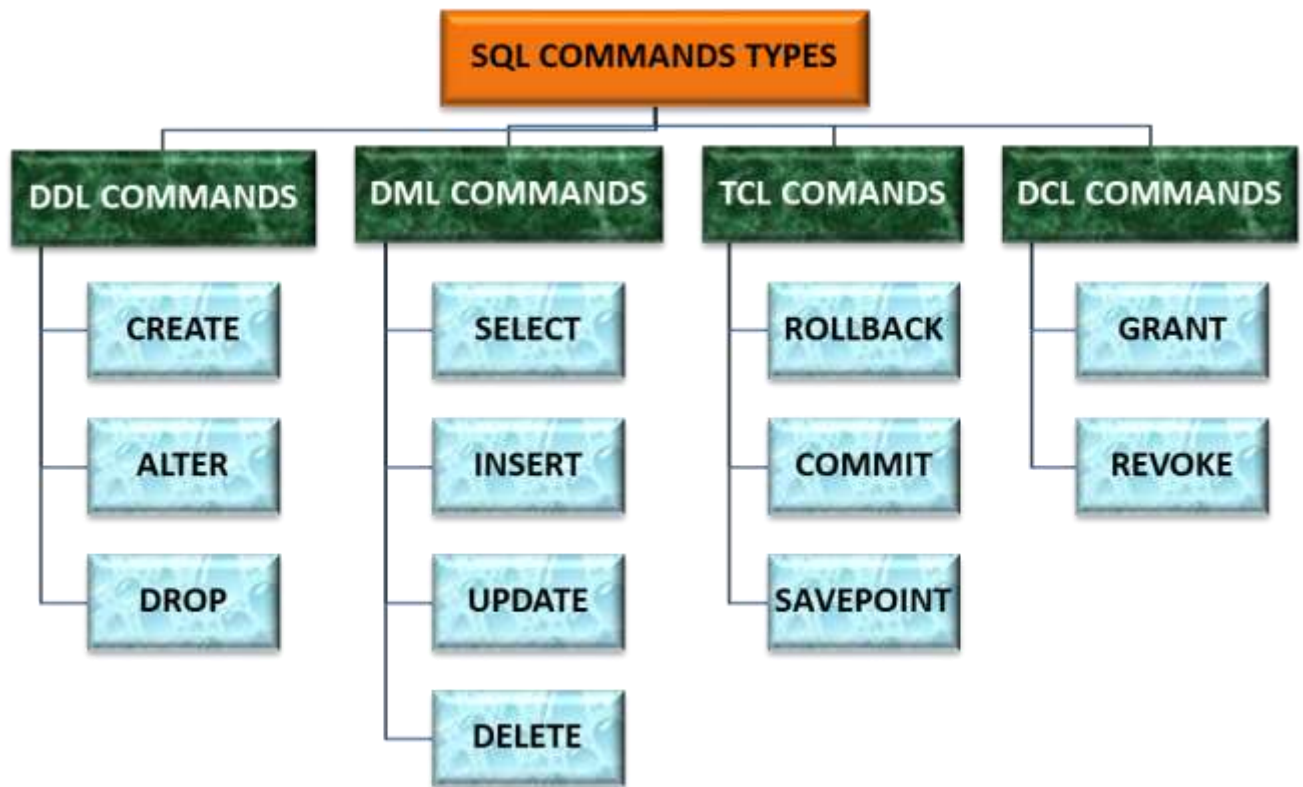


## MYSQL & Its COMMANDS

### SQL (STRUCTURED QUERY LANGUAGE ) COMMANDS

SQL, Structured Query Language, is a programming language designed to manage data stored in relational databases. SQL commands are grouped into four major categories depending on their functionality:



**Data Definition Language (DDL)** - are used for creating, modifying, and dropping the physical structure of database objects.

Examples : **CREATE, ALTER, DROP, RENAME**

NAME OF COMMANDS	USE
<b>CREATE</b>	To create physical structure of database Objects like Table, Function, Views etc.
<b>ALTER</b>	To alteration or modification in database Objects
<b>DROP</b>	To delete/remove in database Objects
<b>RENAME</b>	To is used to rename an object existing in the database.

**Data Manipulation Language (DML)** - are used for storing, retrieving, modifying, and deleting data.

Example : **SELECT, INSERT, UPDATE, and DELETE.**

NAME OF COMMANDS	USE
<b>SELECT</b>	To retrieve data from the a database/Table
<b>INSERT</b>	To insert row into a table
<b>DELETE</b>	To delete row(s) from Table
<b>UPDATE</b>	To update existing data within a table

**Transaction Control Language (TCL)** - These SQL commands are used for managing changes done through DML commands.

Example COMMIT, ROLLBACK, and SAVEPOINT.

