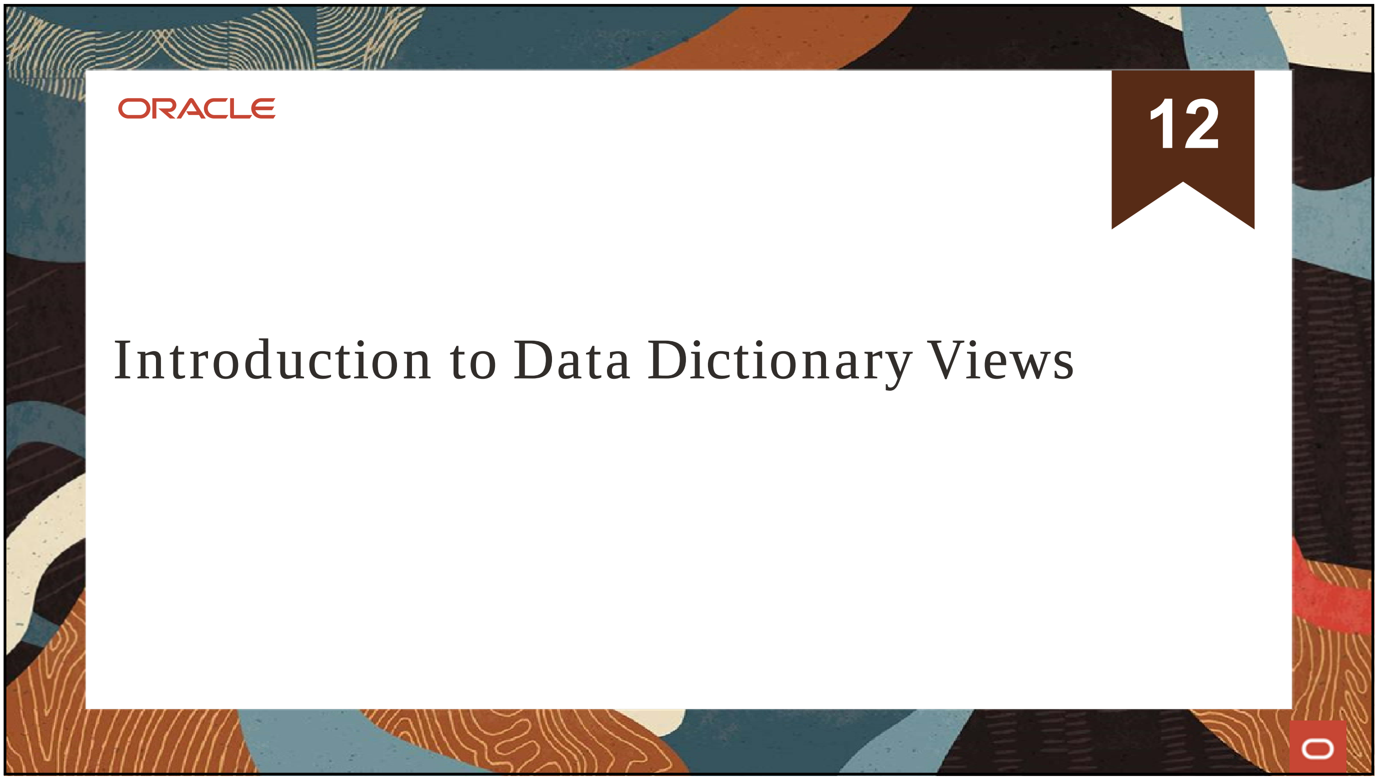
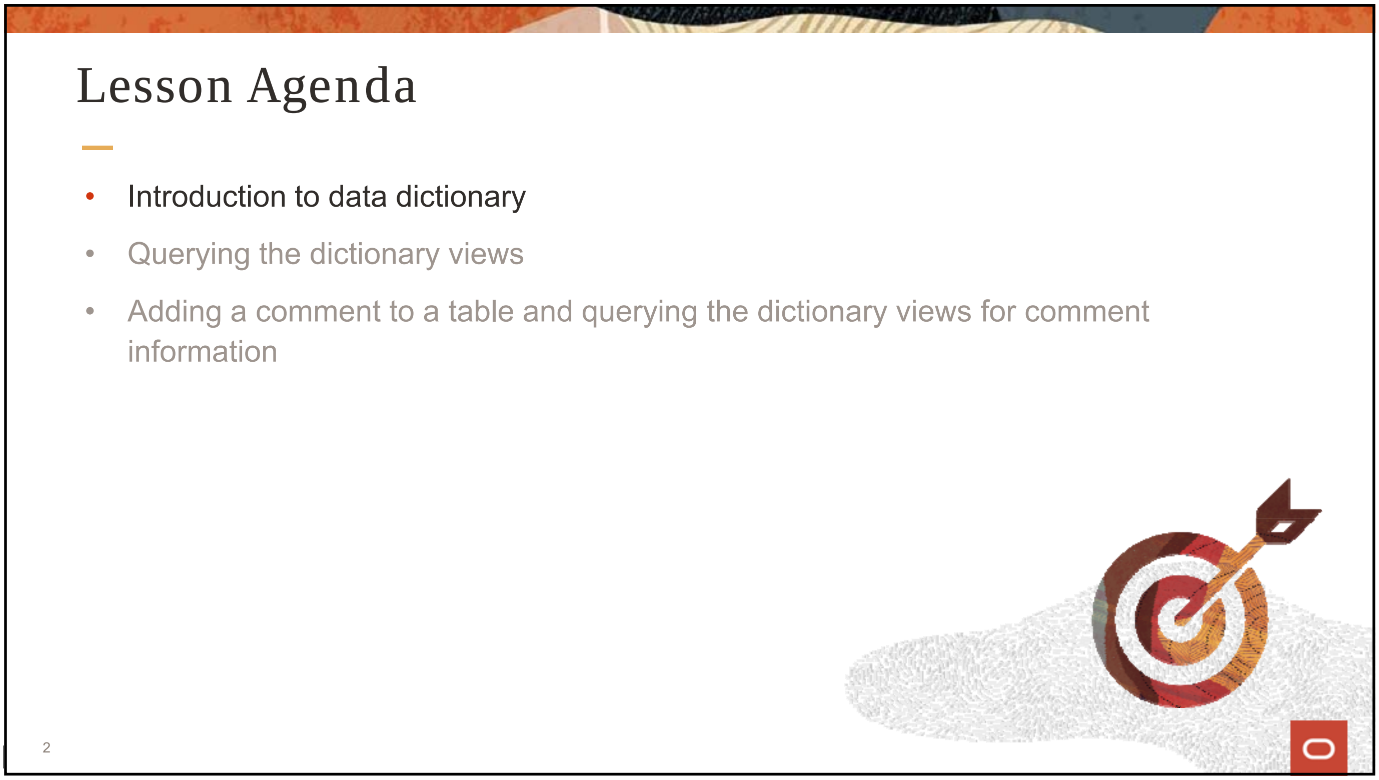


