

SESSION 2022-23

DBMS

(DataBase Management System)

Lab File

**COURSE:- BCA**

**ROLL NO :- 41221139**

**SUBMITTED BY :- SUBMITTED TO:-**

**Sachin Rajbhar Ms. Bushra Jamal**

Index

|  |  |  |  |
| --- | --- | --- | --- |
| S. No | Name | Date | Sign |
| 1 | Installation of My SQL WorkBench | 12/10/22 |  |
| 2 | Basic Commands to start with MySql | 12/10/22 |  |
| 3 | Write SQL command to create & describe table | 12/10/22 |  |
| 4 | Write SQL command for implementing alter and drop | 12/10/22 |  |
| 5 | Write SQL command to insert values into tables | 18/10/22 |  |
| 6 | Create a database with tables Employee and Department & apply following constraints-  1) Primary Key  2) Foreign Key  3) Not NULL | 2/11/22 |  |
| 7 | Use of Update Statements ( by create table copy\_emp as select \* from employee)   1. update copy\_emp set salary=salary+1000; 2. update copy\_emp set dept\_name=”electrical” where emp\_name=”james”; 3. update copy\_emp set phone=123456, dept\_id=20 where emp\_id=10; 4. select \* from copy\_emp; | 2/11/22 |  |
| 8 | Difference between truncate & drop   1. truncate table copy\_emp; 2. drop table copy\_emp; | 2/11/22 |  |
| 9 | Illustrate use of Primary key & Foreign key | 09/11/22 |  |
| 10 | Use of aggregate functions (min,max,sum,avg,count) | 09/11/22 |  |
| 11 | Delete record of 1 employee according to his emp\_id | 09/11/22 |  |
| 12 | Delete data from base table (that is reffered by foreign key) | 09/11/22 |  |
| 13 | Illustrate use of order by | 09/11/22 |  |
| 14 | Draw an ER Diagram in ERDplus.com | 09/11/22 |  |
| 15 | Use of AND, OR, NOT operator | 16/11/22 |  |
| 16 | Use of NULL (IS NULL, IS NOT NULL) | 16/11/22 |  |
| 17 | Use of TOP/LIMIT clause | 16/11/22 |  |
| 18 | Use of LIKE operators with wildcards(\_,%) | 16/11/22 |  |
| 19 | Use of IN, NOT IN operator | 16/11/22 |  |
| 20 | 1. Demonstrate use of UNION 2. Apply joins: inner,left,right,full on the given tables 3. Perform all operations of emp and dept table 4. Database Schema for a customer-sale scenario of a general store | 30/11/22 |  |
| 21 | 1)Demonstrate the application of AUTO\_INCREMENT in MySql  2)Execute following functions for mySql | 07/12/22 |  |
| 22 | To create and update views  i) from single table  ii) from multiple tables | 14/12/22 |  |
| 23 | To perform various Date functions in MySQL | 14/12/22 |  |
| 24 | For a given database with following attributes:  EMPLOYEE ID, FIRST NAME, LAST NAME EMAIL PHONE NUMBER, HIRE\_DATE, JOB\_ID, SALARY, MANAGER\_ID, DEPARTMENT\_ID  Execute given queries | 21/12/22 |  |

