Name: Sayali Bhosale
Date: NOV 19, 2024
Course: Foundations Of Databases & SQL Programming

Assignment 06- Views

Introduction:

In this module, we learned that a VIEW in SQL is a virtual table that displays data from a query without storing it. Views simplify access to data from one or more tables and are valuable for abstracting complex queries, ensuring consistent results, and enhancing security by allowing limited access to underlying tables. They provide reusable data representations, making retrieval, analysis, and sharing easier.

Explain when you would use a SQL View.

SQL Views make data access easier, more secure, and consistent, especially when dealing with complex databases or sensitive information.

- Simplify Complex Queries: Use a view to combine data from multiple tables or perform calculations. For example, instead of writing a long 'JOIN' query every time, you can save the query as a view and query the view directly.

- Enhance Security: Views allow you to control what users can see. For example, you can create a view that shows only specific columns (e.g., 'Name' and 'Email') without exposing sensitive information like 'Salary'.

- Reusability: If you need to use the same query multiple times, creating a view avoids duplication. You can use the view wherever you need the query results.

- Provide a Simplified Interface: Use views to present data in a more user-friendly way, hiding the complexity of the underlying tables and relationships.

- Maintain Data Consistency: By using views, you ensure that all users access the same version of a query, reducing errors and inconsistencies.

Explain the differences and similarities between a View, Function, and Stored Procedure.

Here's a brief explanation of the differences and similarities between a View, Function, and Stored Procedure:

View
- Purpose: A virtual table that displays data from one or more tables based on a predefined query.
- Use Case: Simplify complex queries or restrict access to specific columns/rows.
- Key Features:
  - Can be queried like a table ('SELECT * FROM ViewName').
  - Does not store data; it dynamically retrieves it when queried.
  - Cannot accept parameters.
  - Primarily used for data abstraction and security.

Function
Purpose: A reusable piece of code that performs a specific task and returns a value or result set.
Use Case: Calculate values, process data, or return a table.
Key Features:
- Must return a value (scalar value or a table).
- Can accept input parameters.
- Typically used in 'SELECT' statements or as part of a query.
- Cannot modify data (read-only).

Stored Procedure
Purpose: A reusable piece of code designed to perform a series of operations (e.g., querying, inserting, updating, deleting data).
Use Case: Execute a sequence of SQL commands or complex business logic.
Key Features:
- Can accept input and output parameters.
- Can perform multiple actions, including modifying data.
- Does not need to return a value but can return one or more result sets or status codes.
- Used for automation and complex operations.

Similarities:
- Reusable: All three are reusable, helping reduce repetitive code.
- Simplify Complexity: They abstract complex SQL logic for easier use.
- Database Objects: All are part of the database schema and can be used to interact with data.

Difference:
The key differences between a View, Function, and Stored Procedure are based on their purposes and capabilities.
- A View is a virtual table that simplifies data retrieval by storing a query definition; it cannot accept parameters or modify data.
- A Function performs calculations or transformations, always returning a scalar or table value. It can be used within queries but cannot modify data or perform multiple operations.
- A Stored Procedure is versatile and can execute complex operations like inserting, updating, or deleting data. It accepts parameters and returns output results, but it cannot be used directly within a query.

Summary:

A view in SQL is a virtual table that displays data from one or more tables based on a predefined query. It does not store data but presents results dynamically when accessed. Views simplify complex queries, enhance data security by limiting access to specific columns or rows, and provide a consistent way to organize and retrieve data, making it easier for users to interact with the database.