ORACLE

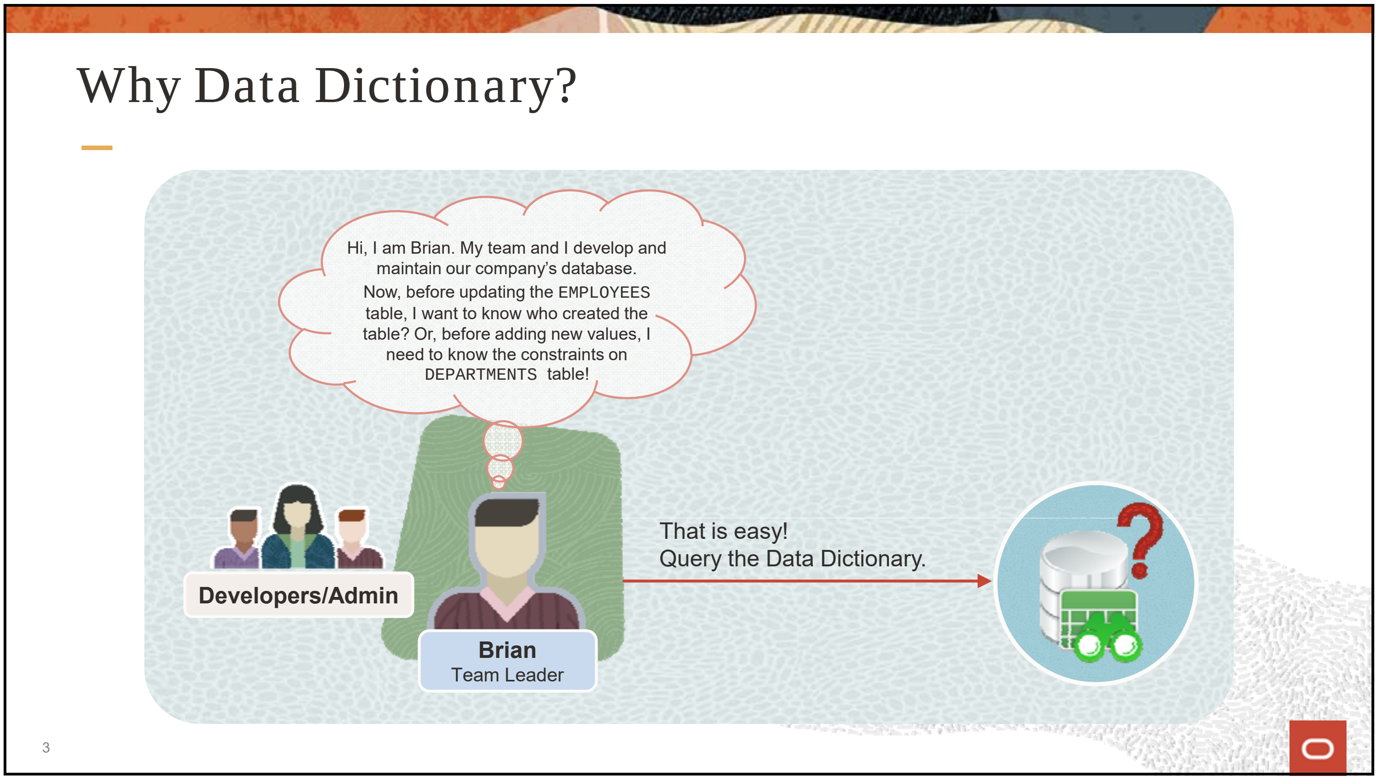
12

Introduction to Data Dictionary Views



Lesson Agenda

- Introduction to data dictionary
- Querying the dictionary views
