View

A note by
Shah Md. Arshad Rahman Ziban

View

Basic Info

A view in database management is a virtual relation that provides a mechanism to hide certain data from users. It allows users to access and manipulate data without directly interacting with the underlying tables. Here are the key points about views:

Definition

A view is a virtual table created from the result of a query that retrieves data from one or more base tables. It does not store data itself but presents it dynamically.

Purpose

Views serve several important functions:

Simplification : They simplify complex queries, making it easier to access and manipulate data.
Security : Views can restrict access to specific data, allowing users to see only what they are permitted to view.
Presentation : They format data in a way that is tailored to specific needs or requirements.

Creation

To create a view, use the following syntax:

```
CREATE VIEW ViewName AS <query expression>;
```

Example:

```
CREATE VIEW Student_Session AS SELECT Name, Session FROM student;
```

Modification

To modify a view, use the ALTER keyword:

```
ALTER VIEW Student_Session AS SELECT ID, Name FROM student WHERE Session='Fall-17';
```

Deletion

To delete a view, use the DROP keyword:

```
DROP VIEW Student_Session;
```

Update

To delete a view, use the UPDATE keyword:

```
UPDATE Student_Session SET Session = 'Spring-18' WHERE Name = 'John';
```

Multiple Tables

To create a view named `student_info` that joins two tables, you can use the following SQL query. Assuming you have two tables, `tblEmployee` and `tblDepartment`, and you want to join them based on the `DepartmentId` from `tblEmployee` and `DeptId` from `tblDepartment`, the SQL statement would look like this:

```
CREATE VIEW student_info AS
SELECT
    tblEmployee.ID,
    tblEmployee.Name,
    tblEmployee.DepartmentId,
    tblDepartment.DeptName
FROM
    tblEmployee
JOIN
    tblDepartment
ON
    tblEmployee.DepartmentId = tblDepartment.DeptId;
```

Why View Virtual Relation.

A view is considered a virtual relation because:

No Physical Storage: It does not store data; instead, it retrieves data from underlying tables when queried.
Dynamic Data: It always reflects the current data in the base tables, showing real-time updates.
Abstraction: It simplifies complex queries by providing a straightforward interface to the data.
Security: It can restrict access to specific data, hiding sensitive information from users.
Data Combination: It can combine data from multiple tables, making it easier to work with related information.

In essence, a view acts as a dynamic, query-based representation of data without being a physical table.

Advantages and Disadvantages

Here’s a simplified overview of the advantages and disadvantages of views in database management:

Advantages of Views:

Simplifies Data Access: Views make it easier to work with complex data by presenting it in a simpler format.
Enhances Security: They can hide sensitive information, allowing users to see only what they need.
Always Up-to-Date: Views show the latest data from the underlying tables whenever they are accessed.
Combines Data: Views can pull together data from multiple tables, making it easier to analyze related information.

Disadvantages of Views:

Slower Performance: Accessing data through views can be slower than accessing it directly from tables, especially if the view is complex.
Update Issues: Making changes to data through a view can be tricky, especially if the view is based on multiple tables.
Dependent on Base Tables: If the underlying tables change, the view may need to be updated or could become invalid.
Limited Functionality: Some operations may not be possible with views, which can restrict their use in certain situations.

In short, views are useful for simplifying data access and enhancing security, but they can have performance drawbacks and limitations when it comes to updates.