

MY SQL

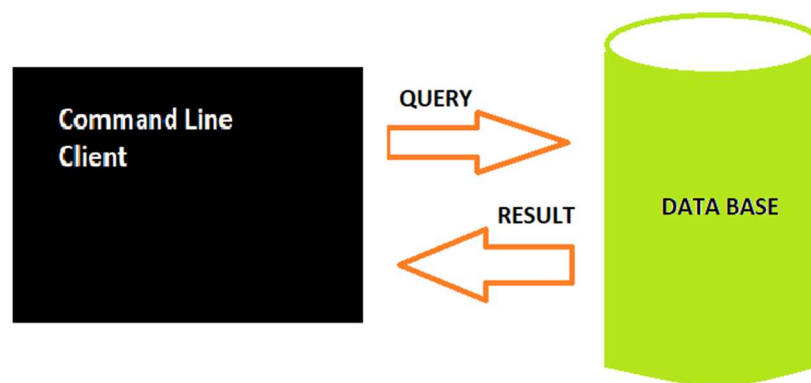
DATABASE- It is a software that stores data in organized manner, so it becomes easy to manage data.

Example- mongoDB, PostgreSQL, ORACLE, MySQL.

→SQL = Structured Query Language.

→MySQL is handled by Oracle.

→It stores data in the form of tables (relational manner).



Queries:

1.To show Databases:

show databases;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql        |
| performance_schema |
| sys          |
+-----+
```

2. To create database:

create database [db_name];

```
mysql> create database learn;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| learn |
| mysql |
| performance_schema |
| sys |
+-----+
```

3. To use database (for creating tables we need to go to database):

use [db_name];

```
mysql> use learn;
Database changed
```

4. To delete a database:

drop database [db_name];

```
mysql> drop database learn;
Query OK, 0 rows affected (0.05 sec)
```

5. To create tables: (first go to database):

create table [table_name] (col1 , col2 , col 3);

```
mysql> create table user (id int(11) primary key , name varchar(100) not null , city varchar(50) );
Query OK, 0 rows affected, 1 warning (0.09 sec)
```

→ need to specify type of each column along with its size:

example- id is of int type of size 11. So : id int(11),

-primary key → making id unique.

-not null → making name mandatory (need to be filled).

6. To view tables:

show tables;

```
mysql> show tables;
+-----+
| Tables_in_learn |
+-----+
| user            |
+-----+
```

7. To show columns of a table:

desc [table_name];

```
mysql> desc user;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    |       |
| name  | varchar(100)  | NO   |     | NULL    |       |
| city  | varchar(50)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

8. To delete table:

drop table [table_name];

9. To rename a table:

alter table [old_table_name] rename to [new_name];

```
mysql> alter table user rename to student;
Query OK, 0 rows affected (0.14 sec)
```

10. To delete / truncate data of the table:

truncate table [table_name];

```
mysql> truncate table student;
Query OK, 0 rows affected (0.16 sec)
```

11. Insert data inside the table:

insert into [table_name] (col1 , col2 , col3..) values (... , ... , ...);

```
mysql> insert into student (id,name,city) values(12,"aman","delhi");
Query OK, 1 row affected (0.01 sec)
```

insert into [table_name] values(...);

```
mysql> insert into student values(22,"sushant","kanpur");
Query OK, 1 row affected (0.01 sec)
```

12. To view all the data:

select * from [table_name];

```
mysql> select * from student;
+----+-----+-----+
| id | name   | city   |
+----+-----+-----+
| 12 | aman   | delhi  |
| 22 | sushant | kanpur |
+----+-----+-----+
```

13. To add column into table:

alter table [table_name] add [column_name and type];

```
mysql> alter table student add country varchar(50);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

14. To rename a column:

alter table [table_name] rename column [old_column] to [new_column];

```
mysql> alter table student rename column country to COUNTRY;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

15. Update details inside a column:

update [table_name] set [column_name] = value {where → sepecify};

```
mysql> update student set COUNTRY='India' where id =12;  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from student;  
+-----+-----+-----+-----+  
| id | name   | city   | COUNTRY |  
+-----+-----+-----+-----+  
| 12 | aman   | delhi  | India   |  
| 22 | sushant | kanpur | NULL    |  
+-----+-----+-----+-----+
```

