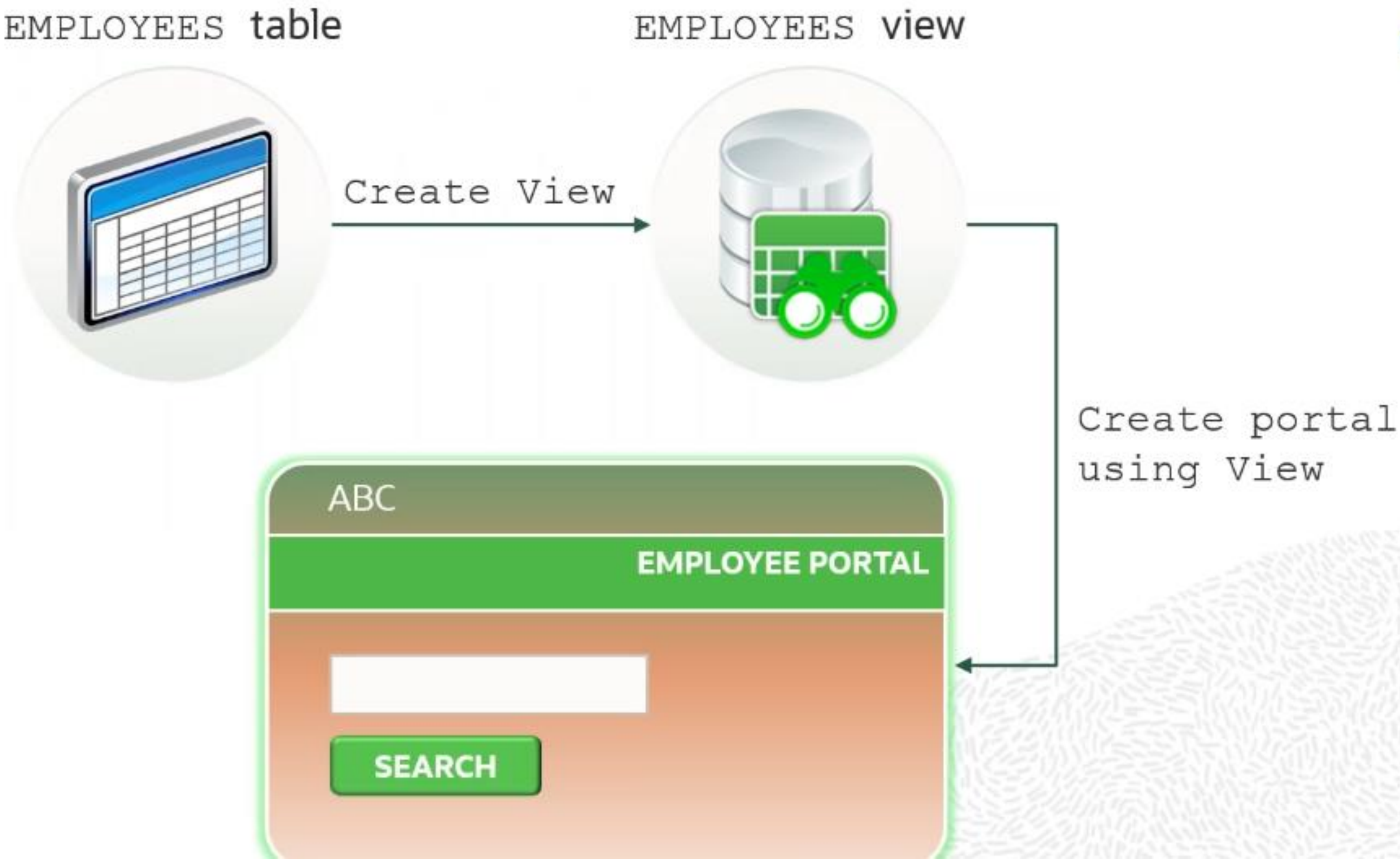
Creating Views

Objectives

After completing this lesson, you should be able to:

- Create simple and complex views
- Retrieve data from views
- Query dictionary views for view information

Why Views?

