Setting up the Oracle and TimesTen systems

Before you can create a cache grid or a cache group, you must first install TimesTen and then configure the Oracle and TimesTen systems

Create users in the Oracle database

Before you can use Oracle In-Memory Database Cache, you must create some Oracle users:

- A user `timesten` owns Oracle tables that store information about cache grids.
- One or more schema users own the Oracle tables to be cached in a TimesTen database. These may be existing users or new users.
- A cache administration user creates and maintains Oracle objects that store information used to manage cache grids and enforce predefined behaviors of particular cache group types.

Start SQL*Plus on the Oracle system from an operating system shell or command prompt, and connect to the Oracle database instance as the `sys` user:

```
% cd TimesTen_install_dir/oraclescripts
% sqlplus sys as sysdba
Enter password: password
```

Use SQL*Plus to create a default tablespace that will be used by both the `timesten` user and the cache administration user. This tablespace should only be used to store objects for Oracle In-Memory Database Cache and should not be shared with other applications. Then run the SQL*Plus script*TimesTen_install_dir*`/oraclescripts/initCacheGlobalSchema.sql` to create the following elements:

- The `timesten` user
- The Oracle tables owned by the `timesten` user to store information about cache grids
- The `TT_CACHE_ADMIN_ROLE` role that defines privileges on these Oracle tables

Pass the default tablespace as an argument to the `initCacheGlobalSchema.sql` script. In the following example, the name of the default tablespace is `cachetblsp`:

```
SQL> CREATE TABLESPACE cachetblsp DATAFILE 'datfttuser.dbf' SIZE 100M;
SQL> @initCacheGlobalSchema "cachetblsp"
```

Next use SQL*Plus to create a schema user. Grant this user the minimum set of privileges required to create tables in the Oracle database to be cached in a TimesTen database. In the following example, the schema user is `oratt`:

```
SQL> CREATE USER oratt IDENTIFIED BY oracle;
SQL> GRANT CREATE SESSION, RESOURCE TO oratt;
```

Then use SQL*Plus to perform the following operations:

- Create a cache administration user.
- Run the SQL*Plus script *TimesTen_install_dir*`/oraclescripts/grantCacheAdminPrivileges.sql` to grant the cache administration user the minimum set of privileges required to perform cache grid and cache group operations.

Pass the cache administration user name as an argument to the `grantCacheAdminPrivileges.sql` script. In the following example, the cache administration user name is `cacheuser` and the name of its default tablespace is `cachetblsp`:

```
SQL> CREATE USER cacheuser IDENTIFIED BY oracle
  2  DEFAULT TABLESPACE cachetblsp QUOTA UNLIMITED ON cachetblsp;
SQL> @grantCacheAdminPrivileges "cacheuser"
SQL> exit
```

The privileges that the cache administration user requires depend on the types of cache groups you create and the operations that you perform on the cache groups.

Create a DSN for the TimesTen database

```
SQL> SELECT value FROM nls_database_parameters WHERE
parameter='NLS_CHARACTERSET';
```

On UNIX, in the `.odbc.ini` file that resides in your home directory or the *TimesTen_install_dir*`/info/sys.odbc.ini` file, create a TimesTen DSN `cachealone1` and set the following connection attributes:

```
[cachealone1]
DataStore=/users/OracleCache/alone1
PermSize=64
OracleNetServiceName=oracledb
```

```
DatabaseCharacterSet=AL32UTF8
```

On Windows, create a TimesTen user DSN or system DSN `cachealone1` and set the following connection attributes:

- Data Store Path + Name: `c:\temp\alone1`
- Permanent Data Size: `64`
- Oracle Net Service Name: `oracledb`
- Database Character Set: `AL32UTF8`

Use the default settings for all the other connection attributes.

Create users in the TimesTen database

In addition to the Oracle users, you must create some TimesTen users before you can use Oracle In-Memory Database Cache:

- A *cache manager user* performs cache grid and cache group operations. The TimesTen cache manager user must have the same name as an Oracle user that can access the cached Oracle tables. For example, the Oracle user must have privileges to select from and update the cached Oracle tables. The Oracle user can be the cache administration user, a schema user, or some other existing user. The password of the cache manager user can be different than the password of the Oracle user with the same name.

  The cache manager user creates and configures the cache grid and creates the cache groups. It may perform operations such as loading or refreshing a cache group although these operations can be performed by any TimesTen user that has sufficient privileges. The cache manager user can also monitor various aspects of the caching environment, such as the grid itself or asynchronous operations that are performed on cache groups such as automatic refresh.