NAME OF COMMANDS	USE
<b>COMMIT</b>	To save DML Transactions permanently
<b>ROLLBACK</b>	To Cancel/Undo DML Transactions
<b>SAVEPOINT</b>	To set a SAVEPOINT within a Transaction so that one can rollback transaction to that point

**Data Control Language (DCL)** - These SQL commands are used for providing security to database objects.

These commands are GRANT and REVOKE.

NAME OF COMMANDS	USE
GRANT	To give user's access privilege to databases
REVOKE	To withdraw access privileges given by GRANT command

### Before Starting with SQL Commands lets understand SQL Datatype

The following table lists the general data types in SQL:

Data type	Description
<b>CHARACTER(n)</b>	Character string. Fixed-length n
<b>VARCHAR(n)</b>	Character string of Variable length. Maximum length <b>n</b>
<b>INTEGER(p)</b>	Integer numerical (no decimal). Precision p
<b>SMALLINT</b>	Integer numerical (no decimal). Precision 5
<b>INTEGER</b>	Integer numerical (no decimal). Precision 10
<b>BIGINT</b>	Integer numerical (no decimal). Precision 19
<b>DECIMAL(p,s)</b>	Precision p, Scale s. Example: decimal(5,2) is a number that has 3 digits before the decimal and 2 digits after the decimal
<b>NUMERIC(p,s)</b>	Same as DECIMAL

<b>FLOAT(p)</b>	Approximate numerical, mantissa precision p. A floating number in base 10 exponential notation. The size argument for this type consists of a single number specifying the minimum precision
<b>REAL</b>	Approximate numerical, mantissa precision 7
<b>FLOAT</b>	Approximate numerical, mantissa precision 16
<b>DOUBLE PRECISION</b>	Approximate numerical, mantissa precision 16
<b>DATE</b>	Stores year, month, and day values
<b>TIME</b>	Stores hour, minute, and second values

# SQL COMMANDS

Let's begin with Creation of a table



We are using My sql , Tables can be created under database by using Create database command

**SYNTAX :**

```
CREATE DATABASE < database name>;
```

**Example :**

```
CREATE DATABASE LIBRARY;
```

TO **MOVE** IN TO DATABASE we use USE command

Syntax

```
USE < DATABASE NAME>
```

EXAMPLE

```
USE LIBRARY
```

**CREATE TABLE:** Command used to create physical structure of a table.

Syntax :

```
CREATE TABLE <Table Name>
(<Col1>    <DataType(Size)>,
 <Col2>    <DataType(size)>,
 <Col3>    <DataType(size)>,
 :
 :
 <Col N>   <DataType(size)>,. );
```

**Example :** Here we are creating table to store records of **Books**

For example-

```
CREATE TABLE Books
(B_NO INT(4) ,
 B_Name CHAR(20),
 price NUMBER (4),
 Author VARCHAR(35),
 Publisher VARCHAR(30) )
```

**B\_NO** : to store Book Number,  
**B\_Name** : to store Book Name,  
**Price** : Price of Book,  
**Author**: to store writer name,  
**publisher** : to store name of publisher

**CONSTRAINT** is a condition applicable on a field or group of fields.

**Two types of constraints:**

**Column Constraint:** - apply only to individual column

**Table Constraint:** - apply to groups of columns

### **TYPES**

UNIQUE CONSTRAINT	It forces every value in a column or set of columns to be unique.  It allows the input of nulls until we also define NOT NULL constraint for the same column.
DEFAULT CONSTRAINT	A column may be given a default value through DEFAULT option.  This constraint prevents nulls from occurring if a row is inserted without a value for the column.
NOT NULL	This constraint ensures that the null values (empty values) are not permitted for a specified column.
PRIMARY KEY CONSTRAINT	This constraint is used to <b>uniquely identify</b> data. It does not accept null values
CHECK CONSTRAINT	It explicitly defines a condition that each row must satisfy
FOREIGN KEY	it is a non -key attribute of one table derived from primary key from other table

### **Applying Constraint**

```
CREATE TABLE Books
(B_NO INT(4) PRIMARY KEY ,
 B_Name CHAR(20) NOT NULL,
 price NUMBER (4),
 Author VARCHAR(35) ,
 Publisher VARCHAR(30) )
```



