

Marina Osiechko

November 19, 2024

Foundations Of Databases & SQL Programming

Assignment 06

GitHubLink TBD

Mastering Structured Query Language: Understanding Views, Functions, and Stored Procedures

Introduction

Structured Query Language (SQL) is essential for database management and data manipulation, forming the backbone of data-driven decision-making. This paper explores the significant roles of views, functions, and stored procedures in enhancing SQL's power and flexibility.

SQL View

In SQL, a view is a virtual table based on the result set of a SQL statement. SQL views provide a versatile mechanism for managing and simplifying database interactions. They can encapsulate complex logic, enhance security, provide a layer of abstraction, enforce data consistency, facilitate reporting, and support legacy systems. Utilizing views effectively can lead to a more manageable, secure, and efficient database environment.

The Differences and Similarities between a View, Function, and Stored Procedure

Views, functions, and stored procedures are essential components in SQL and relational databases. Each has its unique purpose, but they also share some common characteristics. Let's explore their differences and similarities.

Views are virtual tables based on a SELECT query. They do not store data but provide a way to present data from one or more tables. They simplify complex queries by encapsulating them in a virtual table, provide a layer of security by restricting access to specific rows or columns, and maintain a consistent format for reporting and analysis. The following please see an example of a simple view:

1. `CREATE VIEW EmployeeDetails AS`
2. `SELECT EmployeeID, EmployeeName, Department`
3. `FROM Employees;`

Functions are database objects that perform a specific task and return a single value or a table. They can be used in SQL statements wherever expressions are allowed. Functions encapsulate reusable logic for calculations, data transformations, or aggregations. They can be used in SELECT, WHERE, and other clauses to perform operations on data. The following please see an example of a simple function:

1. CREATE FUNCTION CalculateBonus (@salary DECIMAL)
2. RETURNS DECIMAL
3. AS
4. BEGIN
5. RETURN @salary * 0.10;
6. END;

Stored procedures are precompiled collections of one or more SQL statements that can perform complex operations, including data modification and business logic execution. They automate complex sequences of operations, improve performance by reducing the need to send multiple queries over the network, and encapsulate business logic to enforce data integrity. The following please see an example of a stored procedure:

1. CREATE PROCEDURE UpdateEmployeeSalary
2. @EmployeeID INT,
3. @NewSalary DECIMAL
4. AS
5. BEGIN
6. UPDATE Employees
7. SET Salary = @NewSalary
8. WHERE EmployeeID = @EmployeeID;
9. END;

Similarities:

- All three encapsulate SQL logic for reuse and simplify complex operations.
- They can be reused across different queries and applications.
- They can help enforce security and data integrity by restricting direct access to tables and defining controlled interfaces for data access.

Differences:

- Views do not store data, while functions and stored procedures can manipulate data.
- Views always return a table, functions can return a scalar value or a table, and stored procedures can return multiple result sets and output parameters.
- Functions cannot have side effects (i.e., they cannot modify the database state), while stored procedures can perform a wide range of operations, including data modification.

SUMMARY

In summary, views, functions, and stored procedures each serve distinct purposes in SQL, with views providing a virtual table interface, functions encapsulating reusable logic for calculations, and stored procedures automating complex sequences of operations.