- One or more *cache table users* own the cache tables. You must create a TimesTen cache table user with the same name as an Oracle schema user for each schema user who owns or will own Oracle tables to be cached in the TimesTen database. The password of a cache table user can be different than the password of the Oracle schema user with the same name.

  The owner and name of a TimesTen cache table is the same as the owner and name of the corresponding cached Oracle table.

Start the `ttIsql` utility on the TimesTen system from an operating system shell or command prompt as the instance administrator, and connect to the `cachealone1`DSN to create the TimesTen database that will be used to cache data from an Oracle database:

```
% ttIsql cachealone1
```

Use `ttIsql` to create a cache manager user. Grant this user the minimum set of privileges required to create a cache grid and cache groups, and perform operations on the cache groups. In the following example, the cache manager user name is `cacheuser`, which is the same name as the Oracle cache administration user that was created earlier:

```
Command> CREATE USER cacheuser IDENTIFIED BY timesten;
Command> GRANT CREATE SESSION, CACHE_MANAGER, CREATE ANY TABLE TO cacheuser;
```

Then use `ttIsql` to create a cache table user. In the following example, the cache table user name is `oratt`, which is the same name as the Oracle schema user that was created earlier:

```
Command> CREATE USER oratt IDENTIFIED BY timesten;
Command> exit
```

The privileges that the cache manager user requires depend on the types of cache groups you create and the operations that you perform on the cache groups.

Set the cache administration user name and password in the TimesTen database

Start the `ttIsql` utility and connect to the `cachealone1` DSN as the cache manager user. In the connection string, specify the cache manager user name in the `UID`connection attribute. (In this example, the TimesTen cache manager user name is the same as the Oracle cache administration user name.) Specify the cache manager user's password in the `PWD` connection attribute and the cache administration user's password in the `OraclePWD` connection attribute within the connection string.

```
% ttIsql "DSN=cachealone1;UID=cacheuser;PWD=timesten;OraclePWD=oracle"
```

Use `ttIsql` to call the `ttCacheUidPwdSet` built-in procedure to set the Oracle cache administration user name and password:

```
Command> call ttCacheUidPwdSet('cacheuser','oracle');
```

Creating a cache grid

After you have created the Oracle users, the TimesTen database, and the TimesTen users, and set the Oracle cache administration user name and password in the TimesTen database, you need to create a cache grid to define a framework for TimesTen databases that cache tables from an Oracle database.

As the cache manager user, use the `ttIsql` utility to call the `ttGridCreate` built-in procedure to create a cache grid `myGrid`:

```
Command> call ttGridCreate('myGrid');
```

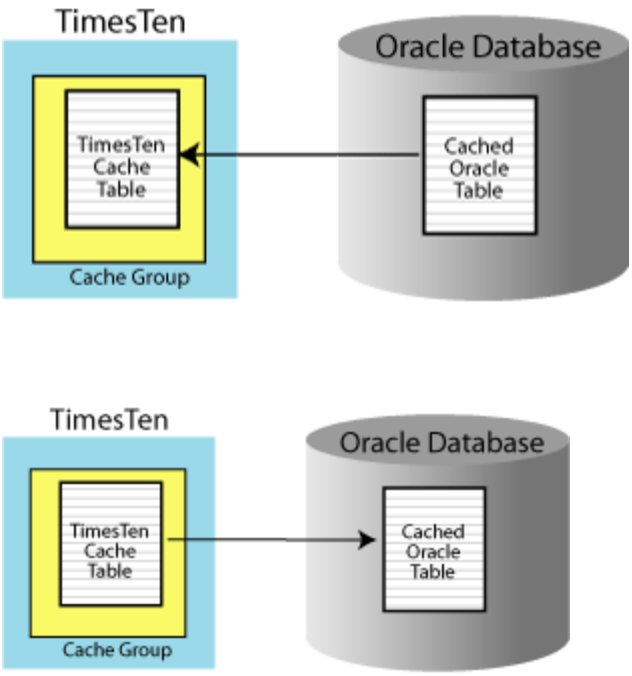
Then use `ttIsql` to call the `ttGridNameSet` built-in procedure to associate the TimesTen database with the `myGrid` cache grid:

```
Command> call ttGridNameSet('myGrid');
```

Creating cache groups

After you have created a cache grid and associated the TimesTen database with the grid, you are ready to create cache groups. You create a read-only cache group. Then you create an asynchronous writethrough (AWT) cache group.

