Oracle 10g for Developers: What You Need to Know

John Jay King
King Training Resources
john@kingtraining.com
Download this paper and code examples from:
http://www.kingtraining.com
Session Objectives

- Learn new Oracle 10g features that are geared to developers
- Be ready to use ISO/ANSI standard SQL to make code more portable
- Know how existing database features have been improved in Oracle 10g
- Become aware of some Oracle 10g DBA oriented features that will impact developers
Oracle 10g

- New and improved data types
- SQL improvements
- SQL*Plus improvements
- iSQL*Plus improvements
- Enhancements to PL/SQL
- Java and XML improvements
10g Binary_double & Binary_float

- Oracle provides support for the IEEE754 floating-point specification
- BINARY_FLOAT is a 32-bit, single-precision floating-point number stored as 5 bytes, including a length byte
- BINARY_DOUBLE is a 64-bit, double-precision floating-point number stored as 9 bytes, including a length byte
- When processing a NUMBER column, floating point numbers have decimal precision. In a BINARY_
- FLOAT or BINARY_DOUBLE column, floating-point numbers have binary precision
- Both binary floating-point numbers support the special values infinity and NaN (not a number)
- binary_double and binary_float should significantly speed some calculations

<table>
<thead>
<tr>
<th></th>
<th>Binary-Float</th>
<th>Binary-Double</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum finite</td>
<td>3.40282E+38F</td>
<td>1.79769313486231E+308</td>
</tr>
<tr>
<td>Minimum finite</td>
<td>1.17549E-38F</td>
<td>2.22507485850720E-308</td>
</tr>
</tbody>
</table>
10g Sdo_GeoRaster

- SDO_GEORASTER is an object type defined as follows:

```sql
CREATE TYPE SDO_GEORASTER AS OBJECT (  
rasterType NUMBER,  
spatialExtent SDO_GEOMETRY,  
rasterDataTable VARCHAR2(32),  
rasterID NUMBER,  
metadata XMLType);
```

- Oracle Spatial’s GeoRaster object-relational model stores a raster grid or image object as a single row - single column of object type SDO_GEORASTER
- Tables of this sort are called GeoRaster tables
  (For more information see: “Oracle Spatial User’s Guide and Reference”)
• SI_Si_StillImage is an object type that represents digital images and metadata including height, width, and format
• SI_Si_StillImage types may be created using one of three constructors:

```
SI_Si_StillImage(content)
```
```
SI_Si_StillImage(content, format)
```
```
SI_Si_StillImage(content, format, height, width)
```

(For more information see: Oracle interMedia Reference)
10g Limitless LOB

- Since first introduced, the Oracle LOB type has been limited to 4GB (enough for most uses).
- Oracle 10g allows LOB data to be limited only by tablespace page size.
- Current limit:
  - 8–128 terabytes
- Supported environments:
  - PL/SQL using DBMS_LOB
  - Java using JDBC
  - C/C++ using OCI
### New “Scalar” Functions, 1

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARDINALITY</td>
<td>Returns number of elements in nested tables</td>
</tr>
<tr>
<td>CV</td>
<td>Current value of dimension in model clause</td>
</tr>
<tr>
<td>ITERATION_NUMBER</td>
<td>Returns iteration number in model rule</td>
</tr>
<tr>
<td>LNNVL</td>
<td>Returns TRUE if condition is FALSE or UNKNOWN</td>
</tr>
<tr>
<td>NANVL</td>
<td>Return alternate value if floating-point is NAN (Not A Number)</td>
</tr>
<tr>
<td>ORA_HASH</td>
<td>Computes hash value for given expression</td>
</tr>
<tr>
<td>POWERMULTISET</td>
<td>Return nested table of non-empty sets in nested table</td>
</tr>
<tr>
<td>POWERMULTISET_BY_CARDINALITY</td>
<td>Returns POWERMULTISET for given cardinality</td>
</tr>
<tr>
<td>PRESENTNNV</td>
<td>Present Value of cell in model clause (nulls converted)</td>
</tr>
<tr>
<td>PRESENTV</td>
<td>Present Value of cell in model clause</td>
</tr>
<tr>
<td>PREVIOUS</td>
<td>Returns cell value at beginning of model clause iteration</td>
</tr>
</tbody>
</table>
New “Scalar” Functions, 2