- Adding a comment to a table and querying the dictionary views for comment information



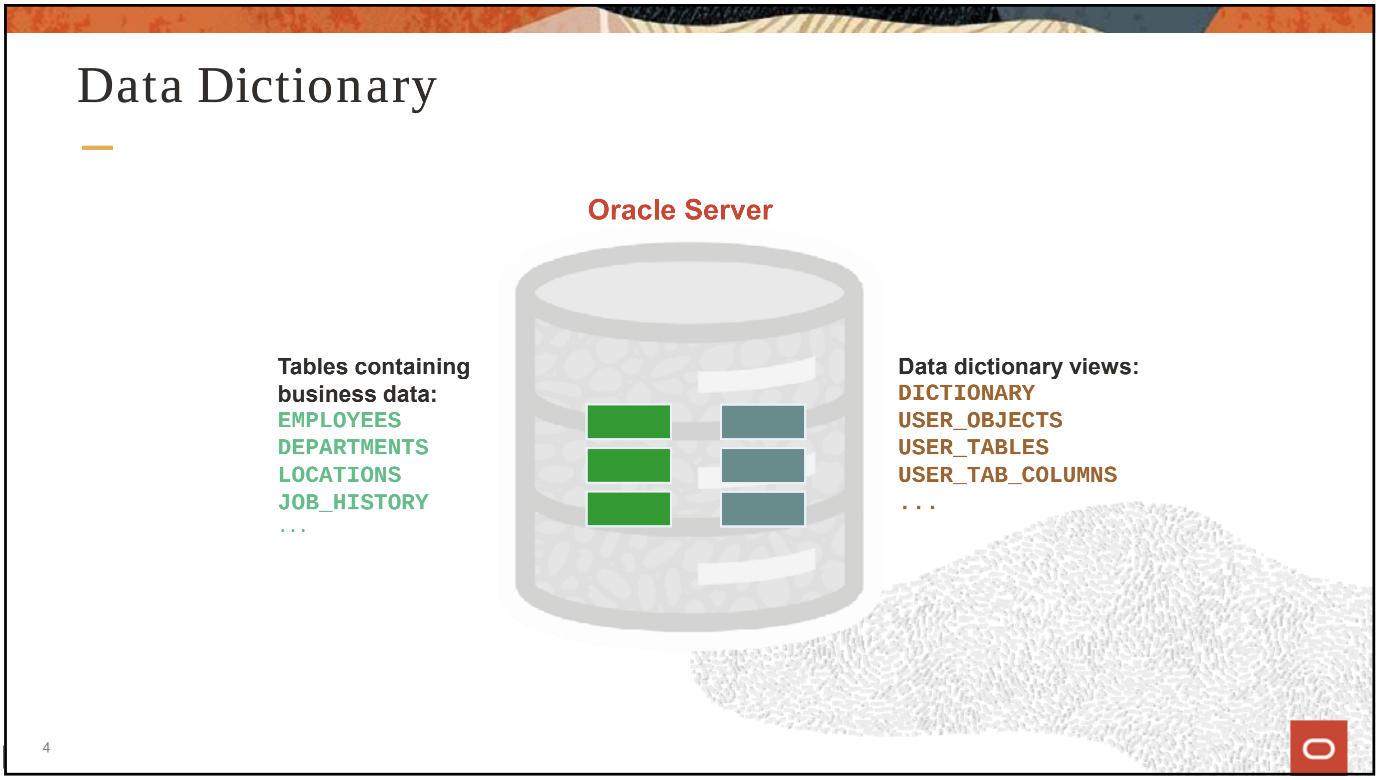
Why Data Dictionary?