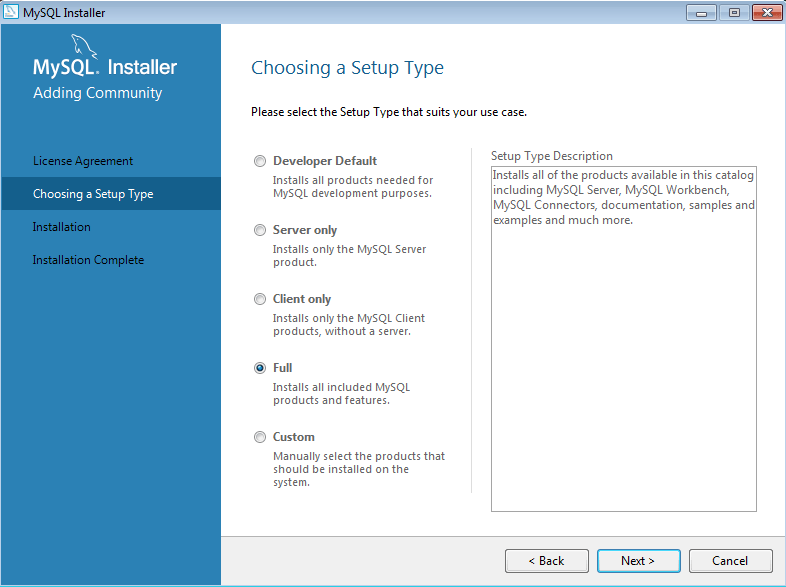
**PROGRAM 1**

**Q:** Installation of MySQL Command mode and Workbench

After downloading the MySQL community edition, double-click the installer file to begin the installation process. It’ll launch the installer window.

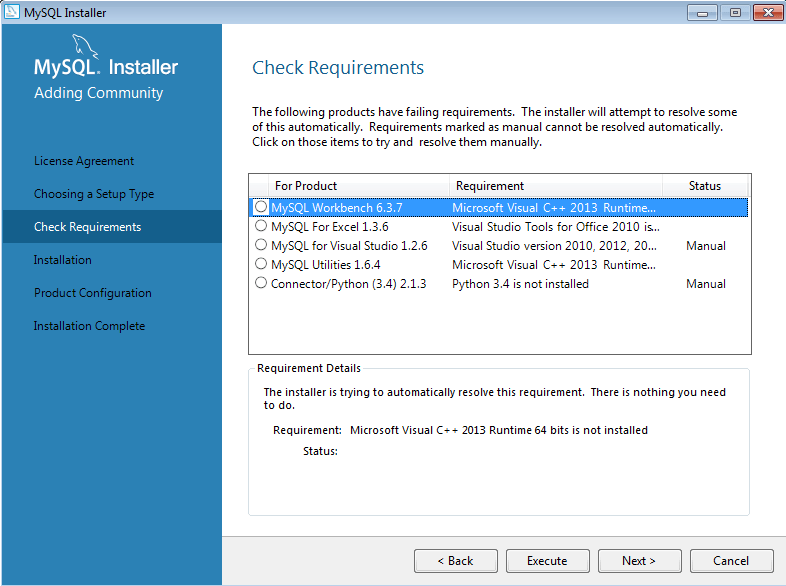
Subsequently, follow the below steps. Take help from the snapshots attached. Make sure to edit or select the option, as shown in the picture.

**Step-1.**



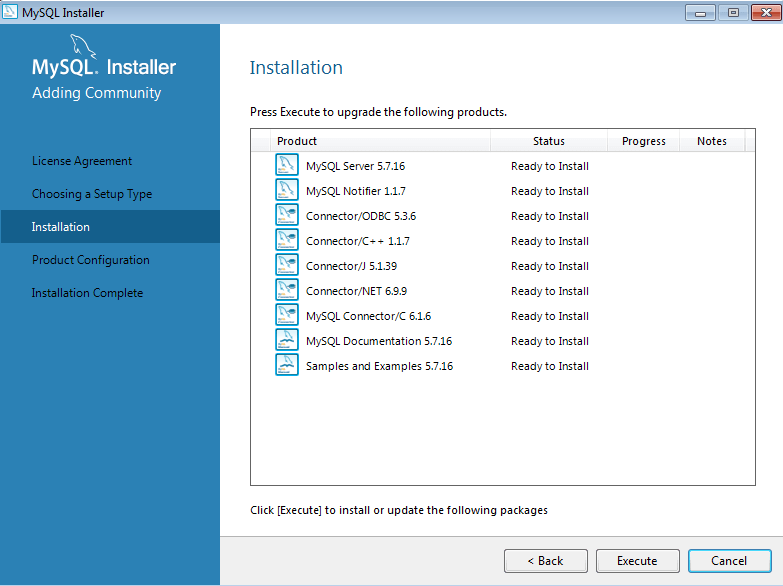
MySQL – Choose a Setup Type.

