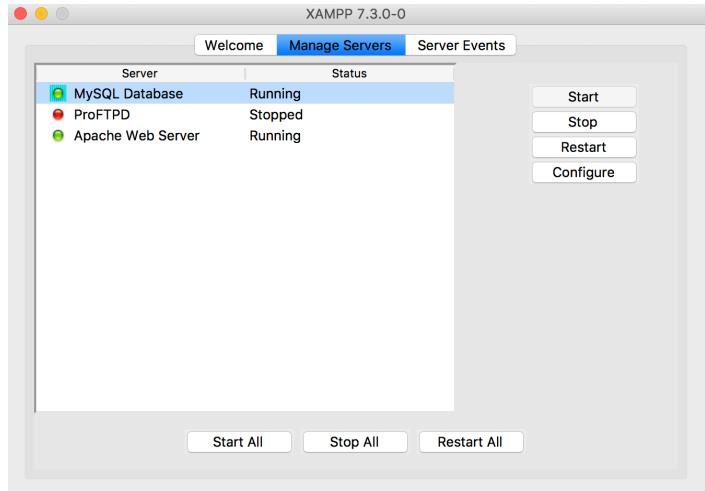


Database Programming using Structured Query Language

Steps to start MySql from Command Line :

Start Mysql service from XAMPP control panel.



Start Terminal / Command Prompt to start MySql.

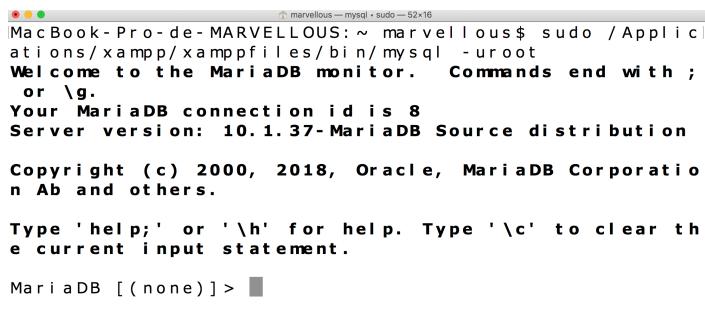


For Windows user enter below command to start MySql

```
cd c:\xampp\mysql\bin
mysql.exe -u root
```

For Linux user enter below command to start MySql

```
/Applications/xampp/xamppfiles/bin/mysql -uroot -p
```



Important concepts of SQL

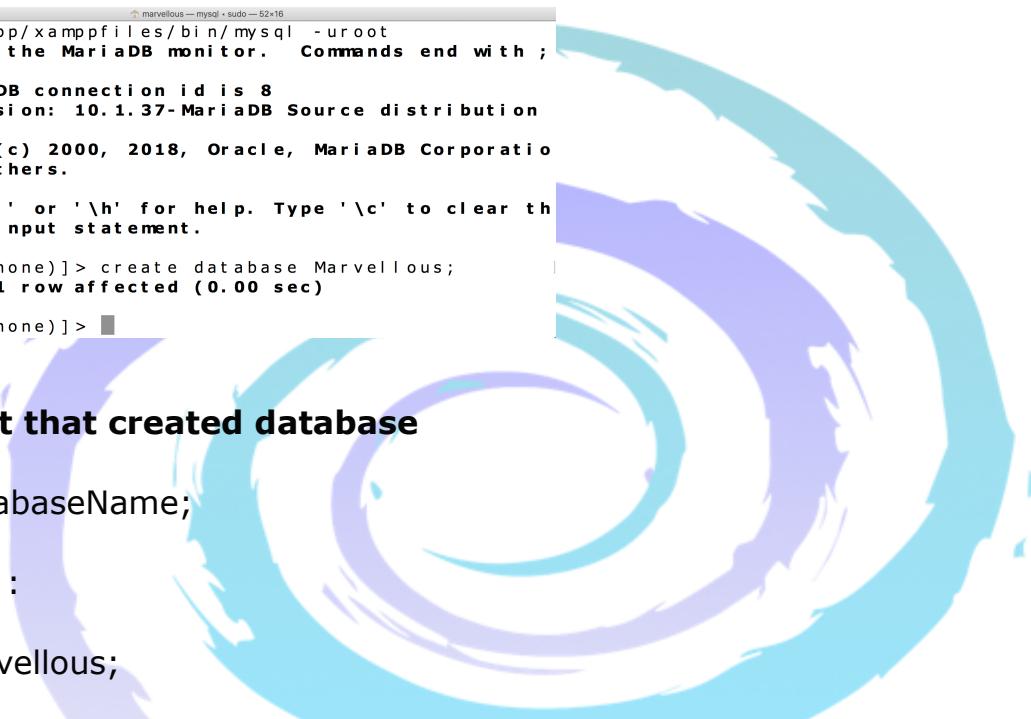
1. Create new Database

```
CREATE DATABASE databasename;
```

