




ORACLE®



ORACLE®

Oracle SQL Developer: Unit Testing, Tuning and Other Advanced Features

Syme Kutz
Architect Database Tools



The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



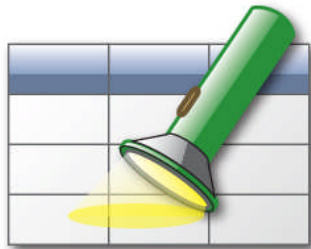
Agenda

- Unit Testing
- Integrated Modeling
- Trace Files
- Real-Time SQL Monitoring
- Refactoring
- Remote Debugging

Oracle SQL Developer

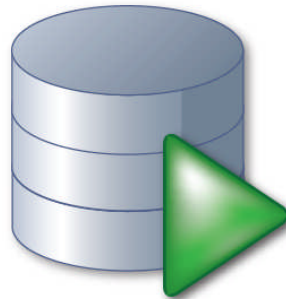
Today

SQL & PL/SQL IDE



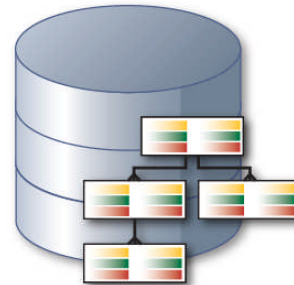
Lightweight, graphical interface that simplifies and enhances database development tasks

Migrations



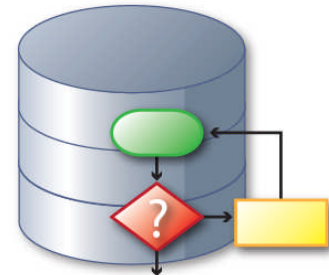
One-step migration of DB objects and data to Oracle. Translation of Sybase and SQL Server T-SQL to Oracle PL/SQL. Migrate from Access, MySQL

Data Modeler



Logical, Relational and Physical database modeling serving Data Architects, DBA's, Developers and Users

Unit Testing

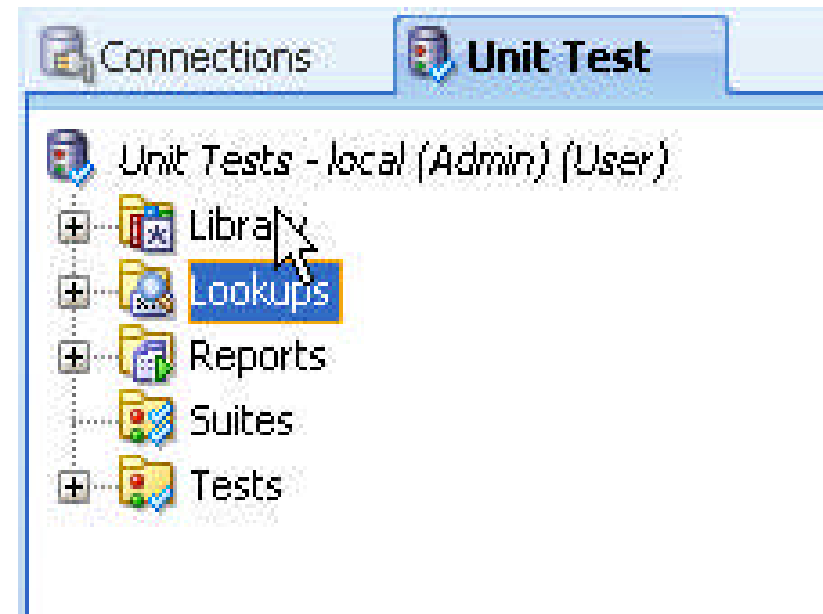


Simplify the automation of SQL and PL/SQL unit testing and code coverage with performance reporting and trending

ORACLE

Unit Testing – Overview

- Tests
- Suites
- Reports
- Library
- Static and dynamic lookups
- Multi user repository based
- Code coverage
- Command line use
- Target any database




Unit Testing - Tests



- Input/Return
 - Static or Dynamic Values
- Startups/TearDown
 - Table Copy/Restore
 - Row Copy/Restore
 - Custom
- Code coverage
- Success or failure testing
- Validation
 - Custom

The screenshot shows the Oracle JUnit interface for testing the procedure `PROCEDURE KLRICE.HI(P_IN IN VARCHAR2)`. The `Startup Process` and `Teardown Process` are both set to `None`. The `Gather Code Coverage Statistics` checkbox is checked. Under the `Test Implementation 1` tab, a table lists the parameter `P_IN` with datatype `VARCHAR2` and direction `IN`. Below the table, the `Dynamic Value Query` is defined as a SQL query that selects five different values from the `DUAL` table. The `Expected Result` is set to `Success` with an error number of `6502`. At the bottom, there are buttons for `Process Validation` (a green plus icon) and a close button (a grey X icon).