Hi, I am Brian. My team and I develop and maintain our company's database. Now, before updating the EMPLOYEES table, I want to know who created the table? Or, before adding new values, I need to know the constraints on DEPARTMENTS table!

Developers/Admin

Brian
Team Leader

That is easy!
Query the Data Dictionary.



Data Dictionary

Oracle Server

Tables containing business data:

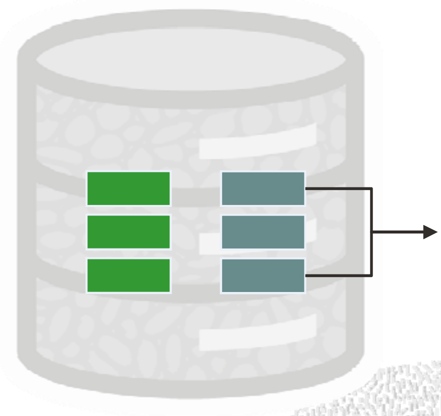
```
EMPLOYEES
DEPARTMENTS
LOCATIONS
JOB_HISTORY
...
```

Data dictionary views:

```
DICTIONARY
USER_OBJECTS
USER_TABLES
USER_TAB_COLUMNS
...
```

Data Dictionary Structure

Oracle Server



Consists of:
- Base tables
- User-accessible views

Data Dictionary Structure

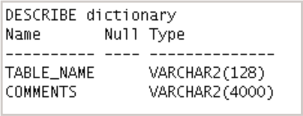
View naming convention is as follows:

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 Dictionary Views

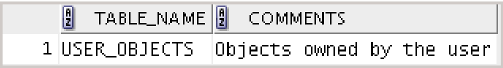
Start with DICTIONARY. It contains the names and descriptions of the dictionary tables and views.

```
DESCRIBE DICTIONARY
```



```
DESCRIBE dictionary
Name       Null Type
---------- ---- --------------
TABLE_NAME      VARCHAR2(128)
COMMENTS        VARCHAR2(4000)
```

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



	TABLE_NAME	COMMENTS
1	USER_OBJECTS	Objects owned by the user

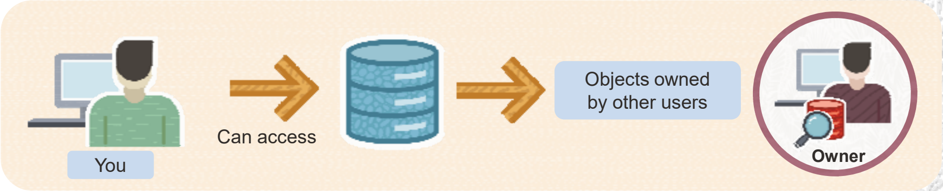
USER_OBJECTS and ALL_OBJECTS Views

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)

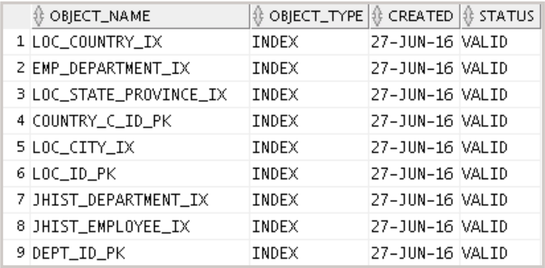
ALL_OBJECTS:

- 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	27-JUN-16	VALID
2	EMP_DEPARTMENT_IX	INDEX	27-JUN-16	VALID
3	LOC_STATE_PROVINCE_IX	INDEX	27-JUN-16	VALID
4	COUNTRY_C_ID_PK	INDEX	27-JUN-16	VALID
5	LOC_CITY_IX	INDEX	27-JUN-16	VALID
6	LOC_ID_PK	INDEX	27-JUN-16	VALID
7	JHIST_DEPARTMENT_IX	INDEX	27-JUN-16	VALID
8	JHIST_EMPLOYEE_IX	INDEX	27-JUN-16	VALID
9	DEPT_ID_PK	INDEX	27-JUN-16	VALID

Owned by the user

. . .

