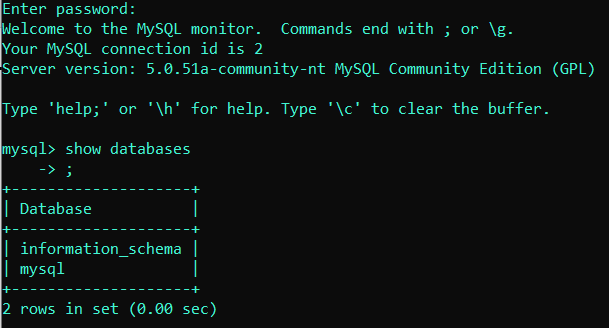
**DBMS LAB – 02**

**Task – 1 :**

**Aim :** Create a simple student database with a simple student info table.

**Procedure :**

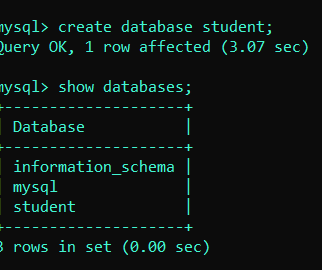
**Command :** show databases;



**Description :**

It will show the list of all Databases in the mysql software.

**Command :** create database student;



**Description :**

It will create a database with the name given by user and you can check your whether your database is created or not by running the previous command..

**Command :** use student;

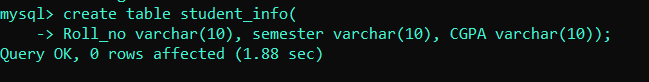


**Description :**

Student is database name, to insert relations/tables you must enter into a database which you have created by running above command.

**Command :** create table student\_info(

Variable1 varchar(max\_limit));

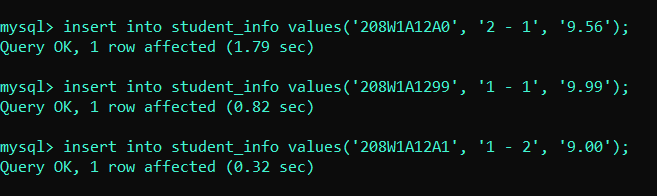


**Description :**

By running above command it will create a columns with column headings also..

**Command :**

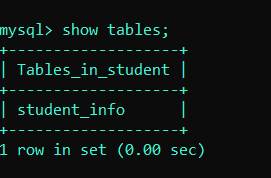
Insert into student\_info values(‘ … ‘, ‘….’, ‘……..’);



**Description :**

By running above command it will create a row and you can insert a user details as values into row/tuple also..

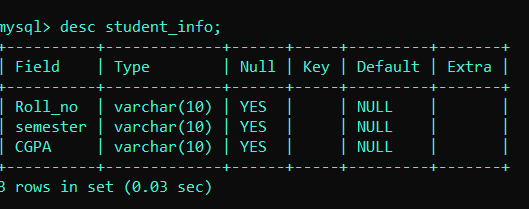
**Command :** show tables;



**Description :**

By running above command it will show the all list of tables which are created in this exsisting database..

**Command :** desc table\_name;



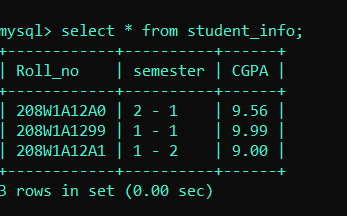
**Description :**

By running above command it will show the no.of fields and data type of the field and default values assigned with them of a table..

**Command :**  select \* from student\_info

**Description :**

By running above command it will show the student\_info table with collection of tuples and attributes of user data inserted…



**Result :** Successfully completed the Aim.

**Task – 02 :**

**Aim :** Different types of datatypes in mysql software…

Table :

