3

Managing Objects with Data Dictionary Views

Objectives

After completing this lesson, you should be able to do the following:

- Use the data dictionary views to research data on your objects
- Query various data dictionary views

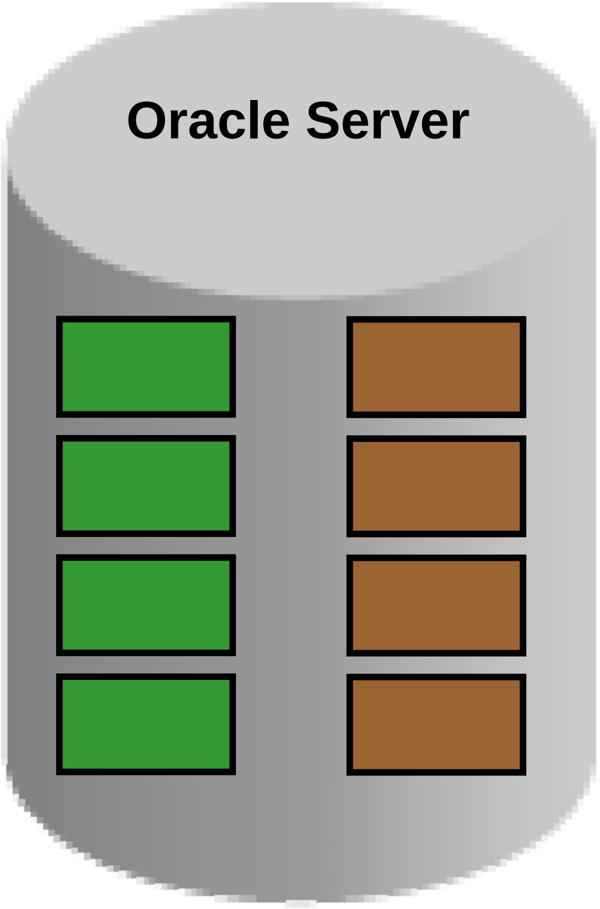
Lesson Agenda

- Introduction to data dictionary
- Querying the dictionary views for the following:
  - Table information
  - Column information
  - Constraint information
- Querying the dictionary views for the following:
  - View information
  - Sequence information
  - Synonym information
  - Index information
- Adding a comment to a table and querying the dictionary views for comment information

Data Dictionary

Tables containing business data:

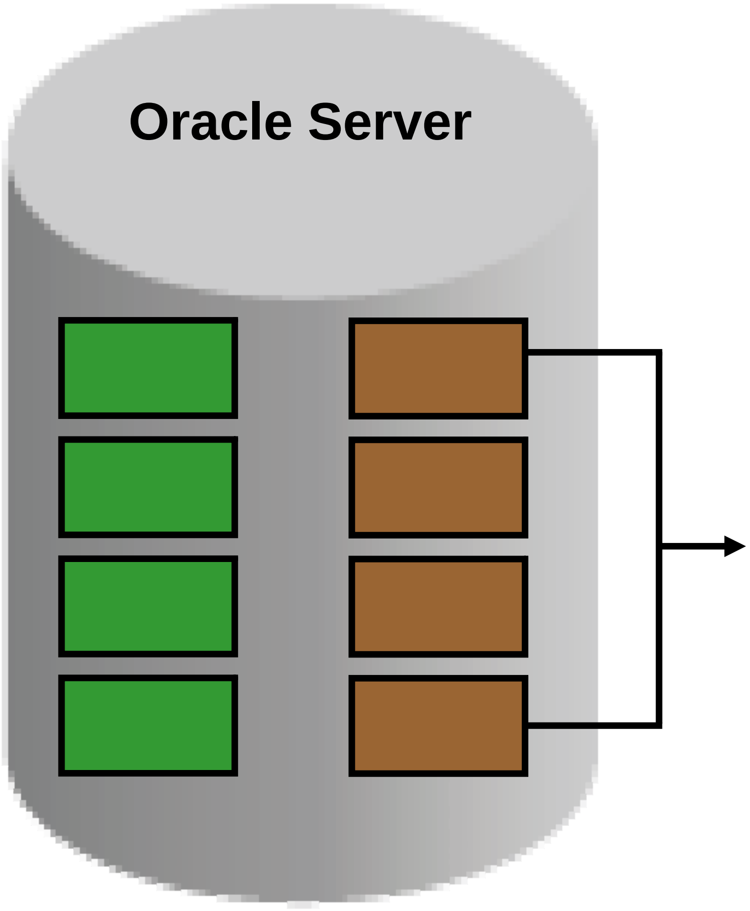
EMPLOYEES
DEPARTMENTS
LOCATIONS
JOB_HISTORY
...



Data dictionary views:

DICTIONARY
USER_OBJECTS
USER_TABLES
USER_TAB_COLUMNS
...

