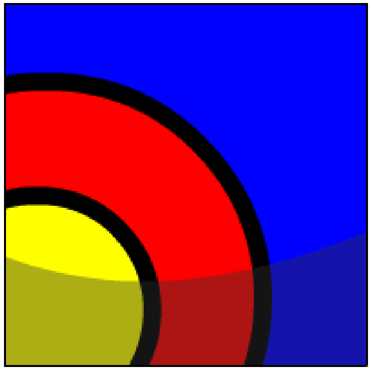
Managing Views

What Will I Learn?

In this lesson, you will learn to:



- Create and execute a SQL statement that removes a view
- Create and execute a query to create an inline view
- Create and execute a top-n-analysis query

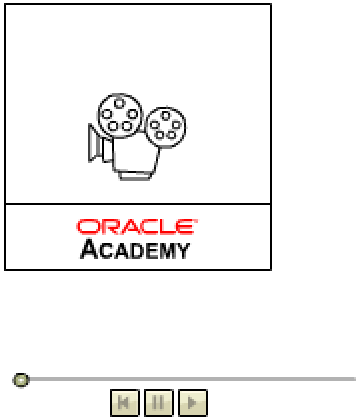
Why Learn It?

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, there has to be a way to make the necessary changes.

If an employee who had access to financial information leaves the company, you probably don’t want his/her 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.



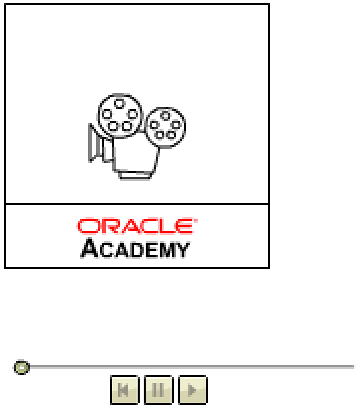
Tell Me / Show Me

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, these 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;
```



Tell Me / Show Me

INLINE VIEWS

Inline views are also referred to as queries in the FROM clause.

You insert a query in the FROM clause just as if the query was a table name.

Inline views are commonly used to simplify complex queries by removing join operations and condensing several queries into a single query.

Tell Me / Show Me

INLINE VIEWS (continued)

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;
```

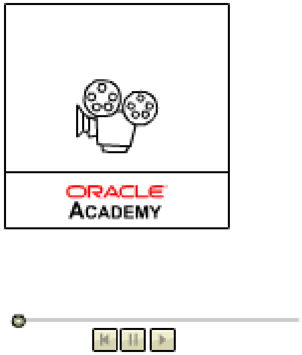
Tell Me / Show Me

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

The top-n-analysis query uses an inline 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;
```



Tell Me / Show Me

```
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 subquery first selects the list of years and titles of the DJ on Demand's CDs:

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

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

```
(SELECT …… ORDER BY year)
```

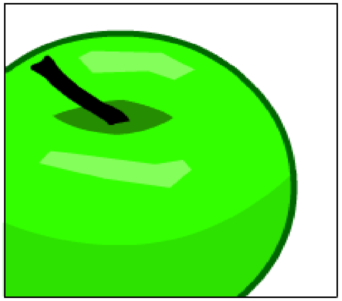
The outer query WHERE clause is used to restrict the number of rows returned and must use a < or <= operator.

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

Tell Me / Show Me

Terminology

Key terms used in this lesson include:

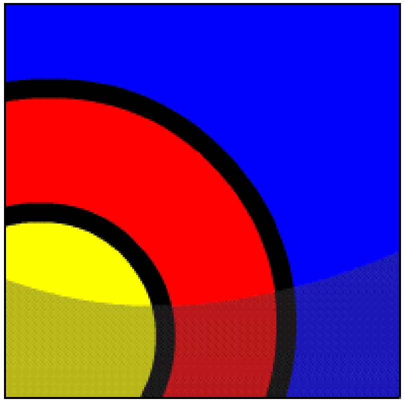


INLINE VIEW

DROP VIEW

TOP-N ANALYSIS

Summary



In this lesson you have learned to:

- Create and execute a SQL statement that removes a view
- Create and execute a query to create an inline view
- Create and execute a top-n-analysis query

Summary

Practice Guide

The link for the lesson practice guide can be found in the course resources in Section 0.