|  |  |
| --- | --- |
| **Data type** | **Description** |
| 1. Varchar(size) | It allows any character from the keyboard with gaps allow also…. |
| 1. Binary(size) | It is equal to the varchar( ), but it stores byte strings…. |
| 1. Varbinary(size) | It is same as varchar() but, it stores binary byte strings. |
| 1. Text(size) | Holds a string with max-limit of 65,535 bytes. |
| 1. Tinytext(size) | Holds a string with max-limit of characters from 0 to 255. |
| 1. Blob(size) | Binary large objects holds upto 65,535 bytes of data. |
| 1. Longtext(size) | Holds a string of max-limit 4,294,967,295 characters. |
| 1. Bit(size) | A bit value type, the number of bits per value specified and max-limit holds from 1 to 64. |
| 1. Bool/Boolean | 0 is considered as false, non-zero values as True. |
| 1. Int/integer | A medium integer of signed range from -2147483648 to 2147483648. Unsigned range from 0 to 4294967295. |
| 1. Float(p) | A floating point number.  P is 0 to 24 i.e; Flaot value.  P is 25 to 53 i.e; double value |
| 1. Date | Format : yyyy – mm – dd , range :  1000-01-01 to 9999-12-31. |
| 1. Datetime(fsp) | Format : yyyy—mm—dd  Hh:mm:ss |
| 1. Time(fsp) | Format: hh : mm : ss  Range : -838:59:59 to 838:59:59. |
| 1. year | A 4 digit format is allowed from 1901 to 2155 and 0000 also. |

**Result :** Successfully completed the Aim.

**DBMS LAB – 03**

**TASK – 01 :**

**Aim :**  create a table and apply Data Defination Language (DDL)

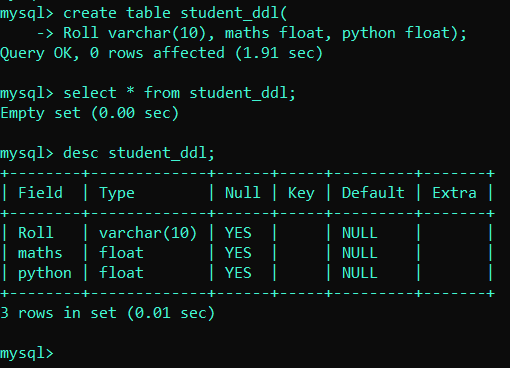
Commands

**Command :** create table student\_ddl(

Roll varchar(10), maths float, python float );

**Description :**

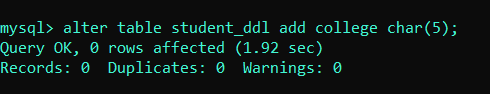
It will create a table with required data types of columns.



**Command :** alter table table\_name add newcolumn\_name datatype;

**Description :**

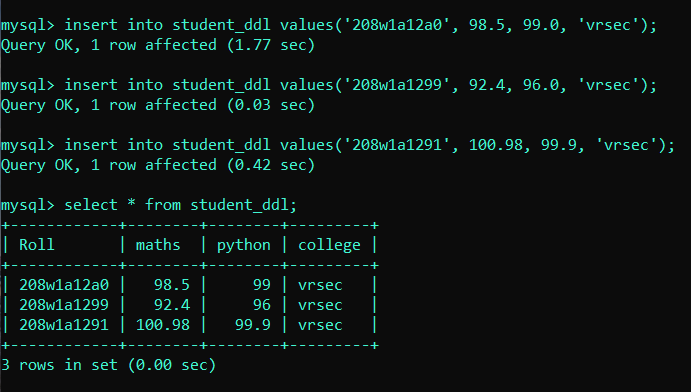
It will create a new column and append to the table at last



**Command :** insert into table\_name values(……., ……….., …….);

**Description:**

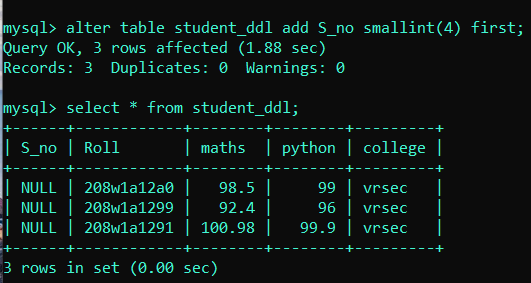
It will create a new record and append to the table with user values..