Data Dictionary Structure



Consists of:

- Base tables
- User-accessible views

Data Dictionary Structure

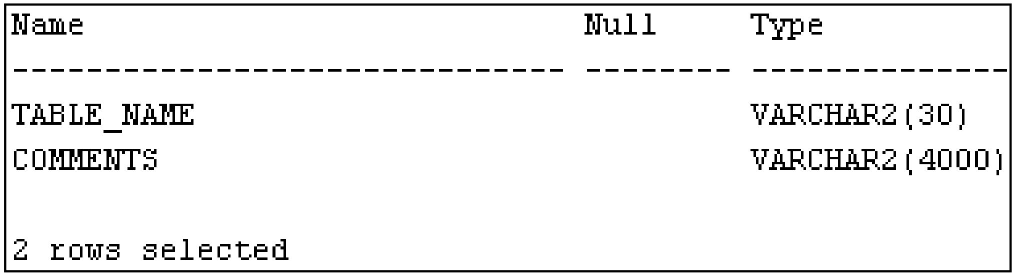
View naming convention:

View Prefix	Purpose
USER	User's view (what is in your schema; what you own)
ALL	Expanded user's view (what you can access)
DBA	Database administrator's view (what is in everyone's schemas)
V$	Performance-related data

How to Use the Dictionary Views

Start with DICTIONARY. It contains the names and descriptions of the dictionary tables and views.

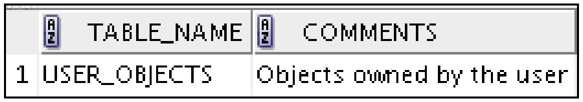
```
DESCRIBE DICTIONARY
```



```
Name                           Null     Type
------------------------------ -------- --------------
TABLE_NAME                              VARCHAR2(30)
COMMENTS                                VARCHAR2(4000)

2 rows selected
```

```
SELECT *
FROM   dictionary
WHERE  table_name = 'USER_OBJECTS';
```



	TABLE_NAME	COMMENTS
1	USER_OBJECTS	Objects owned by the user

USER_OBJECTS and ALL_OBJECTS Views

USER_OBJECTS:

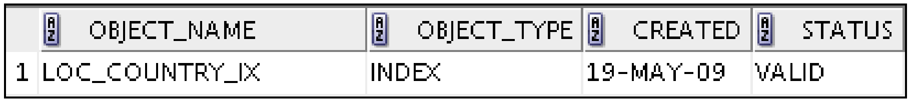
- Query USER_OBJECTS to see all the objects that you own.
- Using USER_OBJECTS, you can obtain a listing of all object names and types in your schema, plus the following information:
  - Date created
  - Date of last modification
  - Status (valid or invalid)

ALL_OBJECTS:

- Query ALL_OBJECTS to see all the objects to which you have access.

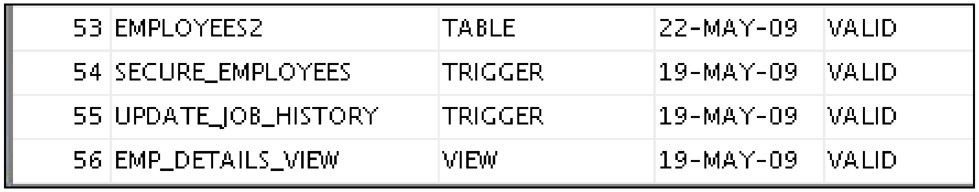
USER_OBJECTS View

```
SELECT object_name, object_type, created, status
FROM   user_objects
ORDER BY object_type;
```



	OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
1	LOC_COUNTRY_IX	INDEX	19-MAY-09	VALID

...



53	EMPLOYEES2	TABLE	22-MAY-09	VALID
54	SECURE_EMPLOYEES	TRIGGER	19-MAY-09	VALID
55	UPDATE_JOB_HISTORY	TRIGGER	19-MAY-09	VALID
56	EMP_DETAILS_VIEW	VIEW	19-MAY-09	VALID

...

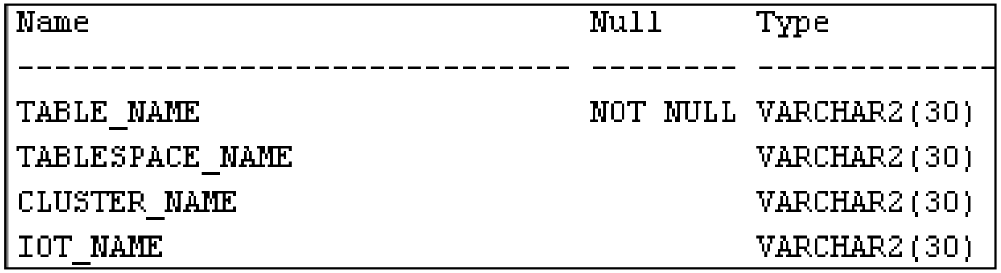
Lesson Agenda

- Introduction to data dictionary
- Querying the dictionary views for the following:
  - Table information
  - Column information
  - Constraint information
- Querying the dictionary views for the following:
  - View information
  - Sequence information
  - Synonym information
  - Index information
- Adding a comment to a table and querying the dictionary views for comment information

Table Information

USER_TABLES:

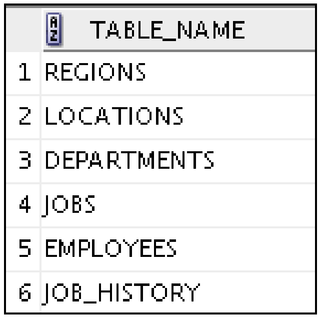
```
DESCRIBE user_tables
```



```
Name                           Null     Type
------------------------------ -------- -------------
TABLE_NAME                     NOT NULL VARCHAR2(30)
TABLESPACE_NAME                         VARCHAR2(30)
CLUSTER_NAME                            VARCHAR2(30)
IOT_NAME                                VARCHAR2(30)
```

…

```
SELECT table_name
FROM   user_tables;
```



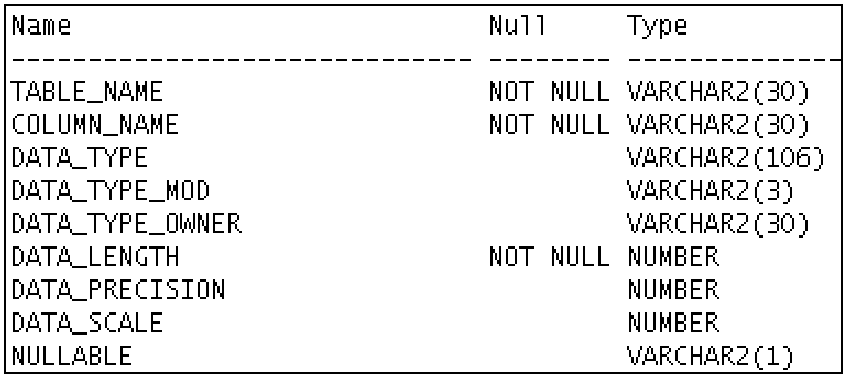
	TABLE_NAME
1	REGIONS
2	LOCATIONS
3	DEPARTMENTS
4	JOBS
5	EMPLOYEES
6	JOB_HISTORY

…

Column Information

USER_TAB_COLUMNS:

```
DESCRIBE user_tab_columns
```

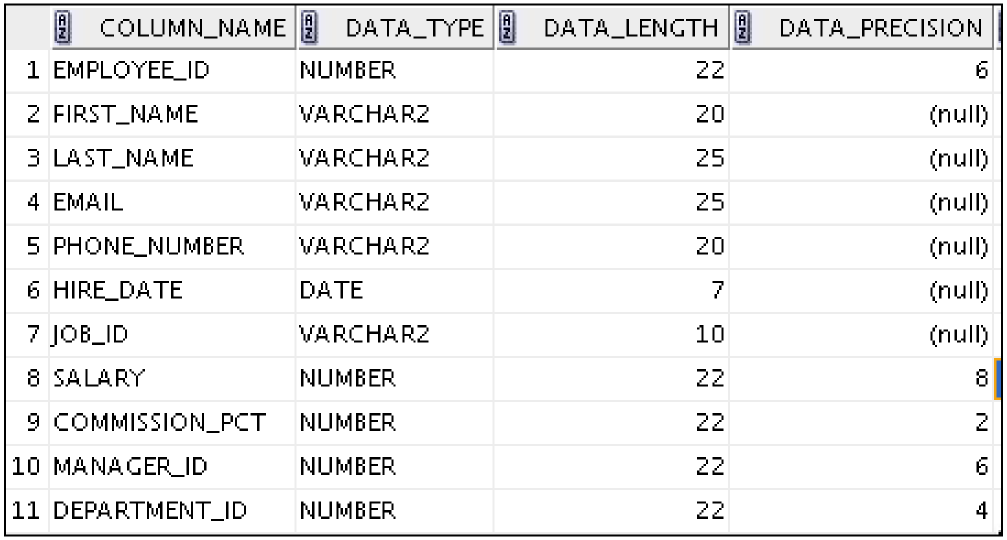


```
Name                           Null     Type
------------------------------ -------- --------------
TABLE_NAME                     NOT NULL VARCHAR2(30)
COLUMN_NAME                    NOT NULL VARCHAR2(30)
DATA_TYPE                               VARCHAR2(106)
DATA_TYPE_MOD                           VARCHAR2(3)
DATA_TYPE_OWNER                         VARCHAR2(30)
DATA_LENGTH                    NOT NULL NUMBER
DATA_PRECISION                          NUMBER
DATA_SCALE                              NUMBER
NULLABLE                                VARCHAR2(1)
```

…

Column Information

```
SELECT column_name, data_type, data_length,
       data_precision, data_scale, nullable
FROM   user_tab_columns
WHERE  table_name = 'EMPLOYEES';
```