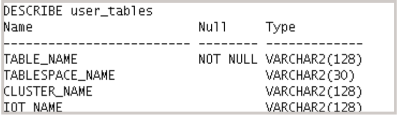
Lesson Agenda

- Introduction to data dictionary
- Querying the dictionary views
- Adding a comment to a table and querying the dictionary views for comment information

Table Information

USER_TABLES:

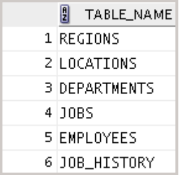
```
DESCRIBE user_tables
```



```
DESCRIBE user_tables
Name                      Null     Type
------------------------- -------- -------------
TABLE_NAME                NOT NULL VARCHAR2(128)
TABLESPACE_NAME                    VARCHAR2(30)
CLUSTER_NAME                       VARCHAR2(128)
IOT_NAME                           VARCHAR2(128)
```

...

```
SELECT table_name
FROM   user_tables;
```



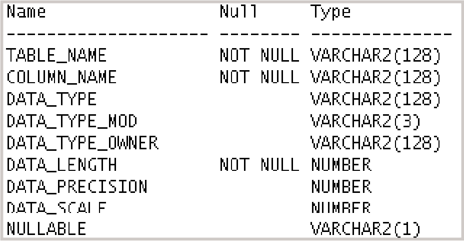
	TABLE_NAME
1	REGIONS
2	LOCATIONS
3	DEPARTMENTS
4	JOBS
5	EMPLOYEES
6	JOB_HISTORY

...

Column Information

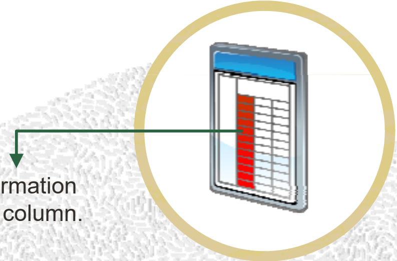
USER_TAB_COLUMNS:

```
DESCRIBE user_tab_columns
```



```
Name                 Null     Type
-------------------- -------- --------------
TABLE_NAME           NOT NULL VARCHAR2(128)
COLUMN_NAME          NOT NULL VARCHAR2(128)
DATA_TYPE                     VARCHAR2(128)
DATA_TYPE_MOD                 VARCHAR2(3)
DATA_TYPE_OWNER               VARCHAR2(128)
DATA_LENGTH          NOT NULL NUMBER
DATA_PRECISION                NUMBER
DATA_SCALE                    NUMBER
NULLABLE                      VARCHAR2(1)
```

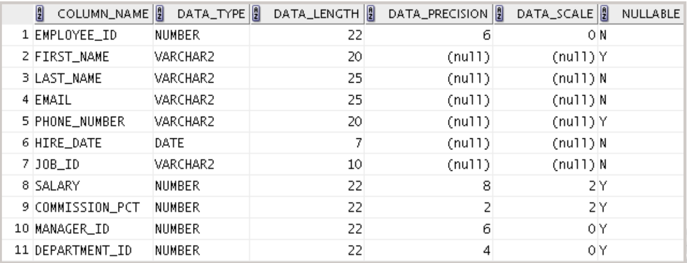
...



Get information about a column.

Column Information

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

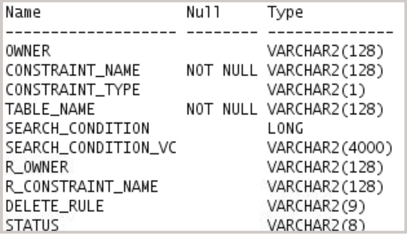


	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION	DATA_SCALE	NULLABLE
1	EMPLOYEE_ID	NUMBER	22	6	0	N
2	FIRST_NAME	VARCHAR2	20	(null)	(null)	Y
3	LAST_NAME	VARCHAR2	25	(null)	(null)	N
4	EMAIL	VARCHAR2	25	(null)	(null)	N
5	PHONE_NUMBER	VARCHAR2	20	(null)	(null)	Y
6	HIRE_DATE	DATE	7	(null)	(null)	N
7	JOB_ID	VARCHAR2	10	(null)	(null)	N
8	SALARY	NUMBER	22	8	2	Y
9	COMMISSION_PCT	NUMBER	22	2	2	Y
10	MANAGER_ID	NUMBER	22	6	0	Y
11	DEPARTMENT_ID	NUMBER	22	4	0	Y

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.

