Day 9: SQL Notes Basic to Advanced

Todays Topic: SQL Views

- Creating a View
- Querying a View
- Updating Data through a View
- Dropping a View

SQL Views

A SQL View is essentially a virtual table that provides a way to look at data from one or more tables in a customized manner.

It doesn't store data itself but rather displays data from the underlying tables based on a query that defines the view.

- A virtual table created by a query on one or more tables.
- Simplifies complex queries by encapsulating them into a single view.
- Can provide a layer of security by restricting access to certain data.
- Allows for easier management and reusability of query logic.

Key Characteristics of SQL Views

Virtual Table: A view behaves like a table in SQL, allowing you to query it as if it were an actual table, but it doesn't physically store the data. Instead, it dynamically retrieves data from the underlying tables each time it is queried.

Simplified Queries: Views can simplify complex queries. For instance, if you have a complex join that you frequently use, you can create a view to encapsulate that logic, allowing you to query the view without repeating the join every time.

Security: Views can be used to restrict access to certain data. For example, if you want to allow users to see only specific columns or rows from a table, you can create a view that only includes the permitted data.

Data Abstraction: Views provide a level of abstraction from the underlying table structures. If the underlying tables change, the view can be updated to reflect those changes without affecting the queries that use the view.

EmployeeID	FirstName	LastName	Department	Salary	HireDate
101	John	Doe	IT	80000	2021-03-15
102	Jane	Smith	HR	75000	2019-06-01
103	Emily	Davis	IT	90000	2020-11-20
104	Michael	Johnson	Marketing	70000	2018-04-10
105	Jessica	Brown	HR	78000	2022-01-22

Creating a View

You create a view using the CREATE VIEW statement, which is followed by the view name and the SELECT statement that defines the data the view will display.

Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Example: Creating a Simple View

Suppose you want to create a view that shows only the employees in the IT department.

```
CREATE VIEW IT_Employees AS
SELECT EmployeeID, FirstName, LastName, Salary
FROM Employees
WHERE Department = 'IT';
```

Querying a View

You can query a view just like a regular table. you can query the view IT_Employees to see the details of employees in the IT department

```
SELECT * FROM IT_Employees;
```

Result:

EmployeeID	FirstName	LastName	Salary
101	John	Doe	80000
103	Emily	Davis	90000

Updating Data through a View

In some cases, you can also update data in the underlying tables through a view, as long as the view meets certain criteria (like having a direct reference to the underlying table's columns).

```
UPDATE IT_Employees
SET Salary = 85000
WHERE EmployeeID = 101;
```

Dropping a View

If you no longer need the view, you can drop it.

```
DROP VIEW view_name;
```

Example

```
DROP VIEW IT_Employees;
```

Benefits of Using Views

- **Simplification:** Views simplify complex SQL queries.

- **Security:** Views can limit access to sensitive data.

- **Consistency:** Views can ensure consistent data presentation across multiple queries.

- **Data Independence:** Views can provide a consistent interface to data even if the underlying table structures change.

Limitations of Views

- **Performance:** Views can impact performance, especially if they are complex or involve multiple joins.
- **Complexity:** While views can simplify individual queries, they can also add a layer of complexity to your database schema.
- **Restrictions:** Not all views are updatable, especially if they involve complex joins, aggregations, or non-deterministic functions.

SQL Views are powerful tools for organizing, securing, and simplifying access to data.

Day 10:SQL Indexes

Will Post Tomorrow Follow For More

Want This PDF?

Join Our Telegram and Download 100+ PDFs and Handwritten Notes