**Command :** alter table table-name add newcolumn-name datatype first;

**Description :**

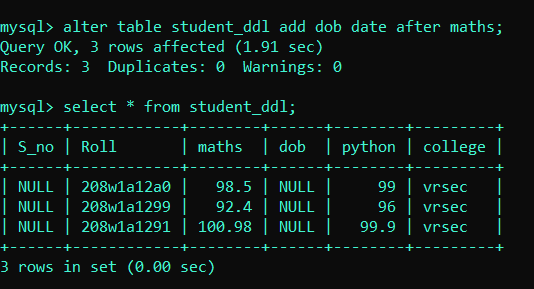
It will create a new column and append to the table as a first column



**Command :** alter table table-name add newcolumn-name datatype after exsistedcolumn-name;

**Description :**

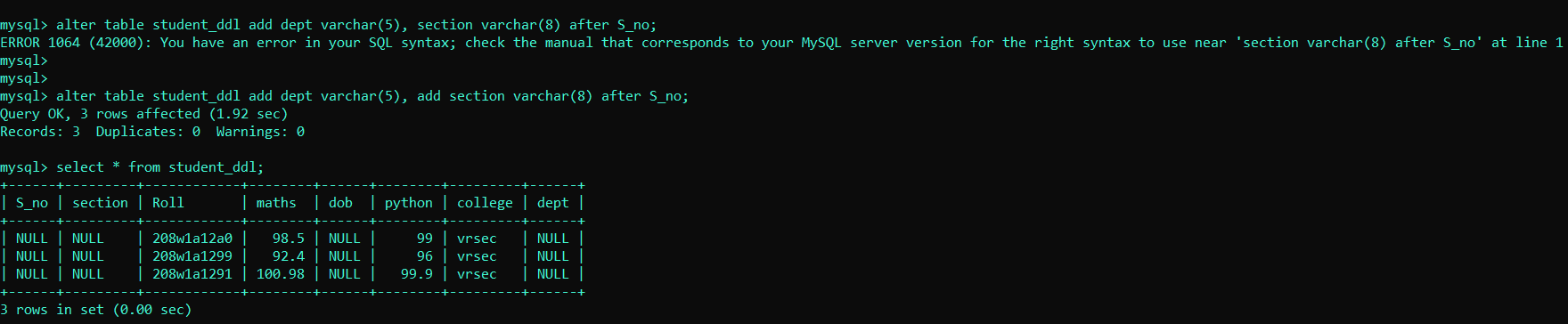
It will create a colum and appends to the table at a specified after a column name which is exsisted one’s.



**Command :** alter table table-name add column1 datatype, add column2 datatype after exsistedcolumn-name;

**Description :**

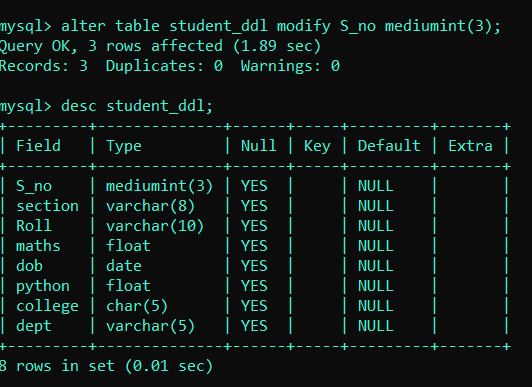
It will create a 2 columns at a time and append to the table at after a particular column;



**Command :** alter table table-name modify exsistedcolumn-name datatype;

**Description :**

It will modifies the datatype of the exsisted column in the table.

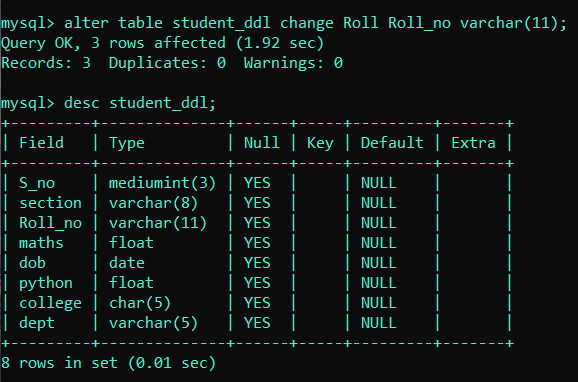


**Command :** alter table table-name **change column** column1

Column2 datatype;

**Description :**

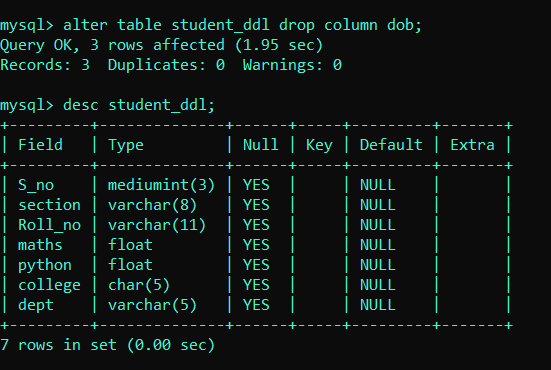
Column1 should be exsisted one’s and it change the exsisted column names of the table.