Example :

Create new database named as Marvellous.

```
CREATE DATABASE Marvellous;
```



```
mysql -u root
Welcome to the MariaDB monitor.  Commands end with ;
or \g.
Your MariaDB connection id is 8
Server version: 10.1.37-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database Marvellous;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> 
```

2. Select that created database

```
USE DatabaseName;
```

Example :

```
USE Marvellous;
```



```
or \g.
Your MariaDB connection id is 8
Server version: 10.1.37-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database Marvellous;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> USE Marvellous;
Database changed
MariaDB [Marvellous]> 
```

3. Now we can create table in that database.

```
CREATE TABLE table_name
```

```
(
```

```
  column1 datatype,
  column2 datatype,
  column3 datatype,
```

```
....
```

```
);
```

Example :

CREATE TABLE Student

```
(  
  RollNo int,  
  Name varchar(255),  
  Address varchar(255),  
  City varchar(255),  
  Marks int  
);
```

```
MariaDB [Marvellous] >  
MariaDB [Marvellous] > create table student(Roll No int,  
Name varchar(255),Address varchar(255),Marks int)  
;  
Query OK, 0 rows affected (0.03 sec)
```

```
MariaDB [Marvellous] >
```

4. Check layout of created table

DESCRIBE Table_name;

Example :

DESCRIBE Student;

```
MariaDB [Marvellous] > DESCRIBE Student;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Ex  
tra |  
+-----+-----+-----+-----+-----+  
| RollNo | int(11) | YES | | NULL |  
| Name | varchar(255) | YES | | NULL |  
| Address | varchar(255) | YES | | NULL |  
| Marks | int(11) | YES | | NULL |  
+-----+-----+-----+-----+-----+
```

5. Now insert record into created table

INSERT INTO table_name VALUES (value1, value2, value3, ...);

Example :

INSERT INTO Student VALUES (11,'Amit','Karve Road Pune',97);

```
marvelous — mysql > insert into student values(11, 'Ami t', 'Karve Road Pune', 78);
Query OK, 1 row affected (0.01 sec)

MariaDB [Marvellous] >
```

6. Now we can display the contents of table using select query.

SELECT * FROM table_name;

Example :

SELECT * FROM Student;

```
marvelous — mysql > select * from student;
+-----+-----+-----+-----+
| Roll No | Name      | Address          | Marks |
+-----+-----+-----+-----+
|     11   | Ami t     | Karve Road Pune |    78  |
|     12   | Sagar     | Singhabad Road Pune |    98  |
|     13   | Sumi t    | Mumbai           |    91  |
|     14   | Akshay    | Nasik            |    71  |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [Marvellous] >
```

7. We can search specific record from the table using where clause

Example :

SELECT * FROM STUDENT WHERE RollNo = 12;

```
marvelous — mysql > select * from student where RollNo=12 ;
+-----+-----+-----+-----+
| Roll No | Name      | Address          | Marks |
+-----+-----+-----+-----+
|     12   | Sagar     | Singhabad Road Pune |    98  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [Marvellous] >
```

