**SQL Assignment 1**

1. **Explain different types of views. Demonstrate with suitable examples.**

- **Simple Views:**

A simple view is a virtual table based on a single base table. It selects certain columns from the base table and presents them as if they were a separate table. This can help to simplify data retrieval by showing only the necessary information.

CREATE VIEW BookTitles AS

SELECT title, author

FROM Books;

Now, users can query the "BookTitles" view to retrieve book titles and authors without needing to know the underlying table structure.

**Complex Views:**

Complex views are derived from multiple base tables or even other views. They can involve joins, subqueries, calculations, and more. Complex views allow users to retrieve data from multiple related tables as if it were a single table, which can be especially useful for complex reporting.

CREATE VIEW BooksAndAuthors AS

SELECT B.title, A.author\_name

FROM Books B

JOIN Authors A ON B.author = A.author\_id;

Now, querying the "BooksAndAuthors" view would yield results that combine information from both the "Books" and "Authors" tables.

**Indexed Views:**

Indexed views are views that have an index associated with them, allowing for improved performance when querying the view. These are typically used for complex views where the cost of computation is high, and indexing can speed up data retrieval.

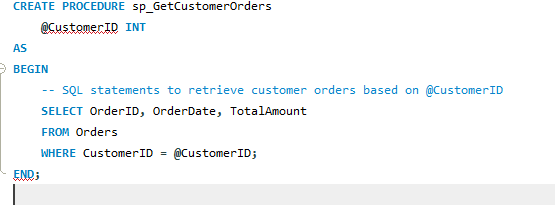
1. **What is the difference between function and stored procedure? Write syntax for creating functions and stored procedures.**

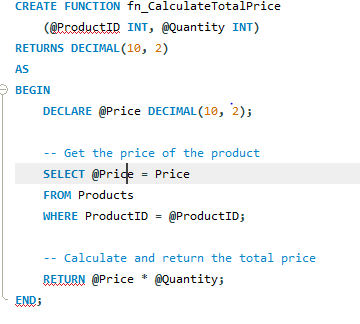
**Stored Procedure:**

A stored procedure is a set of precompiled SQL statements that can accept input parameters and return multiple values through output parameters. They are used for performing specific tasks and are often used for complex data manipulation, transaction management, and other database operations. Stored procedures can also return result sets.

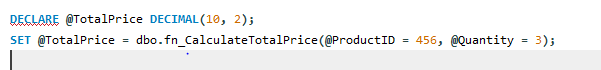
**Function:**

A function is a database object that takes input parameters, processes them, and returns a single value. Functions are primarily used for computations, transformations, and value retrieval based on the input parameters. Functions can be used within SQL statements just like any other expression**.**





Calling Function



1. **What is an index in SQL? What are the different types of indexes in SQL?**

In SQL, an index is a database structure that enhances the retrieval performance of data from a table. It works similar to an index in a book: by providing a quick reference to where specific data is stored in a table, the database system can efficiently locate and retrieve the required data. Indexes play a crucial role in speeding up query performance, especially for tables with large amounts of data.

Different types of indexes in SQL include:

1. B-Tree Index (Clustered and Non-Clustered):

- Clustered Index: In a clustered index, the table's rows are physically sorted on the indexed column(s). Each table can have only one clustered index, and it determines the physical order of the data in the table. Therefore, changing the clustered index may require reorganizing the entire table.

- Non-Clustered Index:Non-clustered indexes store a copy of the indexed column(s) along with a pointer to the corresponding table rows. This allows for faster data retrieval, but the actual data is not physically sorted in the same order as the index.

2. Unique Index:

A unique index enforces the uniqueness of values in the indexed column(s). It ensures that no duplicate values are allowed in the indexed column(s), improving data integrity. Both clustered and non-clustered unique indexes can be created.

3. Composite Index

A composite index consists of multiple columns. It's used when queries frequently involve conditions on multiple columns. The order of columns in a composite index matters as it affects how the index is used for different query scenarios.

4. Covering Inde:

A covering index is designed to cover a query's columns by including all the columns necessary to satisfy the query in the index. This reduces the need to access the actual table rows and can significantly improve query performance.

5. Full-Text Index:

Full-text indexes are used for efficient text-based searches on columns containing large amounts of text data. They enable advanced searches using keywords, phrases, and even complex linguistic rules.

6. Spatial Index:

Spatial indexes are used for optimizing queries that involve spatial data, such as geographic information systems (GIS) applications. They help speed up queries that deal with geographical or location-based data.

7. Filtered Index:

A filtered index is created based on a condition that filters a subset of rows in a table. It's useful for optimizing queries that frequently target specific subsets of data.

8. XML Index:

XML indexes are used to improve the performance of queries involving XML data stored in XML columns. They enhance the retrieval of XML data by creating specialized structures for efficient querying.

1. **What is a temporary and a variable table? Write suitable syntax to create temporary tables and variable tables.**

Temporary Table:

A temporary table is a table that is created and exists for the duration of a session or transaction. Temporary tables can be explicitly created and manipulated like regular tables, and they are useful when you need to store and work with temporary data that persists within the scope of a session or transaction.

Temporary tables can be created using the CREATE TEMPORARY TABLE or CREATE TEMP TABLE statement, depending on the database system you're using.

syntax

CREATE TEMPORARY TABLE temp\_table\_name (

column1 datatype,

column2 datatype,

...

);

Variable Table (Table Variable):

A table variable is a variable that holds a table-like structure in memory. It's typically used within a batch, stored procedure, or function. Table variables are useful when you need to store and manipulate a small amount of data without the need for explicitly creating and dropping tables.

In SQL Server, you can declare table variables using the DECLARE statement and a user-defined table type.

Syntax

DECLARE @TableVariable TABLE (

column1 datatype,

column2 datatype,

...

);