

Oracle® SQL Developer

Oracle TimesTen In-Memory Database Support User's Guide

Release 3.1

E25286-01

January 2012

Copyright © 2009, 2012, 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 RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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.

Contents

Preface	vii
Audience.....	vii
Related documents	vii
Conventions	vii
Documentation Accessibility	viii
Technical support.....	viii
What's New	ix
New features for release of SQL Developer 3.1	ix
1 Connecting to a TimesTen Database	
Setting environment variables for TimesTen and SQL Developer	1-1
Defining a TimesTen database named connection	1-2
Using an existing TimesTen database named connection	1-3
Disconnecting an existing TimesTen database connection.....	1-4
2 Prerequisites to Creating and Accessing a Cache Environment	
Configuring the Oracle and TimesTen databases	2-1
Set the cache administration user name and password	2-1
Configure a cache grid.....	2-2
Start and stop the cache agent.....	2-2
3 Creating Cache Groups	
Creating a cache group	3-1
Defining a cache group and cache table attributes	3-3
Defining the root table.....	3-4
Selecting the columns to cache.....	3-5
Specifying the table attributes.....	3-6
Specifying an aging policy.....	3-8
Verifying the root table definition.....	3-10
Defining the child tables	3-10
Selecting the columns to cache.....	3-11
Specifying the table attributes.....	3-12
Completing the definitions of the child tables.....	3-13

Verifying the cache group definition	3-13
Dropping a cache group	3-14
4 Cache Group Operations	
Loading or refreshing a cache group	4-1
Unloading a cache group	4-2
Flushing a cache group	4-3
5 Cache Group and Cache Table Characteristics	
Viewing the characteristics of a cache group	5-1
Viewing the cache tables	5-2
Viewing the cache group definition	5-3
Viewing the aging attributes	5-3
Altering the automatic refresh attributes of a cache group	5-4
6 Working with TimesTen Objects	
Locating a TimesTen database object	6-1
Specifying and displaying LOB data types	6-2
Specifying the INLINE attribute for columns	6-3
Computing table size information	6-4
Using a REF CURSOR as an OUT parameter	6-6
Viewing the characteristics of a table	6-8
Viewing the columns	6-8
Viewing the indexes	6-9
Viewing the aging attributes	6-9
Viewing the table definition	6-9
Creating an index on a table	6-10
Creating a primary key constraint	6-12
Changing the attributes of primary key	6-13
Altering the aging properties of a table	6-14
Adding an aging policy to a table	6-14
Dropping an aging policy from a table	6-16
Changing the aging state of a table's aging policy	6-16
Changing the memory usage thresholds and LRU aging cycle	6-17
Changing the lifetime and cycle of a table's time-based aging policy	6-18
Granting and revoking object privileges	6-18
Viewing the privileges granted on a database object	6-20
Loading data into tables	6-20
ttIsql commands in SQL worksheet	6-21
7 Interfacing with TimesTen Replication	
Viewing the replication scheme	7-1
Viewing the details of the TimesTen subscriber databases	7-2
Viewing the store attributes of the replicated TimesTen databases	7-3
Starting the replication agent	7-4

8 Interfacing with the TimesTen Query Optimizer

Updating the table and column statistics used by the query optimizer.....	8-1
Viewing the execution plan of a statement	8-4

9 TimesTen Reports

Generating TimesTen Reports	9-1
About Your Database	9-1
All Objects	9-2
Cache Groups	9-2
Database Administration.....	9-2
PLSQL.....	9-3
Security.....	9-3
Table.....	9-3

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/database/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
boldface	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.

Convention	Meaning
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 3.1. There are some features that are only supported in TimesTen Release 11.2.2 or later. If a feature is only supported in TimesTen Release 11.2.2 or later, it will be noted.

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 6-2.
- You can compute table size information. For more information on computing table size information, see "[Computing table size information](#)" on page 6-4.
- 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 9-3.

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 6-3.
- 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 6-6.
- You can load data into tables. For more information, see "[Loading data into tables](#)" on page 6-20.
- There are reports for **Cache Groups**. For more information, see "[Cache Groups](#)" on page 9-2.

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/ttjdbc6.jar` file on Linux systems or the TimesTen `install_dir\lib\ttjdbc6.jar` file on Microsoft Windows systems. SQL Developer 3.0 requires that you have installed Java Development Kit (JDK) 6.

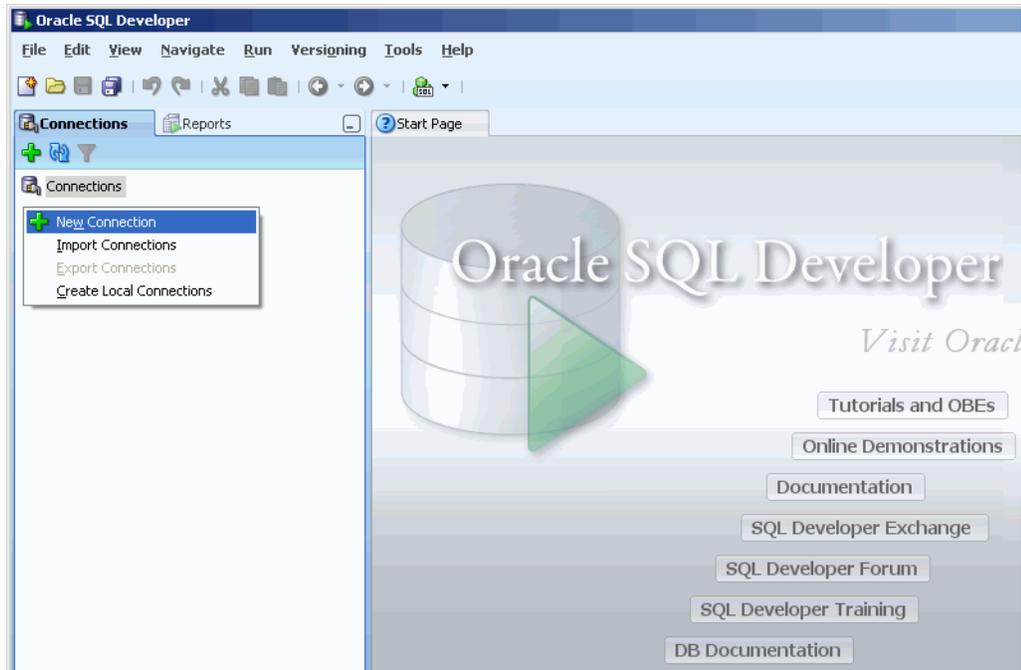
To determine the third party JDBC drivers SQL Developer is using, select **Tools > Preferences**. In the **Preferences** dialog, click the + to the left of the **Database** node. Select **Third Party JDBC Drivers**. View the path name of all third party JDBC drivers including the TimesTen JDBC drivers.

- 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, right-click the **Connections** node and select **New Connection**.

Figure 1–1 *Creating a new connection*



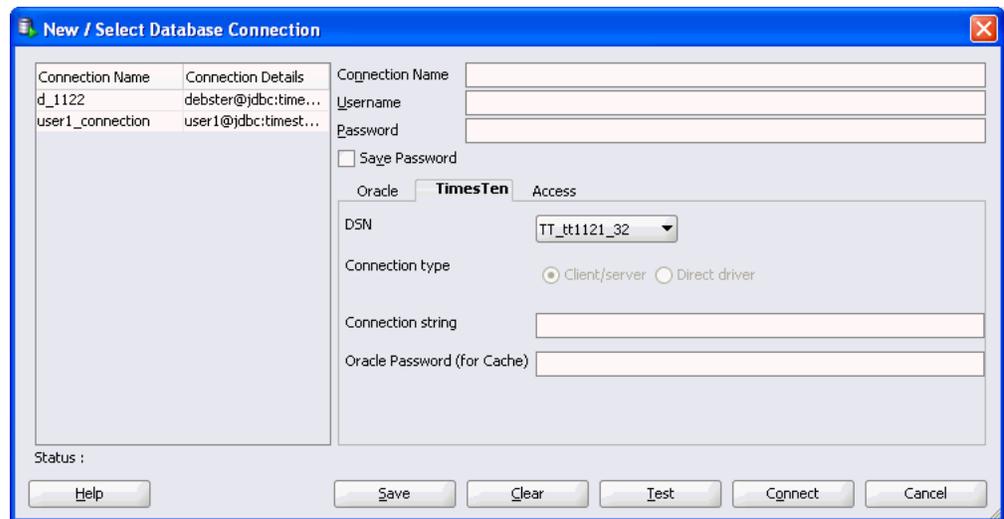
The **New/Select Database Connection** dialog appears. Choose the **TimesTen** tab. You are ready to enter information for the connection.

- In the **Connection Name** field, type the connection name.
- In the **Username** field, type the TimesTen user. If you use the TimesTen database to cache data from an Oracle database, type the TimesTen cache manager user name.
- In the **Password** field, type the password for the TimesTen user.

To save the password, choose **Save Password**. If you save the password, the **Connection Information** dialog does not appear on subsequent connections to the TimesTen database.

- If you use the TimesTen database to cache data from an Oracle database, in the **Oracle Password (for Cache)** field, type the Oracle password for the Oracle user. The Oracle user must be the same name as the TimesTen user you specified in the **Username** field.

Otherwise, leave this field blank.

Figure 1–2 New database connection

For **DSN**, select a predefined DSN that references the TimesTen database. You can select either a Data Manager DSN to establish a direct connection or a client DSN to establish a client/server connection.

If you select **User specified**, then choose either **Client/server** to establish a client/server connection or **Direct driver** to establish a direct connection.

For information about configuring a client/server connection to a TimesTen database, see "Configuring TimesTen Client and Server" in the *Oracle TimesTen In-Memory Database Operations Guide*.

In the **Connection string** field, if you selected **User specified** for the **DSN**, specify a connection string to define the DSN and set attributes for the 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*.

Click **Save** to save the connection name under the **Connections** node so that 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.

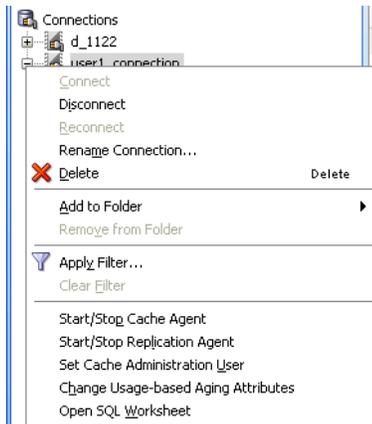
Click **Connect**.

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 1–3 Viewing a TimesTen connection



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

Click **OK**.

Figure 1–4 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 1–3, "Viewing a TimesTen connection"](#).

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.

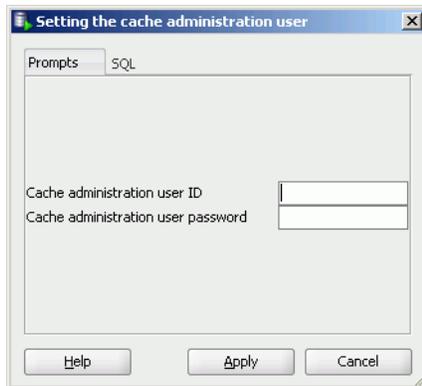
After you create the cache administration user in the Oracle database, you must set the name and password for the user in the TimesTen database.