**Command :** alter table table-name **drop column** column-name;

**Description :**

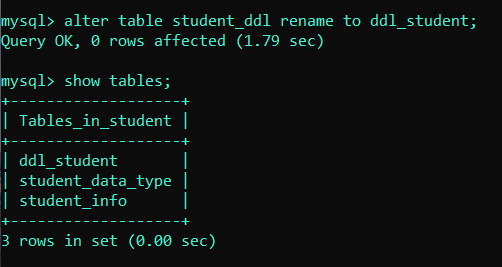
It will deletes the column from the exsisted table.



**Command :** alter table table-name rename to newtable-name;

**Description :**

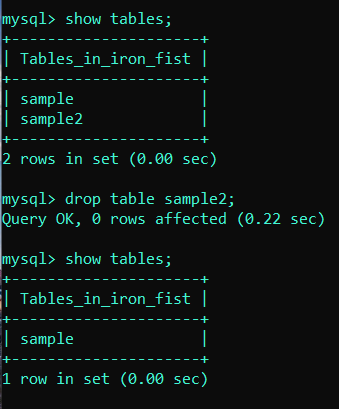
It will renames the exsisted table name to the new table name.



**Command :** drop table table-name;

**Description :**

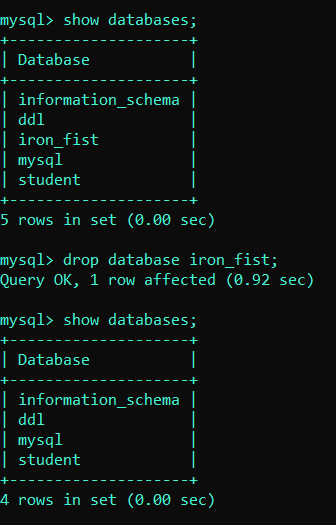
It will deletes the table and its structure from the database.



**Command :** drop database DB\_name;

**Description :**

It will deletes the database permantely from the mysql software.



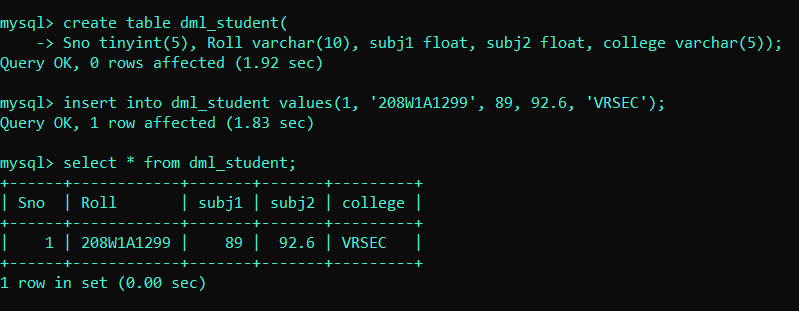
**TASK – 02 :**

**Aim :** create a table and apply all Data Moddelling language (DML) Commands..

**Command :** insert into table-name values(………………);

**Description :**

It will insert a record in to table just like an Row insertion in a table.

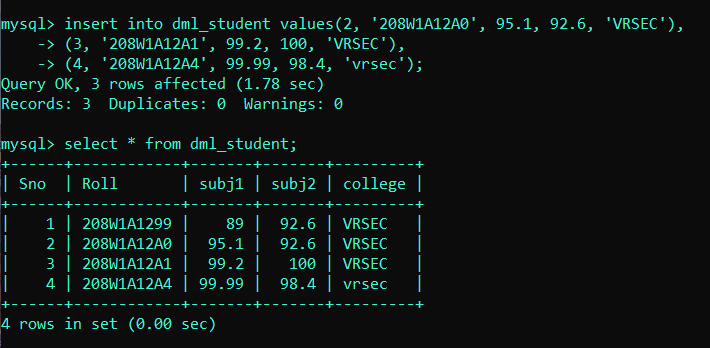


**Command :** insert into table-name values(…………..),

(………..), (…………..), (………………);

**Description :**

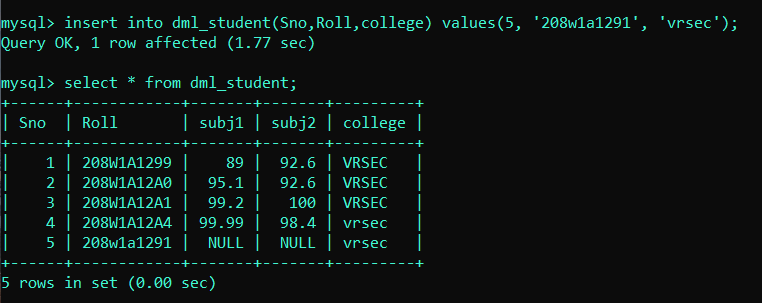
It will insert a multiple records into at an same time into an table.



**Command :** insert into table-name (column-name,column-name) values(………..);

**Description :**

It will insert the values given into an particular column names only and remaining columns filled with a NULL value.

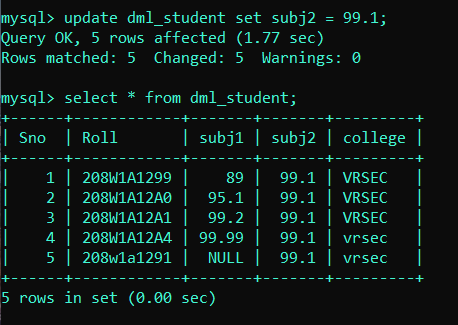


**Command :** update table-name set column-name = value;

**Description :**

Column name should be exsisted ones.

It will update only data present in that particular exsisted column only and it will effects the remaining records in that column also.

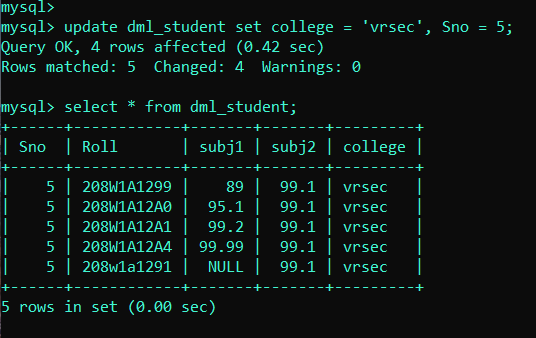


**Command :** update table-name set column1 = value, column2 = value, column3 = value;

**Description :**

It will update the data in multiple records at a same time.

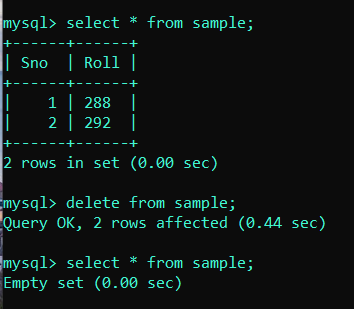
Column name should be exsisted ones, and once we update it will effect all records at a that particular columns which was given in the command.



**Command :** delete from table-name;

**Description :**

It will deletes all data from the exsisted table one’s.



**Result :** successfully excuted the commands of the two tasks

Given in the Aim.

**DBMS LAB – 04**

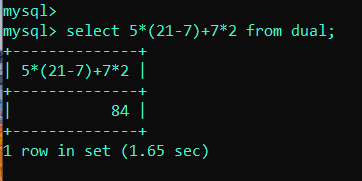
**Task – 01 :**

**Aim :** Evaluate expressions, Logical , Like , sorting techniques on a table.

**Command :** select expression from dual;

**Description :**

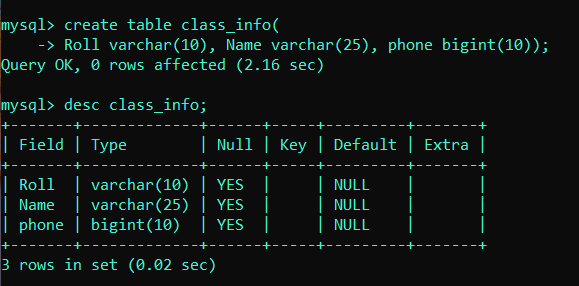
It will analyze the expression and gives the answer in tabular form.



