The Views

- Jayendra Khatod

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**

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	No

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 DESCRIBE command.**

```
DESCRIBE empvu80
```

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.
```

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 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**

Removing a View

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

```
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

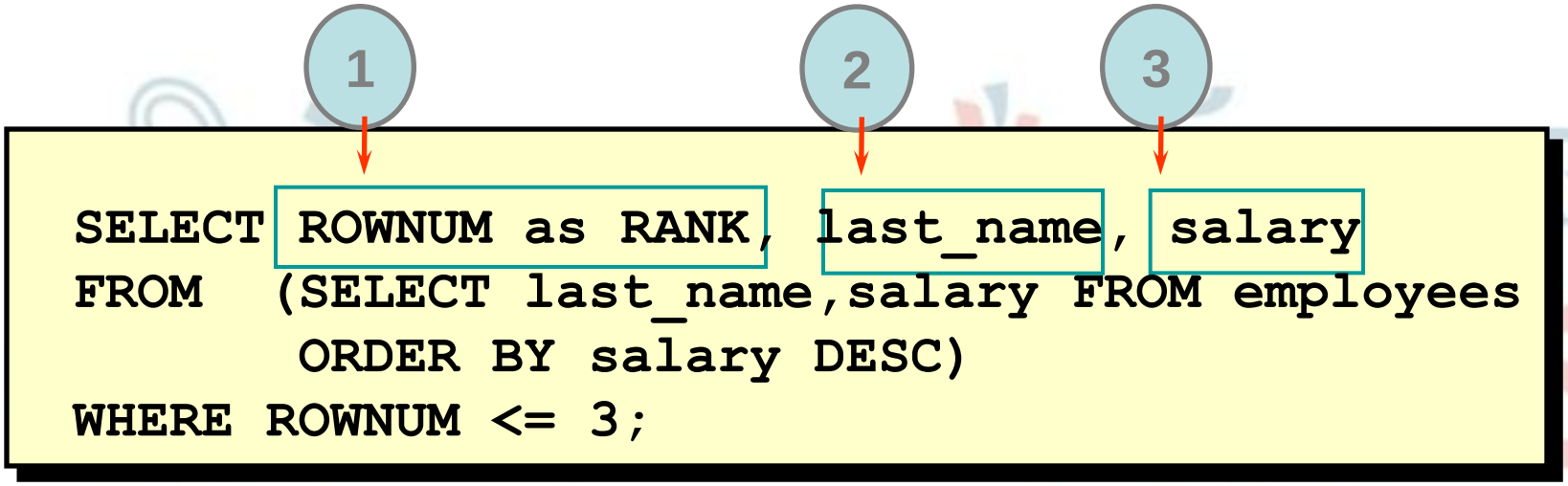
- **Top-N queries ask for the *n* largest or smallest values of a column.**
- **For example:**
  - **What are the ten best selling products?**
  - **What are the ten worst selling products?**
- **Both largest values and smallest values sets are considered Top-N queries.**

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

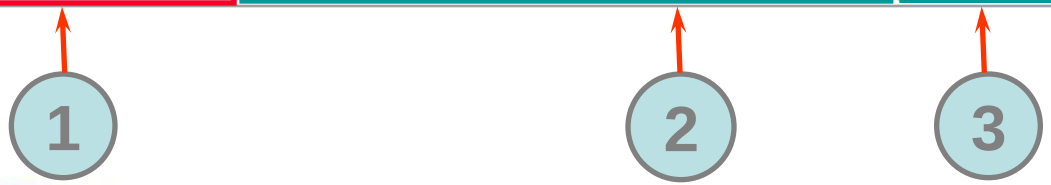
```
SELECT [column_list], ROWNUM
FROM   (SELECT [column_list]
        FROM table
        ORDER  BY Top-N_column)
WHERE  ROWNUM <=  N;
```

Example of Top-N Analysis

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



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



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.

Thank You !