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 1-3, "Viewing a TimesTen connection"](#).

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

Click **Apply**.

Figure 2-1 *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

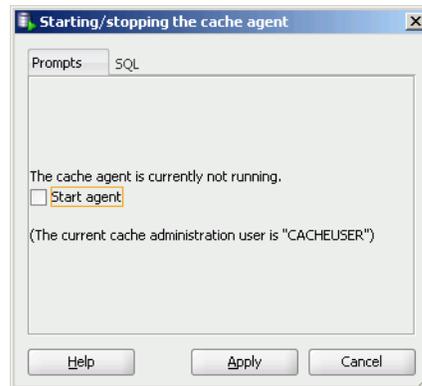
Start and stop the cache agent

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 1-3, "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 2–2 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 2–2, "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.

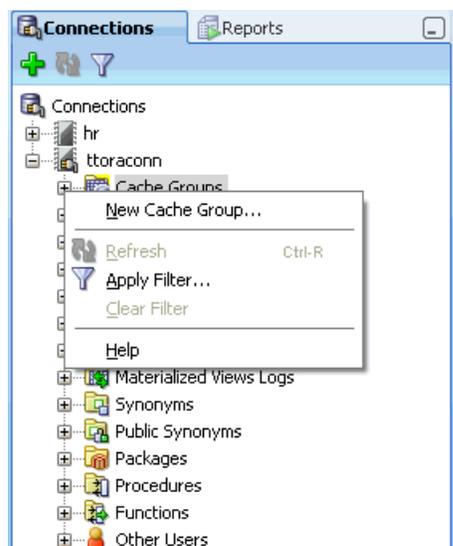
This chapter includes topics:

- [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 3–1 *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*.

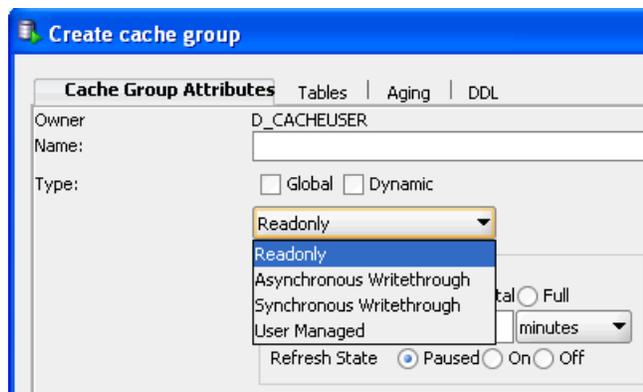
Cache groups are classified as either local or global. If you are creating a global cache group, then select the **Global** in the **Type** section.

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 3–2 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 3–3 Autorefresh attributes

The screenshot shows the 'Cache Group Attributes' dialog box with the 'Autorefresh Attributes' section expanded. The 'Owner' is 'D_CACHEUSER'. The 'Type' is set to 'Readonly'. Under 'Autorefresh Attributes', the 'Mode' is 'Incremental' (selected with a radio button), 'Refresh Interval' is '5 minutes', and 'Refresh State' is 'Paused' (selected with a radio button). Other options include 'Full', 'On', and 'Off' for the Refresh State, and 'Global' and 'Dynamic' for the Type.

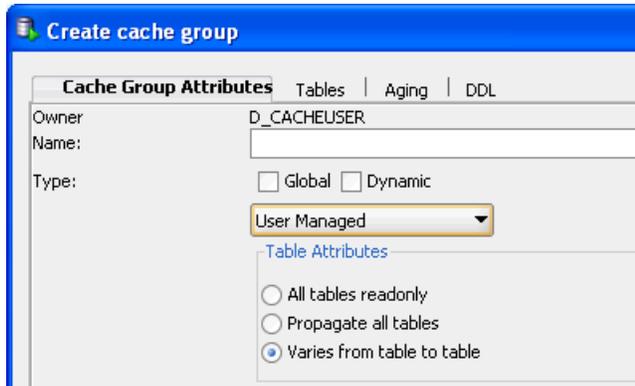
For more information about automatic refresh of cache groups, see "AUTOREFRESH cache group attribute" in the *Oracle In-Memory Database Cache User's Guide*.

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 3–4 User managed cache group attributes

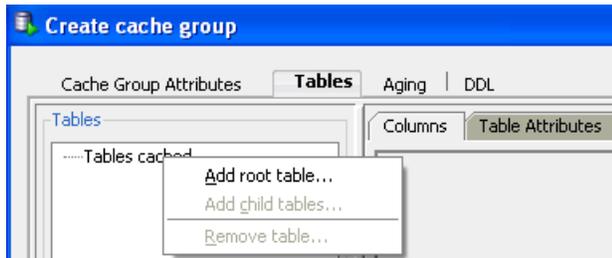


For more information about user managed cache groups, see "User managed cache group" in the *Oracle In-Memory Database Cache User's Guide*.

Defining the root table

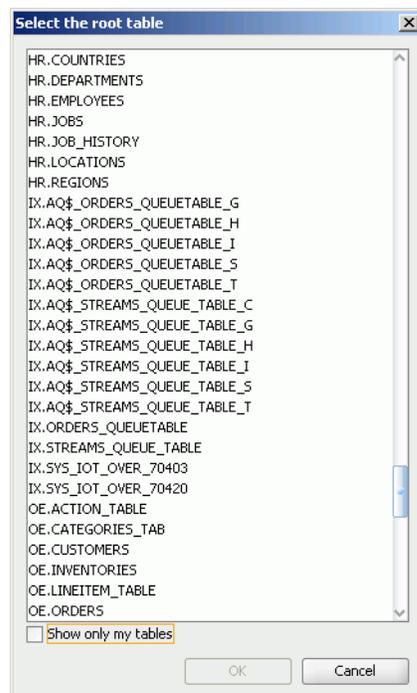
Click the **Tables** tab in the **Create cache group** dialog. Right-click in the **Tables** panel and select **Add root table**.

Figure 3–5 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 3–6 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.

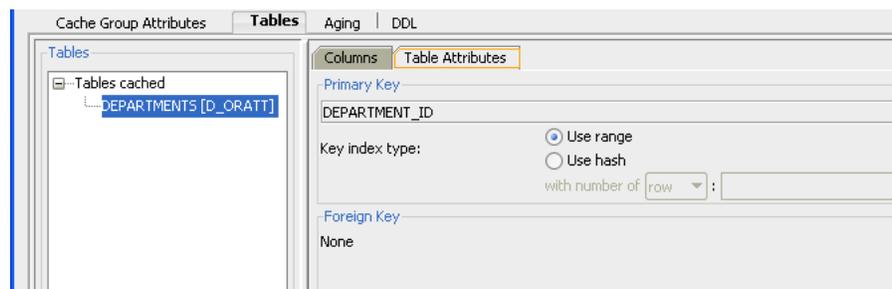
For more information about compatibility between Oracle and TimesTen data types, see "Mappings between Oracle and TimesTen data types" in the *Oracle In-Memory Database Cache User's Guide*.

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 3–7 Primary key attributes



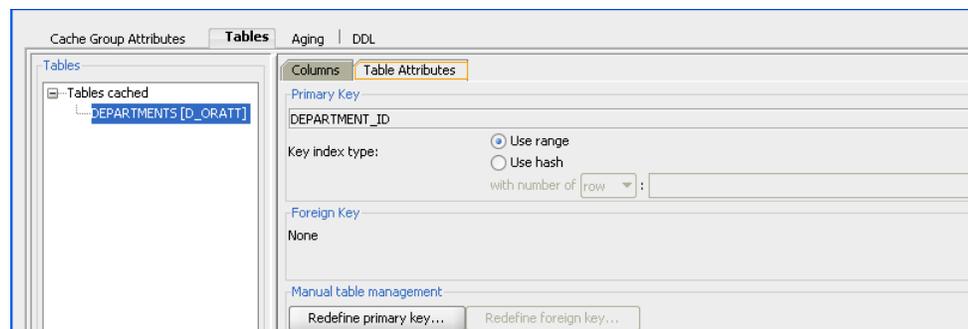
For more information about the index that is automatically created on a cache table's primary key columns, see "CREATE CACHE GROUP" in the *Oracle TimesTen In-Memory Database SQL Reference*.

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.

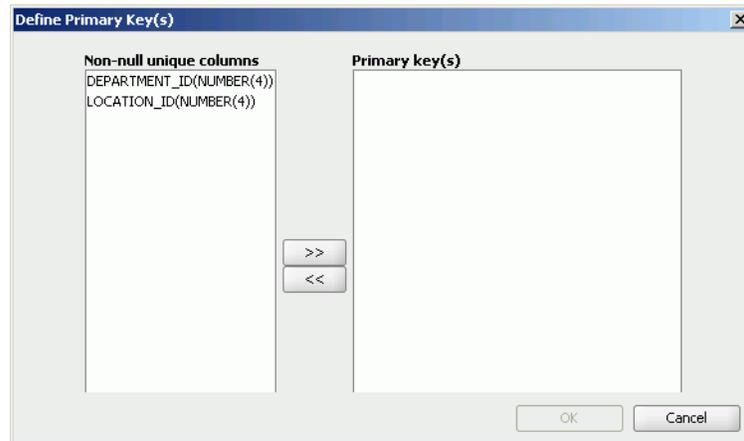
In the **Manual table management** section of the **Table Attributes** tab within the **Tables** tab of the **Create cache group** dialog, click **Redefine primary key**.

Figure 3–8 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.

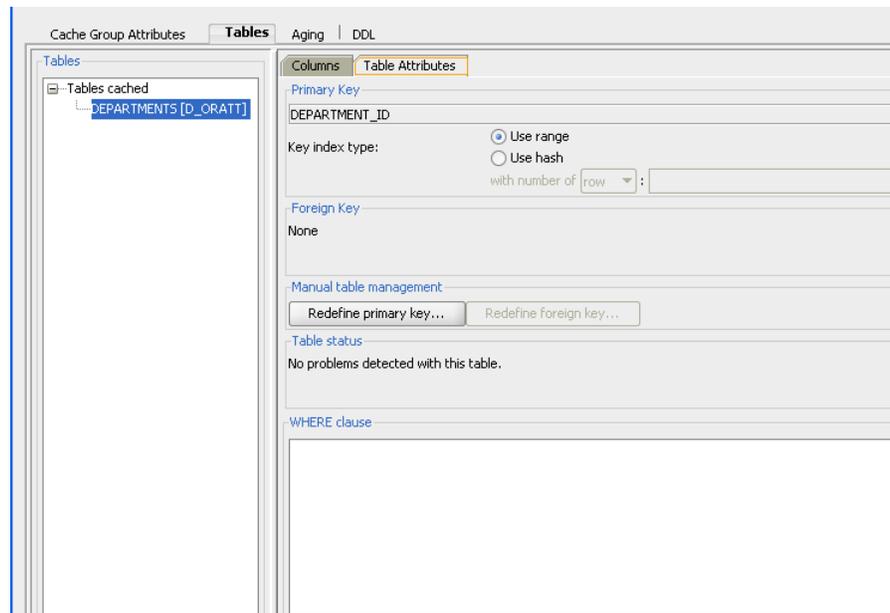
Figure 3–9 Defining a new primary key



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 3–10 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

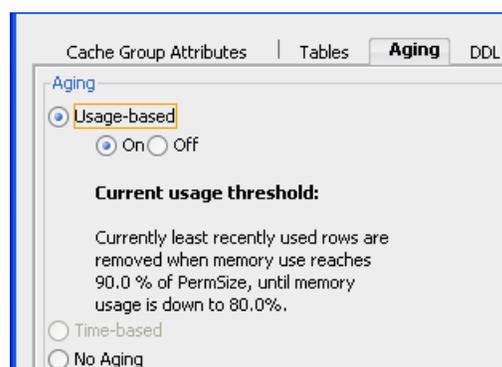
An LRU aging policy can be defined for tables in an explicitly loaded AWT, SWT or user managed cache group that does not have automatic refresh defined. The default

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 6-17.