**Command :** create table class\_info( variable datatype);

**Description :**

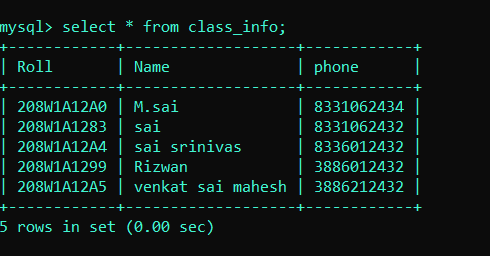
It will create a table with given varibles as column names in a database.



**Command :** select \* from class\_info

**Description :**

It will show all records and attributes in a table present in a database

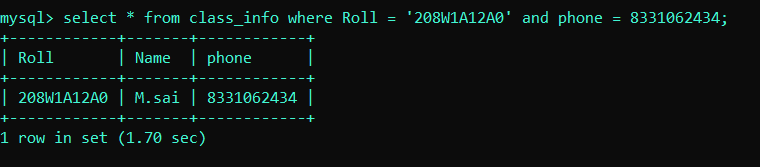


**Command :**select \* from class\_info where roll = ‘208W1A12A0’ and phone =

8331062343;

**Description :**

It will show the records only if given roll number and phone number are satisfied in a table else it throws an error

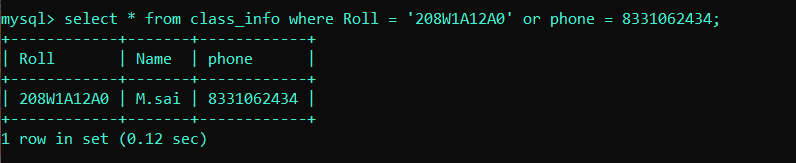


**Command :** select \* from class\_info where roll = ‘208W1A12A0’ or phone =

8331062343;

**Description :**

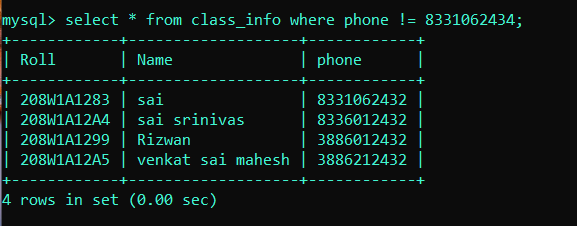
It will show the records in a table if either roll number or phone number is found in a table from the database.



**Command :** select \* from class\_info where phone != 8331062343;

**Description :**

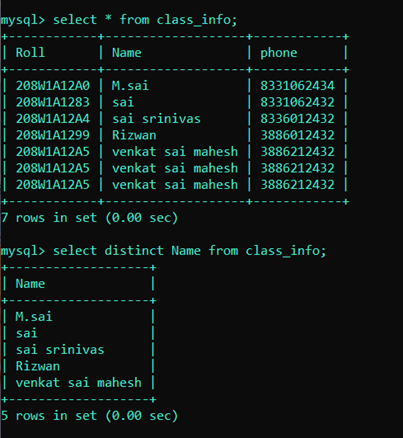
It will show the records in a table only if phone number should not match with given number…



**Command :** select distinct Name from class\_info;

**Description :**

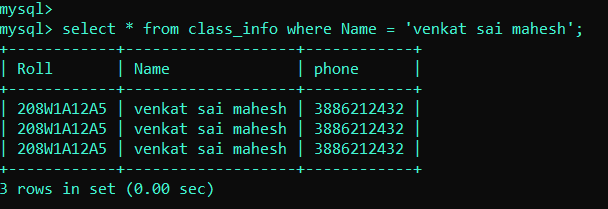
It will show the records by leaving the same names and shows only unique names in a table.



**Command :** select \* from class\_info where Name = value;

**Description :**

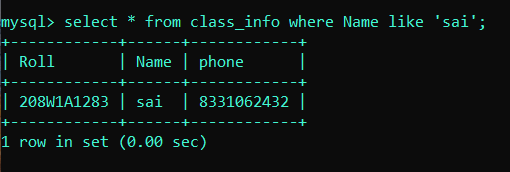
It will display the records if the name present in the table .