**Step-2.**



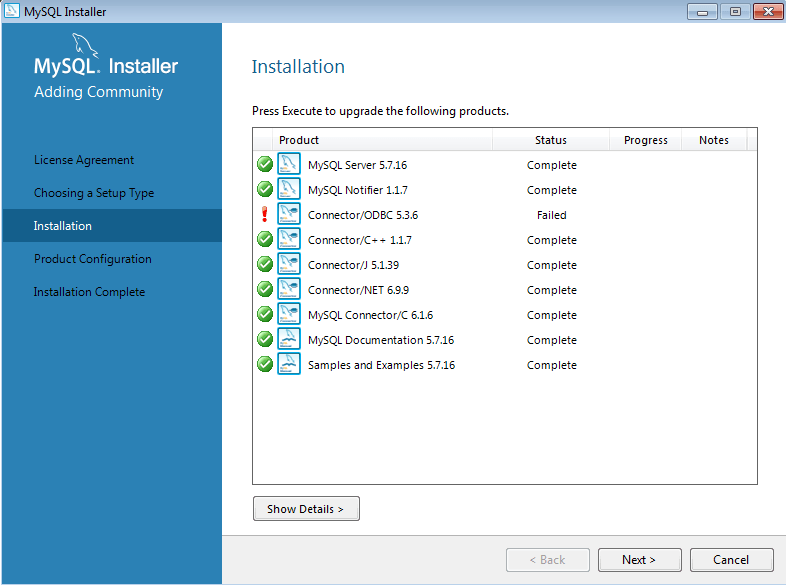
Check Requirements

**Step-3.**



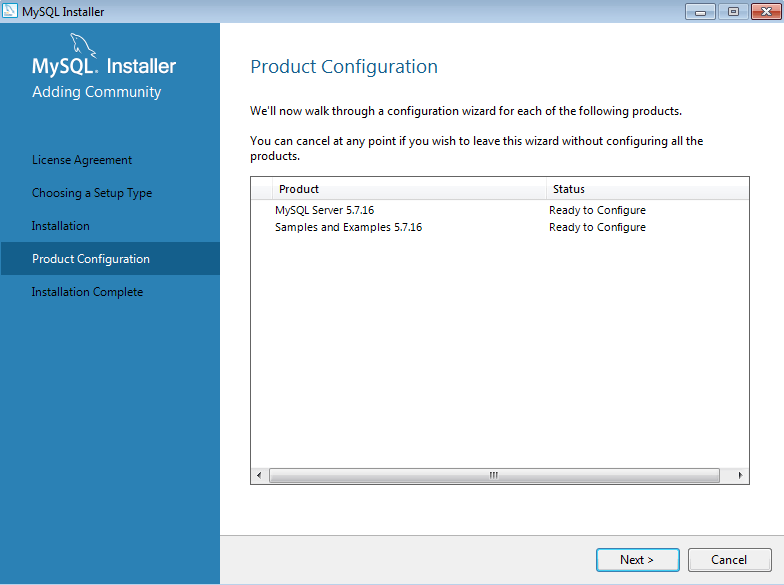
Begin Installation

**Step-4.**



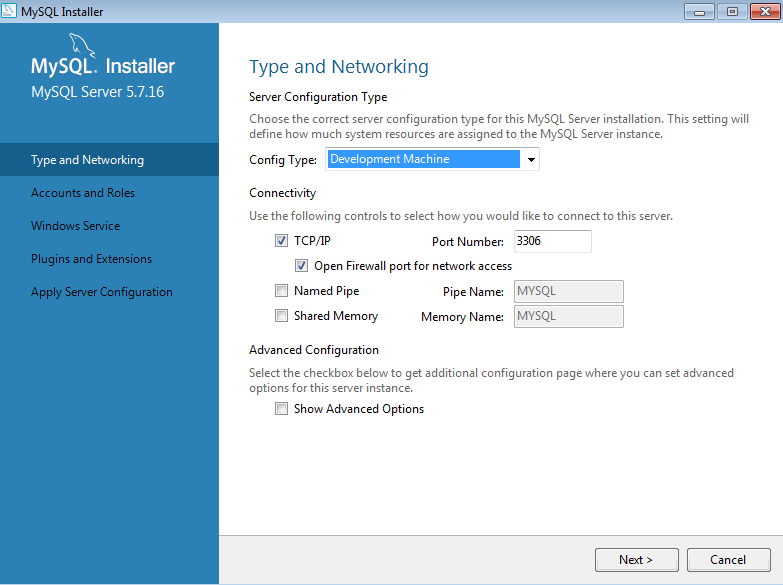
Installation Status.