Figure 2-1 Single-table read-only cache group



Create the Oracle tables to be cached

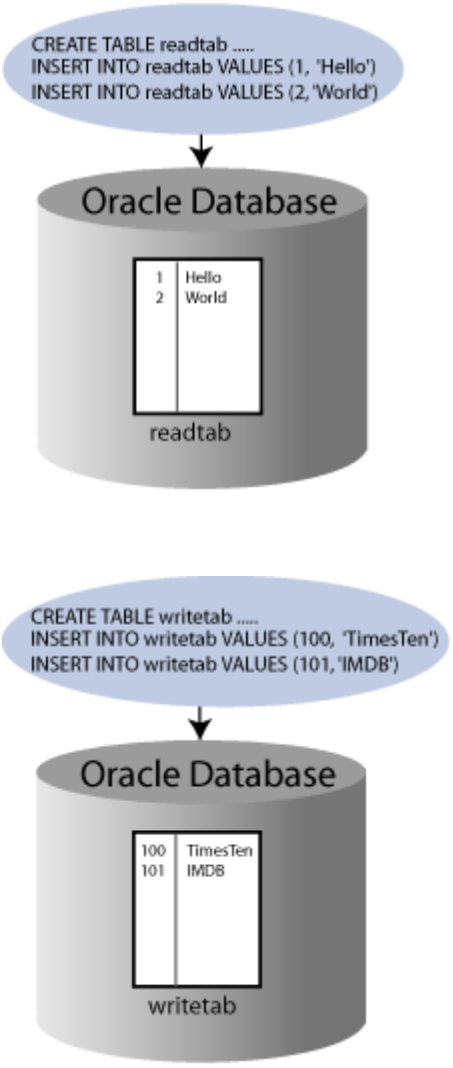
Start SQL*Plus and connect to the Oracle database as the schema user:

```
% sqlplus oratt/oracle
```

Use SQL*Plus to create a table `readtab` and a table `writetab`

```
SQL> CREATE TABLE readtab (keyval NUMBER NOT NULL PRIMARY KEY, str VARCHAR2(32));
SQL> CREATE TABLE writetab (pk NUMBER NOT NULL PRIMARY KEY, attr VARCHAR2(40));
```

Figure 2-3 Creating an Oracle table to be cached in a read-only cache group



Then use SQL*Plus to insert some rows into the `readtab` and `writetab` tables, and commit the changes:

```
SQL> INSERT INTO readtab VALUES (1, 'Hello');
SQL> INSERT INTO readtab VALUES (2, 'World');

SQL> INSERT INTO writetab VALUES (100, 'TimesTen');
SQL> INSERT INTO writetab VALUES (101, 'IMDB');
SQL> COMMIT;
```

Next use SQL*Plus to grant the `SELECT` privilege on the `readtab` table, and the `SELECT`, `INSERT`, `UPDATE` and `DELETE` privileges on the `writetab` table to the cache administration user:

```
SQL> GRANT SELECT ON readtab TO cacheuser;

SQL> GRANT SELECT ON writetab TO cacheuser;
SQL> GRANT INSERT ON writetab TO cacheuser;
SQL> GRANT UPDATE ON writetab TO cacheuser;
SQL> GRANT DELETE ON writetab TO cacheuser;
```

The `SELECT` privilege on the `readtab` table is required to create a read-only cache group that caches this table and to perform automatic refresh operations from the cached Oracle table to the TimesTen cache table.

The `SELECT` privilege on the `writetab` table is required to create an AWT cache group that caches this table. The `INSERT`, `UPDATE` and `DELETE` privileges on the`writetab` table are required to perform writethrough operations from the TimesTen cache table to the cached Oracle table.