16. delete any specific row:

delete from [table_name] where...;

```
mysql> delete from student where id=12;  
Query OK, 1 row affected (0.01 sec)
```

NOTE: * means all columns

17. Select specific data (Use where clause) :

select * from [table_name] where;

```
mysql> select * from student where city = 'kanpur';  
+-----+-----+-----+-----+  
| id | name   | city   | COUNTRY |  
+-----+-----+-----+-----+  
| 22 | sushant | kanpur | NULL    |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

18. To select specific column:

select [column_name] from [table_name];

```
mysql> select COUNTRY from student;
+-----+
| COUNTRY |
+-----+
| NULL    |
+-----+
```

19. To change name of column for viewing (alias name):

select [column_name] as "new_name" from [table_name];

```
mysql> select name as "UserName" from student;
+-----+
| UserName |
+-----+
| sushant  |
+-----+
1 row in set (0.00 sec)
```

20. To select two columns :

use **and** operator :

```
mysql> select * from student where id = 22 and name = "sushant";
+----+-----+-----+-----+
| id | name   | city   | COUNTRY |
+----+-----+-----+-----+
| 22 | sushant | kanpur | NULL    |
+----+-----+-----+-----+
1 row in set (0.11 sec)
```

Similarly, we can use **OR** clause.

21. To select a range of column :

select * from [table_name] where [condition];

```

+-----+-----+-----+-----+
| id      | name   | city   | country |
+-----+-----+-----+-----+
|      23 | ankit  | delhi  | ind     |
|      24 | sanket | lucknow | india   |
|     234 | ishant | lucknow | india   |
|     246 | sumit  | lucknow | india   |
|    2334 | aman   | kanpur  | india   |
| 24234236 | ramsigh | lucknow | india   |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

```
mysql> select * from student where id >=24 and id<=2334;
```

```

+-----+-----+-----+-----+
| id  | name   | city   | country |
+-----+-----+-----+-----+
|   24 | sanket | lucknow | india   |
|  234 | ishant | lucknow | india   |
|  246 | sumit  | lucknow | india   |
| 2334 | aman   | kanpur  | india   |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

OR,

Use between clause:

```
select * from [table_name] where [column_name] between ... and ...;
```

```

mysql> select * from student where id between 24 and 2334;
+-----+-----+-----+-----+
| id  | name   | city   | country |
+-----+-----+-----+-----+
|   24 | sanket | lucknow | india   |
|  234 | ishant | lucknow | india   |
|  246 | sumit  | lucknow | india   |
| 2334 | aman   | kanpur  | india   |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

22. To select random columns:

Way 1: use multiple **or** :

```
mysql> select * from student where id=23 or id=234 or id=246;
```

id	name	city	country
23	ankit	delhi	ind
234	ishant	lucknow	india
246	sumit	lucknow	india

Way 2: use **in** operator:

select * from [table_name] where [column_name] in(...,...,...);

```
mysql> select * from student where id in(23,234,246);
```

id	name	city	country
23	ankit	delhi	ind
234	ishant	lucknow	india
246	sumit	lucknow	india

23. To select number of values from top :

Use : **limit**

select * from [table_name] limit [number of choices];

```
mysql> select * from student limit 4;
```

id	name	city	country
23	ankit	delhi	ind
24	sanket	lucknow	india
234	ishant	lucknow	india
246	sumit	lucknow	india

24. To select number of values after some offsets:

Use : **offset**

select * from [table_name] limit [no of choices] offset [no of choices];

```
mysql> select * from student limit 2 offset 2;
+-----+-----+-----+-----+
| id | name | city | country |
+-----+-----+-----+-----+
| 234 | ishant | lucknow | india |
| 246 | sumit | lucknow | india |
+-----+-----+-----+-----+
```

25. To view column in ascending order:

select * from [table_name] order by [column_name];

```
mysql> select * from student;
+-----+-----+-----+-----+
| id | name | city | country |
+-----+-----+-----+-----+
| 23 | ankit | delhi | ind |
| 24 | sanket | lucknow | india |
| 234 | ishant | lucknow | india |
| 246 | sumit | lucknow | india |
| 2334 | aman | kanpur | india |
| 24234236 | ramsigh | lucknow | india |
+-----+-----+-----+-----+
```

26. To view column in ascending order:

select * from [table_name] order by [column_name] desc;

```
mysql> select * from student order by id desc;
```

id	name	city	country
24234236	ramsigh	lucknow	india
2334	aman	kanpur	india
246	sumit	lucknow	india
234	ishant	lucknow	india
24	sanket	lucknow	india
23	ankit	delhi	ind

% means multiple characters.