**Command :** select \* from class\_info where Name like ‘sai’;

**Description :**

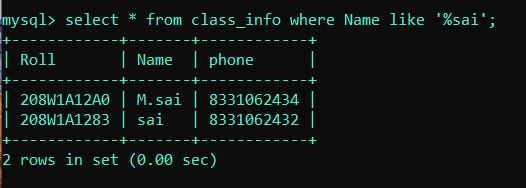
It will check the all names whether the name starts with ‘sai’ or not if true it will display the entire record



**Command :** select \* from class\_info where Name like ‘%sai’;

**Description :**

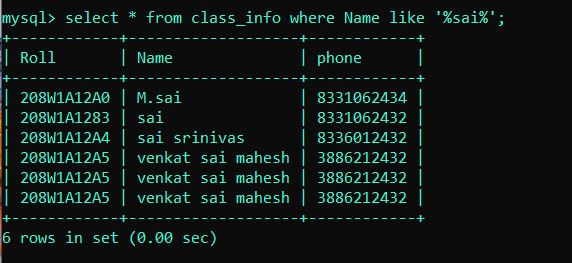
It will check whether the name matches with ‘sai’ in before % means any name it can be their but after it should be filled with ‘sai’ then only it will show the record.



**Command :** select \* from class\_info where Name like ‘%sai%’;

**Description :**

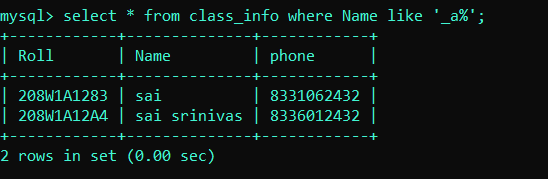
% means it can be any name followed by sai and followed by any name then only it will show or display the records.



**Command :** select \* from class\_info where Name like ‘\_a%’;

**Description :**

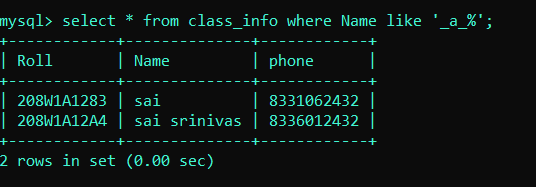
\_ : means it will match with any keyboard input for only single character and followed by second character should be ‘a’ and followed any name .



**Command :** select \* from class\_info where Name like ‘\_a\_%’;

**Description :**

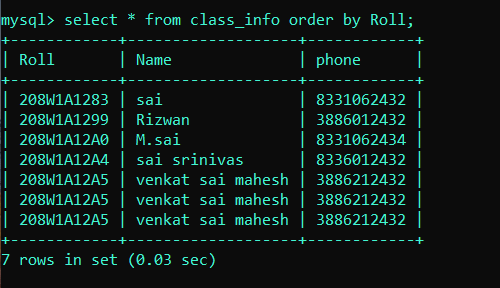
First character will be any character from keyboard second character should be ‘a’ and any single character followed by any name in a table in a database.



**Command :** select \* from class\_info order by Roll;

**Description :**

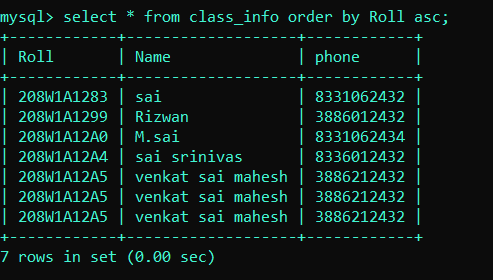
It will sort the table on basis of roll numbers of the students.



**Command :** select \* from class\_info order by Roll asc;

**Description :**

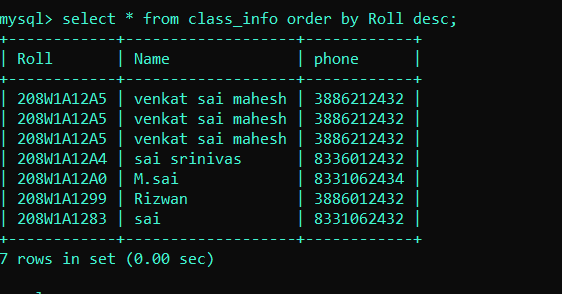
Asc : means ascending order it will sort the table in ascendeing order on basis of roll numbers.



**Command :** select \* from class\_info order by Roll desc;

**Description :**

Desc : means descending order it will sort the table in descending order on basis of roll numbers of the students in a table.



**Result**  : successfully completed the tasks.