Start the cache agent

As the cache manager user, use the `ttIsql` utility to call the `ttCacheStart` built-in procedure to start the cache agent on the TimesTen database:

```
Command> call ttCacheStart;
```

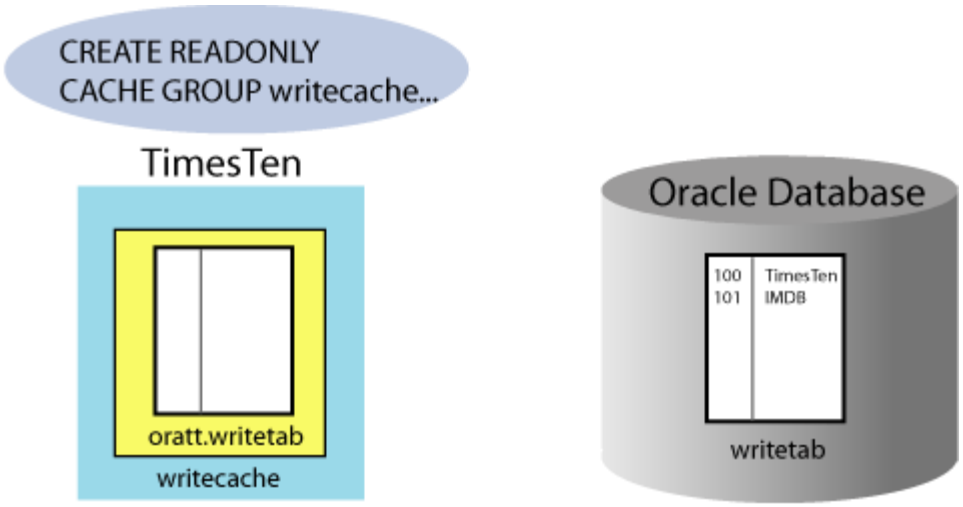
Create the cache groups

As the cache manager user, use the `ttIsql` utility to create a read-only cache group `readcache` that caches the Oracle `oratt.readtab` table and a dynamic AWT global cache group `writecache` that caches the Oracle `oratt.writetab` table:

```
Command> CREATE READONLY CACHE GROUP readcache
       > AUTOREFRESH INTERVAL 5 SECONDS
       > FROM oratt.readtab
       > (keyval NUMBER NOT NULL PRIMARY KEY, str VARCHAR2(32));

Command> CREATE DYNAMIC ASYNCHRONOUS WRITETHROUGH GLOBAL CACHE GROUP writecache
       > FROM oratt.writetab
       > (pk NUMBER NOT NULL PRIMARY KEY, attr VARCHAR2(40));
```

The cache groups `readcache` and `writecache`, and their respective cache tables `oratt.readtab` and `oratt.writetab`, whose owners and names are identical to the cached Oracle tables, are created in the TimesTen database. the `writecache` cache group caches the `oratt.writetab` table.



Use the `ttIsql cachegroups` command to view the definition of the `readcache` and `writecache` cache groups:

```
Command> cachegroups;

Cache Group CACHEUSER.READCACHE:

  Cache Group Type: Read Only
  Autorefresh: Yes
  Autorefresh Mode: Incremental
  Autorefresh State: Paused
  Autorefresh Interval: 5 Seconds
  Autorefresh Status: ok
  Aging: No aging defined

  Root Table: ORATT.READTAB
  Table Type: Read Only
```

```
Cache Group CACHEUSER.WRITECACHE:

  Cache Group Type: Asynchronous Writethrough global (Dynamic)
  Autorefresh: No
  Aging: LRU on

  Root Table: ORATT.WRITETAB
  Table Type: Propagate

2 cache groups found.
```

Start the replication agent for the AWT cache group

As the cache manager user, use the `ttIsql` utility to call the `ttRepStart` built-in procedure to start the replication agent on the TimesTen database:

```
Command> call ttRepStart;
```

The replication agent propagates committed updates on TimesTen cache tables in AWT cache groups to the cached Oracle tables.

Attaching the TimesTen database to the cache grid

If you are creating a local cache group, you do not need to attach the TimesTen database to the cache grid. Before you can perform operations on a global cache group or on its cache tables, you must attach the TimesTen database to the cache grid that it is associated with.

As the cache manager user, use the `ttIsql` utility to call the `ttGridAttach` built-in procedure to attach the TimesTen database to the `myGrid` cache grid:

```
Command> call ttGridAttach(1,'alone1','mysys',5001);
```

In this example, `alone1` is a name that is used to uniquely identify the grid member, `mysys` is the host name of the TimesTen system, and `5001` is the TCP/IP port for the cache agent.Calling the `ttGridAttach` built-in procedure automatically starts the cache agent if it is not already running.

Although the example in this chapter contains only one standalone TimesTen database as the sole grid member, it can be extended to include additional grid

members such as active standby pairs and other standalone TimesTen databases.