Example :

SELECT * FROM STUDENT WHERE Marks = 91;

```
mariaDB [ Marvellous ] > SELECT * FROM STUDENT WHERE Ma
rks=91;
+-----+-----+-----+-----+
| RollNo | Name   | Address | Marks |
+-----+-----+-----+-----+
|     13 | Sumit  | Mumbai  |      91 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

MariaDB [Marvellous] >

Example :

SELECT * FROM STUDENT WHERE Address = 'Mumbai';

```
mariaDB [ Marvellous ] > SELECT * FROM STUDENT WHERE Address='Mumbai';
+-----+-----+-----+-----+
| RollNo | Name   | Address | Marks |
+-----+-----+-----+-----+
|     13 | Sumit  | Mumbai  |      91 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

MariaDB [Marvellous] >

Example :

SELECT * FROM STUDENT WHERE Marks > 90 AND Address = 'Mumbai';

```
mariaDB [ Marvellous ] > SELECT * FROM STUDENT WHERE Marks > 90 AND A
ddress = 'Mumbai';
+-----+-----+-----+-----+
| RollNo | Name   | Address | Marks |
+-----+-----+-----+-----+
|     13 | Sumit  | Mumbai  |      91 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

MariaDB [Marvellous] >

8. We can display records in specific order using Order By Clause.

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

Example :

SELECT * FROM Student ORDER BY Marks;

```
mariaDB [Marvellous] > SELECT * FROM Student ORDER BY Marks;
+-----+-----+-----+-----+
| Roll No | Name | Address | Marks |
+-----+-----+-----+-----+
| 14 | Akshay | Nasik | 71 |
| 11 | Amit | Karve Road Pune | 78 |
| 13 | Sumit | Mumbai | 91 |
| 12 | Sagar | Singhagad Road Pune | 98 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [Marvellous] >
```

Example :

SELECT * FROM Student ORDER BY Marks DESC;

```
mariaDB [Marvellous] > SELECT * FROM Student ORDER BY Marks DESC;
+-----+-----+-----+-----+
| Roll No | Name | Address | Marks |
+-----+-----+-----+-----+
| 12 | Sagar | Singhagad Road Pune | 98 |
| 13 | Sumit | Mumbai | 91 |
| 11 | Amit | Karve Road Pune | 78 |
| 14 | Akshay | Nasik | 71 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [Marvellous] >
```

9. Update any specific record from table.

UPDATE table_name
 SET column1 = value1, column2 = value2, ...
 WHERE condition;

Example :

UPDATE Student SET Name = 'Vishal' WHERE RollNo = 12;

```
mariaDB [Marvellous] > update Student set Name='Vishal' where RollNo=12;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [Marvellous] > select * from Student;
+-----+-----+-----+-----+
| Roll No | Name | Address | Marks |
+-----+-----+-----+-----+
| 11 | Amit | Karve Road Pune | 78 |
| 12 | Vishal | Singhagad Road Pune | 98 |
| 13 | Sumit | Mumbai | 91 |
| 14 | Akshay | Nasik | 71 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [Marvellous] >
```

10. We can Delete specific record from table

DELETE FROM table_name WHERE condition;

Example :

DELETE FROM Student WHERE RollNo=14;

```
marvelous — mysql > delete from student where Roll No=11;
Query OK, 1 row affected (0.00 sec)

MariaDB [Marvellous] > select * from Student;
+-----+-----+-----+-----+
| Roll No | Name      | Address          | Marks |
+-----+-----+-----+-----+
|     12   | Vishal    | Singhagad Road Pune |    98 |
|     13   | Sumit     | Mumbai           |    91 |
|     14   | Akshay    | Nasik            |    71 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [Marvellous] >
```

11. We can call Min function on table

SELECT MIN(column_name)
 FROM table_name
 WHERE condition;

Example :

SELECT MIN(Marks)
 FROM Student;