Figure 3–11 Specifying usage-based aging



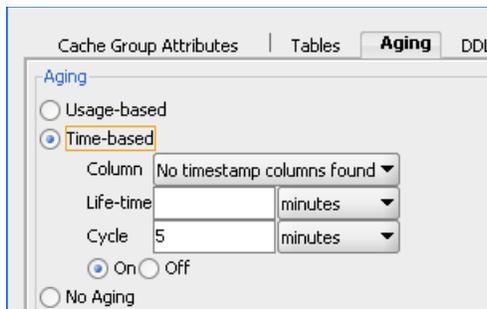
For more information about defining an LRU aging policy on cache tables, see "LRU aging" in the *Oracle In-Memory Database Cache User's Guide*.

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**.

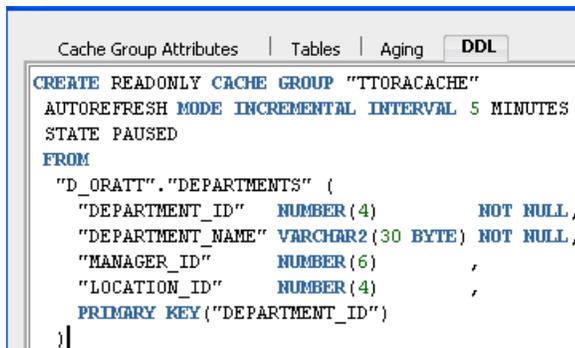
Figure 3–12 Specifying time-based aging

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 3–13 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

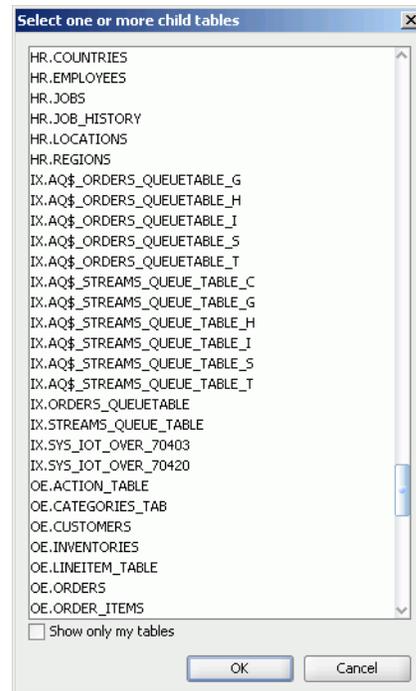
Within the **Tables** tab of the **Create cache group** dialog, right-click in the **Tables** panel and select **Add child tables**.

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 3–14 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.

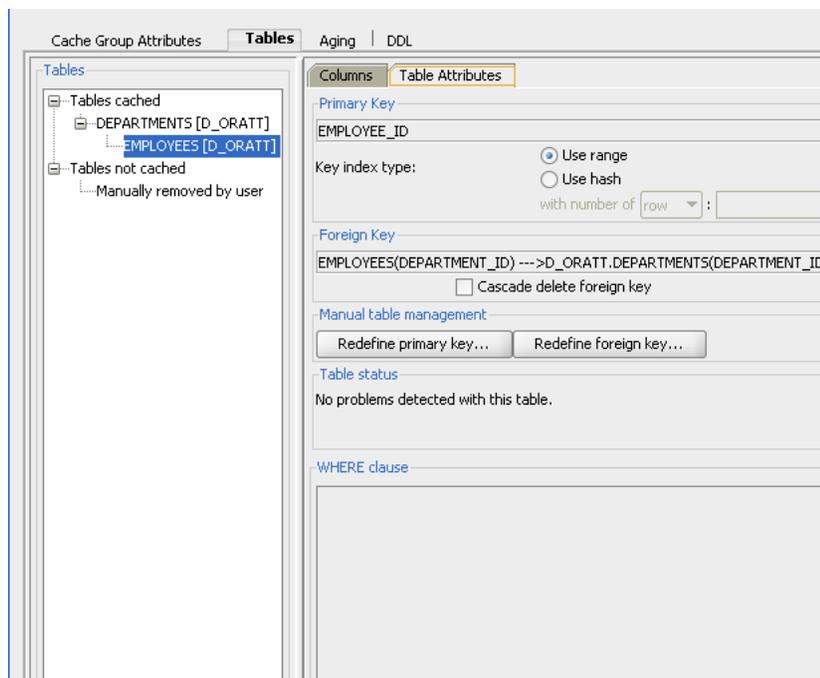
For columns in the Oracle table whose data type can map to multiple TimesTen data types, a drop-down select list appears under the **Data type in TimesTen** field for those columns. Select the desired data type for the column in the child cache table.

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 3–15 Specifying attributes for child table

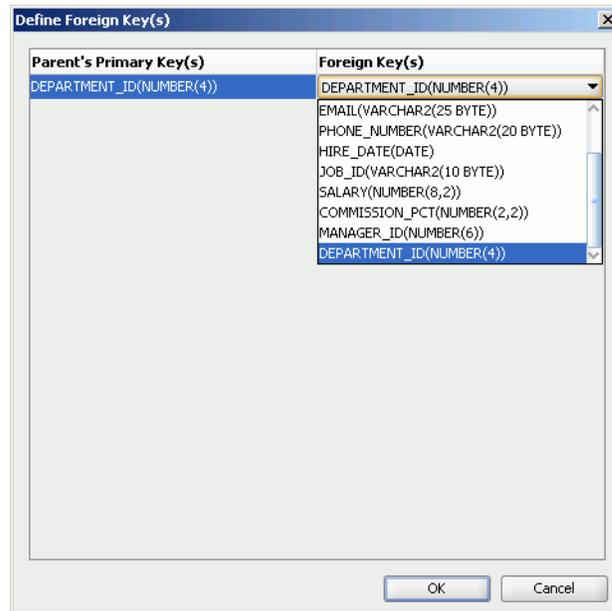


If you would like the foreign key columns of a child table to have cascade delete enabled such that when rows containing referenced key values are deleted from its parent table, rows in the child table with dependent foreign key values are automatically deleted, then select **Cascade delete foreign key**.

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 3–15, "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 3–16 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 3–17 Viewing the cache group definition

```

CREATE ASYNCHRONOUS WRITETHROUGH CACHE GROUP "TTAWTCACHE"
FROM
"HR"."DEPARTMENTS" (
  "DEPARTMENT_ID"    NUMBER(4)          NOT NULL,
  "DEPARTMENT_NAME" VARCHAR2(30 BYTE) NOT NULL,
  "MANAGER_ID"       NUMBER(6)          ,
  "LOCATION_ID"        NUMBER(4)          ,
  PRIMARY KEY("DEPARTMENT_ID")
),
"HR"."EMPLOYEES" (
  "EMPLOYEE_ID"      NUMBER(6)          NOT NULL,
  "FIRST_NAME"       VARCHAR2(20 BYTE),
  "LAST_NAME"        VARCHAR2(25 BYTE) NOT NULL,
  "EMAIL"            VARCHAR2(25 BYTE) NOT NULL,
  "PHONE_NUMBER"     VARCHAR2(20 BYTE),
  "HIRE_DATE"        DATE              NOT NULL,
  "JOB_ID"           VARCHAR2(10 BYTE) NOT NULL,
  "SALARY"           NUMBER(8,2)        ,
  "COMMISSION_PCT"   NUMBER(2,2)        ,
  "MANAGER_ID"       NUMBER(6)          ,
  "DEPARTMENT_ID"   NUMBER(4)          NOT NULL,
  PRIMARY KEY("EMPLOYEE_ID"),
  FOREIGN KEY("DEPARTMENT_ID")
    REFERENCES "HR"."DEPARTMENTS"("DEPARTMENT_ID")
)

```

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 7-4.

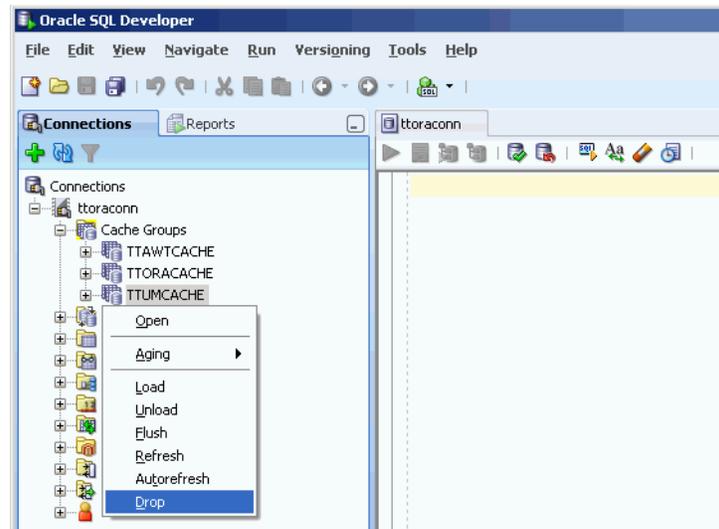
If you created a global cache group, then you must attach the TimesTen database to the cache grid that it is associated with before you can perform operations on the cache group or on its cache tables. This task must be performed either outside of SQL Developer using a command-line interface such as the `ttIsq1` 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 3–18 Drop cache group



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

Figure 3–19 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

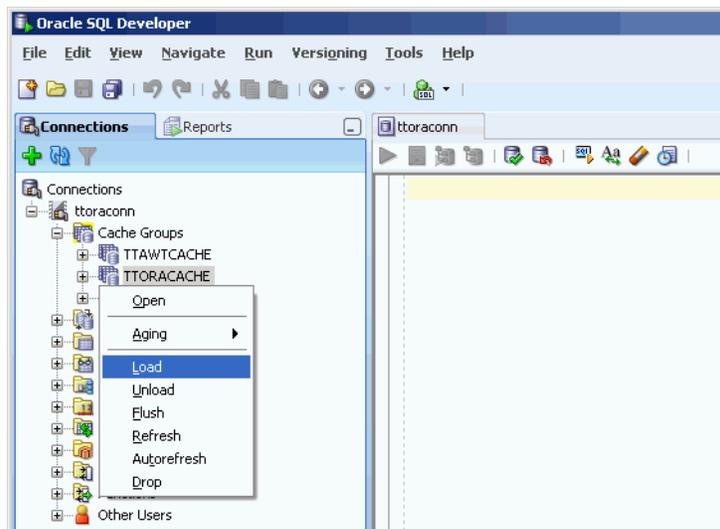
This chapter focuses on the operations you can perform on your cache group. Topics include:

- [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 4-1 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 2-2.

