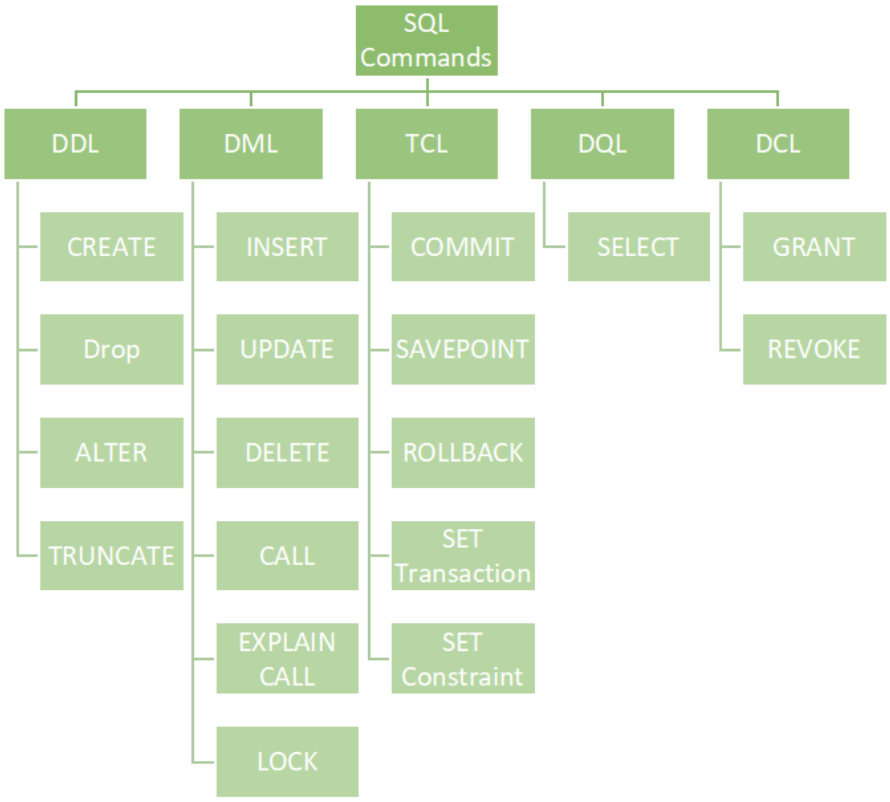
SQL - STRUCTURED QUERY LANGUAGE

- SQL is a program created and formulated in the Relational Database Management System to handle structured data.

- SQL commands are extensively used to interact with databases, enabling users to perform a wide range of actions on database systems.

- This guide will introduce you to the various SQL sublanguage commands, including

    - Data Definition Language (DDL)
    - Data Query Language (DQL)
    - Data Manipulation Language (DML)
    - Data Control Language (DCL)
    - Transaction Control Language (TCL)



DDL – DATA DEFINITION LAYER

CREATE-

CREATE command is used to create a data base or its objects.

- First thing need to create database, for that we used create command
- And use command is used to use the data base.
- Number should be initialised as int,
- And string should be initialised as varchar with number of letters.

Example:

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);
```

DROP

DROP command is to delete the objects from the database.

- We need to use the database
- Next use the Drop command with the object name.

Example:

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
```

```
);
DROP TABLE Emplyoee
```

ALTER

ALTER command is used to alter the structure of the database.

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);

ALTER TABLE Emplyoee add column emp_address varchar(50);
```

This will add one new column to the table.

TRUNCATE

Truncate command is used to remove all the records from the table.

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);
```

ALTER TABLE Emplyoee add column emp_address varchar(50);

TRUNCATE TABLE Emplyoee;

RENAME

RENAME command is used to rename the table.

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);

ALTER TABLE Emplyoee add column emp_address varchar(50);

TRUNCATE TABLE Emplyoee;

RENAME table Emplyoee TO Emplyoee_details;
```

DQL – DATA QUERY LANGUAGE

SELECT

SELECT command is used to retrieve the data from the database.

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);

SELECT * FROM Emplyoee;
```

DML – DATA MANIPULATION LANGUAGE

INSERT

INSERT command is used to insert a data into the table.