```
marvelous — mysql > select MIN(Marks) from Student;
+-----+
| MIN(Marks) |
+-----+
|      71    |
+-----+
1 row in set (0.01 sec)

MariaDB [Marvellous] >
```

12. We can call MAX function on table

SELECT MAX(column_name)
 FROM table_name
 WHERE condition;

Example :

SELECT MAX(Marks)
 FROM Student;

```
● ● ● mariadb [ Marvellous ] > select MAX(Marks) from Student ;
+-----+
| MAX( Marks ) |
+-----+
|      98      |
+-----+
1 row in set (0.00 sec)

mariadb [ Marvellous ] >
```

13. We can call COUNT function to count specific records from table.

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

Example :

```
SELECT COUNT(Name)
FROM Student
WHERE Marks > 90;
```

```
● ● ● mariadb [ Marvellous ] > Select COUNT( Name ) from Student where Marks >
90;
+-----+
| COUNT( Name ) |
+-----+
|      2       |
+-----+
1 row in set (0.00 sec)

mariadb [ Marvellous ] >
```

14. The AVG() function returns the average value of a numeric column.

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

Example :

```
SELECT AVG(Marks)
FROM Student;
```

```
● ● ● mariadb [ Marvellous ] > SELECT AVG( Marks ) from Student ;
+-----+
| AVG( Marks ) |
+-----+
|    86.6667   |
+-----+
1 row in set (0.00 sec)

mariadb [ Marvellous ] >
```

15. The SUM() function returns the total sum of a numeric column.

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

Example :

```
SELECT SUM(Marks)
FROM Student;
```



```
MariaDB [Marvellous] > SELECT SUM(Marks) from Student;
+-----+
| SUM(Marks) |
+-----+
|      260   |
+-----+
1 row in set (0.00 sec)

MariaDB [Marvellous] >
```

16. We can perform Pattern Matching using LIKE operator.

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

Example :

```
SELECT Name
FROM Student
WHERE Name LIKE 'A%';
```



```
MariaDB [Marvellous] > select * from Student;
+-----+-----+-----+-----+
| Roll No | Name    | Address          | Marks |
+-----+-----+-----+-----+
|     12   | Vishal  | Singhagad Road Pune |    98  |
|     13   | Sumit   | Mumbai           |    91  |
|     14   | Akshay   | Nasi k           |    71  |
|     15   | Aniket   | Sangli           |    56  |
|     16   | Amar     | Satara           |    86  |
|     17   | Amey     | Pune             |    76  |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MariaDB [Marvellous] > Select name from student where name like 'A%'
';
+-----+
| name |
+-----+
| Akshay |
| Aniket |
| Amar  |
| Amey  |
+-----+
```

```
SELECT Name
FROM Student
WHERE Name LIKE 'A____';
```

```
● ● ●
2 rows in set (0.00 sec)

MariaDB [Marvellous] > select * from Student;
+-----+-----+-----+-----+
| Roll No | Name      | Address          | Marks |
+-----+-----+-----+-----+
|    12   | Vishal    | Singhabad Road Pune |    98 |
|    13   | Sumit     | Mumbai           |    91  |
|    14   | Akshay    | Nasik            |    71  |
|    15   | Aniket    | Sangli           |    56  |
|    16   | Amar      | Satara           |    86  |
|    17   | Ame y     | Pune             |    76  |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

MariaDB [Marvellous] > Select name from student where name like 'A_';
-- ;
+-----+
| name |
+-----+
| Amar |
| Ame y|
+-----+
```

17. Use of IN operator to search in specific values.

SELECT column_name(s)
 FROM table_name
 WHERE column_name IN (value1, value2, ...);

Example :
 Select Name
 From student
 Where Name IN('Pune','Nasik');

