Name: Sue Duvall
Date: November 19, 2025
Course: Foundations of Databases and SQL Programming

Assignment 6: SQL Views

Introduction

This document provides a brief overview of SQL Views, describing what they are used for. It also compares this functionality with SQL Functions and SQLj Stored Procedures. It discusses the uses and characteristics of each of these tools as well as similarities and differences in the coding syntax.

Uses of SQL Views

A SQL View is saved code in the database that provides a representation of a SQL database table. A View does not actually store the data themselves; rather, it acts as a virtual copy of the table. This functionality is useful because it allows users to call, up, see, and manipulate data through the View, while leaving the underlying database table untouched and unaltered. Since the View does not store data, you do not need to worry about version control issues affecting the underlying database whenever changes are made.

Reporting is one common usage of SQL Views. A simple or complex SQL Select statement for the View can be housed in the database and then called up when needed to produce data for reports. This functionality makes the production of recurring reports easier and faster. It can even speed up the production of one-time reports if the View produces data that are commonly used. Through partitioning, you can use SQL Views to split the way the data are shown, either horizontally or vertically. With partitioning, you can set up and enforce data security controls by restricting user access to certain rows or columns.

Enabling and controlling changes to database table design are another function of SQL Views. Through base views and temporary views, you can modify a table's code and design behind the scenes without disrupting the user experience. You create a base view for users to access the virtual copy of the table. Through permissions, you only allow access to this base view and not the actual database. Doing so allows you to then update the tables without impacting the user experience. Employing Views and setting permissions will also help prevent accidental changes to the tables themselves, though SQL offers other tools for enforcing table versioning, such as schema binding.

Below is an example of how SQL Views are coded. Notice that the big difference from coding an actual SQL table is the use of a separate name for the view (here denoted by "v" before the table names).

```
CREATE VIEW vProductsByCategories
AS
SELECT [CategoryName]
, [ProductName]
, [UnitPrice]
FROM vCategories AS c JOIN vProducts AS p
ON c.CategoryID = p.CategoryID;
GO
```

Similarities and Differences between a View, Function, and Stored Procedure in SQL

SQL Views are joined in the database's abstraction layer with SQL Functions and SQL Stored Procedures. Each of these powerful tools help increase efficiency because they can be called up and reused in future queries, but they have different characteristics and primary purposes. Their syntax also has similarities and differences. The tables below provide a brief overview of these tools (Note that the tables were derived from initial, high-level knowledge, so the information below is not exhaustive and there may be more nuance to how the tool's characteristic should be defined.)

Similarities and Differences in Purpose and Function of SQL Views, Functions, and Stored Procedures

Characteristic	Views	Functions	Stored Procedures
Part of the abstraction layer	Yes	Yes	Yes
Can be saved within database and reused	Yes	Yes	Yes
Provides data storage	No	No	No
Shows virtual copy of specific data set	Yes	No	No
Enforces data security	Yes	No	No
Performs computations	No	Yes	No
Houses task logic, especially complex operations	No	No	Yes
Increases efficiency when coding	Yes	Yes	Yes
Increases efficiency of data processing	Yes, if indexed	Sometimes	Yes

Similarities and Differences in Syntax Used of SQL Views, Functions, and Stored Procedures

Characteristic	Views	Functions	Stored Procedures
Can only contain Select statements	Yes	Yes	No (Variety of statements allowed)
Uses Select statement to call results	Yes	Yes	No (Execute is used)
Requires schema name (e.g., dbo)	Yes	Yes	Yes
Uses parameters	No	Yes - Required	Yes – Optional
May use Begin/End syntax	No	Yes	Yes

Summary

SQL Views are one useful tool for improving how users interact with data in a SQL database. These virtual tables help simplify and streamline reporting. They also allow developers and system administrators to implement privacy and security protections on the underlying data, as well as perform database updates behind the scenes without impacting user experience. When views are used together with SQL Functions and Stored Procedures, you reap important gains in efficiency.