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 5-4.

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*.

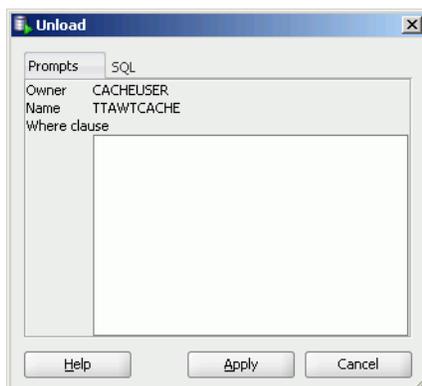
Unloading a cache group

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 4–1, "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 4–2 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 4-1, "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 4-2, "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

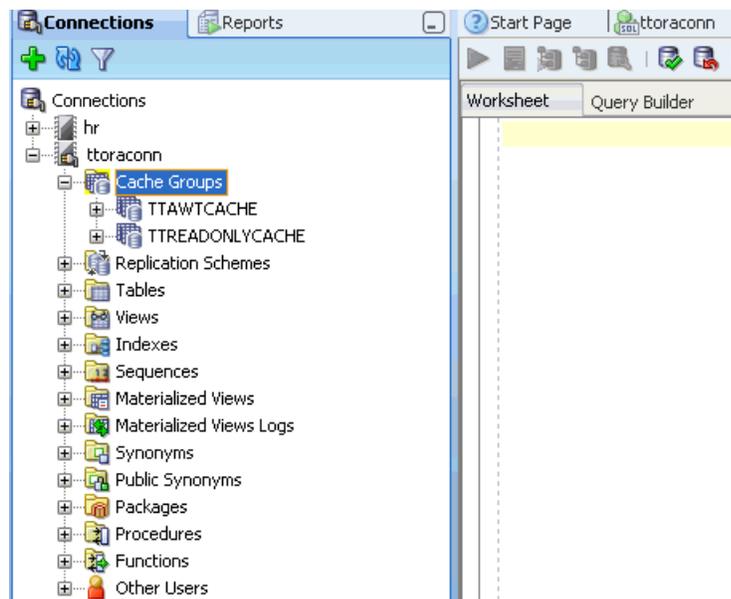
This chapter focuses on viewing cache table characteristics. After you create a cache group, you can view the characteristics of the cache group including the type of cache group, the automatic refresh attributes and the aging policy (if any). You can also alter the automatic refresh attributes.

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](#)

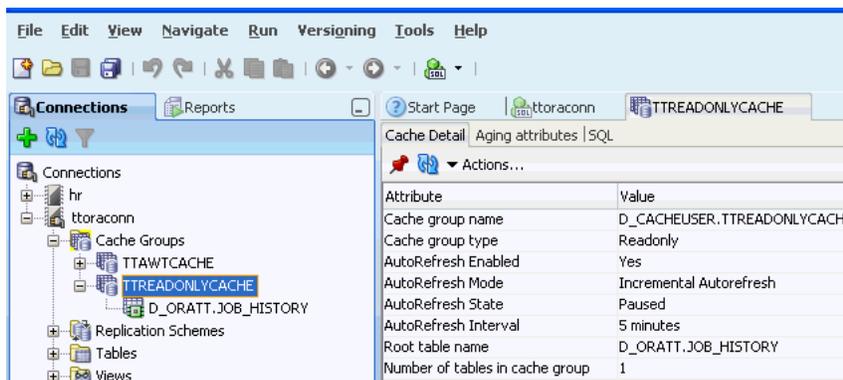
Viewing the characteristics of a cache group

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 5–1 List of cache groups

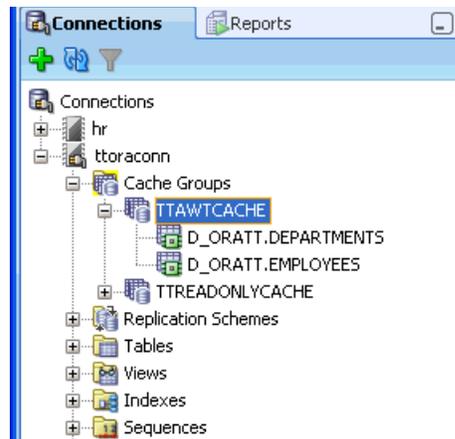
The **Cache Detail** tab shows information such as the cache group type, the name of the root table, and the number of tables in the cache group.

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 5–2 Viewing the cache group in detail

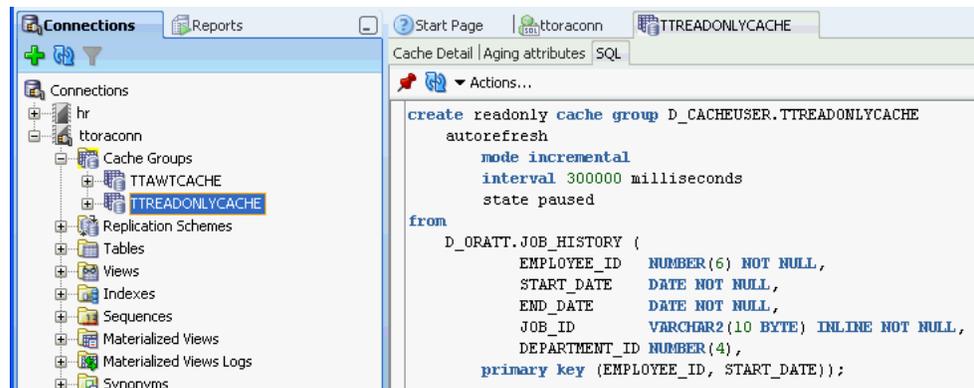
Viewing the cache tables

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 5-3 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 5-4 Viewing the cache group definition

If you are unable to find the cache group, see ["Locating a TimesTen database object"](#) on page 6-1 for information.

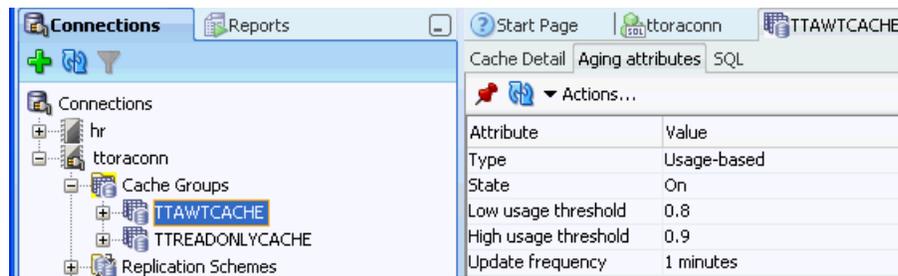
Viewing the aging attributes

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 3-8.

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 5–5 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.

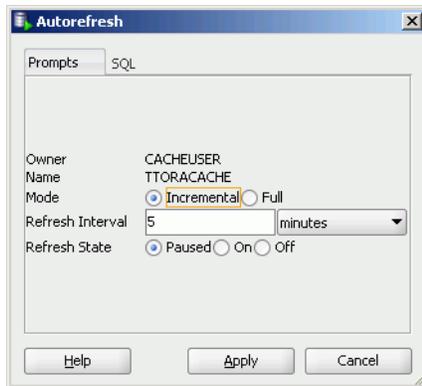
Under the **Cache Groups** node, right-click the name of the cache group and select **Autorefresh**. See [Figure 4–1, "Cache group operations"](#).

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 5–6 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*.

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.

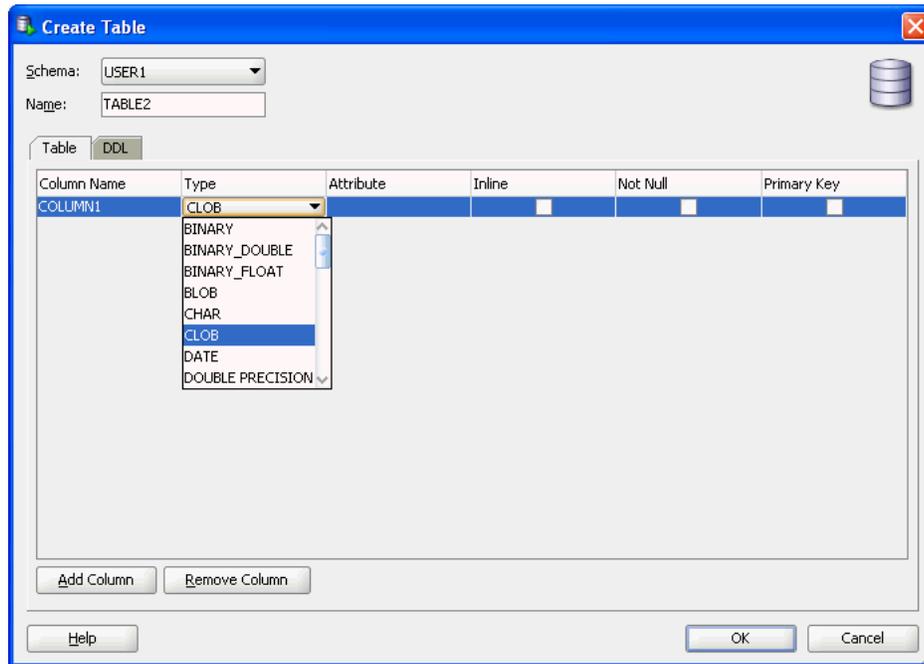
For example, to find a table, click the + to the left of the user's **Tables** node to view the list of tables owned by that user.

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.

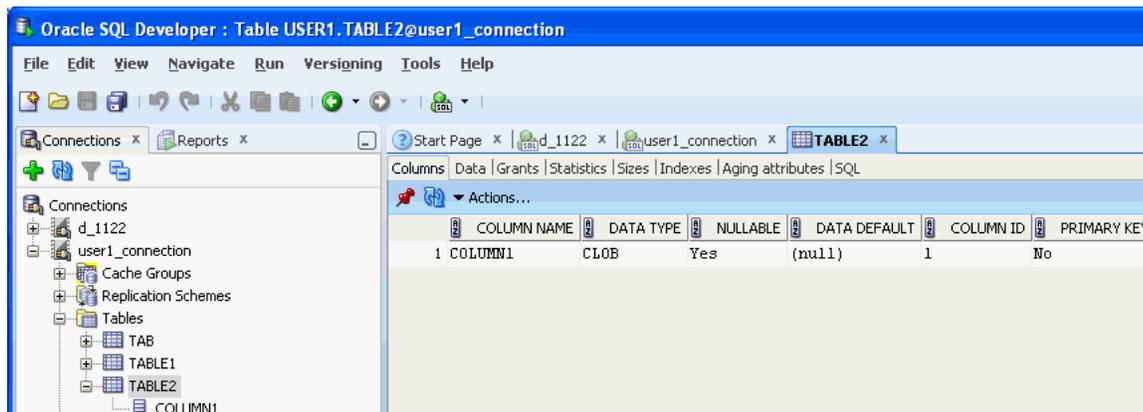
If you wish to create a table with a LOB data type, then in the **Create Table** dialog, expand the **Type** column header. You see CLOB, NCLOB, and BLOB as valid data types.