**Step-5.**



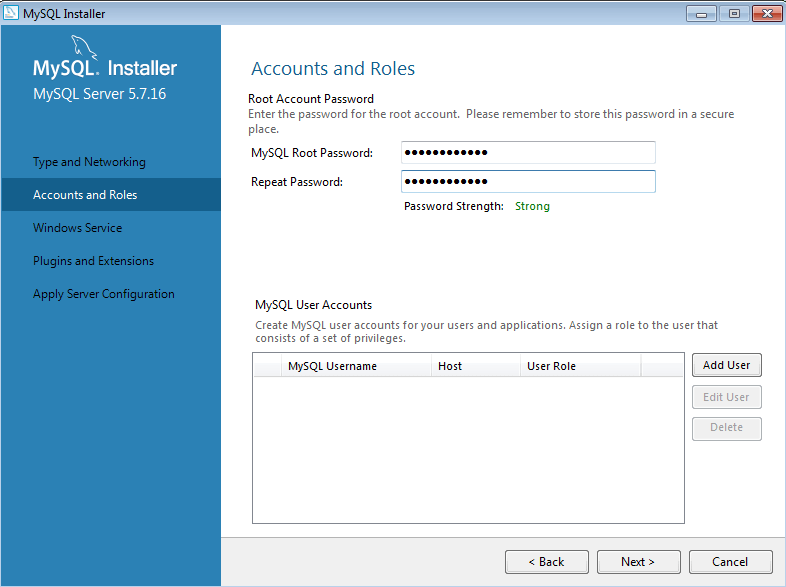
Open Configuration Screen.

**Step-6.**



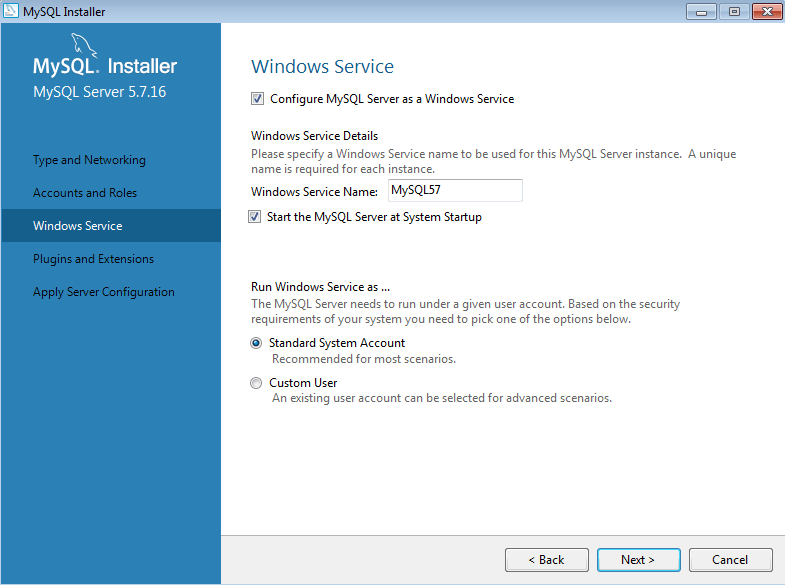
Edit Configuration.

