SQL Server Stored Procedure vs View

What is a stored procedure and what is a view in SQL Server?

[**Stored procedures**](https://www.tsql.info/stored-procedures.php) are SQL statements that can be executed within SQL Server itself. They make it possible for you to perform tasks such as manipulating data or running queries without requiring you to write a significant amount of code.  
Stored procedures improve the protection, performance, and user experience of SQL Server client/server applications. A stored procedure explained by its programmers in aspects of dependent and independent variables.

[**Views**](https://www.tsql.info/views.php) are identical to tables, with the exception that they are constructed from a combination of other tables and views. You can use them to simplify your queries, or to keep your data organized. A view is a virtual table in the SQL Server database and is formed based on a SELECT query.

When would you use a stored procedure over a view, and vice versa?

When you need to manipulate data or run queries, you typically turn to stored procedures for assistance.  
Because they run in memory on the server rather than pulling data across the network like views do, they have the potential to be more efficient than views.  
When you need to query data from multiple tables or views, you typically turn to views for assistance.  
They have the potential to simplify and expedite the querying process, in addition to assisting in the organization of your data.

## What are some performance considerations for stored procedures vs views in SQL Server?

When deciding between using stored procedures or views in SQL Server, there are a number of performance-related factors that should be taken into consideration.  
Stored procedures have the potential to be more effective than views due to the fact that they execute locally on the server rather than pulling data from a remote location over the network. Stored procedures accepts parameters, reduce network traffic, uses SQL statements(insert, update, delete).  
Views uses SELECT statement, can be used to hide complexity and enforce security rules.  
Using views can be helpful for maintaining the organization of your data, and they can also make your queries easier and run more quickly. In conclusion, you should always test your code to ensure that it operates efficiently even when it is under a heavy load.

What is the difference between a Stored procedure and a function?

There are many differences between the Stored procedure and function. Following are the key differences between a Stored procedure and function in SQL Server:  
  
1. Stored procedure can return numerous values, but a function can return a single value.  
2. Stored procedure's return value is optional, but a function must return a value.  
3. Stored procedure can have input and output parameters, but a function can have only input parameters.  
4. Stored procedures can handle errors through the TRY-CATCH block but a function cannot handle errors via the TRY-CATCH block.  
5. Stored procedure supports SELECT, INSERT, UPDATE, and DELETE statements but a function supports only SELECT as it doesn't support [**DML statements**](https://www.tsql.info/statements/dml.php) namely INSERT, UPDATE, and DELETE.  
6. Stored procedure assists to execute [**transactions**](https://www.tsql.info/transactions.php) but a function doesn't assist to execute transactions.  
7. A function can be executed within a stored procedure, but a stored procedure doesn't work within a function.  
8. SELECT statements cannot be used to call a Stored procedure but SELECT statements can be used to call a function.

Which To Use in Crud Operations

A diagram of a process

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Aspect** | Views | Stored Procedures | Functions |
| Performance | Depends on underlying query complexity | Pre-compiled and stored on the server | Performance impact with repeated invocation |
| Use Cases | Provide simplified and controlled access | Encapsulate business logic and complex ops | Data transformation and calculations |
| Data Storage | Virtual tables derived from underlying data | No direct data storage | No direct data storage |
| Abstraction | Abstracts complex queries and joins | Encapsulates multiple operations as a unit | Offers modular and reusable code |
| Security | Controls access to specific columns or rows | Enhances security with centralized logic | N/A |
| Network Traffic | Limited to retrieving data from underlying tables | Reduced network traffic | N/A |
| Input Parameters | No input parameters | Can accept input parameters | Can accept input parameters |
| Number of Parameters | N/A | Can accept multiple parameters | Can accept multiple parameters |
| Return Value | Returns data through querying the view | Can return values, result sets, or status | Returns a single value or result set |