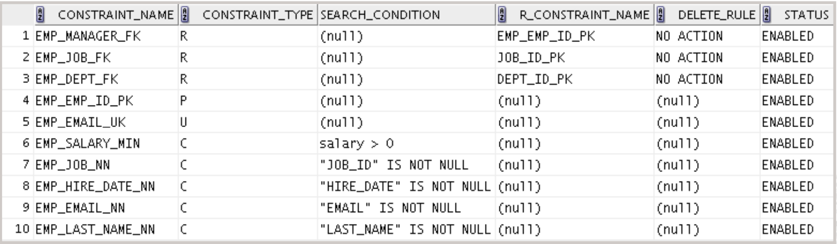
```
DESCRIBE user_constraints
```



```
Name                 Null     Type
-------------------- -------- --------------
OWNER                         VARCHAR2(128)
CONSTRAINT_NAME      NOT NULL VARCHAR2(128)
CONSTRAINT_TYPE               VARCHAR2(1)
TABLE_NAME           NOT NULL VARCHAR2(128)
SEARCH_CONDITION              LONG
SEARCH_CONDITION_VC           VARCHAR2(4000)
R_OWNER                       VARCHAR2(128)
R_CONSTRAINT_NAME             VARCHAR2(128)
DELETE_RULE                   VARCHAR2(9)
STATUS                        VARCHAR2(8)
...
```

USER_CONSTRAINTS: Example

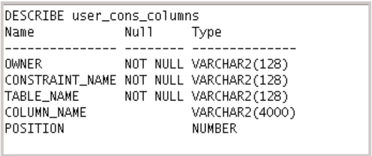
```
SELECT constraint_name, constraint_type,
       search_condition, r_constraint_name,
       delete_rule, status
FROM   user_constraints
WHERE  table_name = 'EMPLOYEES';
```



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

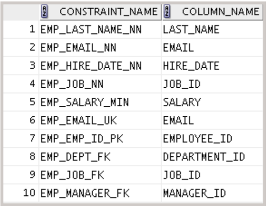
Querying USER_CONS_COLUMNS

```
DESCRIBE user_cons_columns
```



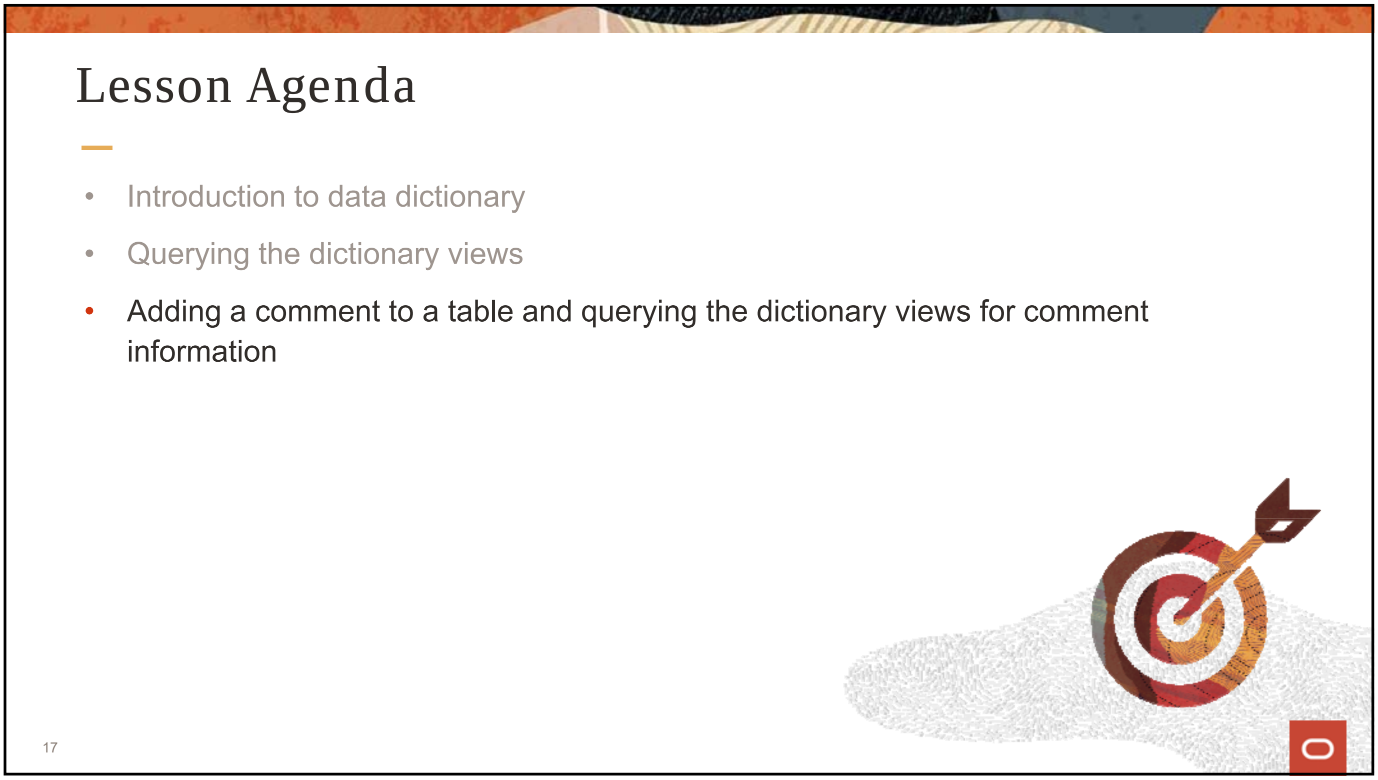
```
DESCRIBE user_cons_columns
Name            Null     Type
--------------- -------- --------------
OWNER           NOT NULL VARCHAR2(128)
CONSTRAINT_NAME NOT NULL VARCHAR2(128)
TABLE_NAME      NOT NULL VARCHAR2(128)
COLUMN_NAME              VARCHAR2(4000)
POSITION                 NUMBER
```