Figure 6–1 Specifying LOB data types in the Create Table dialog



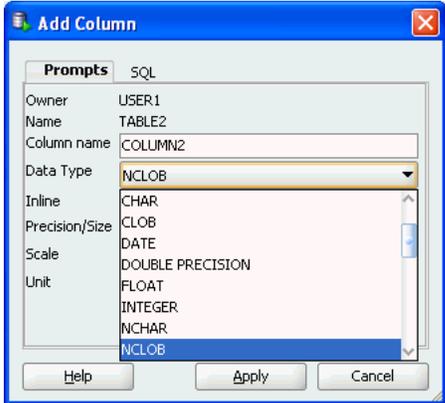
To view the data type for your columns, select the table and then choose the **Columns** tab. You see the LOB data type for your column.

Figure 6–2 Displaying Column data types



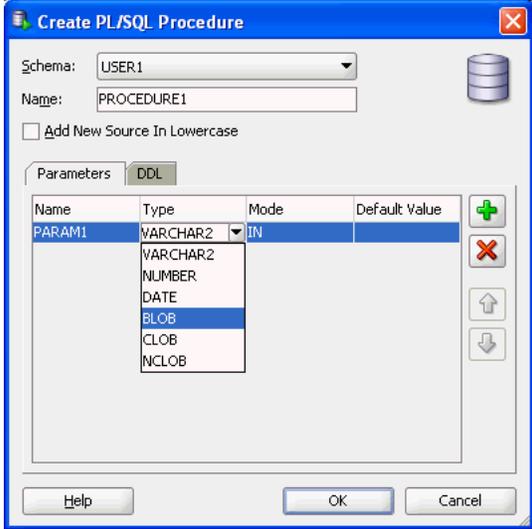
If you wish to add a column and specify a LOB data type, in the **Add Column** dialog, expand **Data Type**. You see CLOB, NCLOB, and BLOB among the possible choices.

Figure 6-3 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.

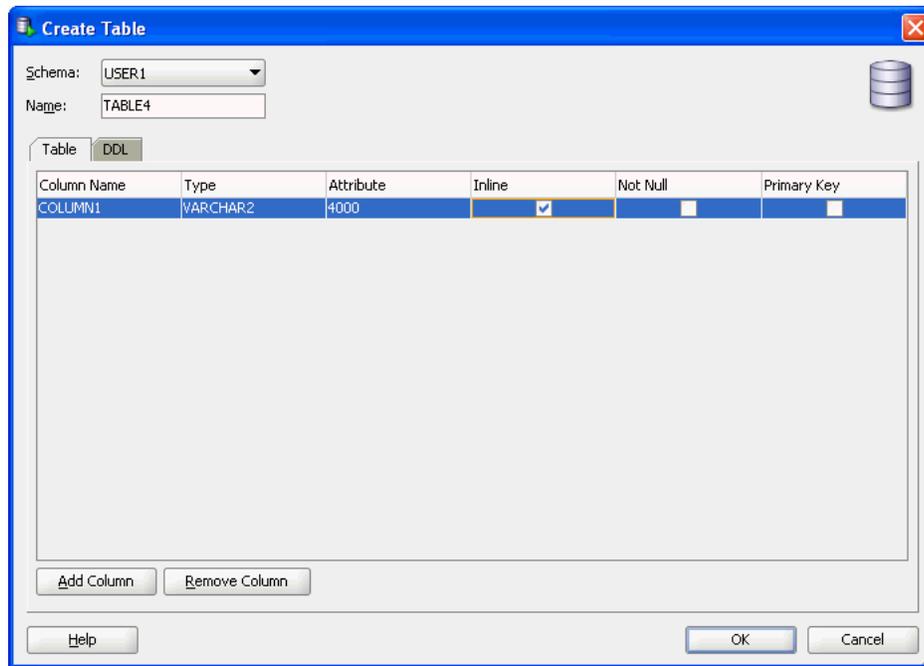
Figure 6-4 Choosing a parameter type



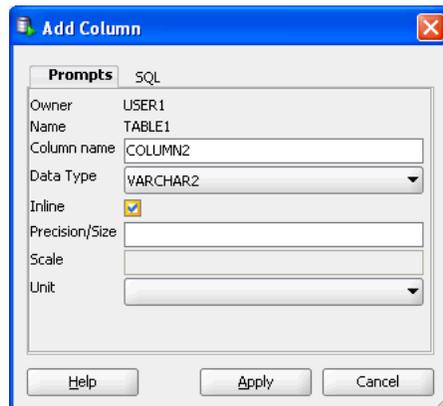
Specifying the INLINE attribute for columns

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 checkbox to define the column with the `INLINE` attribute.

Figure 6–5 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, then choose **Column>Add**. The **Add Column** dialog appears. For columns of type `VARCHAR2`, `NVARCHAR2`, and `VARBINARY`, click in the **Inline** checkbox to add the column with the `INLINE` attribute.

Figure 6–6 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. Choose **Table > Compute Size**.

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

Figure 6–7 Compute Size dialog



4. Click in the checkbox 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 6–8 Displaying table size information

Columns Data Grants Statistics Sizes Indexes Aging attributes SQL	
Actions...	
Name	Value
1 INLINE BYTES	17296
2 OUT OF LINE BYTES	824
3 METADATA BYTES	840

You can choose the **Actions** menu to generate or regenerate the table size. If you choose the **Actions** menu, choose **Table > 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, choose the **TimesTen Reports > Table > Table Sizes** report. For more information on TimesTen reports, see "[Generating TimesTen Reports](#)" on page 9-1.

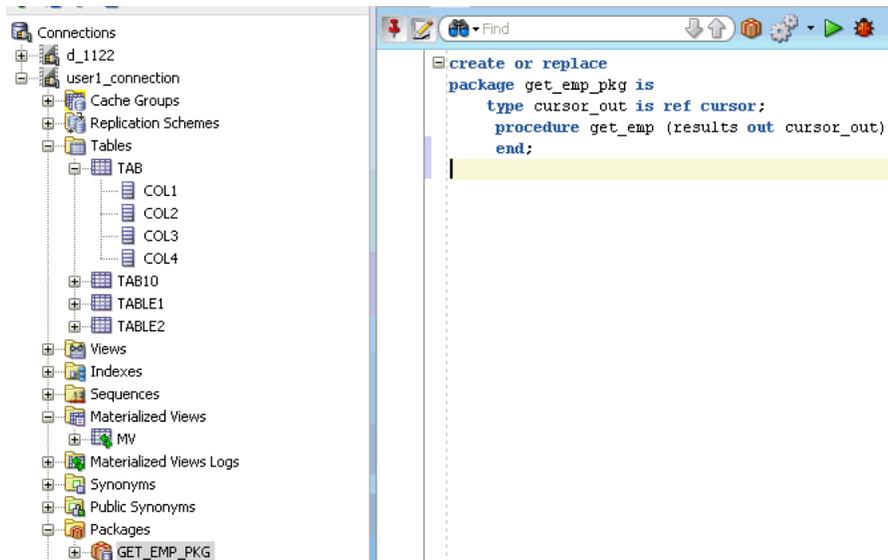
For more information on the `ttComputeTabSizes` built-in procedure, see "`ttComputeTabSizes`" in *Oracle TimesTen In-Memory Database Reference*.

Using a REF CURSOR as an OUT parameter

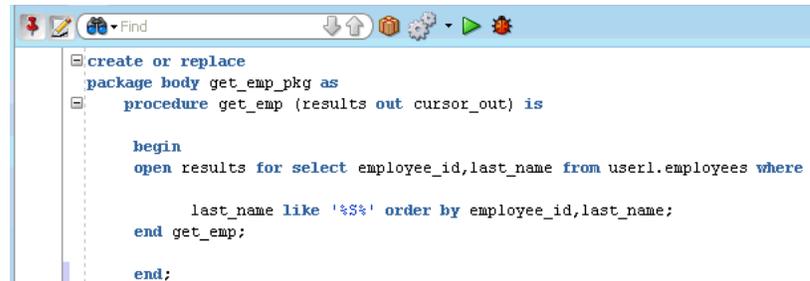
You can test a PL/SQL function, procedure, or package by defining a REF CURSOR as an OUT parameter in your PL/SQL function, procedure, or package. After you define a REF CURSOR, compile and run your PL/SQL function, procedure, or package. The **Run** dialog appears and when you choose **OK**, the details 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 6–9 *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.

