Oracle Data Dictionary

Objectives

These notes introduce concepts about the Oracle data dictionary.

- Learn to use data dictionary views and learn how they are
- created. Write queries for the data dictionary and dynamic performance views.

Database Objects

One of the steps completed when creating a database is the execution of the catalog.sql and catproc.sql scripts.

- These scripts create and populate the Data Dictionary with a number of database objects.
- The Data Dictionary stores metadata; that is, data about data.
- Other objects created by these scripts include the Dynamic Performance Tables that enable a DBA to monitor and tune an Oracle Database Instance.
- PL/SQL packages are also created as well as Database event triggers. These latter two sets of objects are not covered in this course.

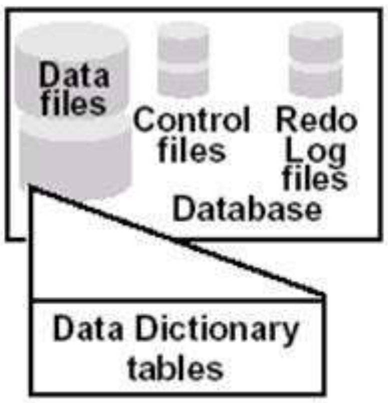
Data Dictionary

The figure below details the important characteristics of the Data Dictionary.

- Typically, system users are not allowed to access Data Dictionary tables and views.
- However, as DBAs, you are authorized access to the data dictionary for the database in order for you to learn more about the data dictionary.

Data Dictionary

- Central to every Oracle database
- Describes the database and its objects
- Contains read-only tables and views
- Stored in the SYSTEM tablespace
- Owned by the user SYS
- Maintained by the Oracle server
- Accessed with SELECT



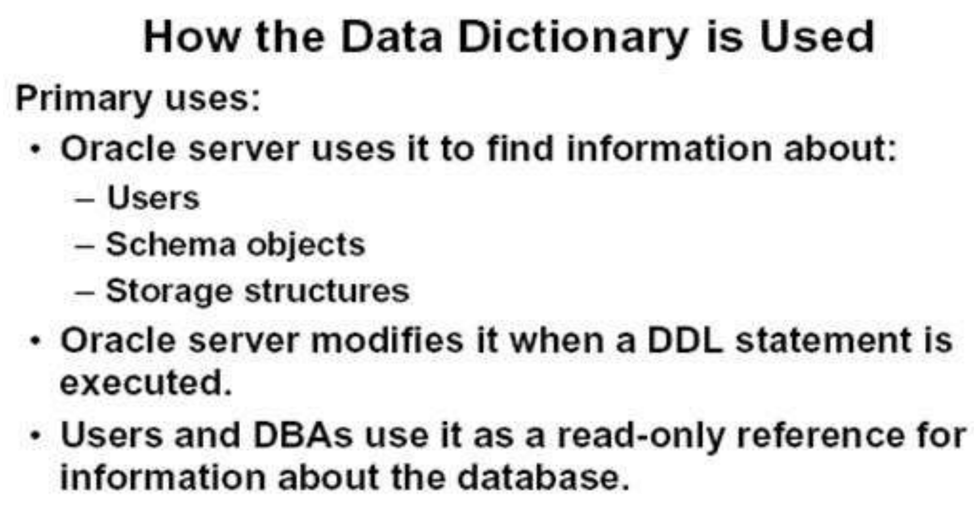
The Data Dictionary consists of two components:

- Base Tables: These tables store descriptions of objects in the database.
  - These are the first objects created in the data dictionary.
  - These are created when the oracle RDBMS software runs a special script named sql.bsq when a database is created by the CREATE DATABASE command – you do not see the sql.bsq script execute, but it does.
  - You should never attempt to write to these tables – never use DML commands to attempt to update base tables directly.
  - An example Base Table is IND$ that stores index information.
- User-Accessible Views: These views summarize information in the base tables in order to make the information easier for a DBA to use.
  - These views are created by the catalog.sql script.
  - When using the Oracle Universal Installer to create a database, then the catalog.sql and catproc.sql scripts are run automatically.
  - An example data dictionary user-accessible view is TABS – it stores information about tables you create as a system user. TABS is a synonym for the view ALL_TABLES.

The Data Dictionary stores information about all database objects created by system users and information systems professionals including tables, views, indexes, clusters, procedures, functions, synonyms, sequences, triggers and the like. For each object, the Data Dictionary stores:

- Disk space allocation (usually in bytes).
- Integrity constraint information.
- Default values for columns.
- Oracle user names (accounts)
- Privilege and role information for users.
- Auditing information – who has accessed/updated objects.

