SQL:

SQL (Structured Query Language) is used to perform operations on the records stored in the database, such as updating records, inserting records, deleting records, creating, and modifying database tables, views, etc.

SQL is a query language. It is mainly designed for maintaining the data on the relational database management system. Data is usually stored in the form of tables (rows and columns).

SQL Command types:

SQL commands are instructions used to perform specific operations and used to communicate with databases.

Types of SQL Commands:

There are 4 types of commands.

1. DDL:

- Data definition language.
- It defines the schema of the table like creating a table, altering a table, dropping a table, truncate a table and renaming a table.
- It is auto committed means it permanently saves the changes.

DDL commands:

- Create: to create new database or table
- Alter: used to alter the structure of the database.
- Drop: used to delete both structure and records of the table.
- Truncate: used to remove all the records of the table but schema remains as it is.
- Rename: used to rename database or table.

2. DML:

- Data manipulation language.
- DML commands are used to modify the database. It is responsible for all the changes to the database.
- DML commands are not auto committed means changes are not permanently saved.

DML Commands:

- Insert: used to insert the data into the rows of the table.
- Update: used to update or modify the value of the column.
- Delete: used to delete one or more rows from the table.
- Select: for accessing the records of the table.

3. DCL:

- DCL commands are used to grant and take back authority from the database user.
- DCL commands are data control language.

DCL Commands:

- Grant: it is used to give user access privileges to the database.
  - GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
- Revoke: is used to take back access permission from the user.
  - REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

4. TCL:

- Transaction Control Language consists of commands that are manage transaction within a database.
- Commands are COMMIT, ROLLBACK, SAVEPOINT.

TCL commands:

1. COMMIT:

- This command is used to permanently save the changes made during transaction.
- Once a commit command executed changes becomes permanent and can't ROLLBACK.

2. ROLLBACK: Is used to undo the transactions before COMMIT.

3. SAVEPOINT: Is used to set a point within a transaction to which we can ROLLBACK.

Datatypes:

Datatypes are used to represent the nature of the data that can be stored in the database table.

- String datatype
- Numeric datatype
- Date and time datatype

Create Database:

This statement helps us to create a new database in the database system with a specified name for the database.

Syntax: CREATE DTATBASE database_name;

- If a database exists with the same name in the database system, then it won't create another database with the same name.

- To verify whether the database is created or not then we use command.

  ```
  SHOW DATABASES;
  ```

- To replace the existing database.

  ```
  CREATE OR REPLACE DATABASE database_name;
  ```

- To drop existing database.

  ```
  DROP DATABASE database_name;
  ```

SQL Tables:

Create table.

Table is a collection of data which is in the form of rows and columns. In DBMS table is a relation and rows are tuple.

Table is created inside the database, which is under usage, if want to create a table then should have to mention table table_name and columns for the table along with datatype of the column.

Syntax:

```
CREATE TABLE table_name (column1 datatype, column2 datatype,……, columnN datatype);
```

To create duplicate of the existing table for future reference is has follows.

Syntax:

```
CREATE TABLE table_dup SELECT * FROM  existing table;
```

Drop table

To drop existing table from the database is by using the statement. The drop table statement is used to delete a table definition and all data from the table, once the table is dropped all the information related with that table is deleted permanently.

Syntax:

```
DROP TABLE table_name;
```

Delete table

To delete a particular row from the table then we use delete statement. If you want to delete a specific row of the table, then should use where clause" otherwise it deletes all the rows of the table if we don't specify which are rows as to be delete.

Syntax:

```
DELETE FROM table_name [WHERE condition];
```

```
DELETE FROM table_name;
```

Truncate table

Truncate statement is used to delete all the rows and frees containing space.

Syntax:

```
TRUNCATE TABLE table_name;
```

Rename table

For renaming the table we use statement.

Syntax:

```
ALTER TABLE table_name RENAME TO new_table_name;
```

Column add and delete

To add a new column to the existing table we use statement.

Syntax:

```
ALTER TABLE table_name ADD COLUMN column_name datatype;
```

To delete a column from the existing table.

Syntax:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

Modify table

To modify column datatype of the existing table then we use statement.

Syntax:

```
ALTER TABLE table_name MODIFY COLUMN column_name datatype;
```

Key constraints:

Primary key:

A column or columns are called primary keys that uniquely identifies each row in a table. It must be unique. It can't be duplicated, there should be only one primary key in a table. If two or columns are marked as a primary key, then it is a composite primary key. It can't be null.

Primary key column should be mentioned at the time of table creation or modifying a table.

Advantage of the primary key is fastest access of records of the table.

Syntax:

CREATE TABLE table_name(id int,……..,primary key(id));

If table is already created, then the statement used to add primary key is.

Syntax:

ALTER TABLE table_name ADD PRIMARY KEY(id);

To drop primary key is.

Syntax:

ALTER TABLE table_name DROP PRIMARY KEY(id);

Foreign key:

The foreign key is a field, or a column used to establish a link between two tables.

The foreign key in one table to point primary key in another table. It will always reference the primary key of the other table.

To drop foreign key from the existing table.

Syntax:

ALTER TABLE table_name DROP FOREIGN KEY column_name;

Difference between primary and foreign keys:

Primary key	**Foreign key**
Not null	Can be null
Should be unique	Duplicates are allowed
Only one primary key in a table	Can be more then one foreign keys are allowed
Uniquely identifies a record of the table	References primary key of another table

Update command:

Update command is used to update the value of the existing record of the table.

