Lesson 1
A DBAs Introduction to the Oracle Data Dictionary
Lesson Objectives

- Answer the question, “what is the data dictionary?”
- Understand the types of data dictionary tables and views available to the DBA
- Gain hands-on experience querying the data dictionary
  - Primarily DBA_ and “V$” views
- Prepare you to use the data dictionary views later in the course
- Show you where to get more information

In this lesson you will learn what the Oracle data dictionary is and what types of tables and views make up the dictionary. You will also do a hands-on workshop, which will give you confidence you will need to query the data dictionary in the subsequent lessons of this course and in practical situations at your workplace.
Introduction

- All successful DBAs must have a solid command of the data dictionary
- Must know how to query
- Answer to many questions are in the dictionary

```
1  SELECT owner, table_name
2  FROM dba_tables
3  WHERE tablespace_name='USERS'

system@CLASS2> /

OWNER                          TABLE_NAME
------------------------------  -----------
SCOTT                          DEPT
SCOTT                          EMP

```

So many of the questions you will be faced with as an administrator will be answered by querying the data dictionary. For example, a simple question such as “what tables are in the USERS tablespace?” is answered with the query:

```
SELECT owner, table_name
FROM dba_tables
WHERE tablespace_name='USERS'
```

Thus, it is imperative that we have a solid command of the data dictionary.

(If you are not sure what a tablespace is, don’t worry, you will learn all about tablespaces later in this course. However, if you are not sure what a table is, before you continue with this course attend an Introduction to Oracle class or read Part II of Oracle9i A Beginner’s Guide by Michael Abbey. ISBN 0-07-219279-8.)
The Oracle Data Dictionary is a set of tables and views containing information about the database. This information is stored inside the database, so the database contains information about itself, often called “metadata.”

There are hundreds of data dictionary tables and views. The views are built on underlying base tables such as SYS.TAB$. However, the underlying tables such as SYS.TAB$ are difficult to use, in part because they are not documented and in part because the column names are not intuitive. (For fun, execute the SQL*Plus DESCRIBE command on SYS.TAB$. I think you will agree that the column names are not intuitive.)

Rather than use the SYS tables, we will use the more convenient — and documented — data dictionary views.
...What is the Dictionary?

- User-friendly views are available
  - e.g. USER_TABLES
    - One row for every table in your schema
- More than 1200 data dictionary views in 9i
  - Static views, e.g. USER_TABLES
  - Dynamic views, e.g. V$PARAMETER
  - Global dynamic views, e.g. GV$PARAMETER
  - Undocumented views, e.g. X$KSPPI
- See Oracle9i Database Reference

The static and dynamic views built on the SYS tables are user-friendly. “USER_TABLES”, for example, contains one row for every table owned by you (i.e. created in your schema).

In this lesson, you will learn about static views and dynamic views. Other types of dictionary views are also available:

- Global dynamic views are used in Oracle Parallel Server environments (called Real Applications Clusters in Oracle9i).
- The “X$” views are undocumented views, and are typically used by the database itself or Oracle support.

Refer to the Oracle9i Database Reference for descriptions of all documented dictionary views.

Optional Mini-Workshop

To become familiar with USER_TABLES, and the data dictionary in general, execute the following two commands from a SQL*Plus session while connected with your DBA userid:

```
desc user_tables
select table_name, num_rows from user_tables;
```
Static Views

- Static views
  - Contain data about objects, privileges, users, etc.
  - Contents change with DDL operations
- Three types
  - USER_
    - Displays only information about current user objects
    - e.g. USER_TABLES
  - ALL_
    - Displays info about users objects and objects privileged to use
    - e.g. ALL_TABLES
  - DBA_
    - Displays all objects, etc.
    - e.g. DBA_TABLES

The static data dictionary views contain information about database objects such as tables, indexes and procedures. They are called “static” because the contents of the views change only when DDL operations are executed. For example, a row is added to the DBA_TABLES view when any user successfully executes a CREATE TABLE statement. (There are also dynamic views, which we will discuss later in this lesson.)

There are three types of static views, USER, ALL and DBA. Every user can query the USER level views, which contain information about the current user’s objects. Every user can also query the ALL-level views, which contain information about the users objects and objects in other schemas that the user has some privilege on.

Supplemental Information

The DBA-level views can only be accessed by users who have been granted the SELECT_CATALOG_ROLE role, the SELECT ANY DICTIONARY privilege or SELECT privilege on specific views. The SELECT ANY DICTIONARY privilege is, by default, granted to the DBA role. You will study more about privileges and roles later in this course in the lesson Managing Security.
Here is a short-list of commonly used static data dictionary views. Do not try to remember every detail about these views now; you will use these views in subsequent lessons in this course. Here’s a brief description of some commonly used dictionary 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). NUM_ROWS and AVG_ROW_LEN are statistics and must be created and maintained with DBMS_STATS (formerly ANALYZE).