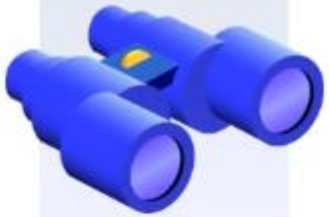


EMPLOYEES table

Create View

EMPLOYEES view

Create portal using View

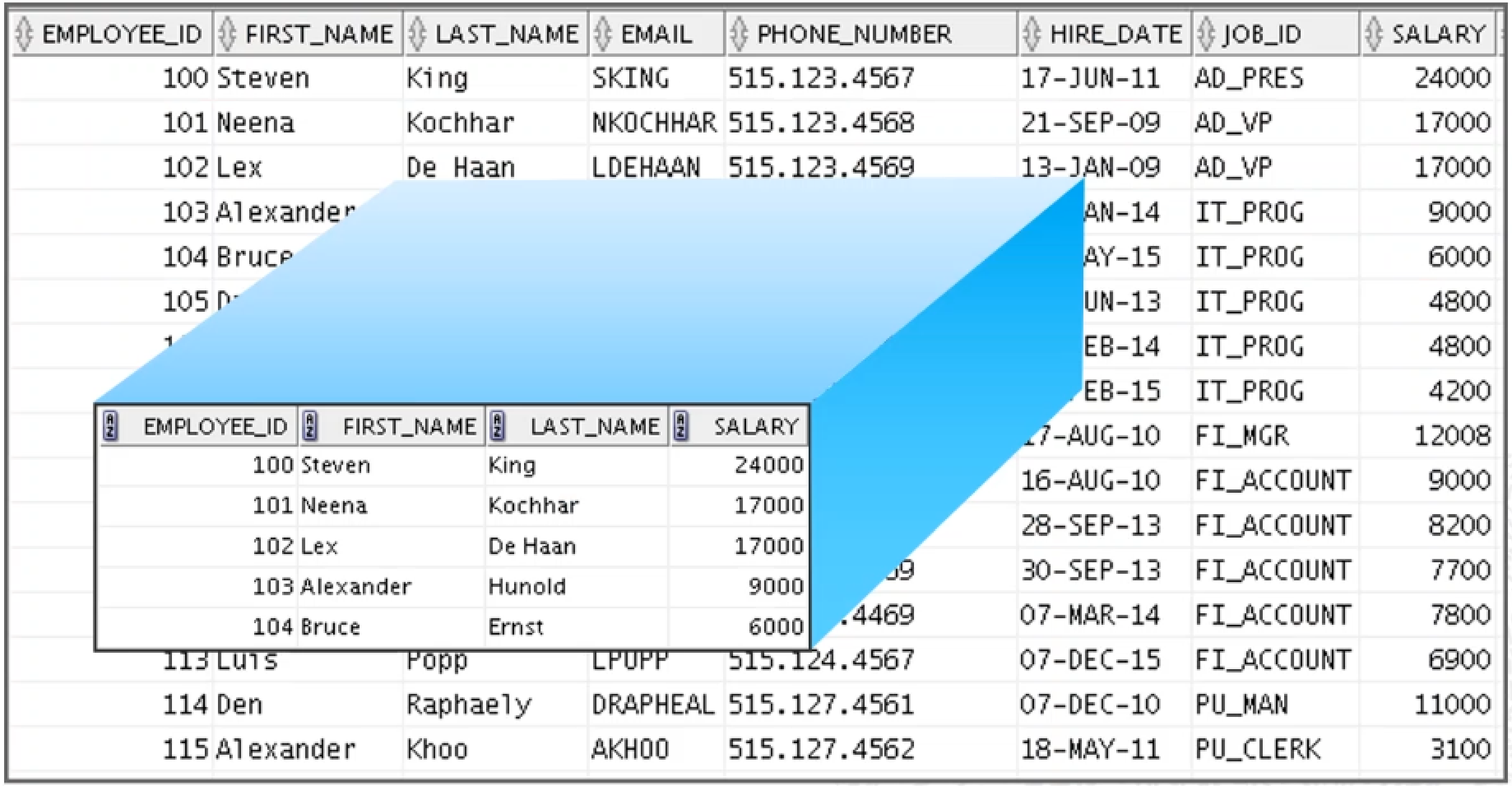


Database Objects

Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of data retrieval queries
Synonym	Gives alternative names to objects

What Is a View?