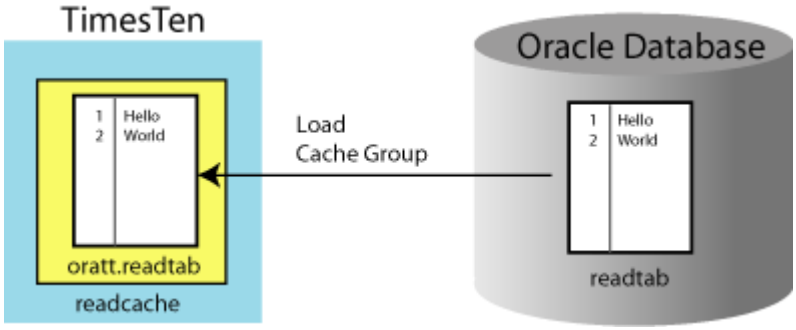
Performing operations on the read-only cache group

This section shows how to manually load the read-only cache group. Then it shows the TimesTen cache table being automatically refreshed with committed updates on the cached Oracle table.

Manually load the cache group

As the cache manager user, use the `ttIsql` utility to load the contents of the Oracle `oratt.readtab` table into the TimesTen `oratt.readtab` cache table in the`readcache` cache group:

```
Command> LOAD CACHE GROUP readcache COMMIT EVERY 256 ROWS;
2 cache instances affected.
Command> exit
```



Start the `ttIsql` utility and connect to the `cachealone1` DSN as the instance administrator. Use `ttIsql` to grant the `SELECT` privilege on the `oratt.readtab` cache table to the cache manager user so that this user can issue a `SELECT` query on this table.

```
% ttIsql cachealone1
Command> GRANT SELECT ON oratt.readtab TO cacheuser;
Command> exit
```

Start the `ttIsql` utility and connect to the `cachealone1` DSN as the cache manager user. Use `ttIsql` to query the contents of `oratt.readtab` cache table.

```
% ttIsql "DSN=cachealone1;UID=cacheuser;PWD=timesten;OraclePWD=oracle"
Command> SELECT * FROM oratt.readtab;
< 1, Hello >
```

```
< 2, World >
2 rows found.
```

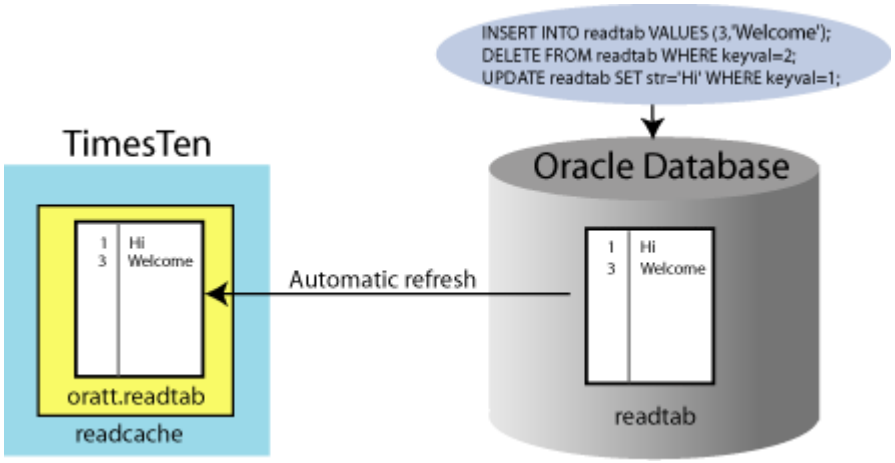
Update the cached Oracle table

Use SQL*Plus, as the Oracle schema user, to insert a new row, delete an existing row, and update an existing row in the Oracle `readtab` table, and commit the changes:

```
SQL> INSERT INTO readtab VALUES (3, 'Welcome');
SQL> DELETE FROM readtab WHERE keyval=2;
SQL> UPDATE readtab SET str='Hi' WHERE keyval=1;
SQL> COMMIT;
```

After 5 seconds, the `oratt.readtab` cache table in the `readcache` cache group is automatically refreshed with the committed updates on the cached Oracle`oratt.readtab` table

Figure 2-7 Automatically refresh the TimesTen cache table with Oracle updates



As the cache manager user, use the `ttIsql` utility to query the contents of the `oratt.readtab` cache table after the `readcache` cache group has been automatically refreshed with the committed updates on the cached Oracle table:

```
Command> SELECT * FROM oratt.readtab;
< 1, Hi >
< 3, Welcome >
2 rows found.
Command> exit
```

Performing operations on the dynamic updatable global cache group

This section shows how to dynamically load the AWT cache group. Then it shows committed updates on the TimesTen cache table being automatically propagated to the cached Oracle table.