Parameter	Datatype	in/out
P_IN	VARCHAR2	IN

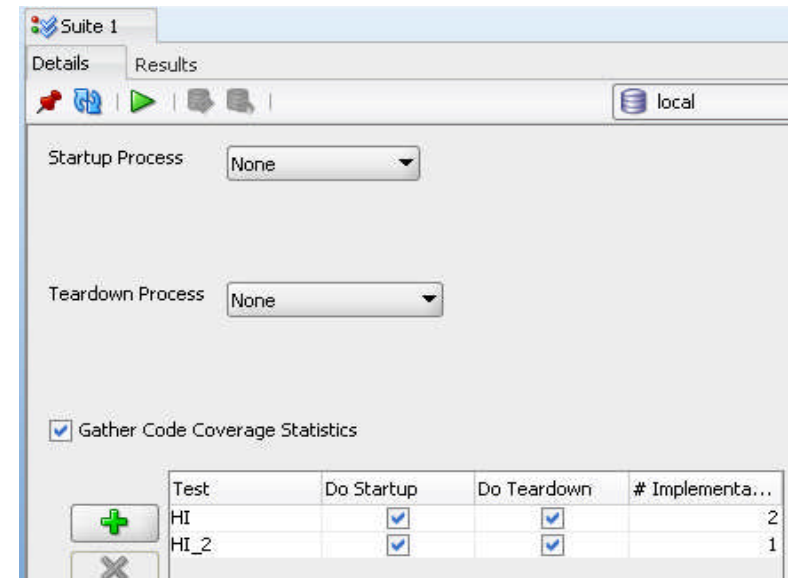
Dynamic Value Query  select 1 as P_IN from dual
union all
select 2 as P_IN from dual
union all
select 3 as P_IN from dual
union all
select 4 as P_IN from dual
union all
select 5 as P_IN from dual

Expected Result: Success Enter expected error number

 Process Validation 

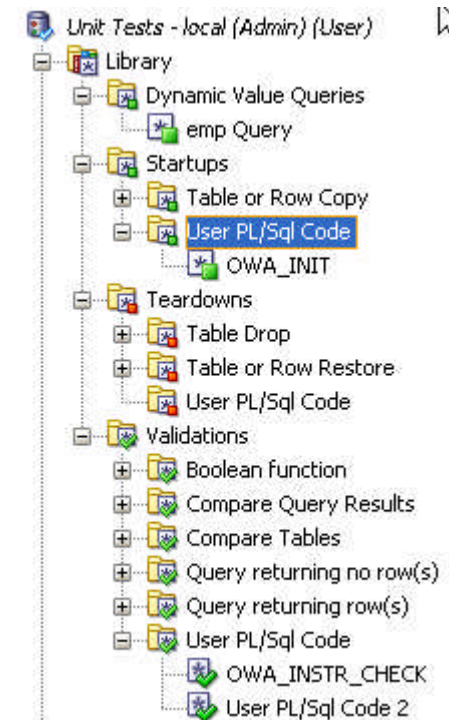
Unit Testing - Suites

- Startups/TearDown
 - Table Copy/Restore
 - Row Copy/Restore
 - Custom
- Code Coverage
- Tests are run sequentially
- Startup and teardowns for tests can be turned off