```
● ● ●
MariaDB [Marvellous] > select name from student where Address in('Pune','Nasik');
+-----+
| name |
+-----+
| Akshay |
| Ame y |
+-----+
2 rows in set (0.00 sec)
```

18. We can use BETWEEN operator to search in specific range.

SELECT column_name(s)
 FROM table_name
 WHERE column_name BETWEEN value1 AND value2;

SELECT Name
 FROM Student
 WHERE Marks BETWEEN 80 AND 95;

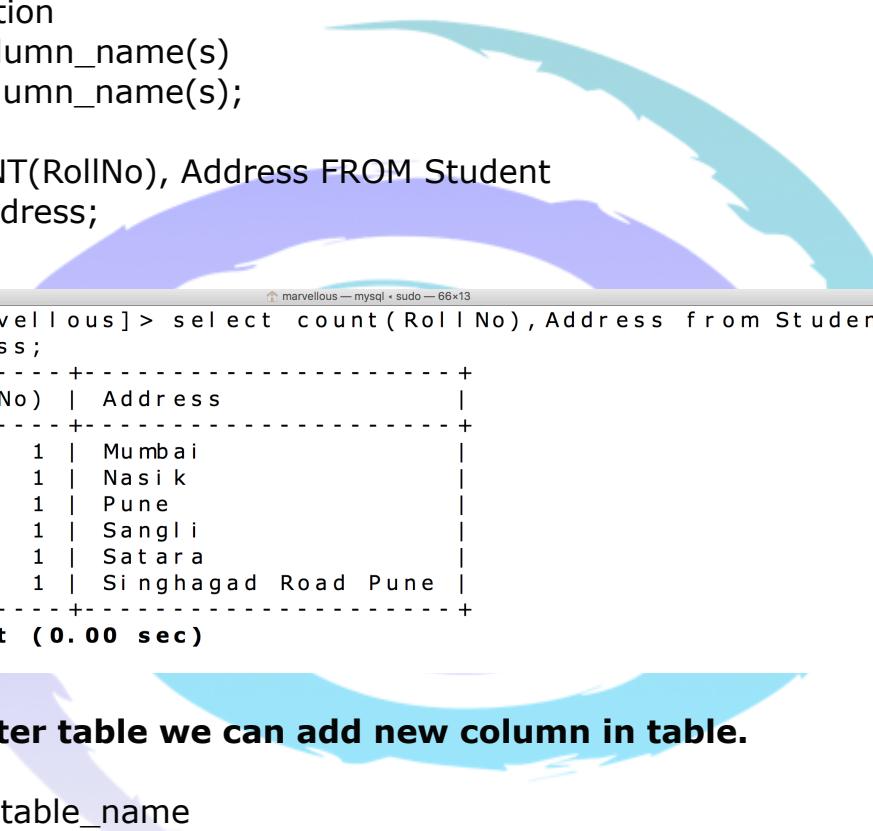
```
mariaDB [Marvellous]> select name from student where Marks BETWEEN
  80 AND 95;
+-----+
| name |
+-----+
| Sumit |
| Amar  |
+-----+
2 rows in set (0.00 sec)

MariaDB [Marvellous]>
```

19. The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

```
SELECT COUNT(RollNo), Address FROM Student
GROUP BY Address;
```



```
mariaDB [Marvellous]> select count(Roll No), Address from Student Group by Address;
+-----+-----+
| count(Roll No) | Address      |
+-----+-----+
|          1    | Mumbai       |
|          1    | Nasik        |
|          1    | Pune         |
|          1    | Sangli       |
|          1    | Satara       |
|          1    | Singhagad Road Pune |
+-----+-----+
6 rows in set (0.00 sec)
```

20. Using Alter table we can add new column in table.