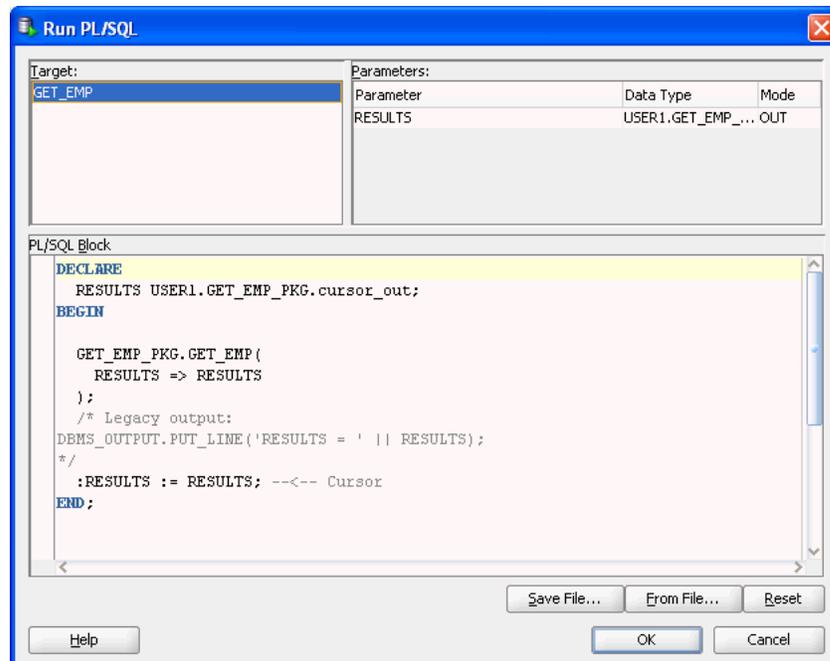
Figure 6–10 *get_emp_pkg* package body definition


```

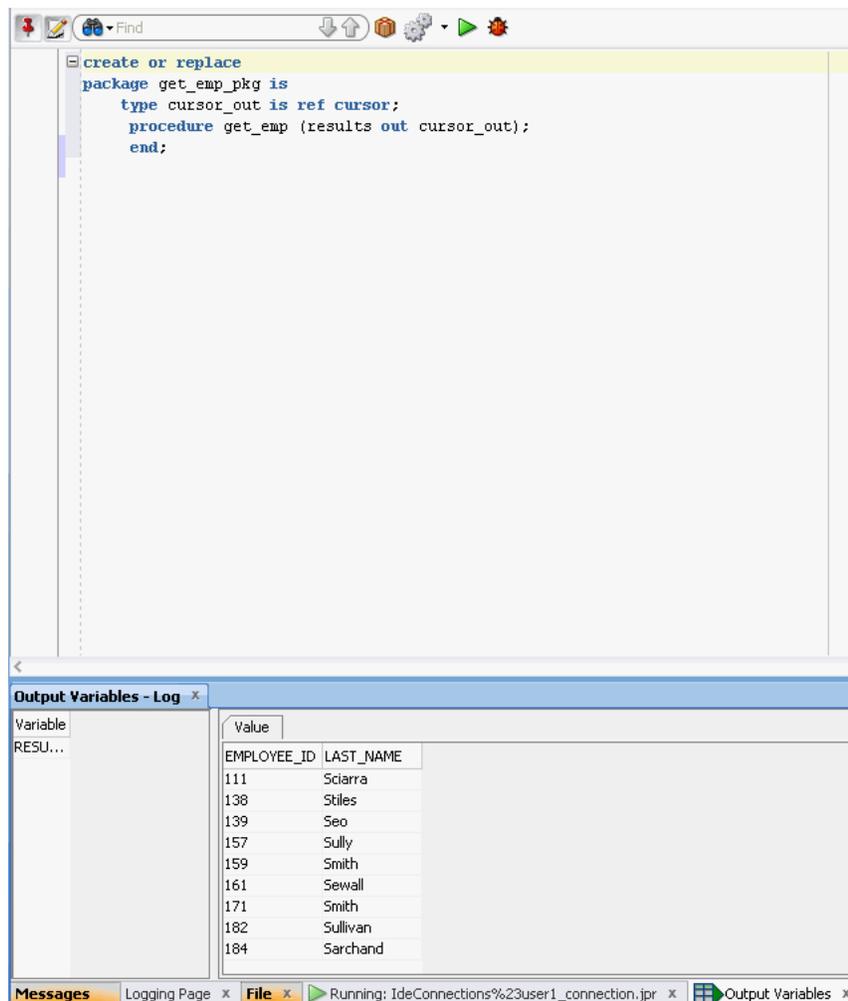
create or replace
package body get_emp_pkg as
  procedure get_emp (results out cursor_out) is
  begin
    open results for select employee_id,last_name from user1.employees where
      last_name like '%S%' order by employee_id,last_name;
  end get_emp;
end;

```

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 6–11 *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 6–12 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 6-1.

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 6–13 Viewing the columns

COLUMN NAME	DATA TYPE	NULLABLE	DATA DEFAULT	COLUMN ID	PRIMARY KEY
DEPARTMENT_ID	NUMBER(4)	No	(null)	1	Yes
DEPARTMENT_NAME	VARCHAR2(30 BYTE) inline	No	(null)	2	No
MANAGER_ID	NUMBER(6)	Yes	(null)	3	No
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 3-8.

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 6–14 Viewing the aging attributes for a table

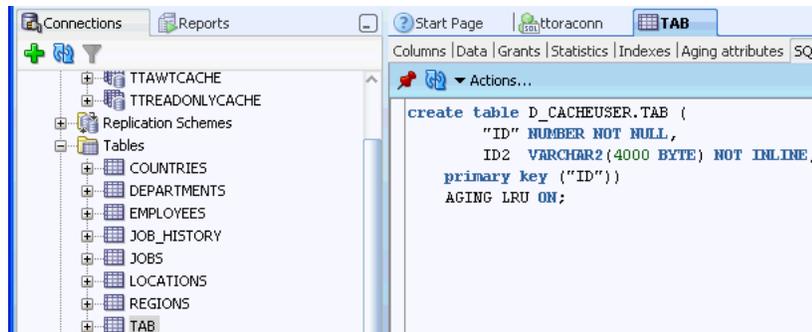
Attribute	Value
Type	Time-based
State	On
Timestamp column	COL2
Life-time	1 days
Cycle	10 minutes

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 5-3.

Figure 6–15 Viewing the SQL for the table definition

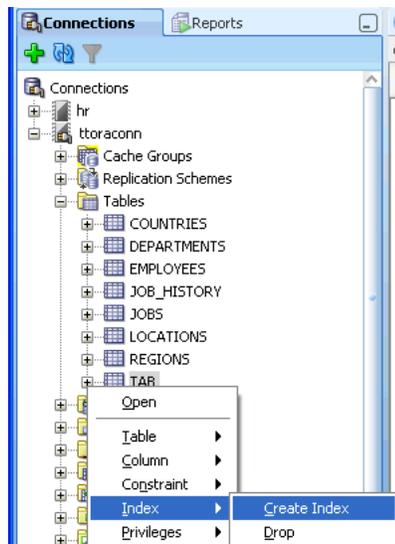


Creating an index on a table

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

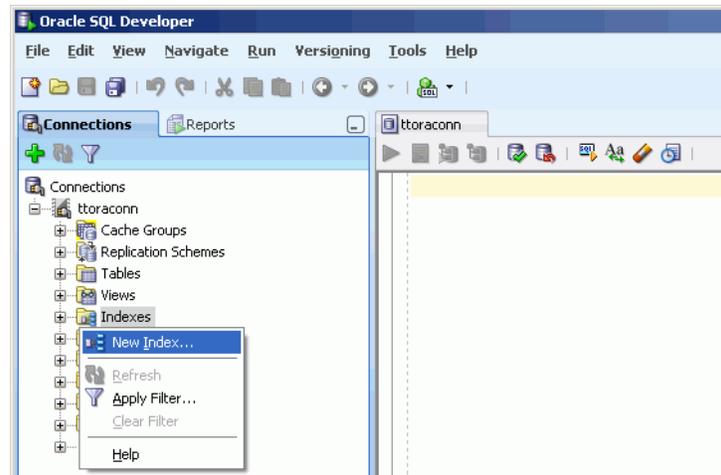
To create an index, click the + to the left of the **Tables** node to view the list of tables. Then right-click the name of the table to create an index on and select **Index > Create Index**.

Figure 6–16 Creating an index from Tables node



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

Figure 6–17 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**:

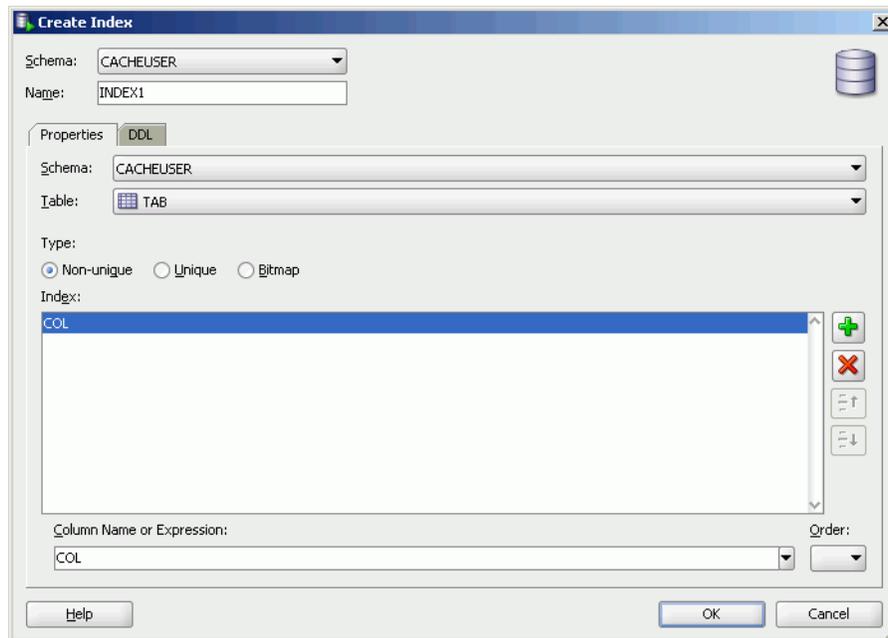
- To create a non-unique range index, choose **Non-unique**
- To create a unique range index, choose **Unique**
- To create a bitmap index, choose **Bitmap**

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. The columns can be selected 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.

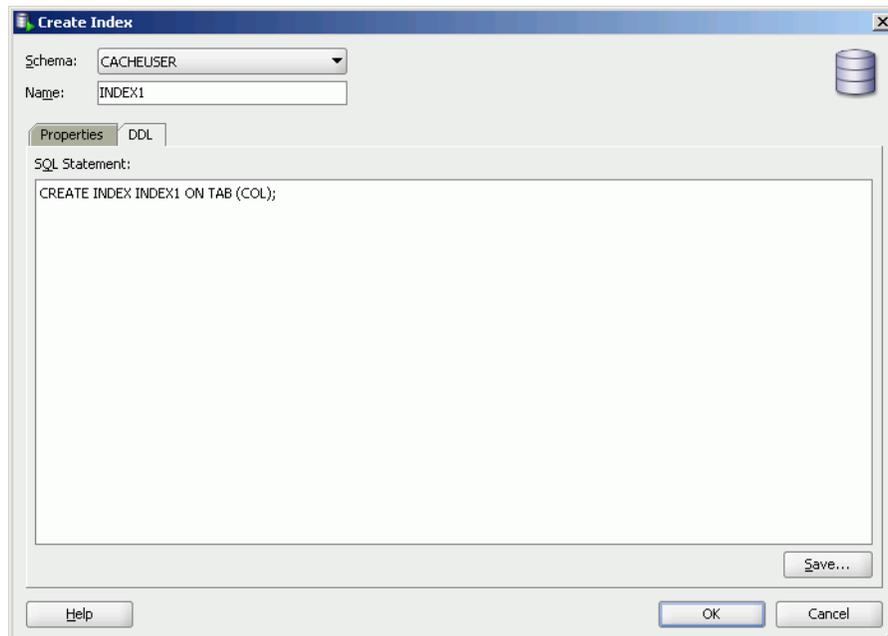
Figure 6–18 Create index dialog



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 6–19 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. Choose **Constraint > 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 (:)**, type 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, type 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. Choose **Constraint > 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 (:)**, type 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*.

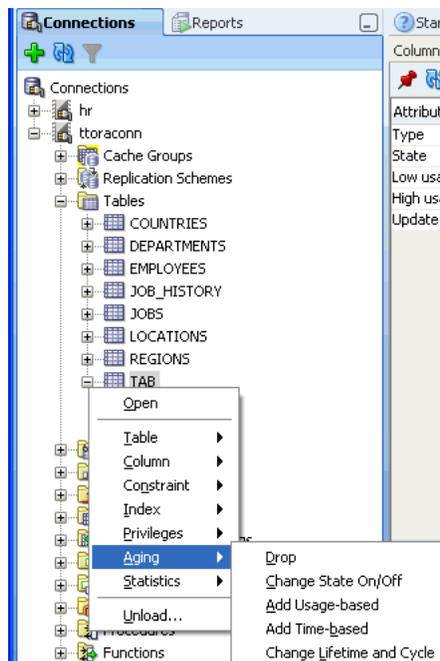
Altering the aging properties of a table

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

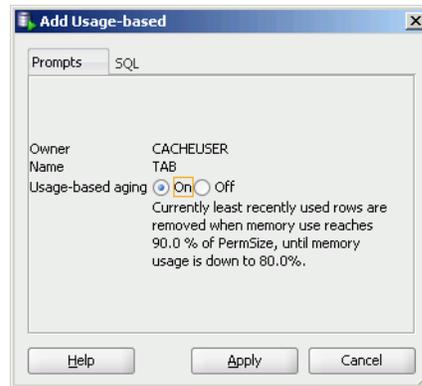
To add an aging policy to a regular table or a cache table, under the **Tables** node, right-click the name of the table to add an aging policy to and then choose **Aging > Add Usage-based** to add an LRU aging policy or choose **Aging > Add Time-based** to add a time-based aging policy. An aging policy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

Figure 6–20 Aging attributes



If you are unable to find the table that you are looking for, see "[Locating a TimesTen database object](#)" on page 6-1.