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGEXP_INSTR</td>
<td>INSTR using regular expression syntax</td>
</tr>
<tr>
<td>REGEXP_REPLACE</td>
<td>REPLACE using regular expression</td>
</tr>
<tr>
<td>REGEXP_SUBSTR</td>
<td>SUBSTR using regular expression</td>
</tr>
<tr>
<td>REMAINDER</td>
<td>Returns remainder after expression is evaluated</td>
</tr>
<tr>
<td>SCN_TO_TIMESTAMP</td>
<td>Returns timestamp for given SCN</td>
</tr>
<tr>
<td>SESSIONTIMEZONE</td>
<td>Timezone of current session</td>
</tr>
<tr>
<td>SET</td>
<td>Converts nested table to set by removing duplicates</td>
</tr>
<tr>
<td>TIMESTAMP_TO_SCN</td>
<td>Returns SCN for given timestamp</td>
</tr>
<tr>
<td>TO_BINARY_DOUBLE</td>
<td>Return binary_double for given char, number, binary</td>
</tr>
<tr>
<td>TO_BINARY_FLOAT</td>
<td>Return binary_float for given char, number, binary</td>
</tr>
</tbody>
</table>
Oracle 10g SQL Changes

• Oracle 10g adds the following SQL statements
  – ALTER DISKGROUP
  – CREATE DISKGROUP
  – DROP DATABASE
  – DROP DISKGROUP
  – FLASHBACK DATABASE
  – FLASHBACK TABLE
  – PURGE

• Oracle 10g removes the following SQL statement
  – CREATE TEMPORARY TABLESPACE

• Oracle 10g modifies 119 SQL statements (too many to list here)
Really Cool 10g Change!!

- Oracle 10g adds two exciting changes to the ALTER SYSTEM statement
- When testing and tuning statements, the values returned by tuning tools are impacted by the actions of Oracle’s shared SQL area and buffer cache – take a look at these!!!

ALTER SYSTEM FLUSH BUFFER_CACHE
ALTER SYSTEM FLUSH SHARED_POOL
(should probably be used in test systems only)
### Statistical Group Functions, 1

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR</td>
<td>Returns the coefficient of correlation of a set of number pairs</td>
</tr>
<tr>
<td>CORR_S</td>
<td>Calculates the Spearman's rho correlation coefficient</td>
</tr>
<tr>
<td>CORR_K</td>
<td>Calculates the Kendall's tau-b correlation coefficient</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>Calculates the statistical median</td>
</tr>
<tr>
<td>STATS_BINOMIAL_TEST</td>
<td>An exact probability test</td>
</tr>
<tr>
<td>STATS_CROSSTAB</td>
<td>Method used to analyze two nominal variables</td>
</tr>
<tr>
<td>STATS_F_TEST</td>
<td>Tests whether two variances are significantly different</td>
</tr>
<tr>
<td>STATS_KS_TEST</td>
<td>Compares two samples see if they are from the same population or from populations that have the same distribution</td>
</tr>
<tr>
<td>STATS_MODE</td>
<td>Returns most frequently occurring value from a set</td>
</tr>
</tbody>
</table>
• **STATS_MW_TEST**  
  Mann Whitney test compares two independent samples to test the null hypothesis that two populations have the same distribution function against the alternative hypothesis that the two distribution functions are different.

• **STATS_ONE_WAY_ANOVA**  
  Tests differences in means (for groups or variables) for statistical significance by comparing two different variance estimates.

• **STATS_T_TEST_ONE**  
  Is a one-sample $t$-test.

• **STATS_T_TEST_PAIRED**  
  Is a two-sample, paired $t$-test (also known as a crossed $t$-test).

• **STATS_T_TEST_INDEP**  
  Is a $t$-test of two independent groups with the same variance (pooled variances).

• **STATS_T_TEST_INDEPU**  
  A $t$-test of two independent groups with unequal variance (unpooled variances).