EMPLOYEES table



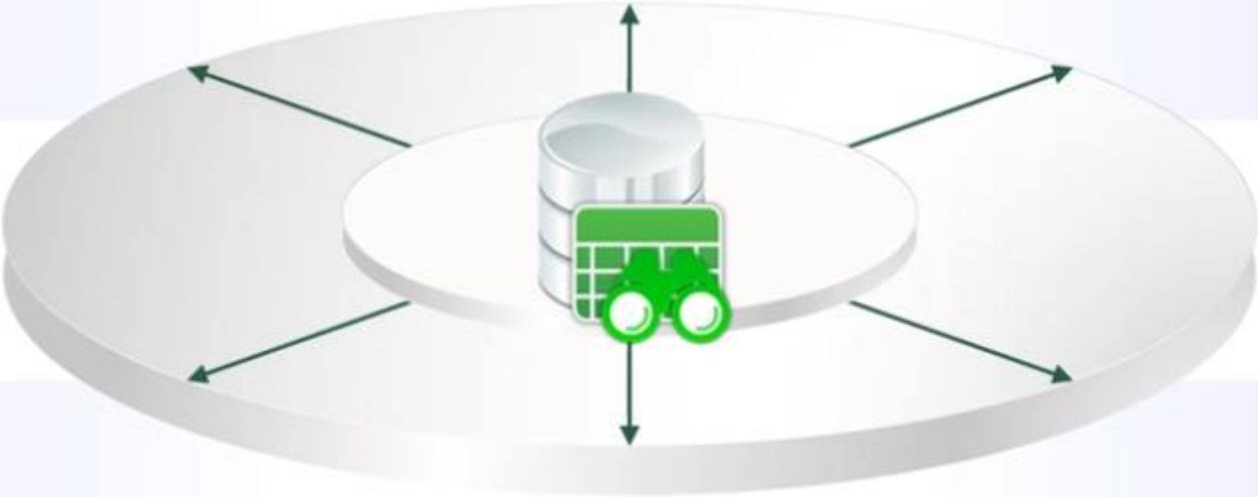
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17-JUN-11	AD_PRES	24000
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-09	AD_VP	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-09	AD_VP	17000
103	Alexander				AN-14	IT_PROG	9000
104	Bruce				AY-15	IT_PROG	6000
105	D				UN-13	IT_PROG	4800
1					EB-14	IT_PROG	4800
					EB-15	IT_PROG	4200
					7-AUG-10	FI_MGR	12008
					16-AUG-10	FI_ACCOUNT	9000
					28-SEP-13	FI_ACCOUNT	8200
				9	30-SEP-13	FI_ACCOUNT	7700
				.4469	07-MAR-14	FI_ACCOUNT	7800
113	Luis	Popp	LPOPP	515.124.4567	07-DEC-15	FI_ACCOUNT	6900
114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-10	PU_MAN	11000
115	Alexander	Khoo	AKHOO	515.127.4562	18-MAY-11	PU_CLERK	3100

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	17000
102	Lex	De Haan	17000
103	Alexander	Hunold	9000
104	Bruce	Ernst	6000

Advantages of Views

To restrict data access

To make complex queries easy



To provide data independence

To present different views of the same data

Simple Views and Complex Views

Feature	Simple Views	Complex Views
Number of tables	One	One or more
Contain functions	No	Yes
Contain groups of data	No	Yes
DML operations through a view	Yes	Not always

Lesson Agenda

- Overview of views
- Creating, modifying, and retrieving data from a view
- Data manipulation language (DML) operations on a view
- Dropping a view

Creating a View

- You embed a subquery in the `CREATE VIEW` statement:

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
  [(alias[, alias]...)]
 AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

- The subquery can contain complex `SELECT` syntax.

FORCE

Specify FORCE if you want to create the view regardless of whether the base tables of the view or the referenced object types exist or the owner of the schema containing the view has privileges on them. These conditions must be true before any SELECT, INSERT, UPDATE, or DELETE statements can be issued against the view.

If the view definition contains any constraints, CREATE VIEW ... FORCE will fail if the base table does not exist or the referenced object type does not exist. CREATE VIEW ... FORCE will also fail if the view definition names a constraint that does not exist.

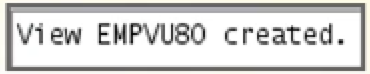
NO FORCE

Specify NOFORCE if you want to create the view only if the base tables exist and the owner of the schema containing the view has privileges on them. This is the default.

