**SQL Commands**

These are the keywords that tell the database what kind of action to perform.

Ex: Create a table, insert data, update, etc.

**SQL Statement**: This is the complete instruction that is given to the database.

It starts with a SQL command and ends with a semicolon.

**SQL Query**: This is the SQL statement which retrieves the data from the database.

Ex: SELECT command….;

**SQL Clause**: It is the part of the query which performs a specific function or defines certain conditions.

These are the keywords which are used within a SQL query or statement to perform different operations like retrieving, manipulating, updating and deleting the data.

**DDL - Data Definition Language**

These commands are involved in creating the structure, updating and deleting the structure of a database.

These commands are only executed once.

1. **CREATE:** It is used to create a database or a table.
2. **Database:** **CREATE** DATABASE database\_name;
3. **Table:** **CREATE** TABLE table\_name(

column\_name1 data type <constraint>,

column\_name2 data type <constraint>,

column\_name3……………………….

.

.

.

column\_nameN data type <constraint>);

Ex: Create a Product database having Product information.

CREATE DATABASE product;

USE product; {if using MySQL}

CREATE TABLE product\_details(

p\_id INT PRIMARY KEY,

p\_name VARCHAR(50) NOT NULL,

unit\_price NUMERIC(6,2), {use float if using MySQL)

mf\_date DATE);

1. **ALTER:** This is used to modify or delete the structure of tables.

It includes ADD, MODIFY, RENAME, DROP.

**ALTER** TABLE table\_name

1. Adding new column:

**ADD** column\_name data type <constraint>;{MySQL}

**ADD** COLUMNcolumn\_name data type <constraint>;{Postgres}

1. Rename a column:

**RENAME** old\_column\_name **TO** new\_column\_name;{MySQL}

**RENAME** COLUMN old\_column\_name **TO** new\_column\_name;{Postgres}

1. Modify a column:

**MODIFY** COLUMNcolumn\_name new data type; {MySQL}

**ALTER** COLUMNcolumn\_name **TYPE** data type; {Postgres}

**ALTER** COLUMN column\_name **DROP DEFAULT**;{Postgres}

**ALTER** COLUMN column\_name **SET NOT NULL**;{Postgres}

1. Add or modify constrainst:

**ADD** **CONSTRAINT** constraint\_name **UNIQUE**(column\_name);

**DROP** **CONSTRAINT** constraint\_name;

**MODIFY** column\_name data type new\_constraint;{MySQL}

{In Postgres to modify a constraint we have to first drop it

and then add a new constraint.}

1. Delete a column:

**DROP** COLUMN column\_name;

Ex: Rename unit\_price column to price.

ALTER TABLE product

RENAME COLUMN unit\_price to price;

1. **DROP:** Used to entirely delete the structure.

It can be a database, table or column.

1. **Database: DROP DATABASE** database\_name;
2. **Table: DROP TABLE** table\_name;
3. **Column: ALTER TABLE** table\_name **DROP COLUMN** column\_name;
4. **TRUNCATE:** It is used to delete the tables but not the structure of the table.

It is used to delete all the fields and rows from the table.

**TRUNCATE TABLE** table\_name;

***Interview Question: What is the difference between DROP and TRUNCATE?***

**Answer:**

**DROP** deletes everything from the schema and it has no physical existence left.

**TRUNCATE** deletes the data from the tables while retaining the structure of the table.