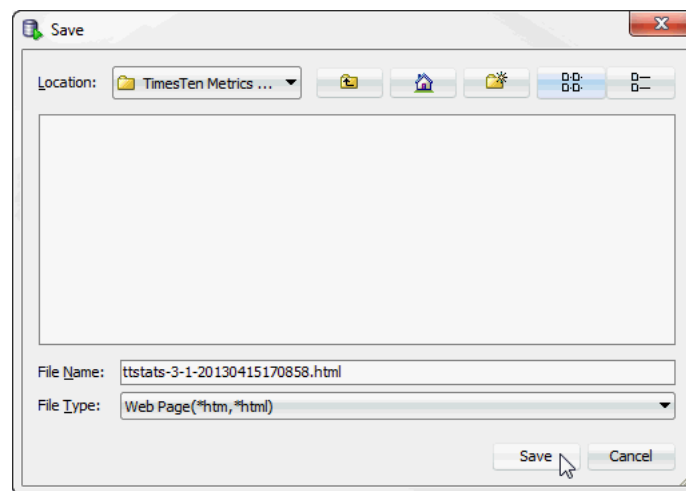
2. Select the directory where you want to save the TimesTen metrics report.

3. In the **File Name** field, define the name of your TimesTen metrics report.

The default filename of the TimesTen metrics report is `ttstats-snapshot_id_1-snapshot_id_2-YYYYMMDDHHMISS.html`, where `snapshot_id_1` and `snapshot_id_2` are the snapshot IDs of the two snapshots you are comparing in your report. `YYYYMMDDHHMISS` is a timestamp of when the report was generated, where `YYYY` is the year, `MM` the month, `DD` the day, `HH` the hour, `MI` the minutes, and `SS` the seconds.

4. Click **Save**.

**Figure 11-13 Save TimesTen metrics report**



SQL Developer saves your report.





---

## TimesTen Reports

You can generate reports about your database and its objects. You can also create your own user-defined reports.

This chapter provides information about generating both TimesTen supplied reports and user-defined reports.

### Generating TimesTen Reports

To generate or create reports, click the **Reports** tab on the uppermost left corner of the SQL Developer navigator. If this tab is not visible, select **View** and then **Reports**. For more information about how SQL Developer generates and creates reports, see the *Oracle SQL Developer User's Guide*.

You can either choose to create your own user-defined reports or you can choose to generate TimesTen supplied reports.

To create your own user-defined reports, click the + sign to the left of the **User Defined Reports** node.

To view the list of TimesTen supplied reports, click the + sign to the left of the **TimesTen Reports** node.

TimesTen reports are grouped into the categories:

- [About Your Database](#)
- [All Objects](#)
- [Cache Groups](#)
- [Database Administration](#)
- [PLSQL](#)
- [Security](#)
- [Table](#)

### About Your Database

There are two reports you can generate:

- **TimesTen configuration**
- **Version Banner**

Choose **TimesTen configuration** to generate a report on the attribute settings for your current database connection. For detailed information on these attributes, see the

built-in procedure "ttConfiguration" in the *Oracle TimesTen In-Memory Database Reference*.

Choose **Version Banner** to generate a report on your database release. For more information, see the built-in procedure "ttVersion" in the *Oracle TimesTen In-Memory Database Reference*.

## All Objects

The **All Objects** reports list information about database objects accessible to the user connected to the TimesTen database. There are three types of reports:

- **All Objects**
- **Object Count by Type**
- **Public Synonyms**

Choose **All Objects** to generate a report for each database object accessible to the user. This report includes information on the owner of the object, the name of the object, the type of the object (for example, table, view), the date the object was created, and the date when the last DDL operation was performed on it. The date of the last DDL operation is useful in determining changes to the definition of the object.

Choose **Object Count by Type** to generate a report on the count of all database objects by owner and type.

Choose **Public Synonyms** to generate a report on the names of public synonyms and underlying object type for the public synonyms.

## Cache Groups

**Cache Groups** reports include information about the cache groups owned by the user associated with the specified connection. The reports include:

- **Cache Group Details**
- **User Cache Group Count by Type**

Choose **Cache Group Details** to view reports for the cache groups:

- **Asynchronous Writethrough Cache Groups**
- **Readonly Cache Groups**
- **Synchronous Writethrough Cache Groups**
- **User Managed Cache Groups**

Choose **User Cache Group Count by Type** to view a report that shows the type of cache group and the count for each type of cache group.

## Database Administration

**Database Administration** reports include information on:

- **All Tables**
- **Users**

Choose **All Tables** to generate reports on the names of tables, columns, constraints, and indexes accessible to the user connected to the TimesTen database.

Choose **Users** to generate a report on all internal and external database users.

## Performance

Performance reports include information on:

- **Statistics Configuration**
- **Statistics Report**
- **Statistics Snapshots**

Choose **Statistics Configuration** to generate a report on the current statistics configuration of the TimesTen database.

Choose **Statistics Report** to generate an HTML report comparing the two most recent statistics snapshots with the same capture level.

Choose **Statistics Snapshots** to generate a report of all the currently available statistics snapshots. This report includes information on the ID of the snapshot, timestamp of when the snapshot was taken, and the capture level of the snapshot.