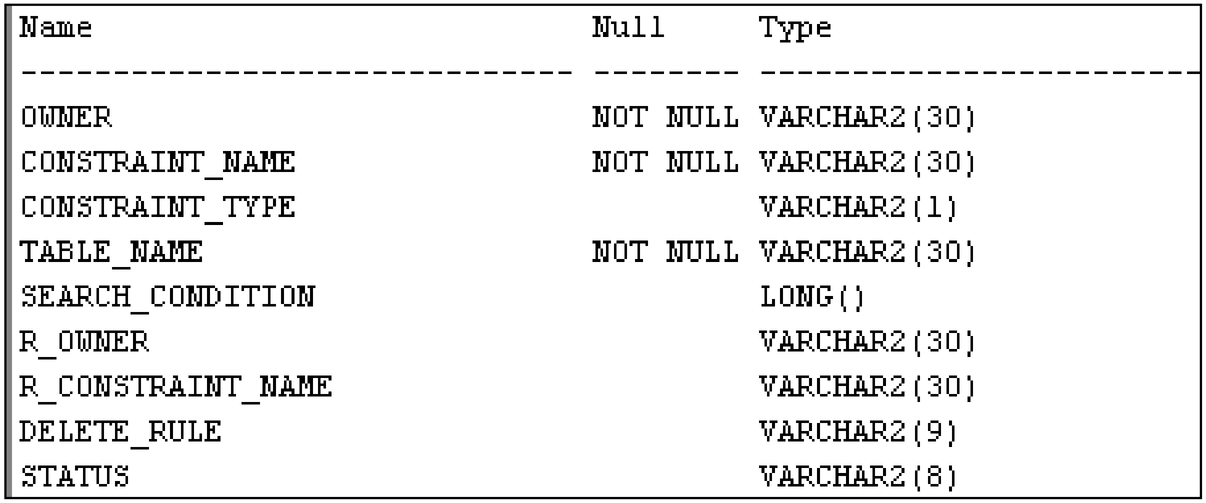


	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION
1	EMPLOYEE_ID	NUMBER	22	6
2	FIRST_NAME	VARCHAR2	20	(null)
3	LAST_NAME	VARCHAR2	25	(null)
4	EMAIL	VARCHAR2	25	(null)
5	PHONE_NUMBER	VARCHAR2	20	(null)
6	HIRE_DATE	DATE	7	(null)
7	JOB_ID	VARCHAR2	10	(null)
8	SALARY	NUMBER	22	8
9	COMMISSION_PCT	NUMBER	22	2
10	MANAGER_ID	NUMBER	22	6
11	DEPARTMENT_ID	NUMBER	22	4

Constraint Information

- `USER_CONSTRAINTS` describes the constraint definitions on your tables.
- `USER_CONS_COLUMNS` describes columns that are owned by you and that are specified in constraints.

```
DESCRIBE user_constraints
```

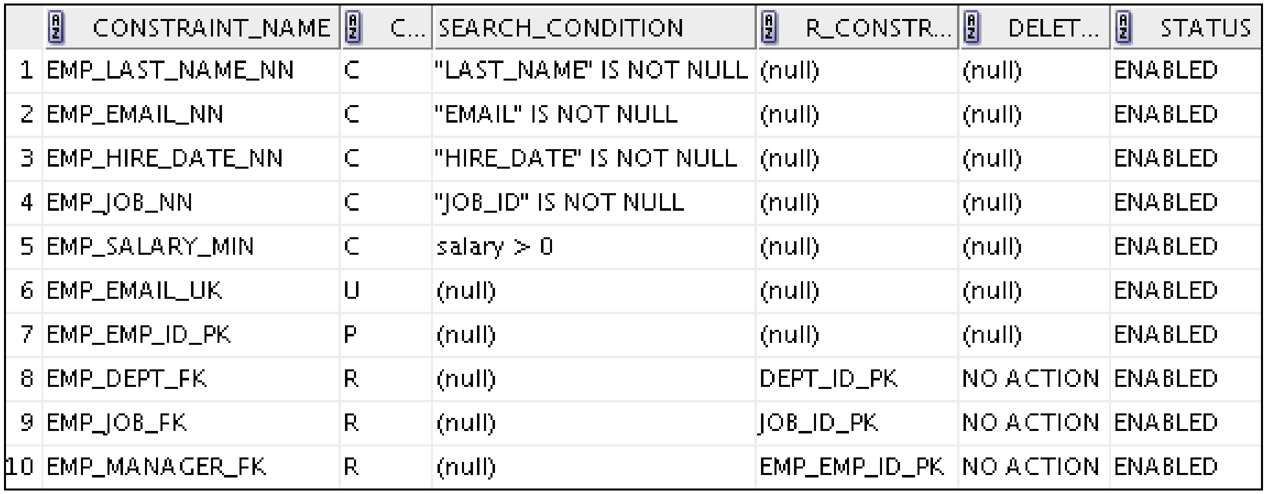


```
Name                           Null     Type
------------------------------ -------- ------------------------
OWNER                          NOT NULL VARCHAR2(30)
CONSTRAINT_NAME                NOT NULL VARCHAR2(30)
CONSTRAINT_TYPE                         VARCHAR2(1)
TABLE_NAME                     NOT NULL VARCHAR2(30)
SEARCH_CONDITION                        LONG()
R_OWNER                                 VARCHAR2(30)
R_CONSTRAINT_NAME                       VARCHAR2(30)
DELETE_RULE                             VARCHAR2(9)
STATUS                                  VARCHAR2(8)
```

