

SQL Statements

DDL [Data Definition Language]

- > CREATE
- > ALTER
- > DROP
- > TRUNCATE

DML [Data Manipulation Language]

- > INSERT
- > UPDATE
- > DELETE

TCL [Transaction Control Language]

- > COMMIT
- > ROLLBACK
- > SAVEPOINT

DCL [Data Control Language]

- > GRANT
- > REVOKE

DQL [Data Query Language]

- > SELECT

> JOINS

Database: It is a place where we store the data in the systematic and organized manner.

Creation of Database

To create the database,
`CREATE DATABASE database_name;`

To access the database,

USE database_name;

To display the databases present in MySQL,

SHOW DATABASES;

To display the tables present in the current database,

SHOW TABLES;

DDL [Data Definition Language]

CREATE : This command is used to create the database and its objects such as tables, users, views, indexes etc..

Syntax to create a table,

```
CREATE TABLE table_name
(
    column_name_1 DATATYPE CONSTRAINT NULL/NOT
NULL,
    column_name_2 DATATYPE CONSTRAINT NULL/NOT
NULL,
    :
    column_name_n DATATYPE CONSTRAINT NULL/NOT
```

NULL

);

CREATE TABLE ACCOUNTS

(

ACC_NO BIGINT PRIMARY KEY,

ACC_HOLDER_NAME VARCHAR(20) NOT NULL,

PHONE BIGINT UNIQUE NOT NULL

CHECK(LENGTH(PHONE)=10),

MAIL VARCHAR(30) UNIQUE,

GENDER ENUM('Male', 'Female', 'Others'),

NATIONALITY VARCHAR(30) DEFAULT 'Indian'

);

CREATE TABLE BRANCH

```
(  
    BID INT PRIMARY KEY AUTO_INCREMENT,  
    BNAME VARCHAR(20) NOT NULL,  
    AREA VARCHAR(30) UNIQUE NOT NULL,  
    CITY VARCHAR(20) DEFAULT 'BENGALURU'  
);
```

ALTER : This command is used to modify the structure of the table.

1. To add a column,

ALTER TABLE table_name

```
ADD column_name DATATYPE CONSTRAINT NULL/NOT  
NULL [AFTER column_name_x];
```

```
ALTER TABLE ACCOUNTS  
ADD BID INT NULL;
```

```
ALTER TABLE BRANCH  
ADD IFSC_CODE VARCHAR(15) UNIQUE  
CHECK(LENGTH(IFSC_CODE)=15) AFTER BNAME;
```

2. To drop a column,

```
ALTER TABLE table_name  
DROP column_name;
```



```
ALTER TABLE ACCOUNTS  
DROP NATIONALITY;
```

```
ALTER TABLE BRANCH  
DROP AREA;
```

3. To modify the datatype,

```
ALTER TABLE table_name  
MODIFY column_name NEW_DATATYPE NULL/NOT NULL;
```

```
ALTER TABLE ACCOUNTS  
MODIFY ACC_HOLDER_NAME CHAR(10) NOT NULL;
```

```
ALTER TABLE BRANCH
```

```
MODIFY CITY CHAR(20) NOT NULL;
```

4. To modify NULL/NOT NULL,

```
ALTER TABLE table_name  
MODIFY column_name existing_datatype NULL/NOT  
NULL;
```

```
ALTER TABLE ACCOUNTS  
MODIFY MAIL VARCHAR(30) NOT NULL;
```

```
ALTER TABLE BRANCH  
MODIFY BNAME VARCHAR(30) NULL;
```

5. To change the column_name,

```
ALTER TABLE table_name  
CHANGE old_column_name new_column_name  
existing_datatype NULL/NOT NULL;
```

```
ALTER TABLE ACCOUNTS  
CHANGE acc_holder_name NAME CHAR(10) NOT NULL;
```

```
ALTER TABLE BRANCH  
CHANGE CITY AREA VARCHAR(30) NOT NULL;
```

6. To change the table_name,

```
ALTER TABLE table_name  
RENAME new_table_name;
```

```
ALTER TABLE ACCOUNTS  
RENAME ACC;
```

```
ALTER TABLE BRANCH  
RENAME EXT;
```

7. To add a constraint,

```
ALTER TABLE table_name  
ADD CONSTRAINT PRIMARY KEY(column_name);  
ADD CONSTRAINT UNIQUE(column_name);  
ADD CONSTRAINT CHECK(condition);  
ADD CONSTRAINT FOREIGN KEY(column_name)  
REFERENCES parent_table_name(column_name);
```

```
ALTER TABLE ACC  
ADD CONSTRAINT FOREIGN KEY(BID) REFERENCES  
EXT(BID);
```

8. To remove the Primary Key

```
ALTER TABLE table_name  
DROP PRIMARY KEY;
```

```
ALTER TABLE ACC  
DROP PRIMARY KEY;
```

9. To remove the UNIQUE constraint

```
ALTER TABLE table_name  
DROP INDEX column_name;
```

```
ALTER TABLE ACC  
DROP INDEX MAIL;
```

10. To remove the Foreign Key and Check
Constraint

```
ALTER TABLE table_name  
DROP CONSTRAINT constraint_name;
```

```
ALTER TABLE ACC  
DROP CONSTRAINT acc_chk_1;
```

3. DROP: This command is used to delete the database and its objects.

Syntax:

```
DROP TABLE table_name;
```

```
DROP TABLE ACC;
```