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



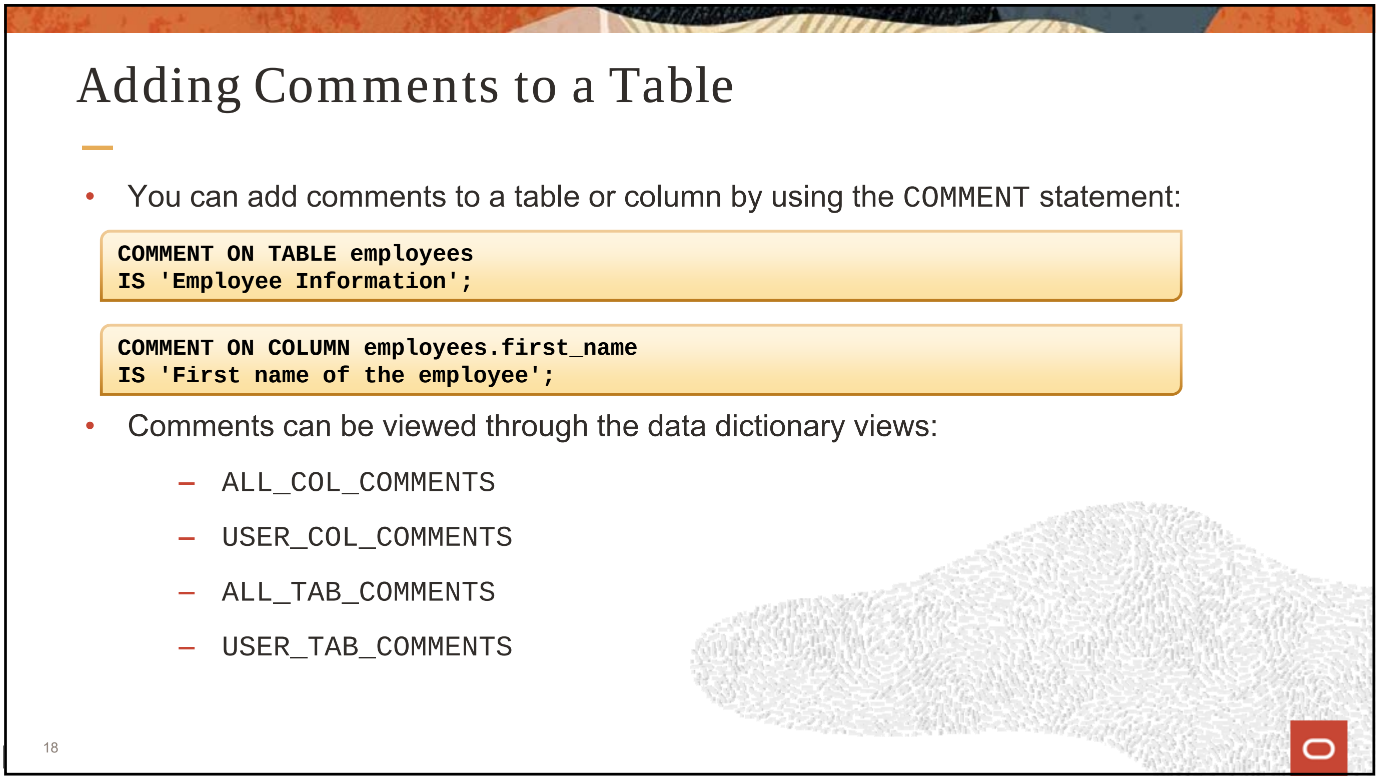
	CONSTRAINT_NAME	COLUMN_NAME
1	EMP_LAST_NAME_NN	LAST_NAME
2	EMP_EMAIL_NN	EMAIL
3	EMP_HIRE_DATE_NN	HIRE_DATE
4	EMP_JOB_NN	JOB_ID
5	EMP_SALARY_MIN	SALARY
6	EMP_EMAIL_UK	EMAIL
7	EMP_EMP_ID_PK	EMPLOYEE_ID
8	EMP_DEPT_FK	DEPARTMENT_ID
9	EMP_JOB_FK	JOB_ID
10	EMP_MANAGER_FK	MANAGER_ID

...



