Introduction to SQL

Structure Query Language(SQL) is a programming language used for storing and managing data in RDBMS. SQL was the first commercial language introduced for E.F Codd's**Relational** model. Today almost all RDBMS(MySql, Oracle, Infomix, Sybase, MS Access) uses **SQL** as the standard database language. SQL is used to perform all type of data operations in RDBMS.

SQL Command

SQL defines following data languages to manipulate data of RDBMS.

DDL : Data Definition Language

All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

Command	Description
Create	to create new table or database
Alter	for alteration
Truncate	delete data from table
Drop	to drop a table
Rename	to rename a table

DML : Data Manipulation Language

DML commands are not auto-committed. It means changes are not permanent to database, they can be rolled back.

Comma	Description
Insert	to insert a new row
Update	to update existing row

Delete	to delete a row
Merge	merging two rows or two tables

TCL : Transaction Control Language

These commands are to keep a check on other commands and their affect on the database. These commands can annul changes made by other commands by rolling back to original state. It can also make changes permanent.

Command	Description
Commit	to permanently save
Rollback	to undo change
Savepoint	to save temporarily

DCL : Data Control Language

Data control language provides command to grant and take back authority.

Command	Description
Grant	grant permission of right
Revoke	take back permission.

DQL: Data Query Language

Comma	Description
Select	retrieve records from one or more table

Data Definition Language

DDL : Data Definition Language

Create command

create is a DDL command used to create a table or a database.

Creating a Database

To create a database in RDBMS, *create* command is uses. Following is the Syntax,

create database *database-name*;

Example for Creating Database

create database Test;

The above command will create a database named **Test**.

Creating a Table

create command is also used to create a table. We can specify names and datatypes of various columns along.Following is the Syntax,

create table command will tell the database system to create a new table with given table name and column information.

```
create table table-nam{column-name1 datatype1,column-name2 datatype2,column-name3datatype3,column-name4 datatype4};
```

Example for creating Table

```
create table boys (id number(10),name varchar(20),age number(2));
```

The above command will create a new table **Boys** in database system with 3 columns, id number, name and age.

ID	NAME	AGE
1	mani	22
2	mahi	24
3	appy	20

Alter command

alter command is used for alteration of table structures. There are various uses of *alter* command, such as,

- to add a column to existing table
- to rename any existing column
- to change datatype of any column or to modify its size.
- *alter* is also used to drop a column.

To Add Column to existing Table

Using alter command we can add a column to an existing table. Following is the Syntax,

```
alter table table-name add(column-namedatatype);
```

Here is an Example for this,

```
alter table boys add (address varchar(50));
```

ID	NAME	AGE	ADDRESS
1	mani	22	-
2	mahi	24	-
3	appy	20	-

The above command will add a new column *address* to the **Boys** table

To Add Multiple Column to existing Table

Using alter command we can even add multiple columns to an existing table. Following is the Syntax,

```
alter table table-name add(column-name1datatype1, column-name2datatype2);
```

Here is an Example for this,

```
alter table boys add (dob number(10),city varchar(10));
```

The above command will add two new columns to the **boys** table

ID	NAME	AGE	ADDRESS	DOB	CITY
1	Mani	22	-	-	-
2	Mahi	24	-	-	-
3	Appy	20	-	-	-

To Add column with Default Value

Alter command can add a new column to an existing table with default values. Following is the Syntax.Here is an Example for this,

alter table *table-name* add(**column-name1***datatype1***default***data*);

```
alter table boys add (branch varchar(10) default 'EE');
```

ID	NAME	AGE	ADDRESS	DOB	CITY	BRANCH
1	mani	22	-	-	-	EE
2	mahi	24	-	-	-	EE
3	appy	20	-	-	-	EE

The above command will add a new column with default value to the **Student** table

To Modify an existing Column

alter command is used to modify data type of an existing column . Following is the Syntax,

alter table *table-name* modify(**column-name** *datatype*);

Here is an Example for this,

```
alter table boys modify (address char(30));
```

The above command will modify *address* column of the **Boys table**

To Rename a column

Using alter command you can rename an existing column. Following is the Syntax,

alter table *table-name***rename** old-column-name to column-name;

Here is an Example for this,

alter table Student rename address to Location;

The above command will rename *address* column to *Location*.

To Drop a Column

alter command is also used to drop columns also. Following is the Syntax,

```
alter table table-name drop(column-name);
```

Here is an Example for this,

```
alter table boys drop (address);
```

ID	NAME	AGE	DOB	CITY	BRANCH
1	Mani	22	-	-	EE
2	Mahi	24	-	-	EE
3	Appy	20	-	-	EE

The above command will drop *address* column from the **Boys table**

SQL queries to Truncate, Drop or Rename a Table

Truncate command

Truncate command removes all records from a table. But this command will not destroy the table's structure. When we apply truncate command on a table its Primary key is initialized. Following is its Syntax,

```
truncate table table-name
```

Here is an Example explaining it.

```
truncate table boys;
```

The above query will delete all the records of **boys** table.

