TOPSTechnologies

Main SQL Commands & Sub-commands (DDL)

Presented for :
TOPs Technologies
Presented by :
Sumit B Yadav

Que 1

SQL Data Definition Language (DDL) refers to the subset of SQL commands used to define, modify, and remove database structures such as tables, views, indexes, and schemas. DDL is primarily concerned with the schema and the overall structure of the database, rather than the data it contains.

Key DDL commands include:

1. CREATE: Used to create a new database object such as a table, view, index, or schema.
   - Example: CREATE TABLE employees (id INT, name VARCHAR(100));
2. ALTER: Used to modify an existing database object, such as adding or removing columns in a table.
   - Example: ALTER TABLE employees ADD COLUMN age INT;
3. DROP: Used to delete an existing database object, such as a table or index.
   - Example: DROP TABLE employees;
4. TRUNCATE: Used to remove all rows from a table, but keeps the structure intact.

Que. 2

The CREATE command in SQL is used to create new database objects, such as tables, views, indexes, and schemas. It defines the structure of the object, specifying the attributes, constraints, and other properties necessary to store and manage data.

Syntax for Creating Different Database Objects:

1. CREATE TABLE

The CREATE TABLE command is used to create a new table in the database. You specify the table name, followed by the column definitions, which include column names, data types, and optional constraints.

```
CREATE TABLE table_name (
column1 datatype [constraint],
column2 datatype [constraint],
...
[table_constraints]
);
```

2. CREATE DATABASE

The CREATE DATABASE command is used to create a new database within the SQL server.

```
CREATE DATABASE database_name;
```

3. CREATE VIEW

The CREATE VIEW command is used to create a virtual table based on the result of a query. The view doesn’t store data physically but stores a SQL query that can be executed when accessed.

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

4. CREATE INDEX

The CREATE INDEX command is used to create an index on one or more columns in a table, improving query performance.

```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```

5. CREATE SCHEMA

The CREATE SCHEMA command is used to create a new schema within a database. A schema is a collection of database objects, such as tables and views, and provides a way to logically group them.

Que. 3

1. Purpose of Specifying Data Types
Data types define the kind of data that can be stored in each column. By specifying data types, you achieve the following:

a. Data Integrity
Data types ensure that only valid data is entered into each column. For example:

- INT ensures that only integers can be stored in the column.
- VARCHAR(100) ensures that only strings with a maximum length of 100 characters are stored.

Without specifying data types, incorrect or incompatible data (e.g., storing text in a numeric column) could be inserted, leading to errors or unexpected behavior.

b. Efficient Storage
Different data types consume different amounts of memory.

c. Performance Optimization
Data types influence how quickly queries are processed. For example:

- Indexing and sorting operations on INT or DATE columns are faster than on VARCHAR columns.
- Operations like joins, comparisons, and aggregations work more efficiently when the appropriate data type is used.

2. Purpose of Specifying Constraints

Constraints enforce rules on the data at the database level, ensuring consistency and maintaining the quality of the stored data. They help prevent erroneous or invalid data from being inserted and maintain relational integrity.

a. Data Integrity

- NOT NULL: Ensures that a column cannot have a NULL value, meaning that every record must have a value for this column.
  - Example: name VARCHAR(100) NOT NULL; ensures that the name column cannot have missing values.
- UNIQUE: Ensures that all values in a column (or a combination of columns) are unique across all rows in the table.
  - Example: email VARCHAR(100) UNIQUE; ensures that no two rows can have the same email address.

b. Referential Integrity

Constraints like foreign keys maintain relationships between tables, ensuring that data across related tables is consistent.

- PRIMARY KEY: Uniquely identifies each row in a table and ensures that no two rows have the same primary key value.
  - Example: id INT PRIMARY KEY; ensures that the id column contains unique values for each row.
- FOREIGN KEY: Ensures that values in a column correspond to valid values in another table (maintaining relationships between tables).
  - Example: FOREIGN KEY (department_id) REFERENCES departments(id); ensures that any department_id in the employees table must exist in the departments table.