• **STATS_WSR_TEST**  
  Test of paired samples to determine whether the median of the differences between the samples is significantly different from zero.
Example CORR Function

```sql
select country, 
corr(sale,cnt) 
from sales_view 
group by country
```
## Example Median Function

- **Given the following data:**

<table>
<thead>
<tr>
<th>ENAME</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMITH</td>
<td>800</td>
</tr>
<tr>
<td>JAMES</td>
<td>950</td>
</tr>
<tr>
<td>ADAMS</td>
<td>1100</td>
</tr>
<tr>
<td>WARD</td>
<td>1250</td>
</tr>
<tr>
<td>MARTIN</td>
<td>1250</td>
</tr>
<tr>
<td>MILLER</td>
<td>1300</td>
</tr>
<tr>
<td>TURNER</td>
<td>1500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ENAME</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLEN</td>
<td>1600</td>
</tr>
<tr>
<td>CLARK</td>
<td>2450</td>
</tr>
<tr>
<td>BLAKE</td>
<td>2850</td>
</tr>
<tr>
<td>JONES</td>
<td>2975</td>
</tr>
<tr>
<td>SCOTT</td>
<td>3000</td>
</tr>
<tr>
<td>FORD</td>
<td>3000</td>
</tr>
<tr>
<td>KING</td>
<td>5000</td>
</tr>
</tbody>
</table>

- **Note the difference between AVG and MEDIAN**

```sql
select avg(sal) average, median(sal) median
from emp;
```

<table>
<thead>
<tr>
<th>AVERAGE</th>
<th>MEDIAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2073.21429</td>
<td>1550</td>
</tr>
</tbody>
</table>
10g R1: XML Enhancements

- Oracle 10g Release 1 improved support for XML data and the XMLtype datatype including:
  - Export/Import support
  - SQL*Loader support for both structured and unstructured XMLtype data
  - DBMS_AQ support, globalization support
  - C and C++ APIs allow XML modification
  - DBMS_XMLGEN allows turning off "pretty print"
  - Hierarchical queries (CONNECT) via DBMS_XMLGEN.newContextFromHierarchy
  - Schema evolution via DBMS_XMLSCHEMA (see next page)
Schema Evolution

- Schema evolution via DBMS_XMLSCHEMA
  - Before Oracle 10g XML registered schemas could not be modified (evolved)
  - Oracle 10g schema "evolution" uses a PL/SQL procedure named CopyEvolve() which is part of the DBMS_XMLSCHEMA package
  - CopyEvolve() copies an XMLType document to temporary tables, drops and re-registers the XML schema, then copies the XMLType data into the new XMLTypes
  - CopyEvolve() has some limits: indexes, triggers, and constraints dependent upon the schemas are lost and must be recreated
  - Changing a top-level element involves additional processing detailed in the Oracle documentation
10g R2: XML Enhancements

- Oracle 10g Release 2 further enhances support for XML:
  - XMLtype is fully supported in Java, C, and C++
  - XML options support XSLT 2.0 with XPath functions and operators
  - Built-in XML support includes a JAXB compiler and XQuery support
  - XMLQUERY and XMLTABLE functions support XQuery
    - XMLQUERY builds XML data, queries XML data and relational data using XQuery
    - XMLTABLE creates relational tables and columns from the results of XQuery (COLUMN_VALUE pseudo-column used to retrieve data from XMLTABLE values)
  - XPATH processing has been improved enhancing speed
  - UpdateXML() function is joined by InsertXML(), AppendChildXML(), InsertXMLBefore(), and DeleteXML()
• 10g Release 2 further enhances support for XML (cont):
  – SOAP services are supported in C and C++
  – More SQL/XML (SQX) 2003 functions added including: XMLPI(), XMLComment(), XMLRoot(), XMLSerialize(), XMLCDATA(), and XMLParse()
  – Enterprise Manager Web Console includes the ability to work with XMLType objects
  – Oracle HTTPS server supports XML DB
  – Oracle XML Developers Kit (XDK) PL/SQL packages have been deprecated: XMLDOM(), XMPARSER(), and XSL_PROCESSOR() have been replaced by DBMS_XMLDOM(), DBMS_XMPARSER(), and DBMS_XSLPROCESSOR()
• Insert, Update, Delete, and Merge add ERROR logging allowing you to capture DML errors and log them

```
INSERT ... /* or UPDATE, DELETE, MERGE */
LOG ERRORS
[ INTO [schema.] table ]
[ (simple_expression) ]
[ REJECT LIMIT { integer | UNLIMITED }]
```