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 6–21 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 3-8.

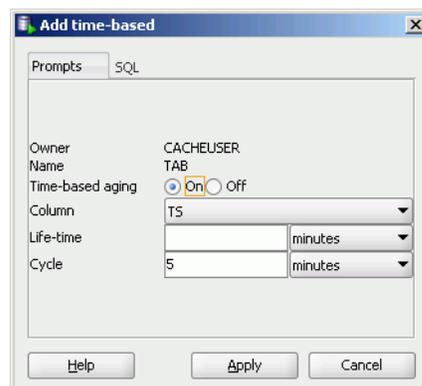
An LRU aging policy can be added to any regular table, and only to cache tables in an AWT, SWT or user managed cache group that does not have automatic refresh defined.

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 6–22 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 then choose **Aging > Drop**. See [Figure 6–20, "Aging attributes"](#).

If you are unable to find the table that you are looking for, see ["Locating a TimesTen database object"](#) on page 6-1.

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

Figure 6–23 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 then choose **Aging > Change State On/Off**. See [Figure 6–20, "Aging attributes"](#).

If you are unable to find the table that you are looking for, see ["Locating a TimesTen database object"](#) on page 6-1.

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.

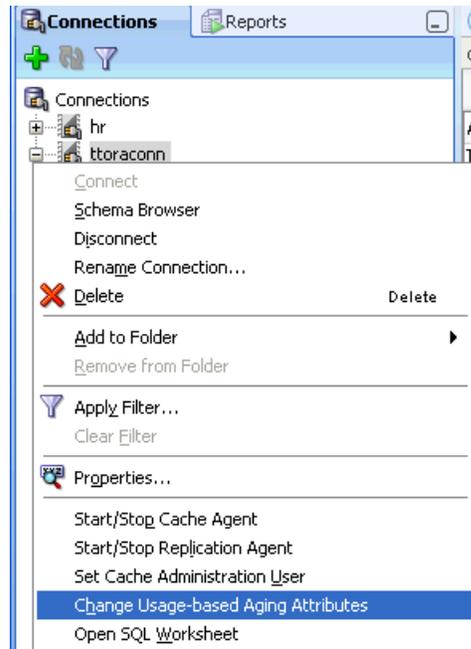
Figure 6–24 Changing aging state



Changing the memory usage thresholds and LRU aging cycle

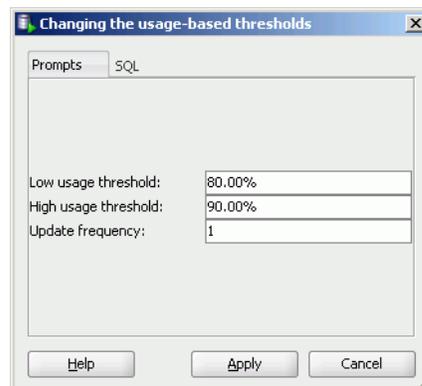
To change the memory usage thresholds and the LRU aging cycle, right-click the node of the connection name for the TimesTen database and choose **Change Usage-based Aging Attributes**.

Figure 6–25 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 6–26 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 then choose **Aging > Change Lifetime and Cycle**. See [Figure 6–20, "Aging attributes"](#).

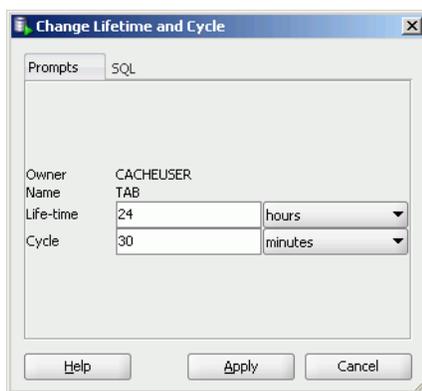
If you are unable to find the table that you are looking for, see "[Locating a TimesTen database object](#)" on page 6-1 for information.

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 6–27 *Changing lifetime and cycle*

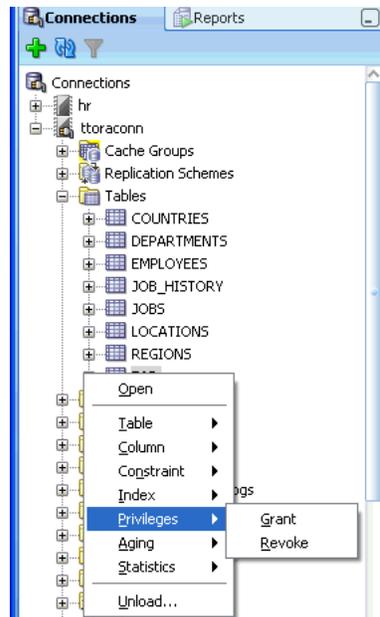


Granting and revoking object privileges

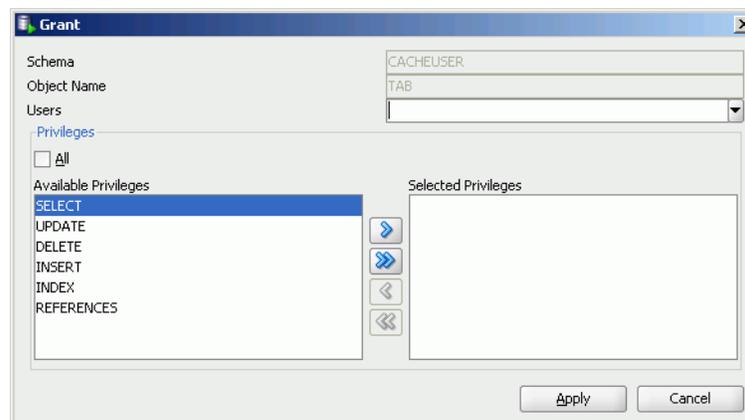
You can grant privileges to and revoke privileges from the following database objects:

- 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 > Grant**.

Figure 6–28 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 6–29 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.

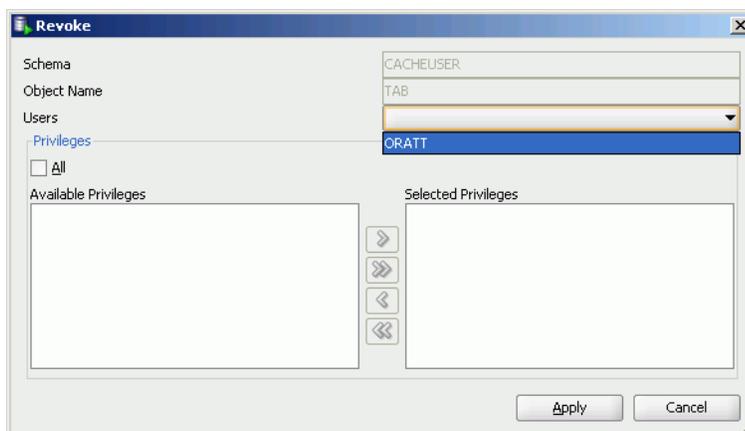
Only users with the ADMIN system privilege or the owner of an object can grant object privileges on the object.

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 > Revoke**. See [Figure 6–28, "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 6–30 Revoke privileges dialog



The **Users** drop-down menu shows only users who have been granted 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 been granted 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. Choose **Table > 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.

ttIsql commands in SQL worksheet

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

- autocommit
- desc
- dssize
- version

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

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:

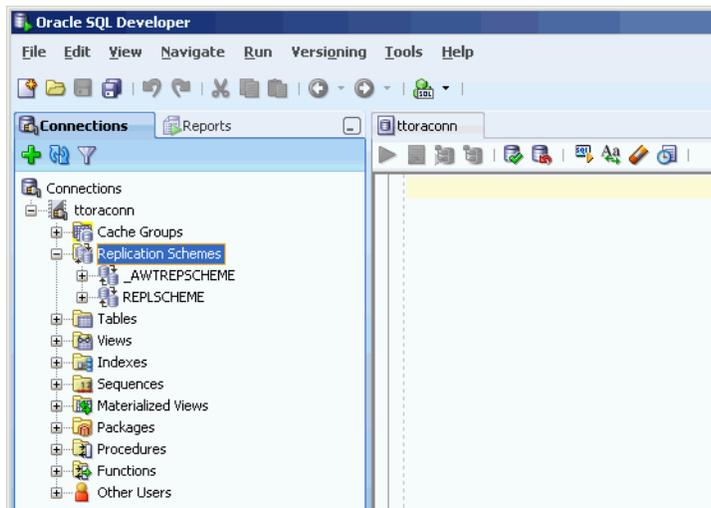
- [Viewing the replication scheme](#) including
 - [Viewing the details of the TimesTen subscriber databases](#)
 - [Viewing the store attributes of the replicated TimesTen databases](#)
- [Starting the replication agent](#)

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 7-1 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.

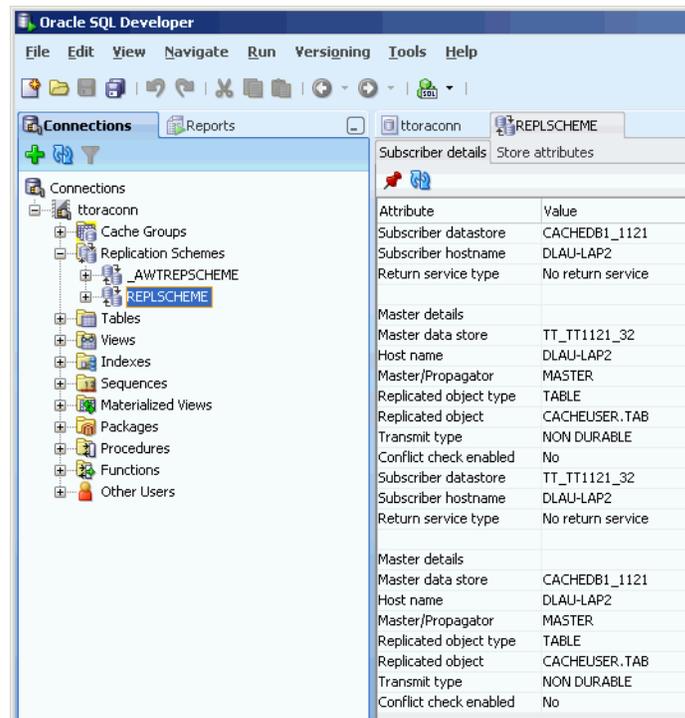
The name of the replication scheme that is automatically created to propagate updates on cache tables in AWT cache groups to the corresponding Oracle tables is `_AWTREPScheme`.