Creating a View

- Create the EMPVU80 view, which contains details of the employees in department 80:

```
CREATE VIEW  empvu80
 AS SELECT  employee_id, last_name, salary
    FROM    employees
    WHERE   department_id = 80;
```



```
View EMPVU80 created.
```

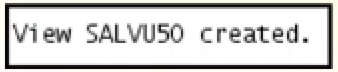
- Describe the structure of the view by using the SQL*Plus DESCRIBE command:

```
DESCRIBE empvu80;
```

Creating a View

- Create a view by using column aliases in the subquery:

```
CREATE VIEW  salvu50
 AS SELECT  employee_id ID_NUMBER, last_name NAME,
            salary*12 ANN_SALARY
    FROM    employees
    WHERE   department_id = 50;
```



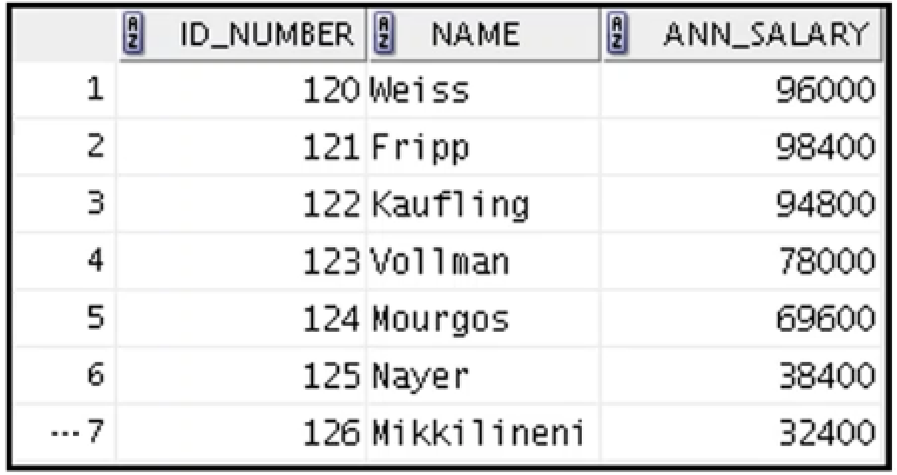
```
View SALVU50 created.
```

- Select the columns from this view by the given alias names.

```
SELECT ID_NUMBER, NAME, ANN_SALARY
FROM salvu50;
```

Retrieving Data from a View

```
SELECT *
FROM   salvu50;
```



	ID_NUMBER	NAME	ANN_SALARY
1	120	Weiss	96000
2	121	Fripp	98400
3	122	Kaufling	94800
4	123	Vollman	78000
5	124	Mourgos	69600
6	125	Nayer	38400
7	126	Mikkilineni	32400

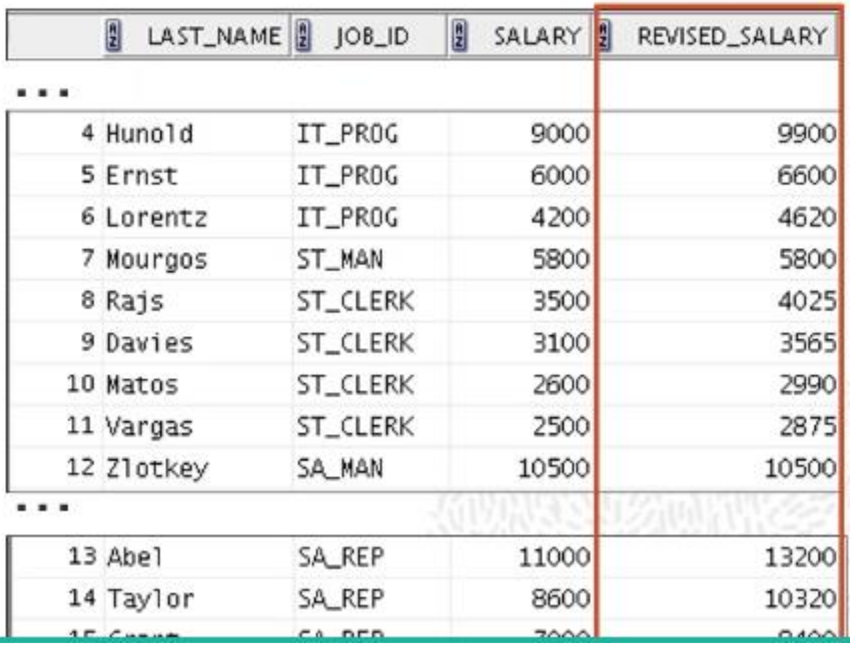
DECODE Function in Oracle

Facilitates conditional inquiries by doing the work of a CASE expression or an IF-THEN-ELSE statement:

```
DECODE(col|expression, search1, result1
                       [, search2, result2,...,]
                       [, default])
```