…

USER_CONSTRAINTS: Example

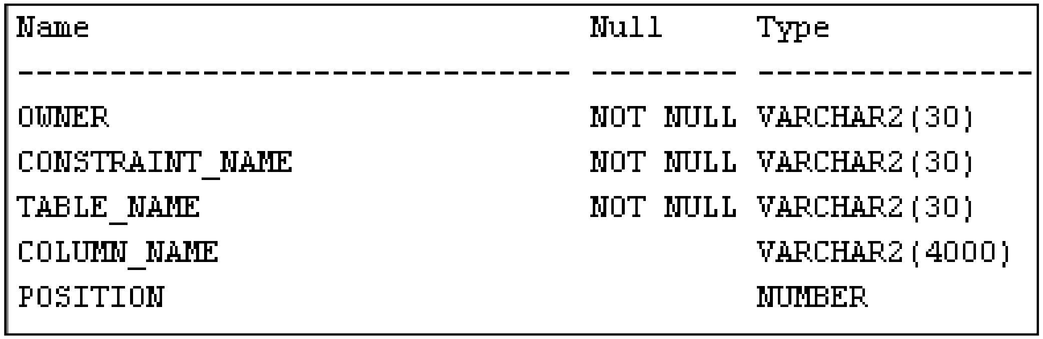
```
SELECT constraint_name, constraint_type,
       search_condition, r_constraint_name,
       delete_rule, status
FROM   user_constraints
WHERE  table_name = 'EMPLOYEES';
```



	CONSTRAINT_NAME	C...	SEARCH_CONDITION	R_CONSTR...	DELET...	STATUS
1	EMP_LAST_NAME_NN	C	"LAST_NAME" IS NOT NULL	(null)	(null)	ENABLED
2	EMP_EMAIL_NN	C	"EMAIL" IS NOT NULL	(null)	(null)	ENABLED
3	EMP_HIRE_DATE_NN	C	"HIRE_DATE" IS NOT NULL	(null)	(null)	ENABLED
4	EMP_JOB_NN	C	"JOB_ID" IS NOT NULL	(null)	(null)	ENABLED
5	EMP_SALARY_MIN	C	salary > 0	(null)	(null)	ENABLED
6	EMP_EMAIL_UK	U	(null)	(null)	(null)	ENABLED
7	EMP_EMP_ID_PK	P	(null)	(null)	(null)	ENABLED
8	EMP_DEPT_FK	R	(null)	DEPT_ID_PK	NO ACTION	ENABLED
9	EMP_JOB_FK	R	(null)	JOB_ID_PK	NO ACTION	ENABLED
10	EMP_MANAGER_FK	R	(null)	EMP_EMP_ID_PK	NO ACTION	ENABLED

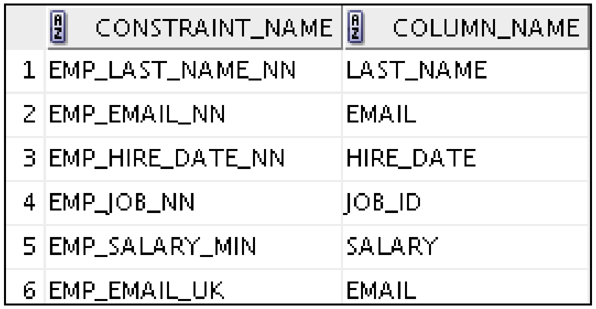
Querying `USER_CONS_COLUMNS`

```
DESCRIBE user_cons_columns
```



```
Name                           Null     Type
------------------------------ -------- ---------------
OWNER                          NOT NULL VARCHAR2(30)
CONSTRAINT_NAME                NOT NULL VARCHAR2(30)
TABLE_NAME                     NOT NULL VARCHAR2(30)
COLUMN_NAME                             VARCHAR2(4000)
POSITION                                NUMBER
```

```
SELECT constraint_name, column_name
FROM   user_cons_columns
WHERE  table_name = 'EMPLOYEES';
```



	CONSTRAINT_NAME	COLUMN_NAME
1	EMP_LAST_NAME_NN	LAST_NAME
2	EMP_EMAIL_NN	EMAIL
3	EMP_HIRE_DATE_NN	HIRE_DATE
4	EMP_JOB_NN	JOB_ID
5	EMP_SALARY_MIN	SALARY
6	EMP_EMAIL_UK	EMAIL

…

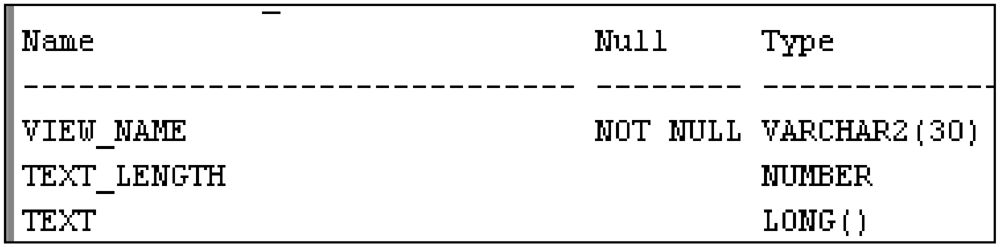
Lesson Agenda

- Introduction to data dictionary
- Querying the dictionary views for the following:
  - Table information
  - Column information
  - Constraint information
- Querying the dictionary views for the following:
  - View information
  - Sequence information
  - Synonym information
  - Index information
- Adding a comment to a table and querying the dictionary views for comment information