For more information on the performance metrics, see [Chapter 11, "Working with TimesTen Performance Metrics"](#).

## PLSQL

There are three reports you can generate:

- **Program Unit Arguments**
- **Search Source Code**
- **Unit Line Counts**

These reports are generated for PL/SQL procedures, functions, and packages.

## Security

**Security** reports list information about system and object privileges granted to the user connected to the TimesTen database. **Security** reports also list information about the privileges granted to the PUBLIC role.

Security reports are grouped into two categories:

- **Grants and Privileges**
- **Public Grants**

Choose **Grants and Privileges** reports to view information about the system and object privileges granted to the user connected to the TimesTen database. Choose **Public Grants** to view information about the privileges granted to the PUBLIC role. **Public Grants** reports include information on:

- **All Grants**
- **Counts by Grantor**
- **Grant EXECUTE**
- **Grant SELECT**

## Table

**Table** reports include information on:

- **Columns**
- **Constraints**
- **Indexes**
- **Table Sizes**
- **User Synonyms**
- **User Tables**

Choose the **Table** reports to generate reports on the tables, columns, constraints, and indexes owned by the user connected to the TimesTen database. In addition, choose the **Table** reports to generate a report on private synonyms.

Reports for Table Sizes are displayed if you are using TimesTen Release 11.2.2 or later. If you are not using TimesTen Release 11.2.2 or later, then if you choose **Table Sizes**, a **No Query Found** dialog appears. You will see the error: No query found for version of selected database.

---

---

# Index

## A

---

- aging
  - altering for table, 2-14
  - changing lifetime and cycle, 2-18
  - dropping for table, 2-16
- aging attributes
  - table, 2-9
- aging policy
  - defined, 7-8
  - LRU, 7-8
  - time-based, 7-9
- aging state
  - changing for table, 2-16
- automatic refresh attributes
  - altering, 9-4

## C

---

- cache agent
  - start and stop, 6-2
  - start and stop dialog, 6-3
- cache grid
  - configuration, 6-2
- cache group
  - create, 7-1
  - create dialog, 7-1
  - dropping, 7-14
  - flushing, 8-3
  - loading, 8-1
  - refreshing, 8-1
  - unloading, 8-2
  - user managed, 7-2
  - viewing the characteristics, 9-1
- cache group operations, 8-1
- Cache Groups, 12-2
- cache groups
  - asynchronous writethrough, 7-2
  - dynamic, 7-2
  - explicitly loaded, 7-2
  - global, 7-2
  - local, 7-2
  - read-only, 7-2
  - root table, 7-4
  - selecting the columns to cache, 7-5
  - synchronous writethrough, 7-2

- viewing the aging attributes, 9-3
  - viewing the definition, 9-3
- cache table
  - aging, 7-8
- cache tables
  - viewing, 9-2
- child table
  - defining foreign keys, 7-12
  - selecting columns to cache, 7-11
  - specifying the table attributes, 7-12
- child tables
  - defining, 7-10
- columns of table
  - viewing, 2-8
- Computing table sizes, 2-4
- configuration
  - creating the cache manager user, 6-1
  - creating the timesten user, 6-1
  - granting privileges, 6-1
  - Oracle and TimesTen databases, 6-1
- connect
  - TimesTen database, 1-1
- connection
  - disconnect, 1-7
  - existing, 1-7
  - new, 1-2
- connection information dialog, 1-7

## E

---

- environment variables
  - CLASSPATH, 1-1
  - LD\_LIBRARY\_PATH, 1-1
  - PATH, 1-1
- execution plan, 3-6

## I

---

- index
  - creating for table, 2-10
- indexes
  - viewing, 2-9
- INLINE attribute, 2-3
- installation
  - scripts, 1-1

## **L**

---

LOB data types, 2-2

## **M**

---

memory usage threshold  
    changing for table, 2-17

## **N**

---

new connection dialog, 1-2

## **O**

---

object privileges, 2-18

## **P**

---

primary key attributes  
    changing, 2-13  
primary key constraint  
    creating, 2-12

## **Q**

---

query optimizer, 3-1  
    explain plan, 3-7  
    statistics detail, 3-3

## **R**

---

REF CURSOR

    OUTPUT parameter, 2-6

replication

    interfacing, 10-1

replication agent

    starting and stopping, 10-3

replication scheme

    viewing, 10-1

reports

    About Your Database, 12-1

    All Objects, 12-2

    Database Administration, 12-2

    generating, 12-1

    introduction, 12-1

    Performance, 12-3

    PLSQL, 12-3

    Security, 12-3

    Table, 12-3

root table

    primary key, 7-6

    where clause, 7-7

root table definition

    verifying, 7-10

root table dialog, 7-4

## **S**

---

store attributes

    viewing, 10-3

subscriber database

viewing, 10-2

## **T**

---

table

    viewing the characteristics, 2-8

table and column statistics

    updating, 3-1

table definition

    viewing, 2-9

time-based aging, 7-9

TimesTen database object

    locating, 2-1

ttIsql commands, 2-21