# Structured Query Language(sql):

Structured Query Language(SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

These SQL commands are mainly categorized into four categories as:

1. DDL – Data Definition Language
2. DQl – Data Query Language/ DRL Data Retrieval Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language

Though many resources claim there to be another category of SQL clauses **TCL – Transaction Control Language**. So we will see in detail about TCL as well.

**DDL (Data Definition Language):**

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.

List of DDL commands:

- **CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- **DROP**: This command is used to delete objects from the database.
- **ALTER:** This is used to alter the structure of the database.
- **TRUNCATE:** This is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT**: This is used to add comments to the data dictionary.
- **RENAME:** This is used to rename an object existing in the database.

**DQL (Data Query Language): Data Retrieval language.**

**DQL** statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. We can define DQL as follows it is a component of SQL statement that allows getting data from the database and imposing order upon it. It includes the SELECT statement. This command allows getting the data out of the database to perform operations with it. When a SELECT is fired against a table or tables the result is compiled into a further temporary table, which is displayed or perhaps received by the program i.e. a front-end.

List of DQL:

- **SELECT:** It is used to retrieve data from the database.

**DML(Data Manipulation Language):**

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

List of DML commands:

- **INSERT** : It is used to insert data into a table.
- **UPDATE:** It is used to update existing data within a table.
- **DELETE** : It is used to delete records from a database table.
- **LOCK:** Table control concurrency.
- **CALL:** Call a PL/SQL or JAVA subprogram.
- **EXPLAIN PLAN:** It describes the access path to data.
-
-
-
-

- **DCL (Data Control Language):**

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

- List of DCL commands:
- **GRANT:** This command gives users access privileges to the database.
- **REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.

Though many resou rces claim there to be another category of SQL clauses TCL – Transaction Control Language. So we will see in detail about TCL as well. TCL commands deal with the transaction within the database.

List of TCL commands:

- **COMMIT:** Commits a Transaction.
- **ROLLBACK:** Rollbacks a transaction in case of any error occurs.
- **SAVEPOINT:**Sets a savepoint within a transaction.
- **SET TRANSACTION:** Specify characteristics for the transaction.

DDL:

***Create***

Syntax: CREATE TABLE *table_name* (
*column1 datatype*,
*column2 datatype*,
*column3 datatype*,
....
);

Ex: create table student (Roll_No int, Name varchar (50), Address varchar (50), Age int,)

Alter

Syntax ALTER TABLE *table_name*
ADD *column_name datatype*;

Ex: Alter table student add Phone_no varchar (20);

Alter table student drop Phone_no;

***Update***

Syntax Uddate *table_name*
SET *column1 = value1*, *column2 = value2*, ...
WHERE *condition*;

Ex update student set pnone_no =’9767043555’ where roll no =’101’;

***Drop***

Syntax: Drop table table_name;

Ex: drop table student;

***Truncate***

Syntax: Truncate table table_name;

Ex: truncate table student;

***Delete***

Syntax: delete from table_name[where condition];

Ex: delete from student where rollno=101;

***Rename***

Syntax: 1) rename old_table_name to new_table_name;

2) alter old_table_name rename to new_table_name;

Ex: rename table student to Learners;

**Select count**

Syntax: 1)select count(column_name)

From table_name

Where condition;

Ex:select score, count(score)from marks;

**Select Distinct**

Syntax: 1) select distinct column1, column2,….

From table_name;

Ex; select distinct name from stud1;

**Limit**

Syntax: 1) select column_name(s)

From table_name

Where condition

Limit number;

Ex; select * from stud1 limit 2;

**In Function**

Syntax: 1) select column_name(s)

From table_name

Where column_name in(value1, value2,…….)

Ex; select city from stud1 where age in(20,21);

**Between Function**

Syntax: select column_name(s)

From table_name

Where column_name between value1 and value2;

Ex; select score from marks where S_no between 2 and6;

## Inner Join

Syntax: select column_name(s)

From table1

Inner join table2

On table1.column_name = table2.column_name;

Ex; select stud1.name, marks.maths from stud1 inner join marks on marks.s_no = stud1.s_no;

## Left join

Syntax: select column_name(s)

From table1

Left join table2

On table1.column_name = table2.column_name;

Ex; select marks.score, stud1.name from marks left join stud1 on marks.s_no = stud1.s_no;

## Right join

Syntax: select column_name(s)

From table1

right join table2

On table1.column_name = table2.column_name;

Ex; select marks.score, stud1.name from marks right join stud1 on marks.s_no = stud1.s_no;

**Union Join**

Syntax: select column_name(s)

From table1

Union

Select column_name(s)

From table2;

Ex; select age from stud1 union select score from marks;

**Group by clause**

Syntax: select column_name(s)

From table_name

Where condition group by column_name(s);

Ex; select max(maths) from marks group by s_no;

**Order by clause**

Syntax: select column_name(s)

From table_name

Where condition

Order by column_name(s);

Ex; select city from stud1 where name = 'paurnima' order by age;

**Where clause**

Syntax: select column(s) from table_name where condition;

Ex; select age from stud1 where name = ‘paurnima’;

**Having clause**

Syntax: select column(s) from table_name where condition;

Ex; select city from stud1 where name = 'paurnima order by age;

**TCL**

**Use**

Syntax: use database_name;

Ex; use class;

**Create table**

Syntax: create table table_name(column(s)_name data type,….);

Ex; create table staff(s_ID int, name varchar(20), city varchar(20), salary int);

**Transaction**

Ex; start transaction;

**Insert**

Syntax: insert into table_name(column(s)_name) values (value1, value2,…….valueN);

Ex; insert into staff values(101, 'Airudha', 'Sagali', 55000 );

**Select**

Syntax: select * from table_name;

Ex; select * from staff;

**Commit**

Ex; commit;

**Savepoint**

Syntax:savepint savepoint_name

Ex;savepoint insertion;

**Update**

Syntax: UPDATE table_name SET column_name WHERE condition[];

Ex; UPDATE staff SET city = 'Pune' WHERE s_ID = 104;

**Rollback**

Syntax: rollback to savepoint;

Ex; rollback to insertion;

**copy**

**Syntax:**

1)select column(s)_name from table_name where condition[];

2)insert into new_table_name select * from old_table_name where condition[];

Ex;

1)create table stud2 like stud1;

2) insert into stud2 select * from stud1 where s_no = 101;

**Like**

Syntax: select column_name from table_name where condition;

Ex; select name from stud1 where name like's%';

**Rand**

Syntax: select column(s)_name from table_name order by rand();

Ex; select ID from teacher order by rand() limit 2;

**Cast**

Syntax: select cast(123 as varchar(20)) [result_name] from [source]

Ex; select s_no , cast(maths as float) float_maths from marks;

**Least**

Syntax: select least(1,2,3,……..n);

Ex; select least(1,2,3,4,……) as least_of_number;

## Subquery

Syntax: select column(s)_name from table_name1 where condition[]comparison_operator(select column(s)_name from table_name from table_name where condition[]);

Ex; select * from stud1 where age >(select avg(age) from stud1);

## Substring

Syntax: select substring(column_name,1,n) from table_name;

Ex; select substring(name ,1,2) from stud2;

## Upper And Lower

Syntax: select upper/lower(column_name)from table_name

Ex; select upper(name) from stud2;

select lower(name) from stud2;

## Second-Highest Value of an Integer

Syntax: select max(column_name) from table_name where column_name not in(select max(column_name) from table_name);

Ex; select max(age) from stud1 where s_no not in(select max(age) from stud1);