Syntax:

```
UPDATE table_name SET column_name = value WHERE [condition];
```

Default value:

To add default value to all the rows of the table we use command. There are two ways to add it one way is while creating the table, and other way is by using alter command to add column with default value for that column.

Syntax: using alter command

```
ALTER TABLE table_name ADD COLUMN column_name datatype DEFAULT value;
```

Aliasing:

Aliases are used to give a table or a column in a table a temporary name. An alias exists only for a duration of query. Are created by using the AS keyword.

Syntax:

```
SELECT column_name AS alias_name FROM table_name;
```

Aggregate functions:

Aggregate functions are used to perform operations on the records of the table. It is applicable only on numerical values. Functions are.

- COUNT ()
- SUM ()
- MAX ()
- MIN ()
- AVG ()
- MEAN ()

SQL select:

Distinct:

For selecting or accessing only specific column of the table or non-duplicate records we use DISTINCT.

Syntax:

SELECT DISTINCT column_name FROM table_name;

JOINS:

Join means combining records of two or more tables. There are multiple types of joins present to perform operations on tables.

1) Inner join: Is used to access the records that are common from both tables.

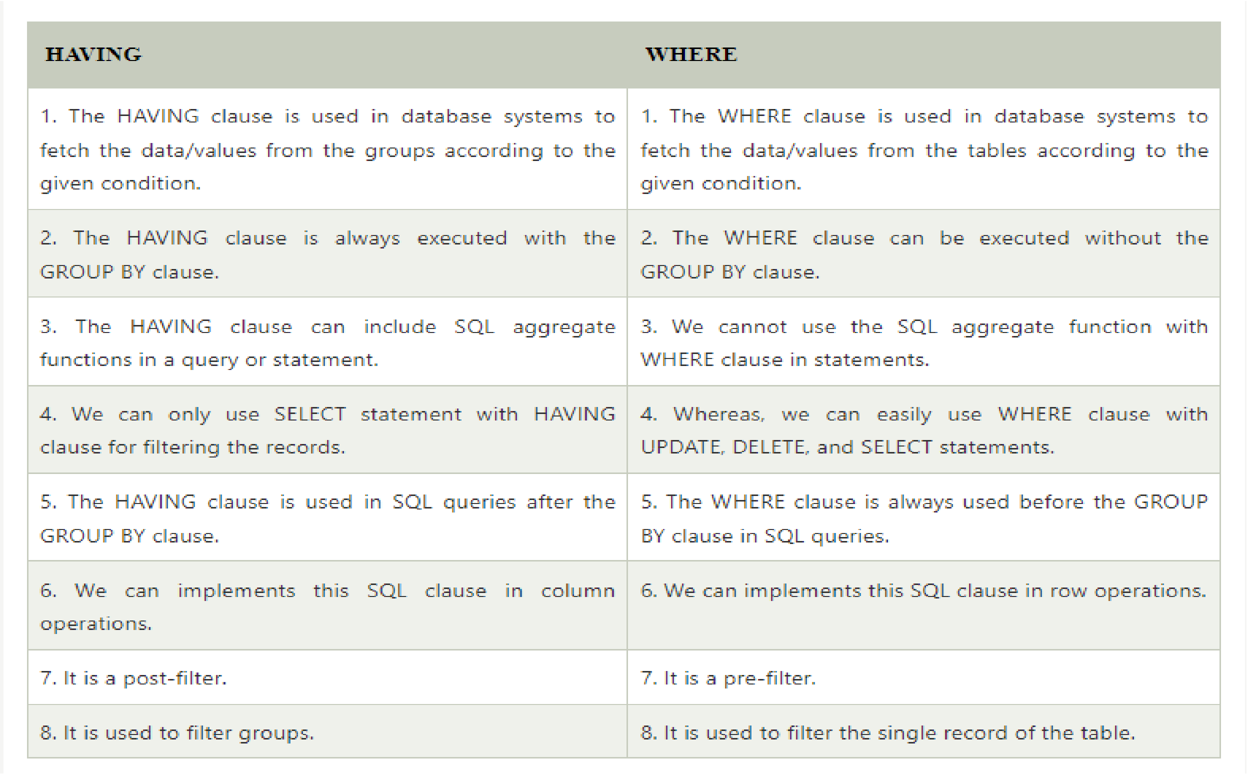
Syntax: select col_1, col_2,..col_n from table1 inner join table2 on

table1.id=table2.id;

2) Left join: Is used to retrieve the records that are matched in both tables and all the columns from the left table.

Syntax: select table1.col1,tab1,col2, tab2.col1, tab2.col2,….tabn.coln from

tabl1 left join tabl2 on table1.col=tabl2.col;



HAVING	WHERE
1. The HAVING clause is used in database systems to fetch the data/values from the groups according to the given condition.	1. The WHERE clause is used in database systems to fetch the data/values from the tables according to the given condition.
2. The HAVING clause is always executed with the GROUP BY clause.	2. The WHERE clause can be executed without the GROUP BY clause.
3. The HAVING clause can include SQL aggregate functions in a query or statement.	3. We cannot use the SQL aggregate function with WHERE clause in statements.
4. We can only use SELECT statement with HAVING clause for filtering the records.	4. Whereas, we can easily use WHERE clause with UPDATE, DELETE, and SELECT statements.
5. The HAVING clause is used in SQL queries after the GROUP BY clause.	5. The WHERE clause is always used before the GROUP BY clause in SQL queries.
6. We can implements this SQL clause in column operations.	6. We can implements this SQL clause in row operations.
7. It is a post-filter.	7. It is a pre-filter.
8. It is used to filter groups.	8. It is used to filter the single record of the table.