- Default error table defined by DBMS_ERRLOG package: ERR$ followed by first 25 characters of DML target table
- Simple expression is value to be used as statement tag (may be result of SQL function call)
- Reject limit default is zero
Creating the Error Log Table

• Oracle provides a PL/SQL packaged procedure to create the logging table (for each table to be logged)

execute DBMS_ERRLOG.CREATE_ERROR_LOG('myemp', 'myemplog');

  – myemp               Table DML is being applied to
  – myemplog            Logging table for rejected rows

• Creates a database table containing:
  – ORA_ERR_NUMBER$    Error number
  – ORA_ERR_MESG$      Error message
  – ORA_ERR_ROWID$     Rowid of impacted rows
  – ORA_ERR_OPTYP$     Operation type (I,U,D,M)
  – ORA_ERR_TAG$       Text from LOG_ERRORS
  – All column values (good & bad) as varchar2(4000)
Error Log Output

```
insert into emp
    select * from myempbig
log errors into myemplog ('Log test3')
    reject limit unlimited;
0 rows created.

ORA_ERR_NUMBER$  12899
ORA_ERR_MESG$ ORA-12899: value too large for column
    "JOHN"."EMP"."JOB" (actual: 13, maximum: 9)
ORA_ERR_ROWID$ I
ORA_ERR_OPTYP$ 1
ORA_ERR_TAG$ Log test3
EMPNO       6543
ENAME       STEPHENSON
JOB         WEB DEVELOPER
MGR        7369
HIREDATE    03-SEP-06
SAL         3000
COMM        
DEPTNO     40
```
10g Fast Dual

• For years, developers have used the Dual table for “quick and dirty” queries only to find during performance tuning that scans involving dual could be expensive
• In Oracle 10g the optimizer knows about dual and implements an operation called “fast dual” greatly speeding access
• Here is the “Explain” output from a simple query:

```sql
select sysdate from dual;
```

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Cost (%CPU)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td>1</td>
<td>2 (0)</td>
<td>00:00:01</td>
</tr>
<tr>
<td>1</td>
<td>FAST DUAL</td>
<td></td>
<td>1</td>
<td>2 (0)</td>
<td>00:00:01</td>
</tr>
</tbody>
</table>
• The login.sql and glogin.sql files are automatically executed upon entering SQL*Plus,

• In Oracle 10g login.sql and glogin.sql are also executed upon execution of CONNECT

• This is either a blessing or a curse, just be aware that it is happening…
10g SQL*Plus Misc.

- SET SERVEROUTPUT ON now works immediately within PL/SQL block where executed
- DBMS_OUTPUT.PUT_LINE text line maximum increased from 255 to 32767 bytes
- Recycle Bin keeps deleted database objects until Purged
- DESCRIBE automatically attempts to validate invalid objects before display
- White space now allowed in file names
- Substitution variables allowed in SET PROMPT
- Three pre-defined SQL*Plus variables added:
  - _DATE Current date or a user defined fixed string.
  - _PRIVILEGE Privilege level of connection (AS SYSDBA, AS SYSOPER or blank)
  - _USER Currently connected userid
- APPEND, CREATE, REPLACE extensions to SPOOL
• It is common for PL/SQL developers to use the DBMS_OUTPUT.PUT_LINE procedure to write to the console during testing and debugging.

• To enable output from DBMS_OUTPUT.PUT_LINE you must enable SERVEROUTPUT.

• In Oracle 10g this command has been enhanced to include a default of UNLIMITED buffer size eliminating the need to specify a buffer size.

• You may also specify “WORD_WRAPPED” to cause DBMS_OUTPUT.PUT_LINE output to be wrapped at clear word breaks.

```
set serveroutput on size 1000000 -- size limited
set serveroutput on unlimited -- size unlimited
set serveroutput on -- size unlimited (default)
```
10g Recycle Bin

- SQL*Plus now provides an “oops” capability for object drops (if database is running in “Flashback mode”)
- After issuing “DROP object xyz” the user may issue the following SQL*Plus command to see “dropped objects:

  ```sql
  SHOW RECYCLEBIN
  ```

- Then, one of these SQL statements might be executed:

  ```sql
  PURGE table xx|index yy |recyclebin|tablespace zz;
  FLASHBACK TABLE xxx TO BEFORE DROP;
  ```
10g Using Recyclebin

- Careful! Dropping tables no longer really drops them… This might be a problem for applications with lots of “temp”-type tables

```sql
drop table myTable;

show recyclebin

ORIGINAL    RECYCLEBIN NAME     TYPE     DROP TIME
myTable      RB$$41506$TABLE$0   TABLE   2004-04-01:22:11:13

flashback table myTable to before drop;

drop table myTable purge;
purge recyclebin;
```
Interesting Recyclebin Artifacts

• Dropping a table:
  – Removes indexes on the dropped table
  – Invalidates views, procedures, functions, and other objects dependent upon the table

• After using “FLASHBACK TABLE”
  – Indexes are restored with different names
  – Primary key constraints are restored using a different name
  – Foreign key constraints are not restored
  – Views, procedures, functions, and other objects dependent upon the table will work once validated (may be done manually or automatically)
iSqlPlus Changes

- Multi-page output
- Prompting for input values
<table>
<thead>
<tr>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>O/S Documentation Set - French</td>
</tr>
<tr>
<td>O/S Documentation Set - Spanish</td>
</tr>
<tr>
<td>O/S Documentation Set - Italian</td>
</tr>
<tr>
<td>O/S Documentation Set - Kanji</td>
</tr>
<tr>
<td>Standard Mouse</td>
</tr>
<tr>
<td>Deluxe Mouse</td>
</tr>
<tr>
<td>Keyboard Wrist Rest</td>
</tr>
<tr>
<td>CD-R Mini Discs</td>
</tr>
<tr>
<td>Music CD-R</td>
</tr>
<tr>
<td>CD-RW, High Speed, Pack of 10</td>
</tr>
<tr>
<td>CD-RW, High Speed Pack of 5</td>
</tr>
<tr>
<td>CD-R, Professional Grade, Pack of 10</td>
</tr>
<tr>
<td>OraMusic CD-R, Pack of 10</td>
</tr>
<tr>
<td>CD-R with Jewel Cases, Pack of 12</td>
</tr>
<tr>
<td>DVD-R Disc with Jewel Case, 4.7 GB</td>
</tr>
<tr>
<td>DVD-RAM Jewel Case, Double-Sided, 9.4 GB</td>
</tr>
<tr>
<td>DVD-R Discs, 4.7GB, Pack of 5</td>
</tr>
<tr>
<td>DVD-R Discs, 4.7GB, Pack of 5</td>
</tr>
<tr>
<td>DVD-RW Discs, 4.7GB, Pack of 3</td>
</tr>
</tbody>
</table>
Input Variables

- SQL*Plus panel below displays for this query:

```
select ename
 from emp
 where deptno = &dpt
```
10g Regular Expressions

- Oracle now has three functions that allow the use of POSIX-compliant regular expressions in SQL
  - `REGEXP_LIKE` Allows pattern matching
  - `REGEXP_INSTR` Search for string matching pattern and return position
  - `REGEXP_REPLACE` Find string matching pattern and replace it
  - `REGEXP_SUBSTR` Search for string matching pattern and return substring
Regular Expression Examples

```sql
select employee_id, phone_number
  from hr.employees
  where REGEXP_LIKE(phone_number,
    '[:digit:]{3}[:punct:][:digit:]{2}[:punct:]');
```

```sql
select first_name, last_name
  from hr.employees
  where REGEXP_LIKE(first_name, '^ste(v|ph)en$');
```
LIKE vs REGEXP_LIKE