Data Dictionary Usage



How the Data Dictionary is Used

Primary uses:

- **Oracle server uses it to find information about:**
  - **Users**
  - **Schema objects**
  - **Storage structures**
- **Oracle server modifies it when a DDL statement is executed.**
- **Users and DBAs use it as a read-only reference for information about the database.**

The Data Dictionary Views are divided into three sets of views. These are differentiated by the scope of the information that they present.

- DBA: These views display information about **objects stored in all schemas** (a schema is a logical organization of objects belong to an individual system user).
  - These views are named DBA_xxx where xxx is the object name.
  - Because these views belong to the DBA and were created by the owner SYS, you may (or may not) need to reference those that do not have public synonyms by qualifying the object name with the owner. Example:

```
SELECT owner, object_name,
object_type
FROM SYS.DBA_OBJECTS;
```

- Other common DBA views:
  - DBA_TABLES contains one row for every table and includes information such as owner, tablespace_name, number of rows (NUM_ROWS) and average row length (AVG_ROW_LEN).
  - DBA_TAB_COLUMNS contains one row for every column of every table. It contains useful information such as data type, length, precision and scale, and average length.
  - DBA_CONSTRAINTS contains one row for every constraint created in the database, i.e. type of constraint for PK, check and not null constraints, etc.) and status of the constraint.
    - Useful to find all references to an Oracle table!
  - DBA_CONS_COLUMNS contains the column names the constraint is built on.

- ALL: These views display "all" information that an individual user of the database is If authorized to access – this will include information about your objects as well as information about objects for which you have access permissions.
  - If you connect to the database, the ALL_xxx views will display all about objects of all database schemas (if you have access permissions).
  - These views also provide information about objects to which you have access by virtue of the assignment of either public or explicit grants of access privileges.

```
SELECT owner, object_name,
object_type
FROM ALL_OBJECTS;
```

  - The ALL_ views obey the current set of enabled roles. Query results depend on which roles are enabled, as shown in the following example:

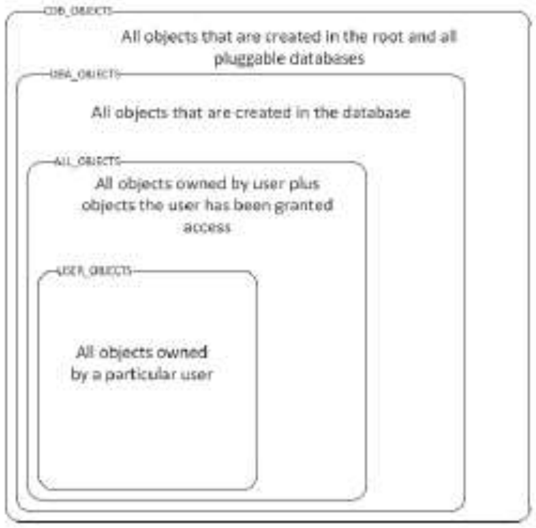
```
SQL> SET ROLE ALL;

Role set.


SQL> SELECT COUNT(*) FROM ALL OBJECTS;
```

```
COUNT(*)
----------
13140

SQL> SET ROLE NONE;
Role set.

SQL> SELECT COUNT(*) FROM ALL_OBJECTS;

COUNT(*)
----------
12941
```

- USER: These views display information that you would most likely want to access.
  - These USER_xxx views refer to your own objects in your own schema.
  - These views display only rows pertinent to you as a user.
  - These views are a subset of the ALL views.
  - These rows do not usually display the OWNER column.

```
SELECT object_name, object_type
FROM USER_OBJECTS;
```

A comparison of data dictionary views


CDB_OBJECTS
All objects that are created in the root and all pluggable databases
DBA_OBJECTS
All objects that are created in the database
ALL_OBJECTS
All objects owned by user plus objects the user has been granted access
USER_OBJECTS
All objects owned by a particular user

In general, Data Dictionary Views answer questions about:

- when an object was created,
- Is the object part of another object,
- who owns the object,
- what privileges do you have as a system user,
- what restrictions are on an object.

Practice selecting information from the following three views: dba_objects, all_objects, user_objects. You may wish to use the SQL*Plus command DESC to describe the views so you know what columns to query.