**Step-7.**



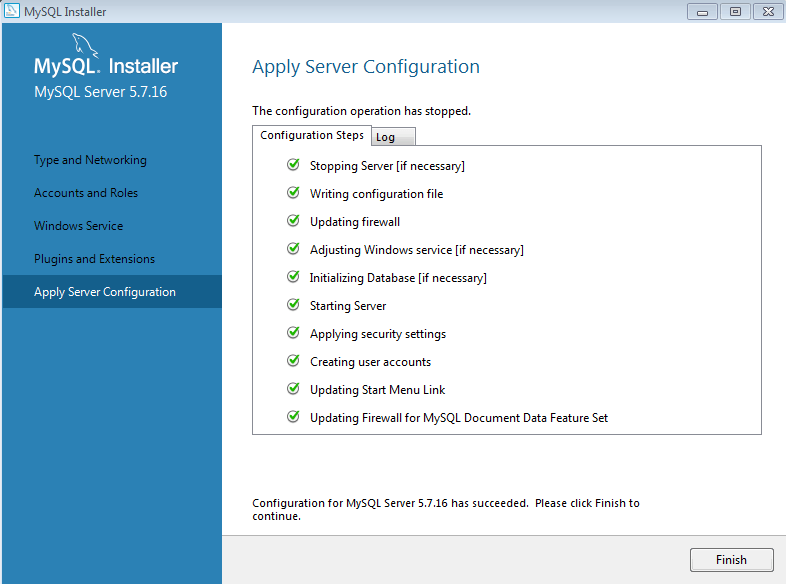
Set Root Password.

**Step-8.**



Set As Windows Service.

**Step-9.**



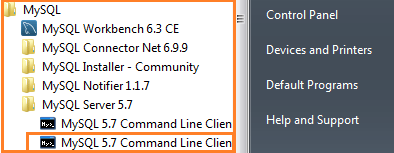
Apply Server Configuration.

**What to do After MySQL installation?**

You’ve completed the MySQL installation on your Windows system. Now, you can perform the following tasks.

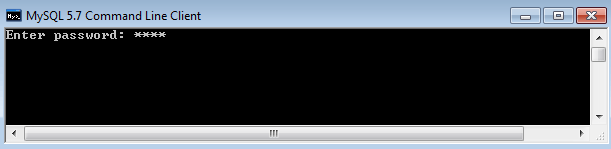
**Using MySQL CLI.**

You can access the MySQL CLI from the <All Programs> section inside the Windows start menu. See the attached screenshot and open the 2nd CLI option as shown on the screen.



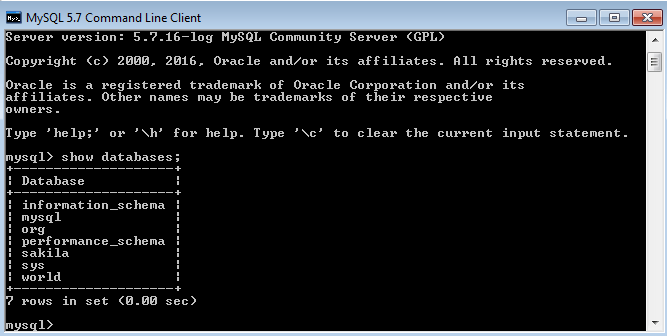
MySQL CLI – Open MySQL Command line

The CLI tool will ask for the root database password.



MySQL CLI – Enter Password**:**

Entering the correct password will open up the MySQL command prompt. Here, you can run simple SQL commands, e.g., show databases.



MySQL CLI – Show Databases

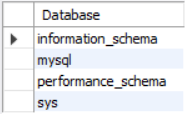
PROGRAM 2

**Q:** Basic Commands to start with MySql

1. Show databases

Command: show databases;

Output:



1. Use mydb

Command: use mysql;

Output:



1. Create database mydb

Command: create database mydb;

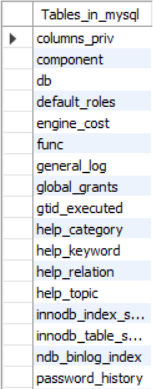
Output:



1. Show tables

Command: show tables;

Output:



1. Select database()

Command: select database();

Output:



PROGRAM 3

**Q:** To create & describe table

1) Command:

create table student

(

Student int;

Fname varchar(20);

Lname varchar(20);

Marks int;

)

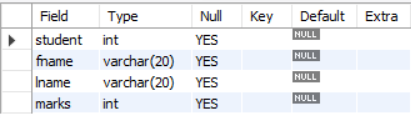
Output:



2) Command:

desc student;

Output:



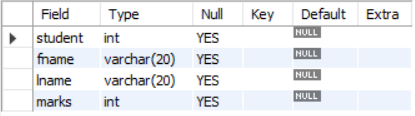
PROGRAM 4

**Q:** Write sql command for implementing alter and drop

1)

