Database Programming

Managing Views

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.

Purpose (cont.)

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. 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 (cont.)

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 (p), which is then used in conjunction with the main query to return selected columns from both query sources.

```
SELECT e.name, e.description, p.maxrange, p.code
FROM d_events e, (SELECT code, max(high_range) maxrange
                                  FROM d_packages
                                  GROUP BY code) p
WHERE e.package_code = p.code
AND e.cost < p.maxrange;
```

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 4 records, or top-n records, of a result set returned by a query.

```
SELECT ROWNUM as RANK, year, title
FROM (SELECT year, title
             FROM d_cds
             ORDER BY year)
WHERE ROWNUM <= 4;
```

TOP-N-ANALYSIS (cont.)

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 four.

```
SELECT ROWNUM as RANK, year, title
FROM (SELECT year, title
             FROM d_cds
             ORDER BY year)
WHERE ROWNUM <= 4;
```

TOP-N-ANALYSIS (cont.)

```
SELECT ROWNUM as RANK, year, title
FROM (SELECT year, title
             FROM d_cds
             ORDER BY year)
WHERE ROWNUM <= 4;
```

In the example above, the inline view first selects the list of years and titles of the DJs on Demand's CDs:

```
(SELECT year, title FROM d_cds ,,,,,,)
```

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

```
(SELECT …… ORDER BY year)
```

TOP-N-ANALYSIS (cont.)

```
SELECT ROWNUM as RANK, year, title
FROM (SELECT year, title
             FROM d_cds
             ORDER BY year)
WHERE ROWNUM <= 4;
```

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

```
WHERE ROWNUM <= 4;
```

Terminology

Key terms used in this lesson included:

- DROP VIEW
- INLINE VIEW
- TOP-N ANALYSIS

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