Example- A % → Aa or Addkdkm or A25456 etc.

_ means single character.

Like operator :- To select similar column details.

select * from [table_name] where [column_name] like 'condition';

```
mysql> select * from student where name like 'a%';
```

id	name	city	country
23	ankit	delhi	ind
2334	aman	kanpur	india

27. To find sum of a particular column:

select SUM(column_name) from [table_name];

```
mysql> select SUM(id) from student;
```

SUM(id)
24237097

28. To find average of columns:

Use : **select AVG(column_name)...**

29. To count number of columns:

Use : **select COUNT(column_name)...**

30. To select column of minimum value:

Use : **select MIN(column_name)...**

31. To select column of minimum value:

Use : **select MIN(column_name)...**

32. Nested Query:

```
mysql> select MIN(id) from student;
+-----+
| MIN(id) |
+-----+
|      23 |
+-----+
1 row in set (0.00 sec)

mysql> select name from student where id = (select MIN(id) from student) ;
+-----+
| name  |
+-----+
| ankit |
+-----+
```

Foreign key and joins

If we have two tables and we need to connect two tables:

Use **foreign key;**

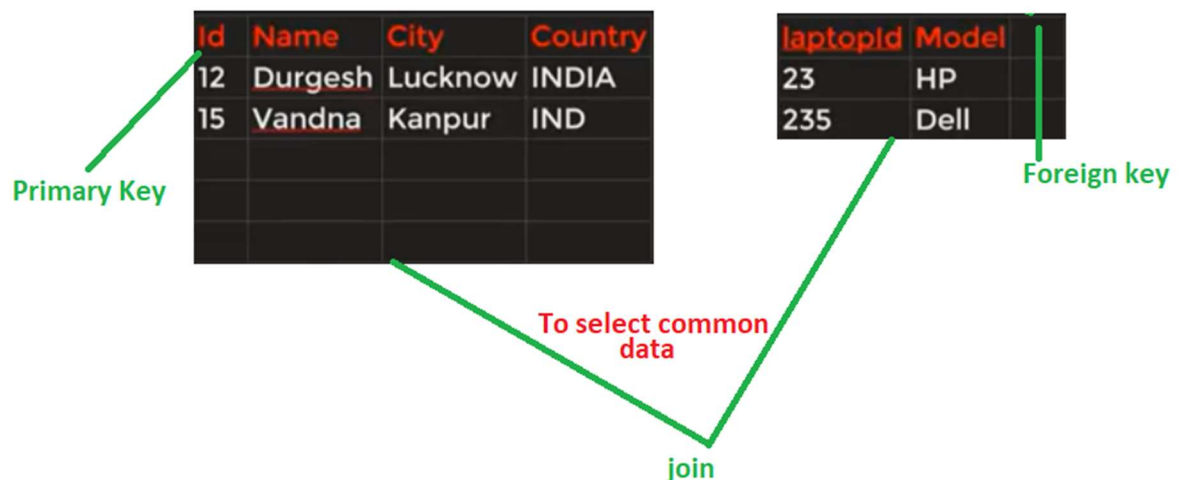
We can make any column as foreign key of the other table:

Foreign key and joins



NOTE: If foreign key is a primary key then we cannot enter any data having same foreign key.

Join can be of many types.



Syntax:

foreign key (key_of_new_table) references [table_name] (key_of_old_table);

```
mysql> create table laptops (Lid int primary key , Lmodel varchar(50) ,  
studentID int , foreign key (studentID) references student(id));
```

Explanation:

```
mysql> create table laptops (Lid int primary key , Lmodel varchar(50) ,  
studentID int , foreign key (studentID) references student(id));
```

This is the foreign key . So now laptop table has 3 columns:

1. Lid
2. Lmodel
3. studentID which is same as id of student-table

Joining Tables:

```
mysql> select student.name , student.city , laptops.Lmodel from  
student , laptops where student.id = laptops.studentID;
```

Explanation:

