ORACLE 10G FOR DEVELOPERS: WHAT YOU NEED TO KNOW

John Jay King, King Training Resources

Introduction

The many new features of Oracle10g extend the capabilities of the database in many ways. Attendees will be introduced to the new and improved features of Oracle that directly impact application development. Special emphasis is placed on features that can reduce development time, make development simpler, improve performance, or speed deployment. This presentation focuses on features added to Oracle10g that enable improved application development. Specific topics include: Model, Regular Expressions, DML logging, XML improvements, and expanded Java capabilities. Attendees will be better equipped to create new applications or modify existing applications to take full advantage of the Oracle database.

Features added or improved with Oracle10g Release 2 are marked in these notes to indicate that they are 10gR2-specific, all other features noted work in both Oracle10g releases. Oracle 10g Release 2 opens up new possibilities which will be examined. Examples of some new features are provided during this presentation, however, you should see the Oracle SQL Reference, Oracle PL/SQL Users Guide and Reference, and other relevant Oracle manuals for complete documentation.

Oracle provides rich tools with new features extending the capabilities of the database in many ways. This paper focuses on those improvements and additions to Oracle10g likely to have the most impact on application developers. This paper assumes a working knowledge of Oracle, SQL, and PL/SQL (from an application developer's perspective).

Developer-oriented features

Oracle10g ("g" is for "grid") allows another evolutionary leap forward in developer productivity. Developer-related features of the new release include: new and improved data types, SQL improvements, SQL*Plus/iSQL*Plus improvements, enhancements to PL/SQL, and enhancements to Java and XML interfaces.

10g New Datatypes

10g Binary_double and Binary_float

Oracle10g provides support for the IEEE754 floating-point specification via two new datatypes, BINARY_FLOAT and BINARY_DOUBLE. 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. Until Oracle 10g all numeric values stored by Oracle used NUMBER datatype. 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 and process more efficiently. Applications with lots of "number-crunching" can gain a significant speed advantage when using the new binary data types. However, should specific decimal rounding (e.g. money) be needed, you should stick with NUMBER data. Both binary floating-point numbers support special values of infinity and NaN (not a number) using the new Oracle values BINARY_FLOAT_NAN, BINARY_DOUBLE_NAN, BINARY_FLOAT_INFINITY, and BINARY_DOUBLE_INFINITY. The chart below illustrates size limitations:

	Binary-Float	Binary-Double	Number
Maximum positive value	3.40282E+38F	1.79769313486231E+308	< 1.0e126
Minimum positive value	1.17549E-38F	2.22507485850720E-308	1.0e-13
Precision	6 - 9	15 - 17	38 - 40

The new floating-point data is supported by SQL, PL/SQL, OCI/OCCI, Pro*C/C++, and JDBC. Functions added or modified to work with the new data types directly include:

- TO_BINARY_DOUBLE converts float or decimal values to double
- TO_BINARY_FLOAT converts double or decimal values to float
- TO_CHAR converts float or double to decimal-formatted string
- TO_NUMBER converts a float, double, or string to a number

10g SDO_GEORASTER

Oracle10g adds a new spatial SDO_GEORASTER object type. Oracle Spatial was created to ease management of spatial data when working with location-enabled applications, geographic information system (GIS) applications, and geoimaging applications. These types of applications are common in the oil and mining industry. Oracle allows spatial data to be stored in the database and retrieved like any other type of data. Oracle Spatial must be installed specifically; it is not part of the default Oracle installation. Spatial data includes the two basic datatypes: SDO_GEOMETRY and SDO_TOPO_GEOMETRY (10gR2) which describe a user-defined table in a single column object-type. SDO_GEOMETRY is used to hold the geometric description of a spatial object. SDO_TOPO_GEOMETRY object describe a topology geometry. GeoRaster objects store a raster grid or image object as a single row in a single column using the new datatype SDO_GEORASTER in a user-defined table. SDO_GEORASTER is defined as follows:

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

SI_StillImage

SI_StillImage is an object type that represents digital images and metadata including height, width, format, and more. The SI_StillImage object type is created in the ORDSYS schema with invoker rights and it is declared as INSTANTIABLE and NOT FINAL. Oracle suggests using the SI_StillImage constructors and methods rather than accessing attributes directly to protect yourself from changes to the internal representation of the SI_StillImage object.

10g "Limitless" LOB

Since first introduced the Oracle LOB type (BLOB, CLOB, BFILE) has been limited to 4GB (enough for most of us). Oracle 10g allows LOB data to be limited only by tablespace page size. This means that with 10g the current limit is between 8 and 128 terabytes depending upon your platform. These new, larger LOBs are supported in PL/SQL (using DBMS_LOB), Java via JDBC, and C/C++ (OCI interface).