- Here are two statement that generate exactly the same output and nearly the same execution plan

```sql
select prod_id
  , substr(prod_name,1,20) prod_name
  , substr(prod_desc,1,30) prod_desc
from sh.products
where prod_name like ('E%')
  or prod_name like ('P%')
order by prod_id;

select prod_id
  , substr(prod_name,1,20) prod_name
  , substr(prod_desc,1,30) prod_desc
from sh.products
where regexp_like (prod_name,('^E|^P'))
order by prod_id;
```
10g Model Clause

• The SQL MODEL clause is a powerful extension of the SELECT statement
• MODEL provides the ability to present the output of a SELECT in the form of multi-dimensional arrays and apply formulas to the array values
• The Model clause defines a multidimensional array by mapping the columns of a query into three groups: partitioning, dimension, and measure columns
  – Partitions define logical blocks of the result set in a way similar to the partitions of the analytical functions; each partition is viewed by the formulas as an independent array
  – Dimensions identify each measure cell within a partition; each column identifies characteristics such as date, region and product name
  – Measures are similar to the measures of a fact table in a star schema, they normally contain numeric values such as sales units or cost; each cell is accessed within its partition by specifying its full combination of dimensions
SELECT
-- rest of SELECT goes here --
MODEL [main]
[reference models]
[PARTITION BY (<cols>)]
DIMENSION BY (<cols>)
MEASURES (<cols>)
[IGNORE NAV] | [KEEP NAV]
[RULES
[UPSERT | UPDATE]
[AUTOMATIC ORDER | SEQUENTIAL ORDER]
[ITERATE (n) [UNTIL <condition>] ]
( <cell_assignment> = <expression> ... )
SELECT substr(country,1,20) country, substr(prod,1,15) prod,
year, sales FROM sales_view
WHERE country IN ('Canada', 'Germany')
MODEL RETURN UPDATED ROWS
PARTITION BY (country)
DIMENSION BY (prod, year)
MEASURES (sales sales)
sales['HulaWhirl',2002] = sales['HulaWhirl',2001],
ORDER BY country, prod, year;

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>PROD</th>
<th>YEAR</th>
<th>SALES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada</td>
<td>HulaZoop Pkg</td>
<td>2002</td>
<td>92613.16</td>
</tr>
<tr>
<td>Canada</td>
<td>ZooperT</td>
<td>2002</td>
<td>9299.08</td>
</tr>
<tr>
<td>Canada</td>
<td>HulaWhirl</td>
<td>2002</td>
<td>83314.08</td>
</tr>
<tr>
<td>Germany</td>
<td>HulaZoop Pkg</td>
<td>2002</td>
<td>103816.6</td>
</tr>
<tr>
<td>Germany</td>
<td>ZooperT</td>
<td>2002</td>
<td>11631.13</td>
</tr>
<tr>
<td>Germany</td>
<td>HulaWhirl</td>
<td>2002</td>
<td>92185.47</td>
</tr>
</tbody>
</table>
The statement on the preceding page calculates sales values for two products and defines sales for a new product based upon the other two products:

- Statement partitions data by country, so formulas are applied to one country at a time, sales fact data ends with 2001, any rules defining values for 2002 or later will insert new cells.
- The second rule defines sales for “HulaWhirl” in 2002 to be the same value they were for 2001.
- Third rule defines "HulaZoop Pkg" that is the sum of the ZooperT and HulaWhirl values for 2002 -- the rules for ZooperT and HulaWhirl must be executed before the HulaZoop Pkg rule.
10g Merge Enhanced

- MERGE now allows:
  - Specification of either update, or insert, or both
  - Deletion of rows during update

```sql
merge into bonus
  using emp
  on ( bonus.ename = emp.ename )
when matched
then  update -- only one update match allowed!
       set bonus.sal = emp.sal,
            bonus.comm = emp.comm
  delete
       where bonus.sal > 3999 -- values after
merge
  when not matched
then insert (ename,job,sal,comm)
  values
    (emp.ename,emp.job,emp.sal,emp.comm);
```
FROM Clause SEED

