

# 11

## Creating Views

# Objectives

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

- **Describe a view**
- **Create, alter the definition of, and drop a view**
- **Retrieve data through a view**
- **Insert, update, and delete data through a view**
- **Create and use an inline view**
- **Perform “Top-N” analysis**

# Database Objects

Object	Description
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Generates primary key values
Index	Improves the performance of some queries
Synonym	Alternative name for an object

# 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-87	AD_PRES	24000
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	5800
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	3500
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	3100
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	2600
144	Timothy	Gietz	TGIEZT	650.121.2014	15-APR-98	ST_CLERK	2500
149	Zlotkey				29-JAN-00	SA_MAN	10500
174	Abel				24-MAY-96	SA_REP	11000
176	Taylor				15-MAR-98	SA_REP	8600
170	Kirderery	Grant	KGRANT	515.144.1644,429263	24-MAY-99	SA_REP	7000
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	6000
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000
206	William	Gietz	WGIEZT	515.123.8181	07-JUN-94	AC_ACCOUNT	8300

20 rows selected.

# Why Use 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

# Creating a View

- You embed a subquery within 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.

# Creating a View

- Create a view, EMPVU80, that contains details of employees in department 80.

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

View created.

- Describe the structure of the view by using the *iSQL\*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 created.

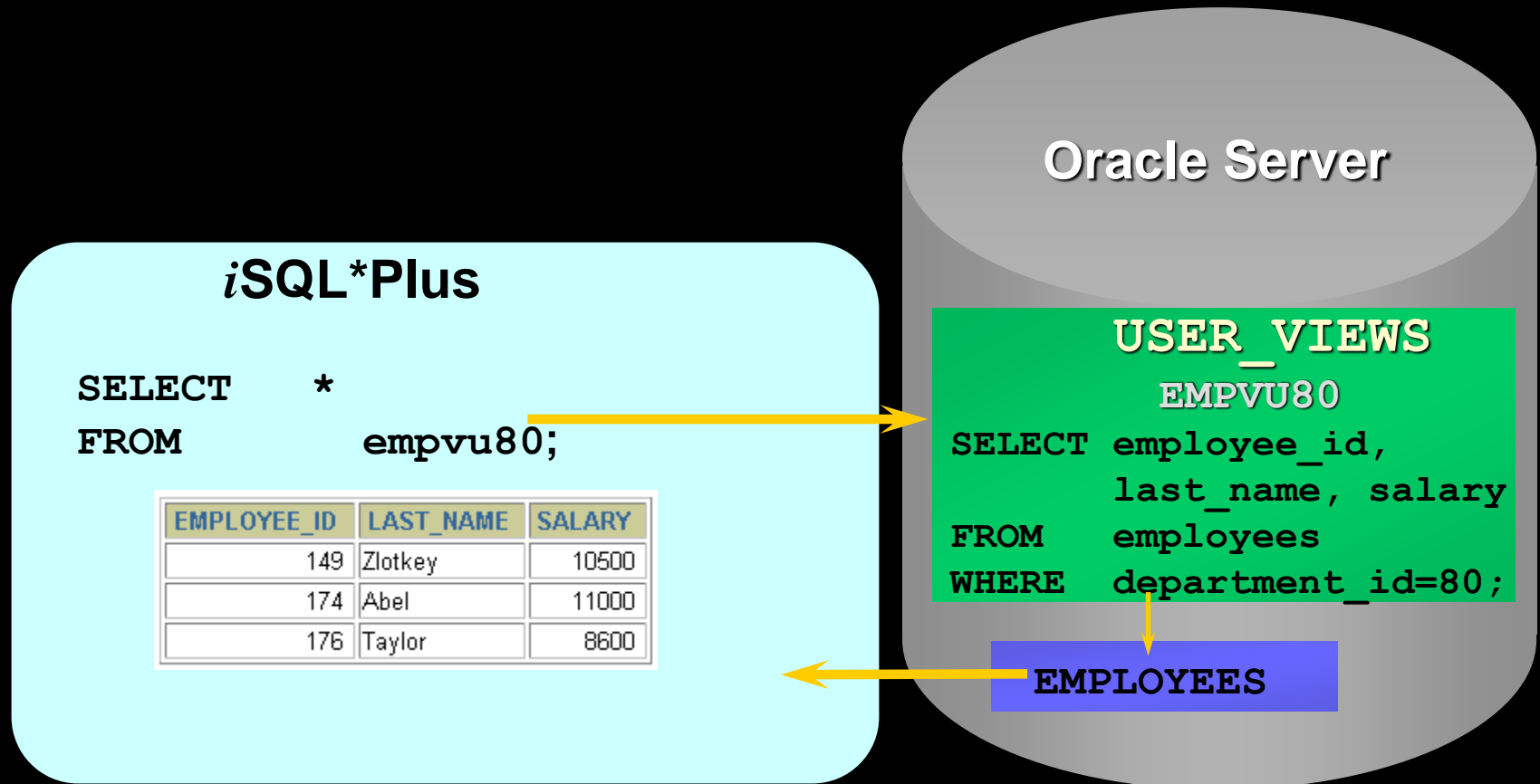
- Select the columns from this view by the given alias names.

