

Views

In Oracle SQL, a view is a virtual table based on the result set of an SQL query. It behaves much like a regular table, with a few key differences. A view does not store data physically; it is a stored query that dynamically retrieves data from one or more tables at runtime. Views can encapsulate complex queries, providing a simpler interface to access data, and can also be used to restrict access to certain rows or columns of data for security purposes.

Creating a View

To create a view, you use the `CREATE VIEW` statement followed by the view name and the query that defines the view. Here is a basic example:

```
CREATE VIEW my_view AS  
SELECT column1, column2  
FROM my_table  
WHERE condition;
```

This creates a view named `my_view` that will show all rows from `my_table` that meet the specified condition, but only the columns `column1` and `column2`.

Updating a View

Oracle allows views to be updated under certain conditions. If a view is based on a single table and includes key-preserved tables, you can perform DML operations (INSERT, UPDATE, DELETE) on the view. However, if the view involves complex calculations, joins, or does not include all key columns, updates may not be directly possible.

With Check Option

When updating a view, you can use the `WITH CHECK OPTION` clause to ensure that all data modifications through the view conform to the view's defining query. This prevents updates that would make the data invisible through the view.

```
CREATE VIEW my_secure_view AS  
  
SELECT column1, column2  
  
FROM my_table  
  
WHERE condition  
  
WITH CHECK OPTION;
```

This ensures that any attempts to insert or update data through `my_secure_view` must satisfy the `condition`.

Dropping a View

To remove a view, you use the `DROP VIEW` statement:

```
DROP VIEW my_view;
```

This deletes the view definition from the database.

Benefits of Using Views

- **Simplification of Complex Queries:** Views can encapsulate complex queries, making it easier for users to access data without understanding the underlying complexities.
- **Security:** Views can restrict access to specific rows or columns, offering a layer of security by limiting data exposure to users.
- **Logical Data Independence:** Views can provide a consistent, unchanging interface to data, even if the underlying database schema changes, thus protecting applications from changes in the schema.

Views are a powerful feature in Oracle SQL, offering flexibility, security, and simplicity for database operations.