QUESTIONS TASK

Nimra Ashraf

110829

MORNING

Submitted to:

Mam Sehrish Khan

11/05/2025

Question # 1:

How to change the name of attribute in my SQL?

In order to change the names of an attribute using the following syntax:

ALTER table table_name RENAME COLUMN attribute_name;

Question #2:

What is View in SQL?

The view in SQL is the virtual tablet designed based on the other SQL state's result set. Views are designed to reduce complications associated with different tables. In addition, it can deliver data in the most simplified manner. The view in SQL can allow you to maintain data integrity & offer data security. As a result, it acts as the perfect security mechanism.

How Does a SQL View Work?

A virtual table whose contents are explained by the query is called a view in SQL. It comprises a set of rows and columns of data. The SQL view doesn't exist as the database's stored assortment of data values. The data columns and rows come from different tables in a query that defines the view. They are produced if the view has been referenced.

Use of Views of SQL

Want to know the use of views in SQL? Highlighted below are the uses of view in SQL:

- They can reduce the complexities of multiple tables, thereby simplifying data.
- They are used to hide data complexity in a database.
- Another important benefit is that SQL views can take little storage as the database comprises a view's statements instead of a copy of tables.
- Views also offer data security

Create a View in SQL

Let's understand SQL views with the help of an example.

CREATE VIEW view_name AS SELECT attribute_name FROM table_name where attribute='value';

Different Types of Views

Considering the SQL Server, there are primarily two types of views in SQL, as highlighted below:

1. System-Defined View

The system-defined view is the pre-established view. It exists in the SQL Server's Master Database. It acts as a template for tables and data. The system-defined view in SQL can be classified into three forms:

- **Information Schema View**

There are 20 schema views, which display data information. The best example is the table. The view's syntax begins with

INFORMATION_SCHEMA. Note that it is succeeded by a view name such as INFORMATION_SCHEMA.[View Name].

- **Catalogue View**

The next one is the catalogue views introduced in 2005 in SQL Server. They are divided into different groups and offer an excellent method for presenting, transforming, and obtaining data. They usually start with “sys.”

- **Dynamic Management View**

The next and last type is the dynamic management view, which was introduced in 2005. In this view, an administrator can diagnose the SQL server by viewing details. There are two main subtypes: Database–Scoped and Server–Scoped.

2. User Defined View

User-defined view comprises the following types:

- **Simple View**

This type of SQL view is based on one table. These operations include the delete and update options.

- **Complex View**

The next is the complex view. When the view is created from

over one table, it is complex. This type of view contains group data. And the operations of update, insert, and delete are impossible here.

.

Pros of using views in SQL

Here are the advantages:

- Seamless to make changes to any underlying table structure
- Using a view in SQL to return data from the tables allows you to hide WHERE clauses or columns
- You may write simplified select statements against views, thereby handling complicated joins and queries

Cons of using views in SQL

Here are the disadvantages:

- Make SQL queries complex & challenging to understand
- Slows down the queries and reduces the performance
- Limited views and certain tasks cannot be performed (like updating or inserting data)
- Imposes security risks when they aren’t secured
- Difficult to maintain, specifically when used in complicated queries based on data from different tables.

Question # 3:

How to change the data type of date?

Using the format the datatype of date can be changed**:**

Syntax:

```
ALTER table table_name MODIFY COLUMN attribute_name
DATE;
```