View Information

1

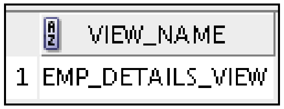
```
DESCRIBE user_views
```



```
Name                           Null     Type
------------------------------ -------- -------------
VIEW_NAME                      NOT NULL VARCHAR2(30)
TEXT_LENGTH                             NUMBER
TEXT                                    LONG()
```

2

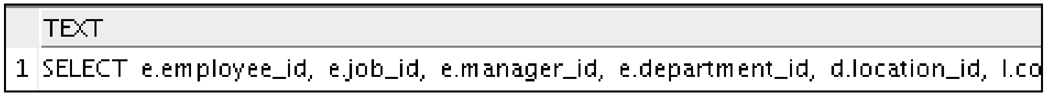
```
SELECT view_name FROM user_views;
```



```
  VIEW_NAME
1 EMP_DETAILS_VIEW
```

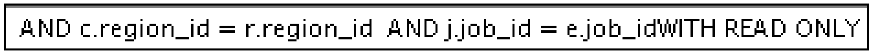
3

```
SELECT text FROM user_views
WHERE view_name = 'EMP_DETAILS_VIEW';
```



```
  TEXT
1 SELECT  e.employee_id,  e.job_id,  e.manager_id,  e.department_id,  d.location_id,  l.co
```

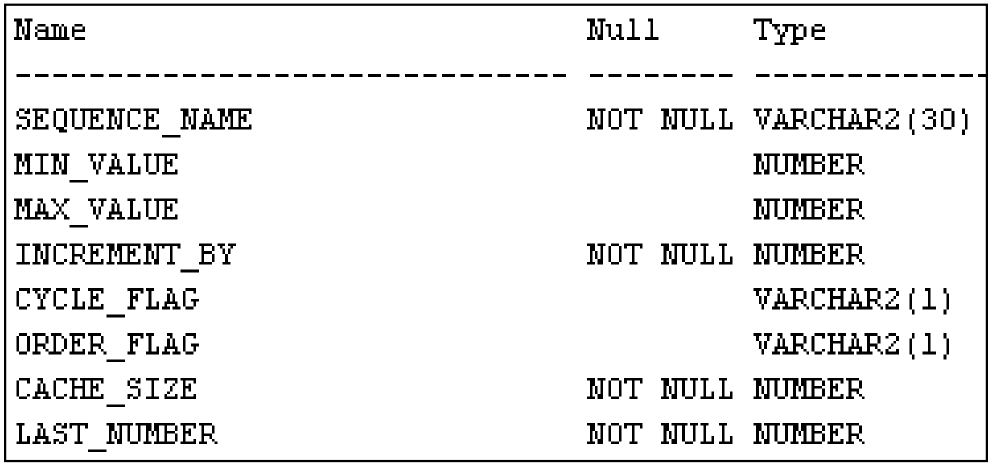
...



```
AND c.region_id = r.region_id  AND j.job_id = e.job_idWITH READ ONLY
```

Sequence Information

```
DESCRIBE user_sequences
```

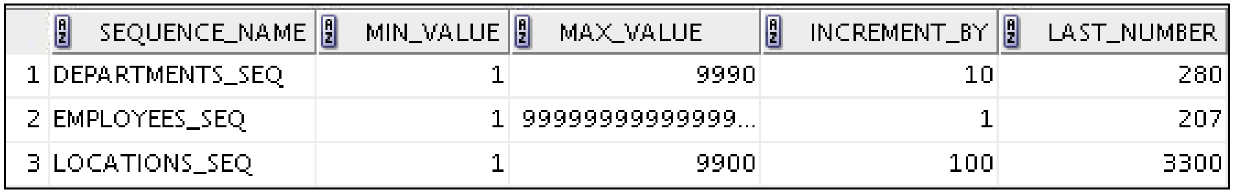


```
Name                           Null     Type
------------------------------ -------- -------------
SEQUENCE_NAME                  NOT NULL VARCHAR2(30)
MIN_VALUE                               NUMBER
MAX_VALUE                               NUMBER
INCREMENT_BY                   NOT NULL NUMBER
CYCLE_FLAG                              VARCHAR2(1)
ORDER_FLAG                              VARCHAR2(1)
CACHE_SIZE                     NOT NULL NUMBER
LAST_NUMBER                    NOT NULL NUMBER
```

Confirming Sequences

- Verify your sequence values in the `USER_SEQUENCES` data dictionary table.

```
SELECT   sequence_name, min_value, max_value,
         increment_by, last_number
FROM     user_sequences;
```