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);
```

```
SELECT * FROM Emplyoee;


INSERT INTO Emplyoee (emp_id,emp_name,emp_age) values
(1,"JIHIN",23);
```

UPDATE

UPDATE command is used to update the value of existing data within a table.

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);


SELECT * FROM Emplyoee;


INSERT INTO Emplyoee (emp_id,emp_name,emp_age) values
(1,"JIHIN",23);
INSERT INTO Emplyoee (emp_id,emp_name,emp_age) values
(1,"JISHI",23);


UPDATE Emplyoee SET emp_id =13 where emp_name='JISHI' ;
```

DELETE

DELETE command is to delete the specific contents from the table

```
create database sql_class;
use sql_class;
create table Emplyoee(
        emp_id int,
   emp_name varchar(50),
   emp_age int
);

SELECT * FROM Emplyoee;

INSERT INTO Emplyoee (emp_id,emp_name,emp_age) values (1,"JIHIN",23);
INSERT INTO Emplyoee (emp_id,emp_name,emp_age) values (1,"JISHI",23);

UPDATE Emplyoee SET emp_id =13 where emp_name='JISHI' ;

DELETE from Emplyoee where emp_name='JIHIN';
```

DCL – DATA CONTROL LANGUAGE

- **GRANT – Grants specific privileges to users.**
- **REVOKE – Removes privileges from users.**

TCL – TRANSACTION CONTROL LANGUAGE

- **COMMIT – Commit the changes**
- **ROLLBACK – Undo the changes in the current transaction.**

- **SAVEPOINT –The SAVEPOINT command allows you to set a point within a transaction to which you can later roll back. This is useful when you want to undo part of a transaction without affecting the entire transaction.**

```
BEGIN;  -- Start a new transaction

INSERT INTO Employee (FirstName, LastName, Age, Gender, Department, HireDate, Salary)
VALUES ('Bob', 'Smith', 28, 'Male', 'HR', '2024-10-18', 55000.00);

SAVEPOINT before_update;

UPDATE Employee
SET Salary = Salary * 1.10
WHERE FirstName = 'Bob';
```

-- If you decide that Bob's raise was a mistake

ROLLBACK TO before_update; -- Undo the salary update

COMMIT; -- Save the new employee record (without the salary update)

IMPORTANT COMMANDS IN SQL

- **SELECT: Used to retrieve data from a database.**
- **INSERT: Used to add new data to a database.**
- **UPDATE: Used to modify existing data in a database.**
- **DELETE: Used to remove data from a database.**
- **CREATE TABLE: Used to create a new table in a database.**
- **ALTER TABLE: Used to modify the structure of an existing table.**
- **DROP TABLE: Used to delete an entire table from a database.**
- **WHERE: Used to filter rows based on a specified condition.**
- **ORDER BY: Used to sort the result set in ascending or descending order.**
- **JOIN: Used to combine rows from two or more tables based on a related column between them.**

JOINS - JOIN is used to combine rows from two or more tables based on a related column between them. There are several types of joins: INNER JOIN, LEFT JOIN, RIGHT JOIN

Sample Tables

Consider two tables, Employee and Department:

Employee Table:

EmployeeID	FirstName	LastName	DepartmentID
1	John	Doe	10
2	Jane	Smith	20
3	Mark	Lee	30
4	Alice	Brown	NULL

Department Table:

DepartmentID	DepartmentName
10	HR
20	IT
30	Finance
40	Marketing

INNER JOIN: Returns records that have matching values in both tables.

```
SELECT Employee.FirstName, Employee.LastName,
Department.DepartmentName FROM Employee INNER JOIN
Department ON Employee.DepartmentID = Department.DepartmentID;
```

FirstName	LastName	DepartmentName
John	Doe	HR
Jane	Smith	IT
Mark	Lee	Finance

LEFT JOIN (or LEFT OUTER JOIN): Returns all records from the left table and the matched records from the right table. If there's no match, NULL values are returned from the right table.

```
SELECT Employee.FirstName, Employee.LastName,
Department.DepartmentName
FROM Employee
LEFT JOIN Department
ON Employee.DepartmentID = Department.DepartmentID;
```

FirstName	LastName	DepartmentName
John	Doe	HR
Jane	Smith	IT
Mark	Lee	Finance
Alice	Brown	NULL

RIGHT JOIN (or RIGHT OUTER JOIN): Returns all records from the right table and the matched records from the left table. If there's no match, NULL values are returned from the left table.

```
SELECT Employee.FirstName, Employee.LastName,
Department.DepartmentName

FROM Employee

RIGHT JOIN Department

ON Employee.DepartmentID = Department.DepartmentID;
```

Result:

FirstName	LastName	DepartmentName
John	Doe	HR
Jane	Smith	IT
Mark	Lee	Finance
NULL	NULL	Marketing