Stephanie Busch

February 28, 2026

IT FDN 130 GP

Assignment 06

GitHub Link: stephb57/DBFoundations

Assignment 06 – Views

Introduction

In modern database management, efficiency and security are paramount. Rather than requiring users to write complex queries repeatedly, SQL provides specialized objects to encapsulate logic. This paper explores the "Reporting View" as a primary tool for data extraction and compares it to other programmatic objects like Functions and Stored Procedures to determine the best use case for each.

When to use SQL View

A “Reporting View” is used to extract date for reporting purposes (Module 06, Views, Functions and Stored Procedures, 2026). The primary advantage of a view lies in its ability to simplify the user experience by structuring data in a way that is natural or intuitive and summarizing data from various tables. (https://www.tutorialspoint.com/sql/sql-using-views.htm, 2026) (External Site). For example, in Assignment06, a single view was created to join five different tables: Categories, Products, Inventories, Employees, and Managers. By encapsulating these complex relationships, a user can retrieve a full inventory report with a simple SELECT statement rather than writing dozens of lines of join logic. They also can restrict data access from the public or allow data access by a specific group, such as HR. (https://learn.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver17, 2026) (External Site). This is demonstrated by creating "Basic Views" (like vProducts or vEmployees) that sit on top of the base tables, allowing a company to share product names and prices while keeping the underlying table structure hidden.

Differences and Similarities between a View, Function and Stored Procedure

While views, functions, and stored procedures share the goal of code encapsulation, they differ in their execution and capabilities. A view is essentially a "stored select statement" that behaves like a table, but it cannot accept input parameters or perform data modifications. (Module 06, Views, Functions and Stored Procedures, 2026). In Assignment 06, this was seen in the view specifically filtered for 'Chai' and 'Chang' products; while it provides a natural experience for a manager interested in those specific items, the view itself is static and cannot be changed by the user to search for different products without altering the underlying code.In contrast, a User Defined Function (UDF) is designed to perform specific calculations, such as tax formatting or string manipulation, and return a value. Unlike views, functions can accept parameters, allowing them to be more dynamic within a SELECT statement. Stored Procedures represent the most powerful of the three; they are capable of executing complex procedural logic, accepting multiple parameters, and performing "write" operations like INSERT or UPDATE which views and functions are generally restricted from doing. (Module 06, Views, Functions and Stored Procedures, 2026).

Summary

Choosing the right tool depends on the intent of the task. Views are the go-to choice for presenting data in a readable, table-like format. Functions are ideal for repeatable math or string manipulations. Finally, Stored Procedures are the "heavy lifters," capable of handling complex logic, parameters, and data modifications that the other two cannot. By leveraging these three objects appropriately, a database administrator can create a layered architecture that is both user-friendly and highly secure.

References

- Course Materials, *Module 06: Views, Functions and Stored Procedures* (Accessed Feb 2026).
- https://en.wikipedia.org/wiki/View_(SQL) (accessed Feb 2026)
- https://www.tutorialspoint.com/sql/sql-using-views.htm (accessed Feb 2026)
- https://learn.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver17 (accessed Feb 2026)