If you are unable to find the replication scheme that you are looking for, see ["Locating a TimesTen database object"](#) on page 6-1.

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 7-2 Viewing the subscriber

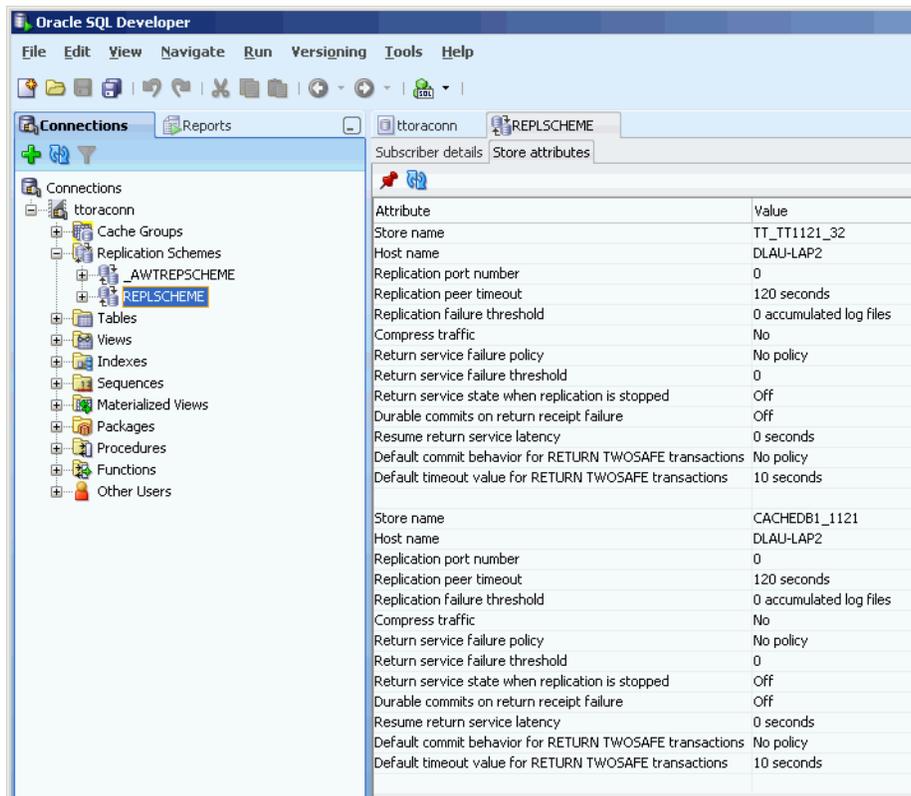


Viewing the store attributes of the replicated TimesTen databases

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 7-3 Viewing the store attributes



For more information about the creating a replication scheme, see "Defining Replication Schemes" in the *Oracle TimesTen In-Memory Database TimesTen to TimesTen 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 1-3, "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 1-3, "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 TimesTen to TimesTen Replication Guide* or "Managing the replication agent" in the *Oracle In-Memory Database Cache User's Guide*.

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](#)
- [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. Table-level statistics such as the number of rows in a table are stored 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.

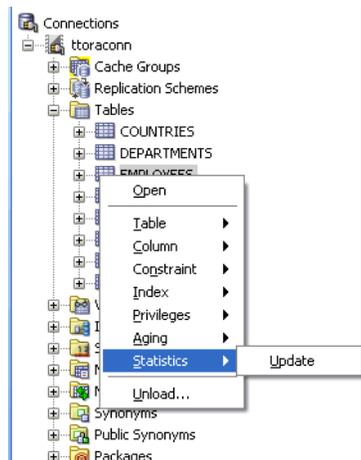
Table and column statistics are not computed as updates occur on regular and cache tables. Instead statistics are updated when an explicit request is made.

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

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

Figure 8–1 Update statistics for a table



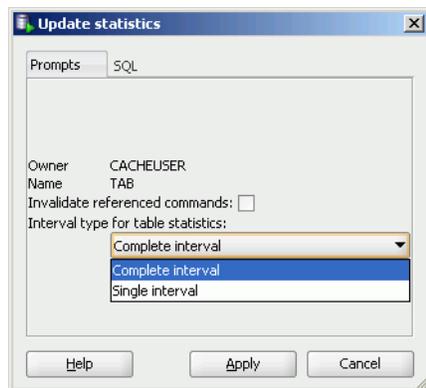
If you are unable to find the table, see ["Viewing the characteristics of a table"](#) on page 6-8.

In the **Prompts** tab of the **Update statistics** dialog, choose **Invalidate referenced commands** to invalidate the execution plans of statements that reference the table on which statistics are updated. Invalidating a statement's execution plan causes that statement to be recompiled or reprepared 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 8–2 Update statistics dialog



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

Figure 8-3 Statistics for employees table

Name	Value
NUMTUPS	107
LASTSTATSUPDATE	Tue Feb 15 02:42:29 2011

TBLNAME	COLNAME	STATS
EMPLOYEES	EMPLOYEE_ID	(6, 0, 107, 107, (20, 20, 1
EMPLOYEES	FIRST_NAME	(1, 0, 89, 107, (89, 107, 0,
EMPLOYEES	LAST_NAME	(6, 0, 102, 107, (19, 19, 2,
EMPLOYEES	EMAIL	(6, 0, 107, 107, (20, 20, 1,
EMPLOYEES	PHONE_NUMBER	(1, 0, 103, 107, (103, 107,
EMPLOYEES	HIRE_DATE	(1, 0, 88, 107, (88, 107, 0,

If you are unable to find the table that you are looking for, see "[Locating a TimesTen database object](#)" on page 6-1.

In the **Statistics** tab within the table's tab, 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 is displayed:

- The name of the column
- The number of intervals the data was 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, the following is displayed:

- 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*.

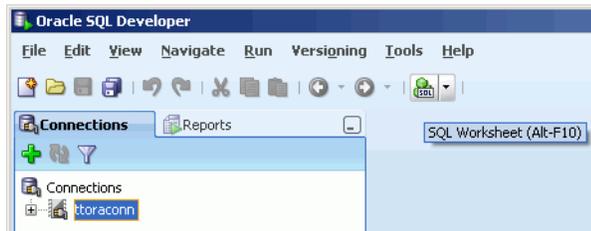
Viewing the execution plan of a statement

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 1–3, "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 8–4 *Selecting the SQL Worksheet*



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

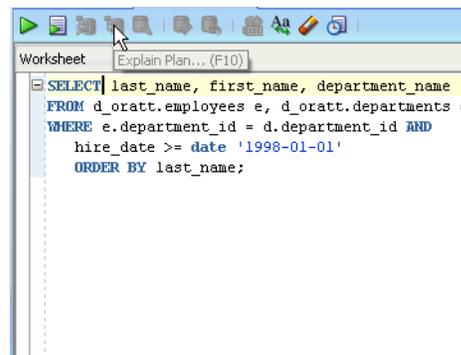
Figure 8–5 *Select Connection dialog*



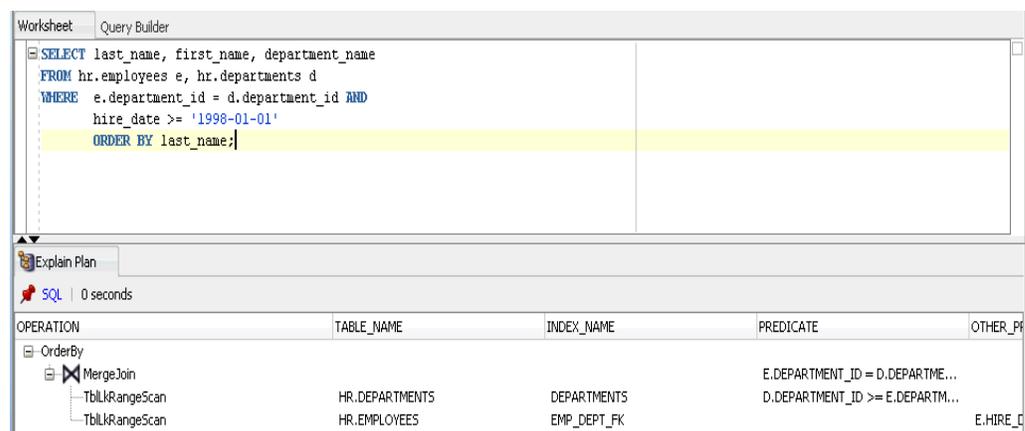
A connection to the TimesTen database is established if a connection associated with the selected connection name does not exist.

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 8-1.

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

Figure 8–6 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 8–7 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*.

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.

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.

A

aging
 altering for table, 6-14
 changing lifetime and cycle, 6-18
 dropping for table, 6-16
aging attributes
 table, 6-9
aging policy
 defined, 3-8
 LRU, 3-8
 time-based, 3-9
aging state
 changing for table, 6-16
automatic refresh attributes
 altering, 5-4

C

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

 viewing the aging attributes, 5-3
 viewing the definition, 5-3
cache table
 aging, 3-8
cache tables
 viewing, 5-2
child table
 defining foreign keys, 3-12
 selecting columns to cache, 3-11
 specifying the table attributes, 3-12
child tables
 defining, 3-10
columns of table
 viewing, 6-8
Computing table sizes, 6-4
configuration
 creating the cache manager user, 2-1
 creating the timesten user, 2-1
 granting privileges, 2-1
 Oracle and TimesTen databases, 2-1
connect
 TimesTen database, 1-1
connection
 disconnect, 1-4
 existing, 1-3
 new, 1-2
connection information dialog, 1-4

E

environment variables
 CLASSPATH, 1-1
 LD_LIBRARY_PATH, 1-1
 PATH, 1-1
execution plan, 8-4

I

index
 creating for table, 6-10
indexes
 viewing, 6-9
INLINE attribute, 6-3
installation
 scripts, 1-1

L

LOB data types, 6-2

M

memory usage threshold
changing for table, 6-17

N

new connection dialog, 1-2

O

object privileges, 6-18

P

primary key attributes
changing, 6-13
primary key constraint
creating, 6-12

Q

query optimizer, 8-1
explain plan, 8-5
statistics detail, 8-3

R

REF CURSOR

OUTPUT parameter, 6-6

replication

interfacing, 7-1

replication agent

starting and stopping, 7-4

replication scheme

viewing, 7-1

reports

About Your Database, 9-1

All Objects, 9-2

Database Administration, 9-2

generating, 9-1

introduction, 9-1

PLSQL, 9-3

Security, 9-3

Table, 9-3

root table

primary key, 3-6

where clause, 3-7

root table definition

verifying, 3-10

root table dialog, 3-4

S

store attributes

viewing, 7-3

subscriber database

viewing, 7-2

T

table

viewing the characteristics, 6-8

table and column statistics

updating, 8-1

table definition

viewing, 6-9

time-based aging, 3-9

TimesTen database object

locating, 6-1

ttIsql commands, 6-21