10g New Statistical Functions

Oracle10g Release 1 added a new set of statistical functions to augment those already in the database. The new functions include:

- CORR returns the coefficient of correlation of a set of number pairs
- CORR_S calculates the Spearman's rho correlation coefficient
- CORR_K calculates the Kendall's tau-b correlation coefficient
- MEDIAN calculates the statistical median
- STATS_BINOMIAL_TEST is an exact probability test
- STATS_CROSSTAB method is used to analyze two nominal variables
- STATS_F_TEST tests whether two variances are significantly different
- STATS_KS_TEST compares two samples to see if they are from the same population or from populations having the same distribution
- STATS_MODE returns the most frequently occurring value from a set

- STATS_MW_TEST - This 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 estimates of variance
- 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 is a t-test of two independent groups with unequal variance (unpooled variances)
- STATS_WSR_TEST tests paired samples to determine whether the median of the differences between the samples is significantly different from zero

Example CORR Function

```
select country,
  corr(sale,cnt)
 from sales_view
 group by country
```

10g Release 2 Data Mining Functions

Oracle10g Release 2 added a new set of data mining functions to use with clustering models created using the DBMS_DATA_MINING package or the Oracle Data Mining Java API. The new functions include:

- CLUSTER_ID returns an Oracle NUMBER containing the cluster identifier of the predicted cluster with the highest probability for predictors specified by the mining_attribute_clause
- CLUSTER_PROBABILITY returns the degree of confidence of membership of an input row in a cluster associated with the specified model
- CLUSTER_SET returns a varray of objects containing all possible clusters that a given row belongs to; each object contains cluster ID and the cluster probability
- FEATURE_ID returns an Oracle NUMBER that is the identifier of the feature with the highest coefficient value
- FEATURE_SET returns a varray of objects containing all possible features; each object in the varray contains the feature ID and the feature value
- FEATURE_VALUE returns the value of a given feature or the highest feature value if no argument is specified
- PREDICTION returns the best prediction for the model depending upon the target value type
- PREDICTION_COST returns a cost measure for a specific prediction
- PREDICTION_DETAILS returns an XML string containing model-specific information related to the scoring of the input row
- PREDICTION_PROBABILITY returns the probability for a given prediction
- PREDICTION_SET returns a varray of objects; each object contains PREDICTION, PROBABILITY, and COST

For more information on the new Data Mining function see the Oracle Data Mining manuals.

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 the Dual table and implements an operation called "fast dual" significantly reducing the logical I/O used to satisfy a query and greatly speeding access.

10g Release 2 DML Error Logging

Oracle 10g Release two implements a new error logging clause for DML statements (INSERT, UPDATE, DELETE, and MERGE). The new error logging clause allows a DML statement to continue processing up to a specified number of errors, logging the errors and placing error rows into an error logging table. The error logging table may be specified directly or defaulted. The default name is "ERR$_" followed by the first 25 characters of the table the DML is directed to. Error logging logs the values of columns involved in the error row (except for LOBs, LONGs, and object types). The DML error logging syntax allows specification of a statement tag to allow differentiation of rows in the error logging table.

The error logging table may be created using a PL/SQL packaged procedure as follows:

```
   EXECUTE DBMS_ERRLOG.CREATE_ERROR_LOG('myemp', 'errlog');
```

Here is an example INSERT using error logging:

```
   INSERT INTO myEMP
      SELECT EMPNO,ENAME,JOB,HIREDATE,SAL
          FROM EMP
          LOG ERRORS INTO errlog ('oops') reject limit 5;
```

In the example above: 'errlog' is the name of the error logging table, 'oops' is the text to appear in error log for rejected rows (could also be a function call), '5' is the maximum number of rejected rows to allow before the DML statement should fail anyway. A row from the error log table might look something like this:

```
   ORA_ERR_NUMBER$     ORA_ERR_MESG$
   1438                ORA-01438: value larger than specified precision allowed for this column
        ORA_ER  ORA_ERR_TAG$   EMPNO
        I       oops           7499
            ENAME   JOB        MGR            HIREDATE       SAL            COMM    DEPTNO
            ALLEN   SALESMAN   7698           20-FEB-81      1600           300     30
```

Some conditions will cause a statement to fail and rollback without error logging including: direct-path INSERT or MERGE violating a unique constraint or index violation, or violation of deferred constraints.

10g Connect login.sql/glogin.sql

As before, in Oracle 10g 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 Miscellaneous new Features

SQL*Plus has been improved in several ways to make it more useful to developers including:

- SET SERVEROUTPUT ON now works immediately within the block where it is executed
- Recycle Bin keeps deleted database objects until PURGEd (see below)
- DESCRIBE now validates any invalid object before display, previously a DESCRIBE on an invalid object caused a failure
- White space is now allowed in file names
- APPEND extension to SPOOL allows data to be APPENDED to an existing SPOOL file
  ```
  SPOOL myspoolfile.txt append
  ```

10g Recycle Bin

SQL*Plus now provides an “oops” capability for object drops if the database is running in “Flashback mode” allowing a DROP TABLE (or other drop) to be undone. After issuing “DROP object xyz” the user may issue the following SQL*Plus command to see “dropped objects:

```
   SHOW RECYCLEBIN
```

Then, one of these SQL statements might be executed to remove the table permanently or "flash back" to the version before the drop as shown by the two examples below:

```
   PURGE table xx|index yy |recyclebin|tablespace zz;


   FLASHBACK TABLE xxx TO BEFORE DROP;
```

Now that 10g uses the recycle bin be careful! Dropping tables no longer really drops them. This might be a problem for applications with lots of “temp”-type tables where successive CREATE TABLE - DROP TABLE statements occur (Temporary Tables function as before).