Truncate command is different from **delete** command. delete command will delete all the rows from a table whereas truncate command re-initializes a table(like a newly created table).

For eg. If you have a table with 10 rows and an auto_increment primary key, if you use *delete* command to delete all the rows, it will delete all the rows, but will not initialize the primary key, hence if you will insert any row after using delete command, the auto_increment primary key will start from 11. But in case of *truncate* command, primary key is re-initialized.

Drop command

Drop query completely removes a table from database. This command will also destroy the table

structure. Following is its Syntax,

```
drop table table-name;
```

Here is an Example explaining it.

```
drop table boys;
```

The above query will delete the **boys** table completely. It can also be used on Databases. For Example, to drop a database,

drop database Test;

The above query will drop a database named **Test** from the system.

Rename query

Rename command is used to rename a table. Following is its Syntax,

```
rename table old-table-name to new-table-name;
```

Here is an Example explaining it.

```
rename boys to students;
```

The above query will rename **Boys** table to **Students.**

Data Manipulation Language

DML : Data Manipulation Language

Data Manipulation Language (DML) statements are used for managing data in database. DML commands are not auto-committed. It means changes made by DML command are not permanent to database, it can be rolled back.

1) INSERT command

Insert command is used to insert data into a table. Following is its general syntax,

```
INSERT into table-namevalues(data1,data2,..)
```

Consider a table **Student** with following fields.

```
insert into boys values(1, 'mani',22);

insert into boys values(2, 'mahi',24);  insert into boys values(3, 'appy',20);
```

Example to Insert NULL value to a column

The statements below will insert NULL value into **age** column of the Student table.

```
insert into students values(4,'padm',null);
```

ID	NAME	AGE
1	Mani	22
2	Mahi	24
3	Appy	20
4	Padm	-

2) UPDATE command

Update command is used to update a row of a table. Following is its general syntax,

```
UPDATEtable-name set column-name = value where condition;
```

Lets see an example,

```
update students set age=22 where id=4;
```

ID	NAME	AGE
1	mani	22
2	mahi	24
3	appy	20
4	padm	22
5	rahul	-

Example to Update multiple columns

```
update students set name='pri',age=25 where id=5;
```

ID	NAME	AGE
1	Mani	22
2	Mahi	24
3	Appy	20
4	Padm	22
5	Pri	25

3) Delete command

Delete command is used to delete data from a table. Delete command can also be used with condition to delete a particular row. Following is its general syntax,

```
DELETE from table-name;
```

a Table

```
delete from students;
```

The above command will delete all the records from **Students** table.

Example to Delete a particular Record from a Table

Consider the following **Students** table

ID	NAME	AGE
1	Mani	22
2	Mahi	24
3	Appy	20
4	Padm	22
5	Pri	25

```
delete from students where id=5;
```

The above command will delete the record where id is 5 from **Students** table.

ID	NAME	AGE
1	Mani	22
2	Mahi	24
3	Appy	20
4	Padm	22

DQL : Data Query Language

SELECT Query

Select query is used to retrieve data from a tables. It is the most used SQL query. We can retrieve complete tables, or partial by mentioning conditions using WHERE clause.

Syntax of SELECT Query

SELECT column-name1, column-name2, column-name3, column-nameN from *table-name*;

Example for SELECT Query

Conside the following **Students** table,

ID	NAME	AGE
1	mani	22
2	mahi	24
3	appy	20

```
selectage,name from students;
```

AGE	NAME
22	mani
24	mahi
20	appy

The above command diisplays age and name column from the table **Students.**

Example to Select all Records from Table

A special character **asterisk*** is used to address all the data(belonging to all columns) in a query. *SELECT* statement uses * character to retrieve all records from a table.

```
select* from students;
```

ID	NAME	AGE
1	mani	22
2	mahi	24
3	appy	20

The above query will show all the records of Studentstable, that means it will show complete Students table as result.

Example to Select particular Record based on Condition

```
select* from students where name='mahi';
```

ID	NAME	AGE
2	mahi	24

Example to Perform Simple Calculations using Select Query

Conside the following **Employee** table.

ID	NAME	DEPT	SALARY
1	mani	EE	20000
3	padm	EC	50000
4	pri	CS	40000

```
select id,name,salary+500 from employee;
```

The above command will display a new column in the result, showing 3000 added into existing salaries of the employees.

ID	NAME	SALARY+500
1	mani	20500
3	padm	50500
4	pri	40500