Student:	Charlene Bane
Date:	November 21, 2022
Course:	IT FDN 130 A Au 22: Foundations of Databases & SQL Programming
Assignment #:	06

Assignment 06 | When to Use a View; Compare Views, Functions, and Stored Procedures

Introduction

The purpose of this paper is to consider the following topics:

1. *Explain when a SQL View should be used.*
2. *Explain the differences and similarities between a View, Function, and Stored Procedure.*

SQL Views | *When to Use*

The purpose of a SQL View is to allow end-users to view the data in a table, or tables, without allowing them to access, edit, or manipulate the actual table. SQL statements and functions can be added to a view so as to present data from multiple tables as if it were coming from a single table.

The basic SQL View syntax is as follows:

```
CREATE VIEW vName AS
SELECT column 1, column2, …
FROM tablename
WHERE condition;
```

A SQL View should be used when you have developed a long or complex query. Instead of retyping the code, it can be stored as a View and run when needed.

A SQL View should also be used to store JOIN queries that you use often.

Finally, to protect the integrity of your data, it is wise to grant public users access to a SQL View instead of the actual database.

Figure 1 | https://www.datacamp.com/tutorial/views-in-sql (2022):

Views, Functions, and Stored Procedures | *Similarities and Differences*

Next, we will describe Views, Functions, and Stored Procedures*, as well as their similarities and differences:*

Table 1 | Views, Functions, and Stored Procedures:

	Views	Functions	Stored Procedures
Description	A View is like a virtual table that contains data from one or multiple tables. It does not hold any data and does not exist physically in the database. A view must be queried (SELECT * FROM vView) to be visible to the end user.	SQL has many built-in functions. However, there are also User Defined functions (UDFs) that are named SELECT statements for which the code is stored in a database. Scalar functions return a single value.	A stored subroutine that can return one or multiple operations on the database and return a value.
Similarities	• Code is stored in a database, not a text file. • Code can be re-used.	• Code is stored in a database, not a text file. • Code can be re-used.	• Code is stored in a database, not a text file. • Code can be re-used.
Differences	• Limited to SELECT statements.	• Limited to SELECT statements.	• Not limited to SELECT statements. • Hold more complex logic such as INSERT, DELETE< and UPDATE.
Use Cases	• To protect the actual database (grant end-users access to a view instead)	• Use a UDF for code that you will need to reuse.	• Use a named procedure for complex code that you wish to reuse.

Conclusion

In conclusion, we discussed three useful SQL tools: Views, Functions, and Stored Procedures. We defined each of these important resources, reviewed their similarities and differences, and determined when to use each. These are important tools for the SQL developer's toolbox.