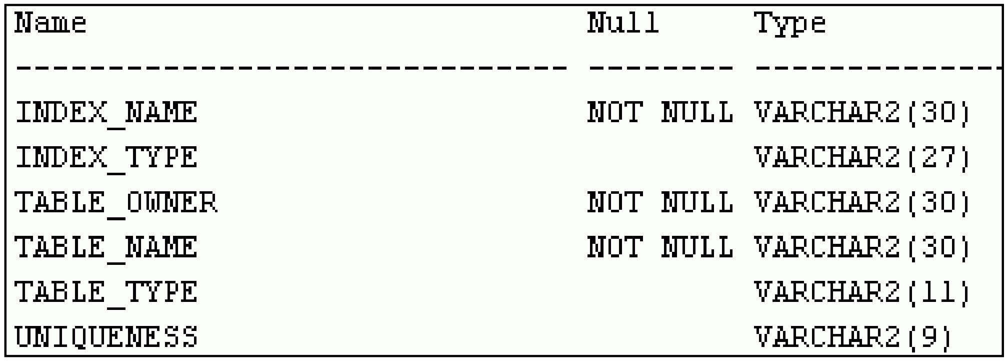
	SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
1	DEPARTMENTS_SEQ	1	9990	10	280
2	EMPLOYEES_SEQ	1	99999999999999...	1	207
3	LOCATIONS_SEQ	1	9900	100	3300

- The `LAST_NUMBER` column displays the next available sequence number if `NOCACHE` is specified.

Index Information

- USER_INDEXES provides information about your indexes.
- USER_IND_COLUMNS describes columns comprising your indexes and columns of indexes on your tables.

```
DESCRIBE user_indexes
```



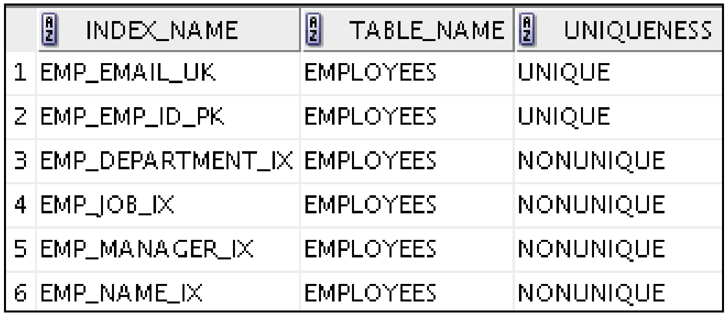
```
Name                            Null     Type
------------------------------- -------- -------------
INDEX_NAME                      NOT NULL VARCHAR2(30)
INDEX_TYPE                               VARCHAR2(27)
TABLE_OWNER                     NOT NULL VARCHAR2(30)
TABLE_NAME                      NOT NULL VARCHAR2(30)
TABLE_TYPE                               VARCHAR2(11)
UNIQUENESS                               VARCHAR2(9)
```

…

USER_INDEXES: Examples

a

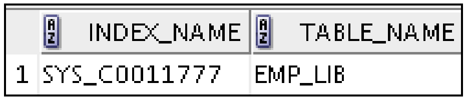
```
SELECT index_name, table_name,uniqueness
FROM   user_indexes
WHERE  table_name = 'EMPLOYEES';
```



	INDEX_NAME	TABLE_NAME	UNIQUENESS
1	EMP_EMAIL_UK	EMPLOYEES	UNIQUE
2	EMP_EMP_ID_PK	EMPLOYEES	UNIQUE
3	EMP_DEPARTMENT_IX	EMPLOYEES	NONUNIQUE
4	EMP_JOB_IX	EMPLOYEES	NONUNIQUE
5	EMP_MANAGER_IX	EMPLOYEES	NONUNIQUE
6	EMP_NAME_IX	EMPLOYEES	NONUNIQUE

b

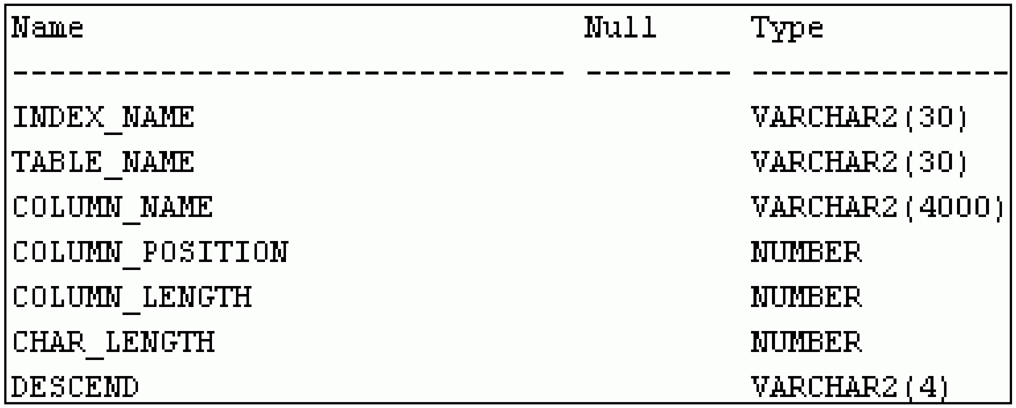
```
SELECT index_name, table_name
FROM   user_indexes
WHERE  table_name = 'emp_lib';
```