After Creating Blank Structure of a Table “BOOK” we would like to store data in it

## BASIC SQL COMMANDS

**1. INSERT INTO:** Command used to insert a row in the specified table

Syntax :

```
INSERT INTO <Table name> VALUES  
<Value1>,<Value>.....<Value n>
```

Inserting a  
row (all  
columns)

Example :

```
INSERT INTO BOOKS VALUES  
(B101,"Computer Science",500,"Sangeeta  
Chauhan","Gwl Pub")
```

Syntax :

```
INSERT INTO <Table name>  
(<col1>, <col>.....)  
( VALUES <Value1>, <Value>.....<Value n>)
```

Example :

```
INSERT INTO BOOK (B_NO, B_NAME)  
VALUES (104,'PYTHON');
```

**2. SELECT :** Command used to Fetch /View data from table

SYNTAX :

```
SELECT [ALL / DISTINCT] expr1 [AS col1], expr2 [AS col2] ;  
FROM tablename WHERE condition
```

[ ]- INDICATES OPTIONAL PART

{ }- indicates EITHER OF THESE CAN BE USED

< option > - write value of option in place of < >

- The query will select rows from the source **tablename** and output the result in table form.
- Expressions **expr1, expr2** can be :
  - (1) a column, or
  - (2) an expression of functions and fields.
- And **col1, col2** are their corresponding column names in the output table.

LET'S UNDERSTAND USE OF **SELECT** COMMAND WITH THE HELP OF FOLLOWING QUERIES:

❖ Write a query

- ❖ to display all records of Book table  
SELECT \* FROM BOOKS;

DISPLAYING ALL RECORDS FROM TABLE

- ❖ to display book name from table  
SELECT B\_NAME FROM BOOK

DISPLAYING SELECTED COLUMNS FROM TABLE

- ❖ to display books with Price greater than 450  
SELECT \* FROM BOOKS  
WHERE PRICE >= 450;

DISPLAYING SELECTED ROWS FROM TABLE

### 3. **DELETE** COMMAND : Used to delete all or Selected Rows from table

SYNTAX :

```
DELETE FROM <TABLE NAME>  
[WHERE <CONDITION>];
```

EXAMPLE :

```
DELETE FROM BOOKS;
```

DELETING ALL ROWS FROM TABLE

```
DELETE FROM BOOKS  
WHERE AUTHOR='XYZ'
```

DELETING ONLY ROWS WITH  
AUTHOR NAME=XYZ



4. **UPDATE** : This command is used to modify/update the data stored in the Table

SYNTAX

```
UPDATE <TABLE NAME>  
SET <column name1>=<value1>  
, [<column name 2>=<value2>]  
[ WHERE <condition>];
```

**EXAMPLE** : Write a query to change the name of publisher from 'XYZ' to 'ABC' of book 'FUN WITH Mathematics'

```
UPDATE Book  
SET PUBLISHER='ABC'  
WHERE B_NAME=' FUN WITH Mathematics';
```

**NOTE** : If we don't write condition with query changes will be applied on all rows means whole table will be updated

5. **ALTER TABLE**: This command is used to modify/update/delete columns from structure of a Table.

SYNTAX

```
ALTER TABLE <TABLE NAME>  
{ ADD/ MODIFY/DROP COLUMN}  
( <column name1 DATA TYPE (Size);
```

Example

```
ALTER TABLE BOOK  
MODIFY B_NO  
INTEGER(5);
```

MODIFYING COLUMN **B\_NO** OF  
TABLE BOOK

```
ALTER TABLE BOOK  
ADD QTY INTEGER(4);
```

ADDING A NEW COLUMN **QTY** TO  
TABLE BOOK

```
ALTER TABLE BOOK  
DROP COLUMN QTY;
```

DELETING THE COLUMN **QTY** FROM  
THE TABLE BOOK

6. **DROP TABLE:** This command is used to drop the whole table with its contents

SYNTAX 

```
DROP TABLE <table name>;
```

EXAMPLE 

```
DROP TABLE BOOK
```

DELETING THE TABLE **BOOK** WITH  
ALL DATA STORED IN IT

7. **ROLLBACK :** This command is used to undo the transactions done using DML commands

```
ROLLBACK;
```

UNDO ALL RECENT TRANSACTION

8. **COMMIT :** This command is used to save the transaction, done by using DML commands, permanently

```
COMMIT;
```

ALL THE CURRENT DML  
TRANSACTIONS WILL BE SAVED

-----continued to PART 3-----