Additional queries you may wish to execute to practice accessing parts of the data dictionary:

```
SELECT table_name, comments
FROM dictionary WHERE
table_name LIKE '%INDEXES%';

SELECT table_name, comments
FROM dictionary WHERE
table_name LIKE 'DBA_SEG%';

DESC dba_users;

COLUMN account_status FORMAT A20; SELECT
username, account_status, lock_date
FROM dba_users;
```

The DUAL Table

Oracle maintains a table named DUAL that is used by Oracle and by user programs to produce a guaranteed known result such as the production of a value through use of an Oracle defined function.

- The table has one column named DUMMY and one row containing the value X.
- SYS owns the DUAL table, but **all users** can select from it.

- Selecting from DUAL is useful for computing a constant expression with a SELECT statement, since DUAL has only one row, the constant is returned only once.

Example: Suppose you want to know the ASCII equivalent of the ASCII value 76? The following SELECT statement fails to execute and generates an error message.

```
SQL> SELECT ASCII(76);

SELECT ASCII(76)
* ERROR at line 1: ORA-00923: FROM keyword not found where
expected
```

You can select the value from DUAL and the SELECT statement succeeds.

```
SQL> SELECT ASCII(76) FROM dual;

ASCII(76)
---------
       55
```

Dynamic Performance Tables and Views

The Oracle Server records database activity to the Dynamic Performance Tables (X$xxx). There are more than 700!

- These are complemented by Dynamic Performance Views (V_$xxx) - virtual tables that exist only in memory when the database is running.
- The tables provide real-time condition information about database operation.
- DBAs use these tables—most users should not be able to access these tables.
- The tables cannot be altered – they are fixed and are owned by the user SYS.
- All Dynamic Performance Tables have names that begin with the letters V_$.
- Views of these tables are created along with public synonyms that begin with the letters

  V$. Did you notice the difference in these names – yes, that's right, the underscore?

What information is stored in the following tables: V$DATAFILE, V$FIXED_TABLE? – Answer: Information about the database's datafiles and information about all of the dynamic performance tables and views.

Examples Dynamic Performance Table views:

- V$CONTROLFILE: Lists the names of the control files
- V$DATABASE: Contains database information from the control file.
- V$DATAFILE: Contains datafile information from the control file
- V$INSTANCE: Displays the state of the current instance
- V$PARAMETER: Lists parameters and values currently in effect for the session
- V$SESSION: Lists session information for each current session
- V$SGA: Contains summary information on the system global area (SGA)
- V$SPPARAMETER: Lists the contents of the SPFILE
- V$TABLESPACE: Displays tablespace information from the control file
  V$THREAD: Contains thread information from the control file
  V$VERSION: Version numbers of core library components in the Oracle server

Note: Refer to the "Oracle12c Database Reference" document for a complete list of dynamic performance views and their columns.

How the Data Dictionary Is Used

Primary uses:

- Oracle (internally) accesses information about users, schema objects, and storage structures.
  - This is done to validate a query executed by a system user.
  - Validates permission and security.
  - Verifies that referenced objects in queries actually exist.
- Oracle modifies the data dictionary for each DDL statement executed.
- Oracle users access the data dictionary as a read-only reference.

Modifying the Data Dictionary:

- Only Oracle (SYS) should ever modify the data dictionary.
- During upgrades to new release versions of the Oracle RDBMS, scripts are provided to upgrade the data dictionary.

- Most (not all) data dictionary views have public synonyms to enable users to access information conveniently. System users must avoid creating public synonyms that conflict with existing public synonyms.
- Oracle software products can reference the data dictionary and add tables and views the product needs to function to the data dictionary.

Fast Data Dictionary Access:
- A lot of data dictionary information is cached in the Dictionary Cache through use of a least recently used (LRU) algorithm.
- Comments columns from the data dictionary are not usually cached.

Administrative Scripts

SQL*Plus scripts provided with Oracle that extend beyond the catalog.sql and catproc.sql scripts. There are four categories of administrative scripts.

cat*.sql: In addition to catalog.sql and catproc.sql, there are other scripts to create information used by Oracle utilities.
- catadt.sql creates data dictionary views to display metadata for types and other objects in the ORDBMS (Object Relational DBMS).
- The catnoadt.sql script drops these tables and views.

dbms*.sql and pvt*.plb: These scripts create objects for predefined Oracle packages that can extend the Oracle server functionality.
- dbmspool.sql is a script that enables a DBA to display the sizes of objects in the shared pool and mark them for retention/removal in the SGA to reduce shared pool fragmentation.

utl*.sql: These scripts are run to add views and tables for database utilities.
- utlxplan.sql creates a table that can be used to view the execution plan for a SQL statement.

END OF NOTES