Using the DECODE Function

```
SELECT last_name, job_id, salary,
       DECODE(job_id, 'IT_PROG',  1.10*salary,
                      'ST_CLERK', 1.15*salary,
                      'SA_REP',   1.20*salary,
              salary)
       REVISED_SALARY
FROM   employees;
```



	LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
...				
4	Hunold	IT_PROG	9000	9900
5	Ernst	IT_PROG	6000	6600
6	Lorentz	IT_PROG	4200	4620
7	Mourgos	ST_MAN	5800	5800
8	Rajs	ST_CLERK	3500	4025
9	Davies	ST_CLERK	3100	3565
10	Matos	ST_CLERK	2600	2990
11	Vargas	ST_CLERK	2500	2875
12	Zlotkey	SA_MAN	10500	10500
...				
13	Abel	SA_REP	11000	13200
14	Taylor	SA_REP	8600	10320
15	Grant	SA_REP	7000	8400

Using the DECODE Function

Display the applicable tax rate for each employee in department 80:

```
SELECT last_name, salary,
       DECODE (TRUNC(salary/2000, 0),
                         0, 0.00,
                         1, 0.09,
                         2, 0.20,
                         3, 0.30,
                         4, 0.40,
                         5, 0.42,
                         6, 0.44,
                            0.45) TAX_RATE
FROM   employees
WHERE  department_id = 80;
```

Modifying a View

- Modify the EMPVU80 view by using a `CREATE OR REPLACE VIEW` clause. Add an alias for each column name:

```
CREATE OR REPLACE VIEW empvu80
  (id_number, name, sal, department_id)
AS SELECT  employee_id, first_name || ' '
           || last_name, salary, department_id
   FROM    employees
   WHERE   department_id = 80;
```

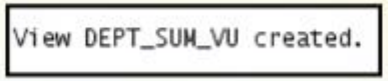
```
View EMPVU80 created.
```

- Column aliases in the `CREATE OR REPLACE VIEW` clause are listed in the same order as the columns in the subquery.

Creating a Complex View

Create a complex view that contains group functions to display values from two tables:

```
CREATE OR REPLACE VIEW dept_sum_vu
  (name, minsal, maxsal, avgsal)
AS SELECT   d.department_name, MIN(e.salary),
            MAX(e.salary),AVG(e.salary)
   FROM     employees e JOIN departments d
   USING    (department_id)
   GROUP BY d.department_name;
```

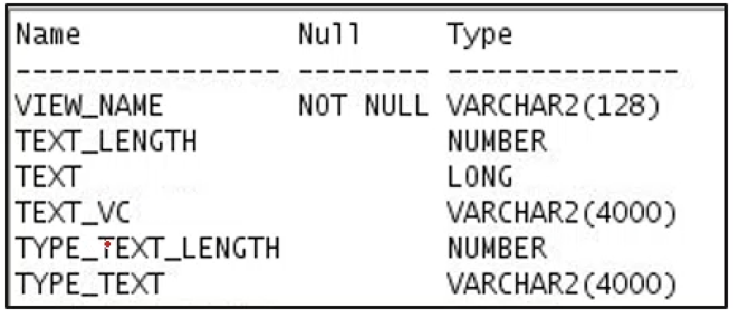


```
View DEPT_SUM_VU created.
```

View Information

1

```
DESCRIBE user_views
```

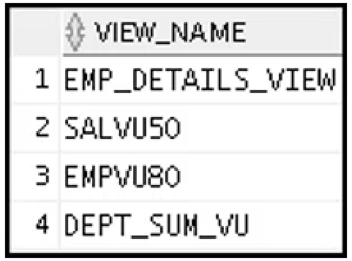


```
Name             Null     Type
---------------- -------- --------------
VIEW_NAME        NOT NULL VARCHAR2(128)
TEXT_LENGTH               NUMBER
TEXT                      LONG
TEXT_VC                   VARCHAR2(4000)
TYPE_TEXT_LENGTH          NUMBER
TYPE_TEXT                 VARCHAR2(4000)
```

...