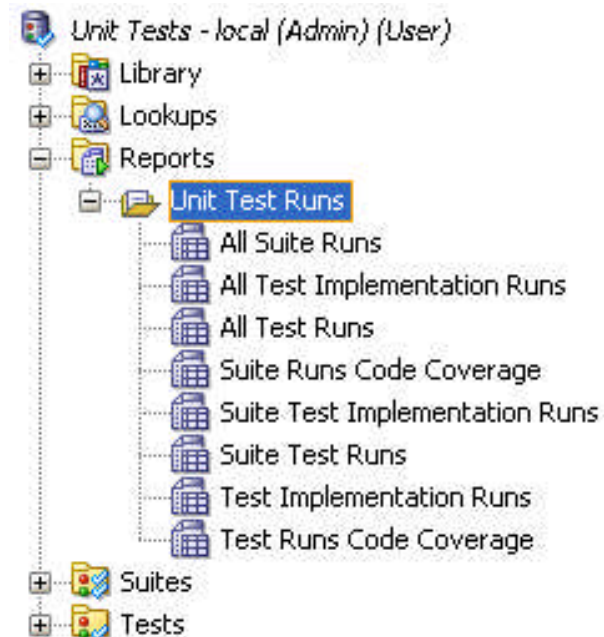
Unit Testing - Library

- Stores reusable items
 - Dynamic Values
 - Startups
 - Teardowns
 - Validations
- Referenced or Copied to local tests



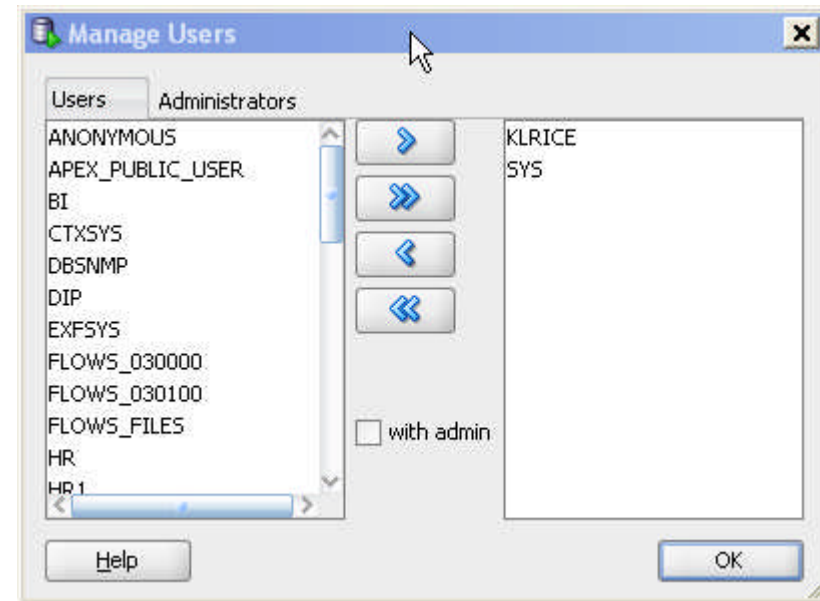
Unit Testing - Reports

- Includes Standard reports
 - Suites
 - Tests
 - Code Coverage
- Reports against the repository
- Users can query the repository directly