- **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. DBA_CONSTRAINTS contains details such as the type of (“P” for primary key, “C” for check and not null constraints, etc.) and status of the constraint.

- **DBA_CONS_COLUMNS** contains the column names the constraint is built on.

Notes for this slide continue on the next page…
Commonly used data dictionary views, continued from the previous page:

- **DBA_INDEXES** contains one row for every index.
- **DBA_TABLESPACES** contains one row for every tablespace.
- **DBA_DATA_FILES** contains one row for every data file.
- **DBA_OBJECTS** contains one row for every object that has been created, including tables, views, indexes, procedures, functions, etc.
- **DBA_SEGMENTS** contains information relating to the initial, next, percent increase, number of extents, and the number of blocks and bytes for the segment.
- **DBA_EXTENTS** contains information relating to each extent within a segment, specifically data file that contains the extent, the starting block number and length of the extent in blocks and bytes.
- **DBA_FREE_SPACE** contains information relating to the free space within the database, specifically, the data file, starting block number and size in blocks and bytes of every area of contiguous free space within the database.
- **DBA_VIEWS** contains a row for every view created in the database and includes the view text (definition).
- **DBA_SYNONYMS** contains a row for every synonym and contains the synonym name and the owner and table the synonym points to.
- **DBA_SYS_PRIVS** contains a row for every system privilege granted (e.g. GRANT CREATE TABLE) to a user or role.
- **DBA_TAB_PRIVS** contains a row for every grant on every table. It contains grantor, grantee and the privilege granted, among other things.
- **DBA_TRIGGERS** contains one row for every trigger defined. It contains the type, event, and body (code) among other things.
The names, account status, creation date and other information about Oracle user accounts on the database can be obtained by querying the DBA_USERS data dictionary view.

The ACCOUNT_STATUS column indicates the following:

- **OPEN** – The username can be used, i.e. you can logon to the database with the username if you know the password.

- **EXPIRED** – The password has expired, either manually with the ALTER USER PASSWORD EXPIRE command, or automatically by exceeding the lifetime of the password (see the PASSWORD_LIFE_TIME parameter on the CREATE PROFILE command.) The user will be prompted for the existing password and a new password upon login.

- **LOCKED** – The username has been locked either manually with the ALTER USER ACCOUNT LOCK command or by exceeding the login attempt limit (see the FAILED_LOGIN_ATTEMPTS parameter of the CREATE PROFILE command).

**Supplemental Notes**

Additional details about user accounts can be accessed from different data dictionary views. For example, DBA_TS_QUOTAS contains information about tablespace quotas (i.e. the amount of disk space a user can use).
This example illustrates how to determine what system-level privileges you have been granted. Specifically, the query finds the system-level privileges held by the user ‘DAVE’ by querying the DBA_SYS_PRIVS view and applying the filter WHERE GRANTEE = ‘DAVE’. It also shows that user DAVE is not an administrator of the system privilege, meaning that this user cannot grant this privilege to other users.

As shown in this example, always use upper-case character strings when searching the data dictionary. All character strings in the dictionary are stored in upper case.

Supplemental Notes

Don’t worry, you will learn all about privileges later in this course in the lesson Managing Security. For now, here is a brief primer on privileges:

- System-level privileges are privileges such as CREATE TABLE and CREATE PROCEDURE; they are not tied to one specific object.

- Object-level privileges on the other hand, are tied to a specific object. For example GRANT SELECT ON CUSTOMER. Query the DBA_TAB_PRIVS to see object privileges granted on tables.
Dynamic performance views are updated at startup time and usually throughout the life of the instance. The names of these views always start with “V$” (There are also like-named views whose names start with the characters “GV$”. These are used for the global, Real Application Clusters version of the view).

The following is a brief description of some common dynamic views. We will use these views in the upcoming lessons in this course:

- **V$PARAMETER**: contains one row for every database initialization parameter and includes parameter name, current value and indicates if the parameter can be dynamically modified.
- **V$DATABASE**: Contains one row describing the database state including database name, database ID (DBID), creation date, archivelog mode and checkpoint information used in recovery.
- **V$INSTANCE**: Contains one row describing the current instance. Contains instance name, host (server) name and status (open, mount, nomount).
- **V$SESSION**: Contains one row for every active session, including database background processes.

Notes for this slide continue on the next page...
Common dynamic performance views, continued from the previous page:

- **V$LOCK** – Contains one row for every lock held and locks waited on.
- **V$TRANSACTION** – Contains rows for every active transaction.
- **V$LOGFILE** – Contains the names, locations of the online redo logs and groups to which they belong.
- **V$LOG** – Contains the size, archive status and other information about the online redo log files.
- **V$ARCHIVED_LOG** – Complete list of archived logs. Contains log name and many details.

