Name: Brian Moran (bmoran)
Date: 8/20/25
Course: IT Foundations of Database Management
Github Repo: https://github.com/bmoran22/DBFoundations

Module 6: SQL Views

Introduction:

As SQL databases grow in complexity, managing and reusing code efficiently becomes critical for both developers and analysts. Rather than rewriting complex queries repeatedly, SQL offers tools to store and abstract code directly within the database itself. This paper explores one of the most common of these tools—SQL Views—and compares it with Functions and Stored Procedures to demonstrate how each serves different but complementary roles in managing and querying relational data.

When you would use SQL Views:

A SQL View is essentially a saved SELECT statement stored in the database. Views are most useful when you want to simplify complex queries or provide a more user-friendly abstraction of a table's structure. They're commonly used in reporting, row- and column-level data partitioning, and to enforce security—by exposing only specific columns or rows and restricting direct access to the underlying tables. For example, a base view might expose public employee information like names and titles, while a separate private view holds confidential data, with permissions controlling access to each. Views do not store data themselves but instead present a virtual table that can be queried like any other table, offering great flexibility without duplicating data.

Differences and Similarities between Views, Functions, and Stored Procedures:

All three—Views, Functions, and Stored Procedures—serve as named, reusable blocks of SQL code stored in the database. However, they differ in purpose and behavior:

- **Views** are limited to a single SELECT statement and cannot accept parameters. They're best for creating reusable query logic and data abstraction layers. They can't be executed, only selected from.
- **Functions** (specifically User Defined Functions or UDFs) can return either a table or a single scalar value. They can accept parameters, making them dynamic and

useful in SELECT, WHERE, and JOIN clauses. Table-valued functions behave similarly to views but allow input filtering directly through parameters.

- **Stored Procedures** (often called "Sprocs") are the most flexible, capable of executing multiple SQL statements—SELECT, INSERT, UPDATE, DELETE, and even control-of-flow logic like IF or BEGIN...END. They are executed using the EXEC command and are ideal for handling more complex business logic.

While Views and Functions both return data and support abstraction, only Functions and Stored Procedures can accept input parameters. Unlike Views or Functions, Stored Procedures are not restricted to returning data—they can also perform actions.

Summary:

SQL Views, Functions, and Stored Procedures are powerful tools for simplifying, reusing, and securing access to data. Views are best suited for simplifying query logic and enforcing abstraction. Functions enhance reusability by allowing parameters and returning either values or tables. Stored Procedures, meanwhile, provide the greatest flexibility for executing logic-driven workflows. Understanding when and how to use each of these tools is essential for building maintainable and efficient database systems.