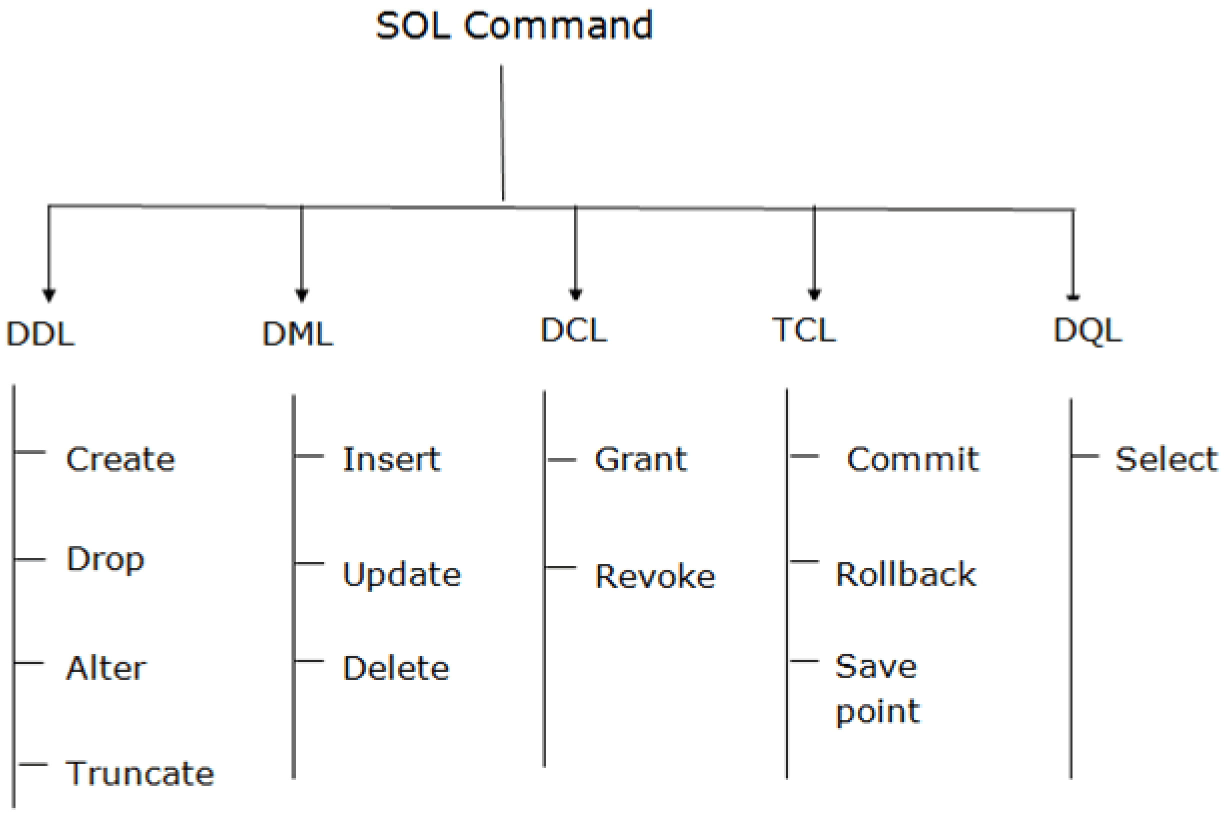
SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



1. Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.

- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

a. CREATE It is used to create a new table in the database.

Syntax:

```
CREATE TABLE TABLE_NAME

(

    COLUMN_NAME1 DATATYPES(size),

    COLUMN_NAME2 DATATYPES(size),

    --------------

    COLUMN_NAMEN DATATYPES(size),

);
```

Example:

```
CREATE TABLE EMP

(

    EMPNo VARCHAR2(20),

    EName VARCHAR2(20),

    Job  VARCHAR2(20),

     DOB DATE

);
```

b. DROP : This statement is used to drop an existing database. When you use this statement, complete information present in the database will be lost.

Syntax

```
DROP DATABASE DatabaseName;
```

Example

```
DROP DATABASE Employee;
```

The 'DROP TABLE' Statement

This statement is used to drop an existing table. When you use this statement, complete information present in the table will be lost.

Syntax

```
DROP TABLE TableName;
```

Example

```
DROP Table Emp;
```

c. ALTER

This command is used to delete, modify or add constraints or columns in an existing table.

The 'ALTER TABLE' Statement

This statement is used to add, delete, modify columns in an existing table.

The 'ALTER TABLE' Statement with ADD/DROP COLUMN

You can use the ALTER TABLE statement with ADD/DROP Column command according to your need. If you wish to add a column, then you will use the ADD command, and if you wish to delete a column, then you will use the DROP COLUMN command.

Syntax

- `ALTER TABLE TableName ADD ColumnName Datatype;`

- `ALTER TABLE TableName DROP COLUMN ColumnName;`

Example

```
--ADD Column MobNo:
    ALTER TABLE Emp ADD MobNo Number(10);
--DROP Column MobNo:
   ALTER TABLE Emp DROP COLUMN MobNo ;
```

The 'ALTER TABLE' Statement with ALTER/MODIFY COLUMN

This statement is used to change the datatype of an existing column in a table.