Dynamically load the cache group

Start the `ttIsql` utility and connect to the `cachealone1` DSN as the instance administrator. Use `ttIsql` to grant the `SELECT` privilege on the `oratt.writetab` cache table to the cache manager user so that this user can issue a dynamic load `SELECT` statement on this table.

```
% ttIsql cachealone1
Command> GRANT SELECT ON oratt.writetab TO cacheuser;
Command> exit
```

Start the `ttIsql` utility and connect to the `cachealone1` DSN as the cache manager user. Use `ttIsql` to load a cache instance on demand from the Oracle`oratt.writetab` table to the TimesTen `oratt.writetab` cache table in the `writecache` cache group.

```
% ttIsql "DSN=cachealone1;UID=cacheuser;PWD=timesten;OraclePWD=oracle"
Command> SELECT * FROM oratt.writetab WHERE pk=100;
< 100, TimesTen >
1 row found.
Command> exit
```

In a dynamic cache group, a cache instance can be loaded into its cache tables on demand with a dynamic load statement. A `SELECT`, `UPDATE`, `DELETE` or `INSERT` statement issued on a TimesTen cache table that uniquely identifies a cache instance results in the cache instance being automatically loaded from the cached Oracle table if the data is not found in the cache table. A dynamically loaded cache instance consists of a single row in the root table of the cache group, and all the related rows in the child tables.

Update the TimesTen cache table

Start the `ttIsql` utility and connect to the `cachealone1` DSN as the instance administrator. Use `ttIsql` to grant the `INSERT`, `DELETE` and `UPDATE` privileges on

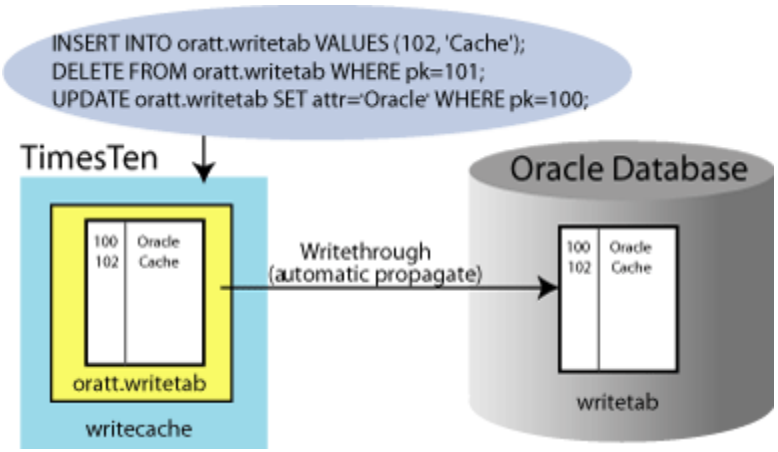
the`oratt.writetab` cache table to the cache manager user so that this user can perform updates on this table.

```
% ttIsql cachealone1
Command> GRANT INSERT ON oratt.writetab TO cacheuser;
Command> GRANT DELETE ON oratt.writetab TO cacheuser;
Command> GRANT UPDATE ON oratt.writetab TO cacheuser;
Command> exit
```

Start the `ttIsql` utility and connect to the `cachealone1` DSN as the cache manager user. Use `ttIsql` to insert a new row, delete an existing row, and update an existing row in the `oratt.writetab` cache table, and commit the changes.

```
% ttIsql "DSN=cachealone1;UID=cacheuser;PWD=timesten;OraclePWD=oracle"
Command> INSERT INTO oratt.writetab VALUES (102, 'Cache');
Command> DELETE FROM oratt.writetab WHERE pk=101;
Command> UPDATE oratt.writetab SET attr='Oracle' WHERE pk=100;
Command> COMMIT;
Command> exit
```

The committed updates on the `oratt.writetab` cache table in the `writecache` cache group are automatically propagated to the Oracle `oratt.writetab` table



As the Oracle schema user, use SQL*Plus to query the contents of the `writetab` table:

```
SQL> SELECT * FROM writetab;

        PK ATTR
---------- --------------------------------
       100 Oracle
```

```
      102 Cache

SQL> exit
```