Lesson Agenda

- Introduction to data dictionary
- Querying the dictionary views
- Adding a comment to a table and querying the dictionary views for comment information



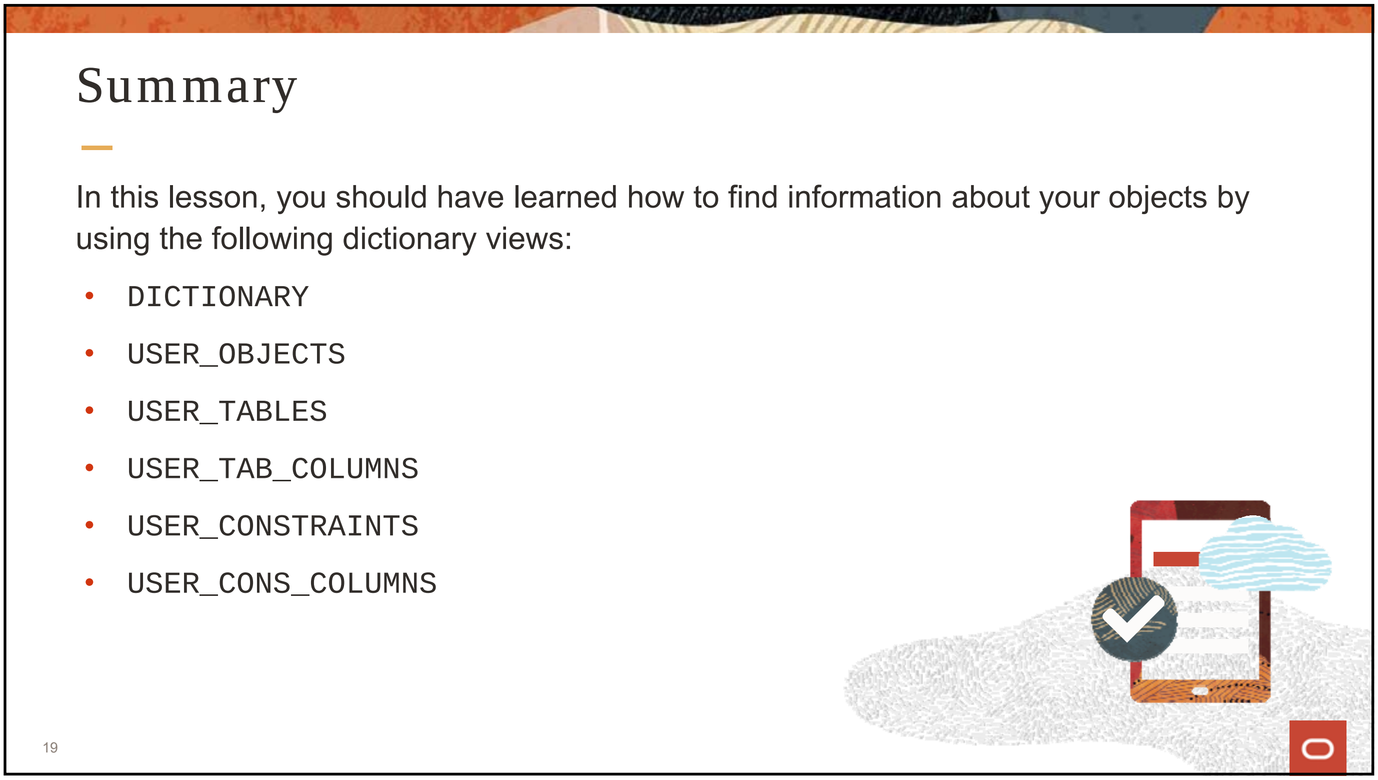
Adding Comments to a Table

- 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



Summary

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

- `DICTIONARY`
- `USER_OBJECTS`
- `USER_TABLES`
- `USER_TAB_COLUMNS`
- `USER_CONSTRAINTS`
- `USER_CONS_COLUMNS`