Refer to the *Oracle9i Database Reference* for a description of all documented data dictionary views. (Remember, http://technet.oracle.com has all the Oracle manuals online; membership is free!)
Example: V$PARAMETER

- Use V$PARAMETER to find the current value of initialization parameters
- Find the location of the Alert log for this instance:

```
1  select value
2  from v$parameter
3  where name = 'background_dump_dest'
SQL> /

VALUE
----------------------------------------
C:\Oracle\admin\dave\bdump
```

The V$PARAMETER view shows the current value for all database initialization parameters. Since there can sometimes be confusion as to what parameter file the instance was started with, this view is invaluable.

The example above shows us how to determine the location of the alert log for the instance. The alert log is typically named “ALERT instance_name.LOG” and is a standard text file that can be viewed with any text editor. The alert log contains messages about significant events and is often used in debugging.

Notes for this slide continue on the next page...
You will find, among other events, these events in the Alert log:

- Instance startup and shutdown
- Log switches
- Instance recovery messages
- Oracle errors, for example:

  ORA-01157: cannot identify/lock data file 1 - see DBWR trace file
  ORA-01110: data file 1: 'C:\ORACLE\ORADATA\DAVE\SYSTEM01.DBF'
  ORA-27041: unable to open file
  OSD-04002: unable to open file
  O/S-Error: (OS 2) The system cannot find the file specified.

This error would be an indication that the SYSTEM datafile had been moved or corrupted, and possibly needs recovery.

We will study how to find and solve some common Oracle errors later in this course.

**Supplemental Notes**

Note that the SQL*Plus SHOW command can also be used to display the database initialization parameters. For example:

```
SQL> show parameter background_dump
NAME                  TYPE     VALUE
-------------------- -------- ------------------
background_dump_dest string   C:\oracle\admin\class2
```
There are several useful views that do not fit neatly into the categories we have discussed in this lesson. These include:

- **SESSION_PRIVS** – Contains system privileges active for the current user and session.
- **DICTIONARY (DICT)** – Contains the name and a brief description (in most cases) of all data dictionary views.
- **DICT_COLUMNS** – Contains the name and a brief description of each column of every data dictionary view.
- **TABLE_PRIVILEGES** – Contains object privileges for the current user.

You will see an example of querying the DICTIONARY view on the next page.

Again, don't concern yourself with trying to remember every detail on each of these views. Later in this course you will be presented each view again when it will help your understanding of a subject.
The DICTIONARY view contains the name of all dictionary views that are available to you (based on privileges granted). It contains two columns, TABLE_NAME and COMMENTS. The COMMENTS column usually contains a descriptive comment about what the view contains. Unfortunately, sometimes the comment is missing or uninformative.

Since there are so many data dictionary views, it is sometimes difficult to find the view with the information that you need. So if you need information about tables, you could use a query like the one shown above to list all dictionary views containing “DBA_TAB”. Use abbreviations, i.e. “DBA_TAB” instead of “DBA_TABLES”, because many data dictionary views use abbreviations in their names.

**Reminder**

Note that we have coded DBA_TAB in upper case. This is because all object names in the data dictionary are stored in upper case. If you a used lower case string, e.g. ‘dba_tab’, no rows would be found.
Since finding data dictionary views is a frequent task, I keep a script handy that assists me. This script, shown above, searches the dictionary view for views that contain the string I provide as an argument.

For example, say I was interested in finding all DBA-level views related to privileges. I would enter:

```
SQL> @dict dba_%priv%
```

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBA_SYS_PRIVS</td>
<td>System privileges granted to users and roles</td>
</tr>
<tr>
<td>DBA_TAB_PRIVS</td>
<td>All grants on objects in the database</td>
</tr>
</tbody>
</table>

This script (DICT.SQL) is available with your supplied scripts.
Data Dictionary Workshop

1. Connect to SQL*Plus with your DBA user (or the SYSTEM user).
2. Display the username and account status of all users on your database.
3. Display the name and status of all tablespaces on your database.
4. For each data file on your database, display the tablespace name, file name, number of bytes used and number of blocks used. If you are familiar with the SQL*Plus BREAK and COMPUTE commands, display the sum of the BYTES column - this is the size of your database (excluding logs, control files and parameter files).
5. Use a V$ view to determine the name of the database you are connected to.
6. Use a V$ view to determine the name of the instance you are connected to.
7. What is the current value of the BACKGROUND_DUMP_DEST parameter?
8. Locate the alert log and view its contents.