

The logo for Oracle Academy. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is centered on a light gray background, which is framed by dark gray horizontal bars at the top and bottom.

ORACLE

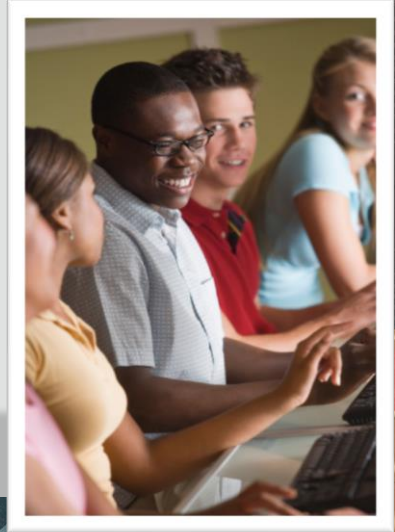
Academy

Database Programming with SQL

15-3

Managing Views

ORACLE
Academy



Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

Objectives

- This lesson covers the following objectives:
 - Create and execute a SQL statement that removes a view
 - Create and execute a query using an inline view
 - Create and execute a top-n-analysis query

Purpose

- Learning to create and replace views wouldn't be complete unless you also knew how to remove them
- Views are created for specific purposes
- When the view is no longer needed or needs to be modified, the means exist to make the necessary changes
- If an employee who had access to financial information leaves the company, you probably don't want his view to remain accessible
- In this lesson, you will learn how to delete a view, create an inline view, and construct a SELECT statement to produce a sorted list of data

Deleting a View

- Because a view contains no data of its own, removing it does not affect the data in the underlying tables
- If the view was used to INSERT, UPDATE, or DELETE data in the past, those changes to the base tables remain
- Deleting a view simply removes the view definition from the database



Deleting a View

- Remember, views are stored as SELECT statements in the data dictionary
- Only the creator or users with the DROP ANY VIEW privilege can remove a view
- The SQL syntax to remove a view is:

```
DROP VIEW viewname;
```

Inline Views

- Inline views are also referred to as subqueries in the FROM clause
- You insert a subquery in the FROM clause just as if the subquery was a table name
- Inline views are commonly used to simplify complex queries by removing join operations and condensing several queries into one

Inline Views

- As shown in the example below, the FROM clause contains a SELECT statement that retrieves data much like any SELECT statement
- The data returned by the subquery is given an alias (d), which is then used in conjunction with the main query to return selected columns from both query sources

```
SELECT e.last_name, e.salary, e.department_id, d.maxsal
FROM employees e,
     (SELECT department_id, max(salary) maxsal
      FROM employees
      GROUP BY department_id) d
WHERE e.department_id = d.department_id
AND e.salary = d.maxsal;
```

This inline view finds the highest salary for each department, and the query then displays the name of the employee with that salary.

An inline view must have an alias (in this example, "d") because it functions like a table name in the FROM clause, and "SELECT department_id, max(salary) ..." is not a valid table name!

TOP-N-ANALYSIS

- Top-n-analysis is a SQL operation used to rank results
- The use of top-n-analysis is useful when you want to retrieve the top 5 records, or top-n records, of a result set returned by a query

```
SELECT ROWNUM AS "Longest employed", last_name, hire_date
FROM employees
WHERE ROWNUM <=5
ORDER BY hire_date;
```

Longest employed	LAST_NAME	HIRE_DATE
1	King	17-Jun-1987
4	Whalen	17-Sep-1987
2	Kochhar	21-Sep-1989
3	De Haan	13-Jan-1993
5	Higgins	07-Jun-1994

The results of this query are not however what you would expect. The reason for this is that the ORDER BY clause always executes last, so the rows are ordered after they are given a number. The next slide demonstrates how to resolve this issue.

TOP-N-ANALYSIS

- The top-n-analysis query uses an inline view (a subquery) to return a result set
- You can use ROWNUM in your query to assign a row number to the result set
- The main query then uses ROWNUM to order the data and return the top five

```
SELECT ROWNUM AS "Longest employed", last_name, hire_date
FROM (SELECT last_name, hire_date
      FROM employees
      ORDER BY hire_date)
WHERE ROWNUM <=5;
```

By using an inline view, the ORDER BY clause is performed before the ROWNUM is added.

TOP-N-ANALYSIS

Longest employed	LAST_NAME	HIRE_DATE
1	King	17-Jun-1987
2	Whalen	17-Sep-1987
3	Kochhar	21-Sep-1989
4	Hunold	03-Jan-1990
5	Ernst	21-May-1991

- In the example above, the inline view first selects the list of last_names and hire_dates of the employees:

```
(SELECT last_name, hire_date FROM employees...
```

- Then the inline view orders the years from oldest to newest

```
...ORDER BY hire_date)
```

TOP-N-ANALYSIS

- The outer query WHERE clause is used to restrict the number of rows returned and must use a < or <= operator

```
SELECT ROWNUM AS "Longest employed", last_name, hire_date
FROM (SELECT last_name, hire_date
      FROM employees
      ORDER BY hire_date)
WHERE ROWNUM <=5;
```

Longest employed	LAST_NAME	HIRE_DATE
1	King	17-Jun-1987
2	Whalen	17-Sep-1987
3	Kochhar	21-Sep-1989
4	Hunold	03-Jan-1990
5	Ernst	21-May-1991

Terminology

- Key terms used in this lesson included:
 - DROP VIEW
 - INLINE VIEW
 - TOP-N ANALYSIS
 - ROWNUM

Summary

- In this lesson, you should have learned how to:
 - Create and execute a SQL statement that removes a view
 - Create and execute a query using an inline view
 - Create and execute a top-n-analysis query

The logo for Oracle Academy. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is centered on a light gray background, which is framed by dark gray horizontal bars at the top and bottom.

ORACLE

Academy