# Retrieving Data from a View

```
SELECT *  
FROM salvu50;
```

ID_NUMBER	NAME	ANN_SALARY
124	Mourgos	69600
141	Rajs	42000
142	Davies	37200
143	Matos	31200
144	Vargas	30000

# Querying a View



# Modifying a View

- **Modify the EMPVU80 view by using 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 created.**

- **Column aliases in the CREATE 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 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, departments d
  WHERE         e.department_id = d.department_id
  GROUP BY     d.department_name;
```

View created.

# Rules for Performing DML Operations on a View

- You can 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 DML 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**
- **Columns defined by expressions**

# Rules for Performing DML Operations on a View

**You cannot add data through 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 in the base tables that are not selected by the view**



# Using the WITH CHECK OPTION Clause

- You can ensure that DML operations performed on the view stay within the domain of the view by 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 created.

- Any attempt to change the department number for any row in the view fails because it violates the WITH CHECK OPTION constraint.

# Denying DML Operations

- You can ensure that no DML operations occur by adding the **WITH READ ONLY** option to your view definition.
- Any attempt to perform a DML 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 created.

# Removing a View

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

```
DROP VIEW view;
```

```
DROP VIEW empvu80;  
View dropped.
```

# Inline Views

- **An inline view is a subquery with an alias (or correlation name) that you can use within a SQL statement.**
- **A named subquery in the `FROM` clause of the main query is an example of an inline view.**
- **An inline view is not a schema object.**

# Top-N Analysis

- **Top-N queries ask for the  $n$  largest or smallest values of a column. For example:**
  - What are the ten best selling products?
  - What are the ten worst selling products?
- **Both largest values and smallest values sets are considered Top-N queries.**

# Performing Top-N Analysis

The high-level structure of a Top-N analysis query is:

```
SELECT [column_list], ROWNUM
FROM   (SELECT [column_list]
        FROM table
        ORDER BY Top-N_column)
WHERE  ROWNUM <= N;
```

# Example of Top-N Analysis

To display the top three earner names and salaries from the EMPLOYEES table:

1 2 3

```
SELECT ROWNUM as RANK, last_name, salary
FROM (SELECT last_name,salary FROM employees
      ORDER BY salary DESC)
WHERE ROWNUM <= 3;
```

RANK	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000

1 2 3



# Summary

**In this lesson, you should have learned that a view is derived from data in other tables or views and provides the following advantages:**

- **Restricts database access**
- **Simplifies queries**
- **Provides data independence**
- **Provides multiple views of the same data**
- **Can be dropped without removing the underlying data**
- **An inline view is a subquery with an alias name.**
- **Top-N analysis can be done using subqueries and outer queries.**

# Practice 11 Overview

**This practice covers the following topics:**

- **Creating a simple view**
- **Creating a complex view**
- **Creating a view with a check constraint**
- **Attempting to modify data in the view**
- **Displaying view definitions**
- **Removing views**