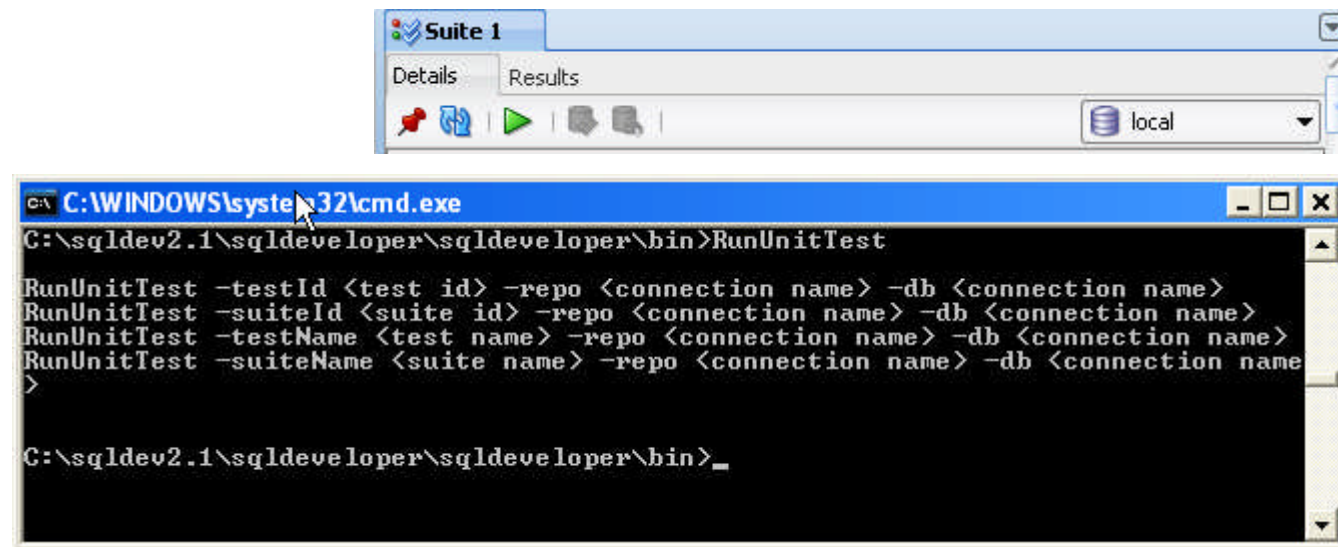
Unit Testing – Multi User

- Use database users
- Control
 - admin vs. user
- Managed with roles



Unit Testing – Running Suites/Tests

- Inside SQL Developer simply change the Combo List
- Command line by passing arguments
- Results are stored in the repository



The screenshot shows two windows. The top window is titled 'Suite 1' and has tabs for 'Details' and 'Results'. It features a toolbar with icons for adding, deleting, and running tests, and a dropdown menu currently set to 'local'. The bottom window is a Windows command prompt titled 'C:\WINDOWS\system32\cmd.exe'. It shows the command 'RunUnitTest' being executed in the directory 'G:\sqldev2.1\sqldeveloper\sqldeveloper\bin'. The command prompt displays the usage syntax for 'RunUnitTest' with placeholders for test id, suite id, test name, suite name, repository, and database connection name.

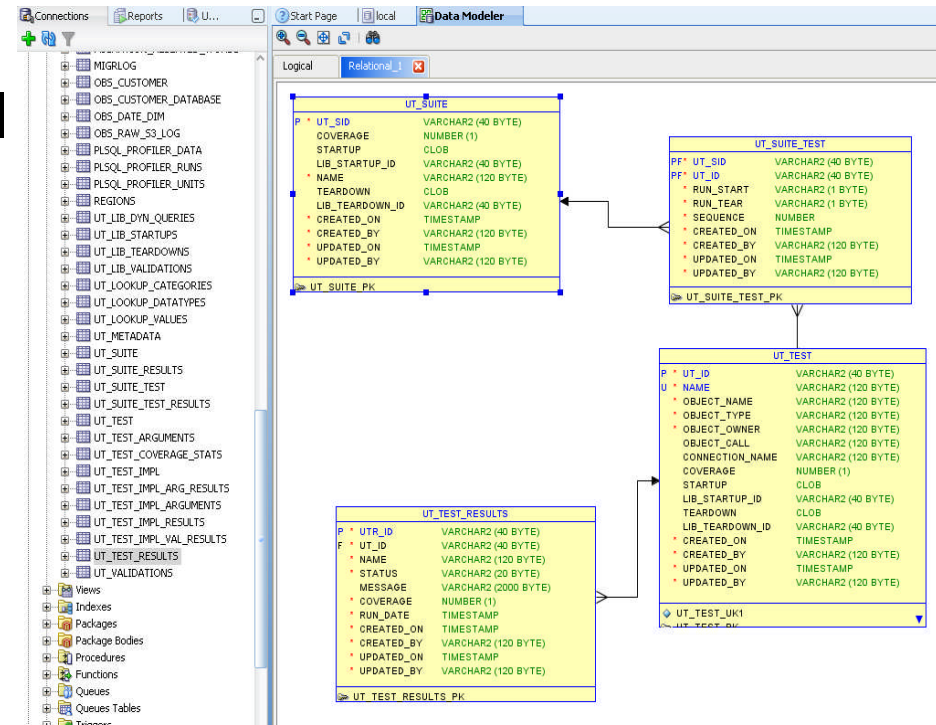
```
C:\WINDOWS\system32\cmd.exe
G:\sqldev2.1\sqldeveloper\sqldeveloper\bin>RunUnitTest

RunUnitTest -testId <test id> -repo <connection name> -db <connection name>
RunUnitTest -suiteId <suite id> -repo <connection name> -db <connection name>
RunUnitTest -testName <test name> -repo <connection name> -db <connection name>
RunUnitTest -suiteName <suite name> -repo <connection name> -db <connection name>
>

C:\sqldev2.1\sqldeveloper\sqldeveloper\bin>_
```