2

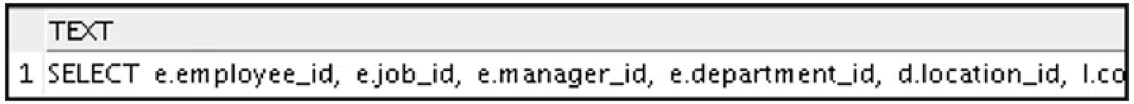
```
SELECT view_name FROM user_views;
```



```
  VIEW_NAME
1 EMP_DETAILS_VIEW
2 SALVU50
3 EMPVU80
4 DEPT_SUM_VU
```

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
```

...

Lesson Agenda

- Overview of views
- Creating, modifying, and retrieving data from a view
- Data manipulation language (DML) operations on a view
- Dropping a view

Rules for Performing DML Operations on a View

- You can usually perform DML operations on simple views.
- You cannot remove a row if the view contains the following:
  - Group functions
  - A GROUP BY clause
  - The DISTINCT keyword
  - The pseudocolumn ROWNUM keyword

Rules for Performing Modify Operations on a View

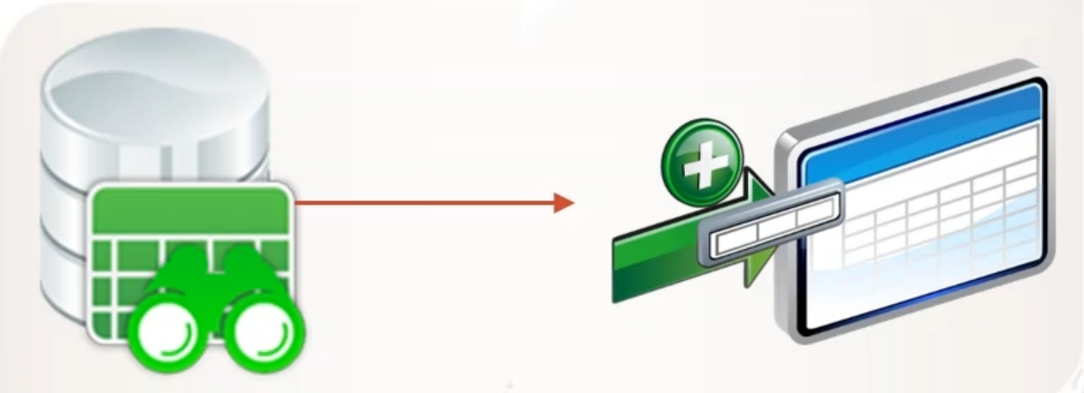
You cannot modify data in a view if it contains:

- Group functions
- A `GROUP BY` clause
- The `DISTINCT` keyword
- The pseudocolumn `ROWNUM` keyword
- Expressions

Rules for Performing Insert Operations Through a View

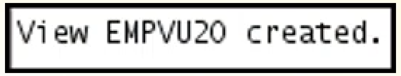
You cannot add data in a view if the view includes:

- Group functions
- A `GROUP BY` clause
- The `DISTINCT` keyword
- The pseudocolumn `ROWNUM` keyword
- Columns defined by expressions
- `NOT NULL` columns without default value in the base tables that are not selected by the view



Using the `WITH CHECK OPTION` Clause

```
CREATE OR REPLACE VIEW empvu20
AS SELECT *
   FROM     employees
   WHERE    department_id = 20
   WITH CHECK OPTION CONSTRAINT empvu20_ck ;
```



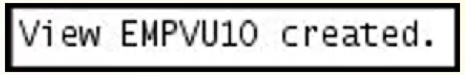
```
View EMPVU20 created.
```

Denying DML Operations

- You can ensure that no `DML` operations occur on your view by adding the `WITH READ ONLY` option to your view definition.
- Any attempt to perform a `DML` operation on any row in the view results in an Oracle server error.

Denying DML Operations

```
CREATE OR REPLACE VIEW empvu10
    (employee_number, employee_name, job_title)
AS SELECT employee_id, last_name, job_id
   FROM     employees
   WHERE    department_id = 10
   WITH READ ONLY ;
```



```
View EMPVU10 created.
```

Summary

In this lesson, you should have learned how to:

- Create, modify, and remove views
- Query the dictionary views for view information

Removing a View

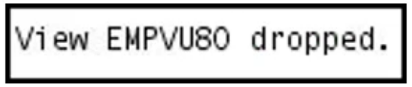
You can remove a view without losing data because a view is based on underlying tables in the database.

Syntax:

```
DROP VIEW view;
```

Example:

```
DROP VIEW empvu80;
```



```
View EMPVU80 dropped.
```