	INDEX_NAME	TABLE_NAME
1	SYS_C0011777	EMP_LIB

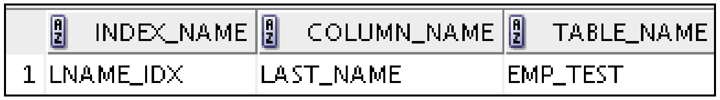
Querying USER_IND_COLUMNS

```
DESCRIBE user_ind_columns
```



```
Name                           Null     Type
------------------------------ -------- --------------
INDEX_NAME                              VARCHAR2(30)
TABLE_NAME                              VARCHAR2(30)
COLUMN_NAME                             VARCHAR2(4000)
COLUMN_POSITION                         NUMBER
COLUMN_LENGTH                           NUMBER
CHAR_LENGTH                             NUMBER
DESCEND                                 VARCHAR2(4)
```

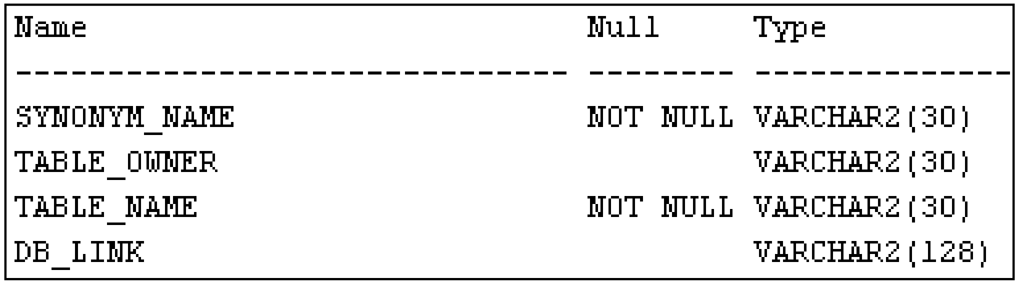
```
SELECT index_name, column_name,table_name
FROM   user_ind_columns
WHERE  index_name = 'lname_idx';
```



	INDEX_NAME	COLUMN_NAME	TABLE_NAME
1	LNAME_IDX	LAST_NAME	EMP_TEST

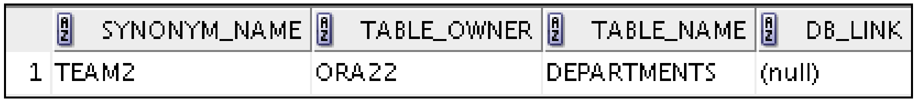
Synonym Information

```
DESCRIBE user_synonyms
```



```
Name                           Null     Type
------------------------------ -------- --------------
SYNONYM_NAME                   NOT NULL VARCHAR2(30)
TABLE_OWNER                             VARCHAR2(30)
TABLE_NAME                     NOT NULL VARCHAR2(30)
DB_LINK                                 VARCHAR2(128)
```

```
SELECT *
FROM   user_synonyms;
```



	SYNONYM_NAME	TABLE_OWNER	TABLE_NAME	DB_LINK
1	TEAM2	ORA22	DEPARTMENTS	(null)

Lesson Agenda

- Introduction to data dictionary
- Querying the dictionary views for the following:
  - Table information
  - Column information
  - Constraint information
- Querying the dictionary views for the following:
  - View information
  - Sequence information
  - Synonym information
  - Index information
- Adding a comment to a table and querying the dictionary views for comment information

ORACLE

Adding Comments to a Table

- You can add comments to a table or column by using the COMMENT statement:

```
COMMENT ON TABLE employees
IS 'Employee Information';
```

```
COMMENT ON COLUMN employees.first_name
IS 'First name of the employee';
```

- Comments can be viewed through the data dictionary views:
  - ALL_COL_COMMENTS
  - USER_COL_COMMENTS
  - ALL_TAB_COMMENTS
  - USER_TAB_COMMENTS

Quiz

The dictionary views that are based on the dictionary tables contain information such as:

1. Definitions of all the schema objects in the database
2. Default values for the columns
3. Integrity constraint information
4. Privileges and roles that each user has been granted
5. All of the above

Summary

In this lesson, you should have learned how to find information about your objects through the following dictionary views:

- `DICTIONARY`
- `USER_OBJECTS`
- `USER_TABLES`
- `USER_TAB_COLUMNS`
- `USER_CONSTRAINTS`
- `USER_CONS_COLUMNS`
- `USER_VIEWS`
- `USER_SEQUENCES`
- `USER_INDEXES`
- `USER_SYNONYMS`

Practice 3: Overview

This practice covers the following topics:

- Querying the dictionary views for table and column information
- Querying the dictionary views for constraint information
- Querying the dictionary views for view information
- Querying the dictionary views for sequence information
- Querying the dictionary views for synonym information
- Querying the dictionary views for index information
- Adding a comment to a table and querying the dictionary views for comment information