Syntax

```
   ALTER TABLE TableName ADD COLUMN ColumnName Datatype;
```

Example

```
--Add a column DOB and change the data type to Date.
  ALTER TABLE Emp ADD DOB date;
```

d. TRUNCATE

This command is used to delete the information present in the table but does not delete the table. So, once you use this command, your information will be lost, but not the table.

Syntax:

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE EMPLOYEE;
```

2. Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

a. INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax:

```
INSERT INTO TABLE_NAME

(col1, col2, col3,.... col N)

VALUES (value1, value2, value3, .... valueN);
```

Or

```
INSERT INTO TABLE_NAME

VALUES (value1, value2, value3, .... valueN);
```

For example:

```
INSERT INTO EMP(EName,Job) VALUES ("SCOTT", "MANAGER");
```

b. UPDATE: This command is used to update or modify the value of a column in the table.

Syntax:

```
UPDATE table_name SET column1= value, column2= value,
columnN = value WHERE CONDITION;
```

For example:

```
UPDATE Emp SET Ename = 'SMITH'  WHERE EmpNo = '1003';
```

c. DELETE: It is used to remove one or more row from a table.

Syntax1:

```
DELETE FROM table_name;
```

Syntax1

```
DELETE FROM table_name WHERE condition;
```

Example1: Delete all rows from emp table

```
DELETE FROM Emp;
```

Example2: Delete all rows from emp table whose Ename is SCOTT

```
DELETE FROM EName WHERE EName="SCOTT";
```

3. Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- Grant
- Revoke

a. Grant: It is used to give user access privileges to a database.

Example

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

b. Revoke: It is used to take back permissions from the user.

Example

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

4. Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the database.

Syntax:

```
COMMIT;
```

Example:

```
DELETE FROM CUSTOMERS WHERE AGE = 25;

COMMIT;
```

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

```
ROLLBACK;
```

Example:

```
DELETE FROM CUSTOMERS WHERE AGE = 25;

ROLLBACK;
```

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

```
SAVEPOINT SAVEPOINT_NAME;
```

5. Data Query Language

DQL is used to fetch the data from the database.

SELECT

This statement is used to select data from a database and the data returned is stored in a result table, called the result-set.

Syntax

```
SELECT Column1, Column2, ...ColumN FROM TableName;

--(*) is used to select all from the table

   SELECT * FROM table_name;
```

```
-- To select the number of records to return use:

   SELECT TOP 3 * FROM TableName;
```

Apart from just using the SELECT keyword individually, you can use the following keywords with the SELECT statement:

-
- *DISTINCT*
- *ORDER BY*
- *GROUP BY*
- *HAVING Clause*
- *INTO*

The ‘SELECT DISTINCT’ Statement

This statement is used to return only different values.

Syntax

```
SELECT DISTINCT Column1, Column2, ...ColumnN FROM TableName;
```

SELECT DISTINCT MobNo FROM Emp;

Example

The ‘ORDER BY’ Statement

The ‘ORDER BY’ statement is used to sort the required results in ascending or descending order. The results are sorted in ascending order by default. Yet, if you wish to get the required results in descending order, you have to use the DESC keyword.

Syntax

```
SELECT Column1, Column2, ...ColumnN FROM TableName

ORDER BY Column1, Column2, ... ASC|DESC;
```

Example

Select all employees from the 'Emp’ table sorted by EmpNo:

```
   SELECT * FROM Emp ORDER BY EmpNo;
```

-- Select all employees from the 'Emp table sorted by EmpNo in Descending order:

```
   SELECT * FROM Employee_Info ORDER BY EmpNo DESC;
```

-- Select all employees from the 'Empl’ table sorted by EmpNo and EName:

```
    SELECT * FROM Emp ORDER BY EmpNo, EName;
```

/* Select all employees from the 'Emp' table sorted bsoEmpNo in Descending order and Ename in Ascending order: */

```
    SELECT * FROM Emp ORDER BY EmpNo ASC, Ename DESC
```

The ‘GROUP BY’ Statement

This ‘GROUP BY’ statement is used with the aggregate functions to group the result-set by one or more columns.

Syntax

```
SELECT Column1, Column2,..., ColumnN FROM TableName

WHERE Condition GROUP BY ColumnName(s) ORDER BY ColumnName(s);
```

Example

To list the number of employees from each city.

```
   SELECT COUNT(EmpNo), City FROM Emp GROUP BY City
```

The ‘HAVING’ Clause

The ‘HAVING’ clause is used in SQL because the WHERE keyword cannot be used everywhere.

Syntax

```
SELECT ColumnName(s) FROM TableName WHERE Condition GROUP BY
ColumnName(s) HAVING Condition ORDER BY ColumnName(s);
```

Example

To list the number of employees in each city. The employees should be sorted high to low and only those cities must be included who have more than 5 employees:*/

```
    SELECT COUNT(EmpNo), City FROM Emp GROUP BY City HAVING COUNT(EmpNo) > 2 ORDER BY
COUNT(EmpNo) DESC;
```

The ‘SELECT INTO’ Statement

The ‘SELECT INTO’ statement is used to copy data from one table to another.

Syntax

```
SELECT * INTO NewTable IN ExternalDB FROM OldTable WHERE
Condition;
```

Example

To create a backup of database 'Employee'

```
SELECT * INTO EmpNo FROM Emp;
```