Recycle Bin sequence of events

Drop a table as before:

```
drop table myTable;
```

To view the recyclebin's contents:

```
show recyclebin

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

To restore the table:

```
flashback table myTable to before drop;
```

To drop a table and avoid the recyclebin:

```
drop table myTable purge;
```

To "empty" the recyclebin:

```
purge recyclebin;
```

iSql*Plus Changes

With Oracle 10g, iSQL*Plus has been improved to include multi-page output and now supports prompting for input values.

10g Regular Expressions

Oracle now has 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

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

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

10g Model Clause

Forecasting is becoming a frequent need in many organizations. The SQL MODEL clause is a powerful extension of the SELECT statement providing the ability to perform spreadsheet-like processing 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.

10g Model Syntax

```
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> ... )
```

10g Model Example

```
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 (sale sales)
     RULES (
         sales['Zoop Tube', 2002] = sales['Zoop Tube', 2001] +
                      sales['Zoop Tube', 2000],
         sales['Hula Twirl', 2002] = sales['Hula Twirl', 2001],
         sales['HulaZoop Pkg', 2002] = sales['Zoop Tube', 2002] +
                         sales['Hula Twirl', 2002])
 ORDER BY country, prod, year;

COUNTRY PROD YEAR SALES
------------------- -------------- --------- ----------
Canada HulaZoop Pkg 2002 92613.16
Canada Zoop Tube 2002 9299.08
Canada Hula Twirl 2002 83314.08
Germany HulaZoop Pkg 2002 103816.6
Germany Zoop Tube 2002 11631.13Germany Hula Twirl 2002 92185.47
```

10g Model Example Explained

The preceding SQL statement calculates sales values for two products and defines sales for a new product based upon the other two products. The 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 first rule defines sales of the "Zoop Tube" game in 2002 as the sum of its sales in 2000 and 2001. The second rule defines sales for "Hula Twirl" in 2002 to be the same value they were for 2001. The third rule defines "HulaZoop Pkg" that is the sum of the Zoop Tube and Hula Twirl values for 2002. The rules for Zoop Tube and Hula Twirl must be executed before the HulaZoop Pkg rule. The Model clause allows forecasting of future values by applying algorithms to existing data.

10g Merge Enhanced

MERGE now allows specification of either update, or insert, or both. Deletion is now allowed during update.

10g Application Express (formerly HTML DB)

Oracle Application Express (formerly HTML DB) is a complete web development and deployment environment built into Oracle 10g. Application Express is based upon the home-grown software that helped make Tom Kyte's "Ask Tom" website so powerful. Application Express 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. Some of the built-in features include Page Rendering and a Processing Engine. Rather than generating code, Oracle Application Express stores user interface properties and data access and logic behaviors in an application definition; when an Application Express 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. 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.

Oracle Documentation

Thorough documentation may be found for Oracle's features in the following on-line references (go to http://tahiti.oracle.com or http://technet.oracle.com if you do not have the disks).

1. Oracle SQL Reference
2. Oracle PL/SQL Users Guide and Reference
3. Oracle Java Developer's Guide
4. Oracle Java Tools Reference
5. Oracle JDBC Developer's Guide and Reference
6. Oracle Supplied Java Packages Reference
7. Lots of papers and examples: at http://technet.oracle.com

Wrapping it all Up

Oracle10g continues the tradition of fine products and significant enhancements we have come to expect from Oracle. The many new features that are geared to the developer will save time, provide enhanced performance, and allow us to create richer applications. The DBA-oriented features of the new release will make performance and reliability easier to guarantee for our users.

About the Author

John King is a Partner in King Training Resources, a firm providing instructor-led training since 1988 across the United States and Internationally. John specializes in application development software on a variety of platforms including Web, Unix, Linux, IBM mainframe, and personal computers. John has worked with Oracle products and the database since Version 4 and has been providing training to Oracle application developers since Oracle Version 5. John develops and presents customized courses in a variety of topics including SOA, Web Services, Oracle, DB2, UDB, Java, XML, C#, and various programming languages. He has presented papers at many industry events including: IOUG-A Live!, UKOUG Conferences, EOUG Conferences, AUSOUG Conferences, RMOUG Training Days, OOUG, TOUG, MAOP-AOTC, NYOUG, and the ODTUG conference.

John Jay King
King Training Resources
6341 South Williams Street
Littleton, CO 80121-2627
U.S.A.

Phone: 1.303.798.5727 1.800.252.0652 (within the U.S.)
Fax: 1.303.730.8542
Email: john@kingtraining.com
Website: www.kingtraining.com