• For years Oracle SQL has allowed “sampling” to occur using randomized value
• Beginning with Oracle 10g you may now use a SEED to get the same (pretty close anyway) randomized value each time

```sql
select count(*) nbr_rows
  from sh.sales;  NBR_ROWS = 918843
select count(*) * 10 nbr_rows
  from sh.sales sample(10);  NBR_ROWS = 921400
select count(*) * 10 nbr_rows
  from sh.sales sample(10);  NBR_ROWS = 917400
select count(*) * 10 nbr_rows
  from sh.sales sample(10) seed(1);  NBR_ROWS = 913280
select count(*) * 10 nbr_rows
  from sh.sales sample(10) seed(1);  NBR_ROWS = 913280
```
PL/SQL Conditional Compile

- Oracle 10g R2 added the ability to conditionally include PL/SQL code at compile time
- PL/SQL conditional compilation occurs before the rest of PL/SQL compilation
- Using the “$” delimiter PL/SQL may include:
  - **Conditional directives** tell the compiler to choose between different code fragments using: $IF, $THEN, $ELSE, $END, and $ELSIF
  - **Inquiry directives** use environment values to determine whether something should be included such as: PL/SQL compile parameters and flags (two predefined: PLSQL_Unit & PLSQL_Line)
  - **Error Directives** allow a developer to force a compile error using the specified VARCHAR2 string (may be concatenation)
Oracle Application Express

- Oracle Application Express (APEX; formerly HTML DB) is a complete web development and deployment environment built into Oracle 10g
- HTML DB is based upon the home-grown software that helped make Tom Kyte's "Ask Tom" website so powerful
- HTML DB is designed to make building web applications easy without compromising flexibility when building web applications
- Pre-built components are assembled using wizards and declarative programming eliminating most need to write code
- Pre-Built components are used with wizards, to assemble applications with forms, reports, and charts without writing code; the pre-built components include: navigational controls, authentication schemes and user interface themes
• Some of the built-in features in Oracle Application Express include:
  – Page Rendering and Processing Engine Rather than generating code, Oracle HTML DB stores user interface properties and data access and logic behaviors in an application definition; when an HTML DB application is run pages are rendered in real time based upon an application definition stored in the database
  – Logic to determine how a user flows from page to page, data validation and form handlers are all built in to the processing engine
  – Deployment is automatic, immediately after an application is built or changed users can start using it
Oracle SQL Developer

- Oracle has released the free SQL Developer tool (formerly known as Project Raptor) and it is super!
- If you’ve used the Free Toad tool this tool will feel familiar plus it has more features and supports Oracle 10g too!
- SQL Developer is a Java-based GUI tool (works in Windows, Unix, and Linux)
  - Power of SQL*Plus without a local Oracle client installation
  - Intuitive software
  - Provides features often found only in expensive third-party software including:
    - GUI browsing of database objects
    - Debugging complete with breakpoints
    - Query execution
    - Database reporting
    - Multiple connections
How Do I Get It?

• SQL Developer is freely downloadable from Oracle using the following URL:
  
Should Others Worry?

- Free products should worry lots, Oracle’s SQL Developer:
  - Supports current SQL & PL/SQL
  - Provides unlimited free licenses
  - Licenses do not “die” periodically
  - Supports many features usually only available in “pay for” products: e.g. PL/SQL debugging, custom reporting, and more!

- The better pay-for products are probably safe for a while…

- This product needs a logo!
Oracle Documentation

• Oracle10g
  – Oracle10g SQL Reference
  – Oracle10g PL/SQL User’s Guide and Reference
  – Oracle10g Application Developer's Guide - Object-Relational Features
  – Oracle10g Concepts
  – Oracle XML DB Developer’s Guide

• Lots of papers and examples:
  http://technet.oracle.com
  http://tahiti.oracle.com
Wrapping it all Up

- Oracle10g adds significant new functionality to the already robust database environment
- While an emphasis is sometimes placed on the features of Oracle that support the Data Base Administrator, this paper shows many Developer-oriented features of great usefulness
Training Days 2007

Mark your calendar for:
February 14-15, 2007!
Oracle 10g for Developers: What You Need to Know

To contact the author:
John King
King Training Resources
6341 South Williams Street
Littleton, CO 80121-2627 USA
1.800.252.0652 - 1.303.798.5727
Email: john@kingtraining.com

Today’s slides and examples are on the web: http://www.kingtraining.com

Thanks for your attention!