```
ALTER TABLE table_name
ADD column_name datatype;
```

```
ALTER TABLE Student
ADD Age int;
```



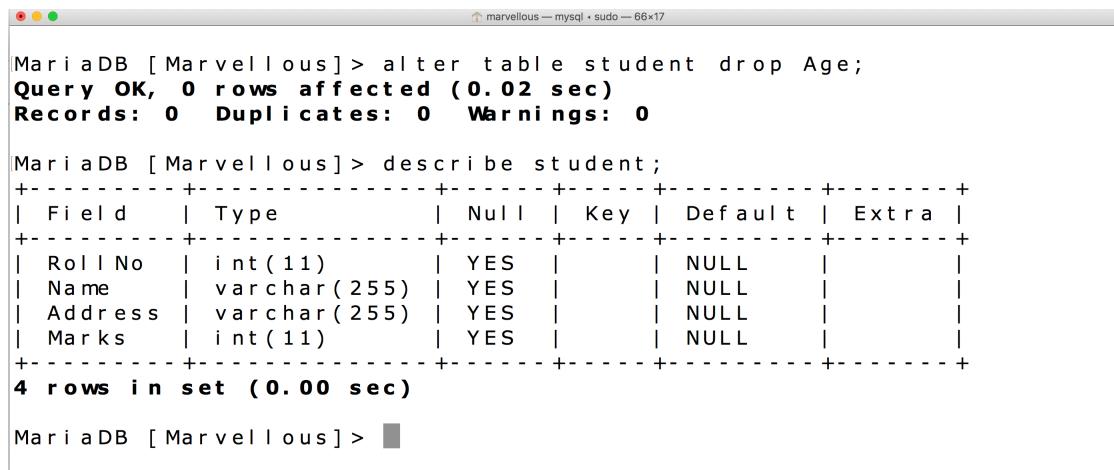
```
mariaDB [Marvellous]> alter table student add Age int;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Marvellous]> describe student;
+-----+-----+-----+-----+-----+
| Field   | Type     | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Roll No | int(11) | YES  |     | NULL    |       |
| Name    | varchar(255)| YES  |     | NULL    |       |
| Address | varchar(255)| YES  |     | NULL    |       |
| Marks   | int(11)  | YES  |     | NULL    |       |
| Age     | int(11)  | YES  |     | NULL    |       |
```

21. Using Alter table we can remove column from table.

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

```
ALTER TABLE Student
DROP Age;
```



```
MariaDB [Marvellous]> alter table student drop Age;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Marvellous]> describe student;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| Roll No | int(11) | YES | NO   | NULL    |       |
| Name    | varchar(255)| YES | NO   | NULL    |       |
| Address | varchar(255)| YES | NO   | NULL    |       |
| Marks   | int(11)  | YES | NO   | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [Marvellous]> 
```

SQL constraints are used to specify rules for the data in a table. There are multiple constraints as

22. PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only one primary key, which may consist of single or multiple fields.

23. UNIQUE Constraint

The UNIQUE constraint ensures that all values in a column are different. Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint. However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

24. NOT NULL Constraint

By default, a column can hold NULL values.

The NOT NULL constraint enforces a column to NOT accept NULL values.

This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

25. DEFAULT Constraint

The DEFAULT constraint is used to provide a default value for a column.

The default value will be added to all new records IF no other value is specified.

26. CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a single column it allows only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

CREATE TABLE Teacher

```
(  
  EID int PRIMARY KEY,  
  Name varchar(255) NOT NULL UNIQUE,  
  Age int,  
  Address varchar(255) DEFAULT 'Pune',  
  CHECK (Age>=25)  
);
```

```
MariaDB [Marvellous] > create table teacher(EID int primary key, Name varchar(255) not null unique, age int, address varchar(255) default 'Pune', check(Age>25));  
Query OK, 0 rows affected (0.01 sec)

MariaDB [Marvellous] > describe teacher;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type   | Null | Key  | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| EID   | int(11) | NO   | PRI  | NULL    |       |  
| Name  | varchar(255)| NO  | UNI  | NULL    |       |  
| age   | int(11)  | YES  |      | NULL    |       |  
| address | varchar(255) | YES  |      | Pune   |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```