**What is SQL?**

It is used in programming and managing data held in relational database management systems such as MySql, MS SQL Server, Oracle Sybase, etc as a medium (instructions) for accessing and interacting with data.

It enables performing several operations such as creating, deleting, modifying, and fetching entries in the database and some other advanced statistical, arithmetic, and mathematical operations. Let us try understanding the basics of SQL now.

**What is SQL Used For?**

This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data that is stored in the form of tables).

It is also designed for **stream processing in RDBMS**. You can easily create and manipulate the database, access and modify the table rows and columns, etc.

It is used in relational database management systems such as MySql, MS SQL Server, Oracle Sybase, etc as a medium (instructions) for accessing and interacting with data.

**Types of SQL Commands**

**DDL(Data Definition Language):** To make/perform changes to the physical structure of any table residing inside a database, DDL is used. These commands when executed are auto-commit in nature and all the changes in the table are reflected and saved immediately. They are **CREATE**, **DROP**, **ALTER**, **TRUNCATE**, and **RENAME**.

**DML(Data Manipulation Language):** Once the tables are created and the database is generated using DDL commands, manipulation inside those tables and databases is done using [DML](https://www.scaler.com/topics/dml-commands-in-sql/) commands. The advantage of using DML commands is, that if in case any wrong changes or values are made, they can be changed and rolled back easily. They are **INSERT, UPDATE, and DELETE**

**DQL(Data Query Language):** Data query language consists of only one command upon which data selection in SQL relies. The **SELECT** command in combination with other SQL clauses is used to retrieve and fetch data from databases/tables based on certain conditions applied by the user.

**DCL(Data Control Language):** DCL commands as the name suggests manage the matters and issues related to the data controller in any database. DCL includes commands such as **GRANT** and **REVOKE** which mainly deal with the rights, permissions, and other controls of the database system.

**TCL(Transaction Control Language):** [Transaction Control Language](https://www.scaler.com/topics/tcl-commands-in-sql/) as the name suggests manages the issues and matters related to the transactions in any database. They are used to roll back or commit the changes in the database.

**Data Definition Language**

In SQL DDL commands are used to create and modify the structure of a database and database objects. These commands are **CREATE**, **DROP**, **ALTER**, **TRUNCATE**, and **RENAME**.

**1. CREATE**

The **syntax** for the create command is:

1. For creating a database:

CREATE DATABASE database\_name;

1. For creating a table:

## CREATE TABLE table\_name( column1 datatype, column2 datatype, ..... columnN datatype );

**2. DROP**

**a. To drop a table:**

DROP TABLE table\_name;

deletes the table named 'table\_name' if present.

**b. To drop a database:**

DROP DATABASE database\_name;

This removes the database named 'database\_name' if present

**3. ALTER**

**a. syntax to alter the contents of a table:**

ALTER TABLE table\_name

ADD COLUMN column\_name column\_definition(data type);

ALTER TABLE table\_name

DROP COLUMN column\_name;

**4. TRUNCATE**

This command is similar to the drop table command. The only difference is that while the drop command removes the table as well as its contents, the truncate command only erases the contents of the table's contents and not the table itself. Let us take an example to be clearer:

Think of truncating a table as emptying it rather than deleting it.

truncate table my\_table;

select \* from my\_table;

**5. RENAME**

RENAME TABLE table\_name TO table\_name\_new;

**Data Manipulation Language**

DML is used for inserting, deleting, and updating data in a database. It is used to retrieve and manipulate data in a relational database. It includes **INSERT, UPDATE, and DELETE**. Let's discuss these commands one at a time.

**1. INSERT**

Insert statement is used to insert data in an SQL table. Using the Insert query, we can add one or more rows to the table. – implicit inserting

INSERT INTO table\_name

(attribute1, attribute2, ...)

VALUES(val1, val2, ...)

INSERT INTO table\_name

VALUES(val1, val2, ...)

Suppose that we need to insert some rows – explicit inserting

INSERT INTO table\_name

VALUES(val1, val2, ...), (val1, val2, ...)

We don't necessarily have to include all column values. We can for example omit the age column as shown below:

INSERT INTO table\_name

(attribute1, attribute3)

VALUES(val1, val3)

**2. UPDATE**

This command is used to make changes to the data in the table. Its syntax is:

UPDATE table\_name

SET column1 = val1,

column2 = val2,

...

WHERE CLAUSE;

**update my\_table**

**set employed=true**

**where age=12;**

**update my\_table**

**set age = age + 1;**

**3. DELETE**

This command is used to remove a row from a table. the syntax for delete is

DELETE FROM table\_name

WHERE CLAUSE;

**delete from my\_table**

**where first\_name = "myName5";**

**Data Query Language**

DQL commands are used for fetching data from a relational database. They perform read-only queries of data. The only command, **'SELECT'** is equivalent to the projection operation in relational algebra. This command selects the attribute based on the condition described by the **WHERE** clause and returns them.

The data returned is stored in a result table. With the SELECT clause of a SELECT command statement, we specify the columns that we want to be displayed in the query result and, optionally, which column headings we prefer to see above the result table.

The select clause is the first clause and is one of the last clauses of the select statement that the database server evaluates. The reason for this is that before we can determine what to include in the final result set, we need to know all of the possible columns that could be included in the final result set.

The syntax for the SELECT statement is:

SELECT column1 as c1

column2 as c2

...

where / like....