Command: alter table student rename to student123;

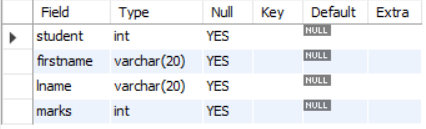
Output:



2)

Command: alter table student123 rename column fname to firstname;

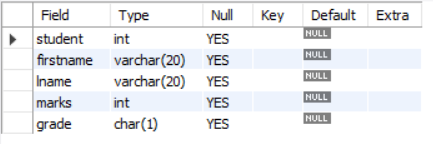
Output:



3)

Command: alter table student123 add grade char(1);

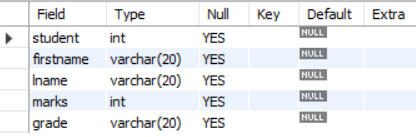
Output:



4)

Command: alter table student123 modify grade varchar(20);

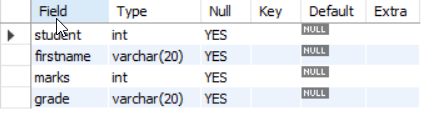
Output:



5)

Command: alter table student123 drop column lname;

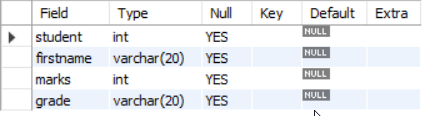
Output:



6)

Command: create table copy\_st as select \* from student123;

Output:



7)

Command: drop table student123;

Output:



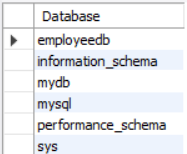
PROGRAM 5

**Q:** Write SQL command to insert values into tables

1)

Command: create database employeedb;

Output:



2)

Command: create table emp (id int ,name varchar(10),salary float ,dept\_name varchar(15), phone int, dept\_id int, date\_of\_joining int);

Output:



3)

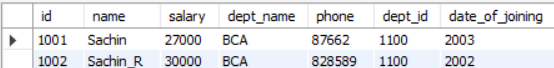
Command: insert into emp values (1001, "Sachin", 27000, "BCA", 87662, 1100 , 2003);

Output:



Command: insert into emp (id,name,salary,dept\_name,phone,dept\_id,date\_of\_joining) values (1002,"Sachin\_R",30000,"BCA",828589,1100,2002);

Output:



4)

Command: select \* from emp where salary is NULL;

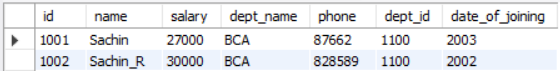
Output:



5)

Command: select \* from emp where salary is NOT NULL;

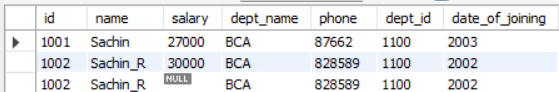
Output:



6)

Command: select distinct \* from emp;

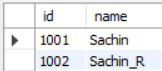
Output:



7)

Command: select distinct id,name from emp;

Output:



8)

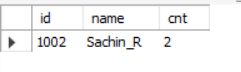
Command: select count(\*) from emp;

Output:

9)

Command: select id, name, count(\*) as cnt from emp group by id, name having count(\*)>1;

Output:



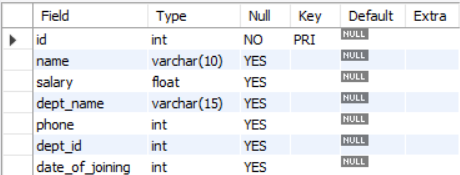
PROGRAM 6

**Q:** Create a database with tables Employee and Department & apply following constraints-

1. Primary Key

Command: alter table emp modify COLUMN id int PRIMARY KEY;

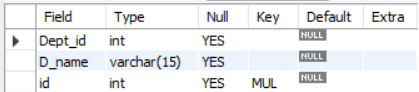
Output:



1. Foreign Key

Command: alter table department ADD foreign key (id) references emp (id);

Output:



1. Not NULL

Command: alter table emp MODIFY salary float NOT NULL;