Integrated Data Modeling

- Open existing models
- Drag and drop relational
- Design Rules
- Inspect all properties
- Thumbnail view
- Read only



Trace Files

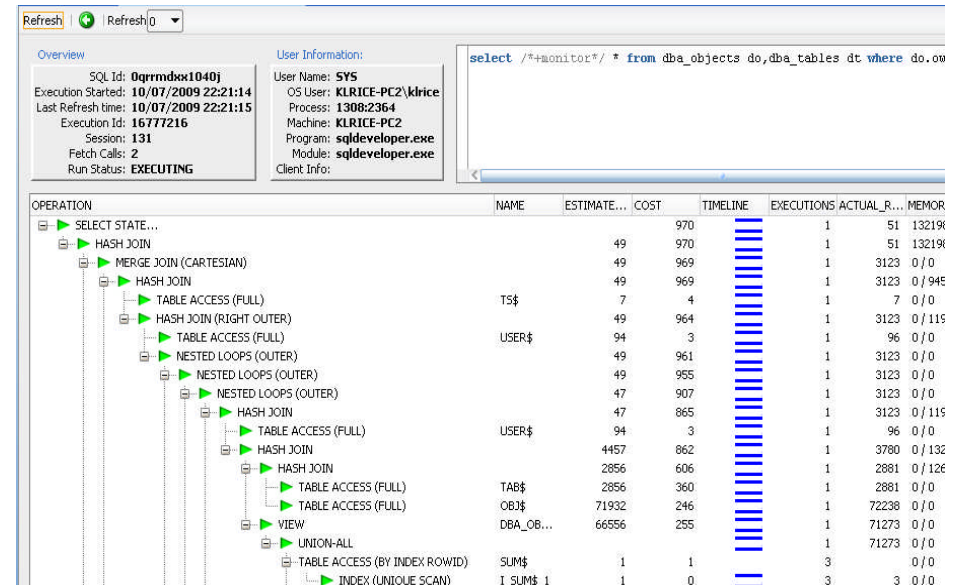
- Open trace files
- Search SQL
- Sort Options
- Filter recursive SQL
- Statistics of executions

SQL	Count	Elapsed	CPU	Phys	Cons.	Logical	Rows
ALTER SESSION SET SQL_TRACE = TRUE	1	0.04	0.00	0.00	0.00	0.00	0.00
select count(*) from dictionary	1	0.00	0.00	0.00	0.00	0.00	0.00
ALTER SESSION SET SQL_TRACE = FALSE	1	0.00	0.00	0.00	0.00	0.00	0.00

Operation	Rows
SORT AGGREGATE (cr=2300 pr=59 pw=0 time=1966214 us)	1
VIEW DICTIONARY (cr=2300 pr=59 pw=0 time=934417 us)	1821
UNION-ALL (cr=2300 pr=59 pw=0 time=930773 us)	1821
FILTER (cr=230 pr=44 pw=0 time=906680 us)	1008

Real-Time SQL Monitoring

- Real time view of SQL
- Use /*+MONITOR*/
- Drill to view details
- Visual indicators for current step
- Queries over 5 seconds monitored
- DBMS_SQLTUNE.REPORT_SQL_MONITOR





Refactoring

- Extract a procedure
- Surround blocks with
 - For
 - While
 - Begin block
- Variable renames
- Extract anonymous PL/SQL blocks from APEX apps



Finding More Detail

www.oracle.com/technology/products/database/sql_developer

- **SQL Developer on OTN**
 - White papers, Oracle by Example (OBE) and online demos
 - Team Blogs: [Blogs, Magazine Articles & Podcasts](#)
 - www.oracle.com/technology/products/database/sql_developer
- **SQL Developer Exchange**
 - Share reports, snippets, code, and add feature requests
 - <http://sqldeveloper.oracle.com>
- **Forums**
 - SQL Developer
forums.oracle.com/forums/forum.jspa?forumID=260



Summary

- Unit Testing
 - Creating, Running, Reporting
- Integrated Modeling
 - Drag and Drop Relational models
 - Trace Files
 - File -> Open
- Real-Time SQL Monitoring
 - Watch SQL as it runs
- Refactoring
 - Convert APEX anonymous blocks into a Package



For More Information

search.oracle.com

SQL Developer



or

www.oracle.com/technology/products/database/sql_developer

ORACLE®



ORACLE IS THE INFORMATION COMPANY