**select age as myAge**

**from my\_table;**

**select \* from my\_table;**

**select \* from my\_table**

**where first\_name like "myName\_";**

**'like' is an operator used for pattern matching**. It uses two special characters embedded in the string to be matched; The **percent sign '%' represents zero, one, or any number of characters, while the underscore sign '\_' represents a single character.**

In the previous example, we used underscore, and therefore all rows except the first were returned because 'myName\_' implies a matching string of the form 'myName' + an additional character. The last 3 strings are of this form and hence are matched, unlike the first one which does not have any additional character present at last.

Similarly, we can obtain all the rows in the first\_name field which ends with zero ('0').

**select \* from my\_table**

**where first\_name like "%0";**

**Data Control Language**

DCL is used to access the stored data. It is used to revoke and grant the user the required access to a database. In the database, this language does not have the feature of rollback. It is a part of the **structured query language (SQL)**. Used by **DBA’s database administrators.**

It helps in controlling access to information stored in a database. It complements the data manipulation language and the data definition language. It is the simplest of three commands. It provides the administrators, to remove and set database permissions to desired users as needed.

These commands are employed to grant, remove and deny permissions to users for retrieving and manipulating a database. There are two relevant commands under this category: grant and revoke.

**a. GRANT**

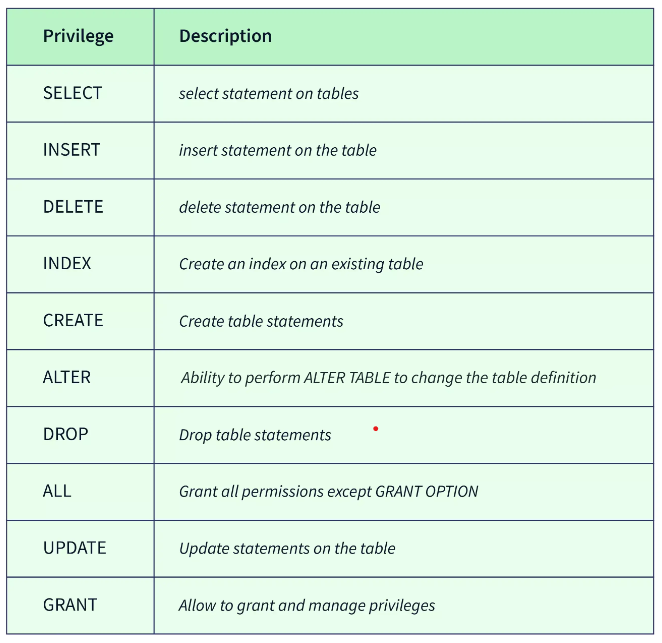
GRANT is a command used to provide access or privileges on the database objects to the users.

GRANT

PRIVILEGES

ON OBJECT

TO USER;

****

**1. SELECT:**

To grant Select Privilege to a table named “tableName", the user name is "userName".

**GRANT SELECT**

**ON tableName**

**TO 'userName'@'localhost';**

**2. Granting multiple privileges to a user:**

To grant multiple Privileges to a user named “username" in table “tableName”, the following GRANT statement should be executed:

**GRANT SELECT, INSERT, DELETE, UPDATE**

**ON tableName**

**TO 'userName'@'localhost';**

**3. Granting all the privileges to a user:**

To Grant all the privileges to a user named “userName” in a table “tableName”, the following Grant statement should be executed.

**GRANT ALL**

**ON tableName**

**TO 'userName'@'localhost';**

**4. Granting a privilege to all users:**

To Grant a specific privilege to all the users in a table “tableName" this Grant statement should be executed.

**GRANT SELECT**

**ON tableName**

**TO '\*'@'localhost';**

**b. REVOKE**

Once you have granted privileges, you may need to revoke some or all of these privileges. To do this, you can run a revoke command. You can revoke any combination of SELECT, INSERT, UPDATE, DELETE, REFERENCES, ALTER, or ALL.

REVOKE privileges

ON object

FROM user;

Object is the name of the database object that you are revoking privileges for. In the case of revoking privileges on a table, this would be the table name. Username of the user that will have these privileges revoked.

Suppose we need to revoke delete permission for the 'tableName' table' from a user named 'userNamed', the following would be the query.

**REVOKE DELETE**

**ON tableName**

**FROM userName;**

To remove every permission use 'ALL'.

**REVOKE ALL**

**ON tableName**

**FROM userName;**

**Transaction Control Language**

TCL includes statements that are used to manage the changes that are made from DML statements i.e, used to control DML staements. It enhances the transactional nature of SQL. The TCL commands in SQL are:

* **COMMIT:** It's a SQL command used in the transaction tables or database to make the current transaction or database statement permanent. It shows the successful completion of a transaction. If we have successfully executed the transaction statement or a simple database query, we want to make the changes permanent. We need to perform the commit command to save the changes, and these changes become permanent for all users. Furthermore, once the commit command is executed in the database, we cannot regain its previous states in which it was earlier before the execution of the first statement. commit;
* **ROLLBACK:** Undoes any changes made to the database. ROLLBACK is the SQL command that is used for reverting changes performed by a transaction. When a ROLLBACK command is issued it reverts all the changes since the last COMMIT or ROLLBACK. ROLLBACK;
* **SAVEPOINT:** This command creates a point in your transaction to which you can roll back. It is a command in SQL that is used with the rollback command. It is a command in Transaction Control Language that is used to mark the transaction in a table. SAVEPOINT some\_name;