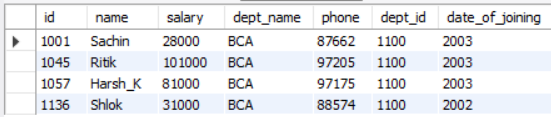
Output:



PROGRAM 7

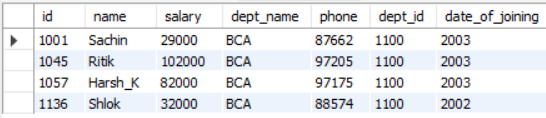
**Q:** Use of Update Statements ( by create table copy\_emp as select \* from employee)





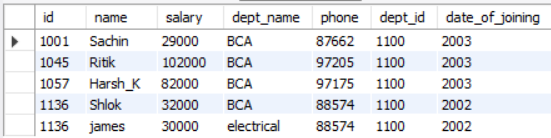
1. Command: update copy\_emp set salary=salary+1000;

Output:



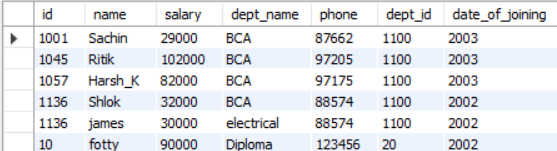
1. Command: copy\_emp set dept\_name=”electrical” where emp\_name=”james”;

Output:



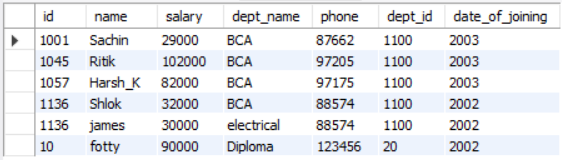
1. Command: update copy\_emp set phone=123456, dept\_id=20 where emp\_id=10;

Output:



1. Command: select \* from copy\_emp;

Output:

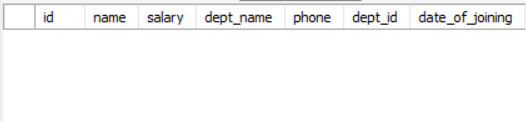


PROGRAM 8

**Q:** Difference between truncate & drop

1. Command: truncate table copy\_emp;

Output:



1. Command: drop table copy\_emp;

Output:



PROGRAM 9

**Q:** Illustrate use of Primary key & Foreign key

**Primary keys**

A *primary key* is a column or a set of columns in a table whose values uniquely identify a row in the table. A relational database is designed to enforce the uniqueness of primary keys by allowing only one row with a given primary key value in a table.

**Foreign keys**

A *foreign key* is a column or a set of columns in a table whose values correspond to the values of the primary key in another table. In order to add a row with a given foreign key value, there must exist a row in the related table with the same primary key value.

PROGRAM 10

**Q:** Use of aggregate functions (min,max,sum,avg,count)

1. min

Command:

select min(salary) as salary\_min from emp ;

Output:



1. max

Command: select max(salary) as salary\_max from emp ;

Output:



1. sum

Command: select sum(salary) as salary\_sum from emp ;

Output:



1. avg

Command: select avg(salary) as salary\_avg from emp ;

Output:



1. count

Command: select count(salary) as salary\_count from emp;

Output:

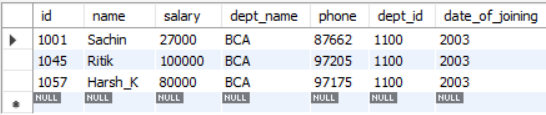


PROGRAM 11

**Q:** Delete record of 1 employee according to his emp\_id

Command: delete from emp where id=1136;

Output:



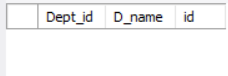
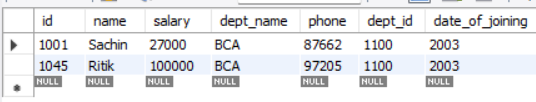
PROGRAM 12

**Q:** Delete data from base table (that is reffered by foreign key)

Command: delete from department where id=1057;

delete from emp where id=1057;

Output:

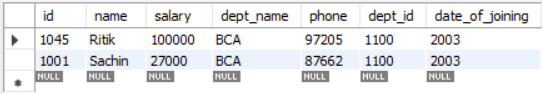
 

PROGRAM 13

**Q:**  Illustrate use of order by

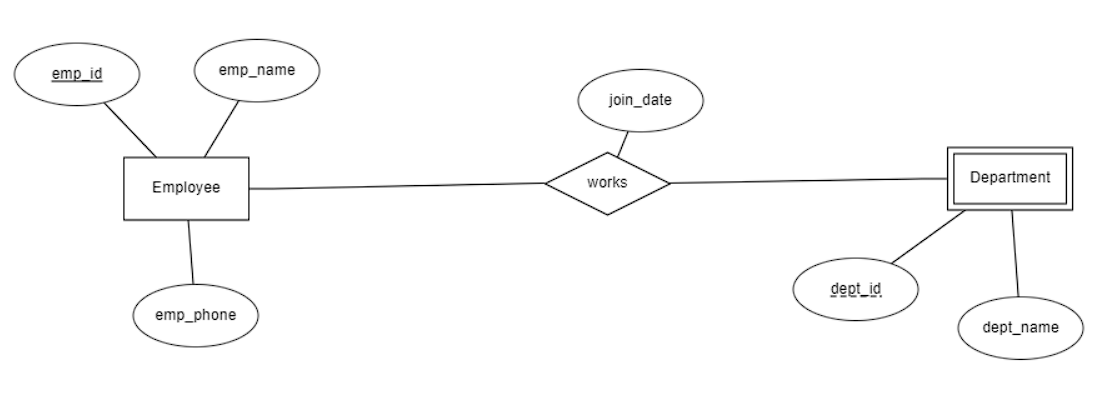
Command: select \* from emp order by name asc;

Output:



PROGRAM 14

**Q:** Draw an ER Diagram in ERDplus.com

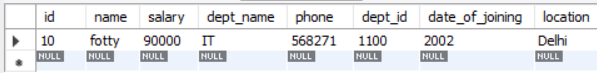


PROGRAM 15

**Q:** Use of AND, OR, NOT operator

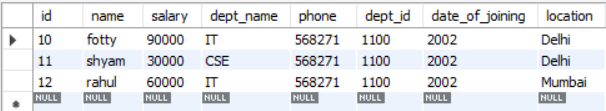
Command: select \* from emp where dept\_name = "IT" AND location = "Delhi";

Output:



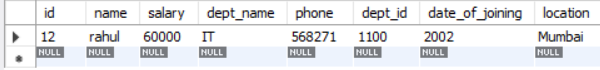
Command: select \* from emp where dept\_name = "IT" OR location = "Delhi";

Output:



Command: select \* from emp where dept\_name = "IT" NOT location = "Delhi";

Output:

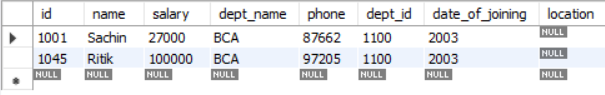


PROGRAM 16

**Q:** Use of NULL (IS NULL, IS NOT NULL)

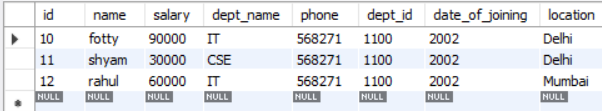
Command: select \* from emp where location IS NULL;

Output:



Command: select \* from emp where location IS NOT NULL;

Output:

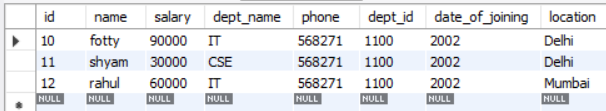


PROGRAM 17

**Q:** Use of TOP/LIMIT clause

Command: select \* from emp LIMIT 3;

Output:

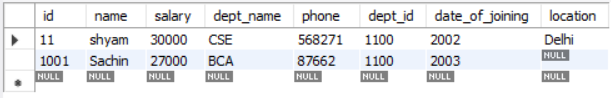


PROGRAM 18

**Q:** Use of LIKE operators with wildcards (\_,%)

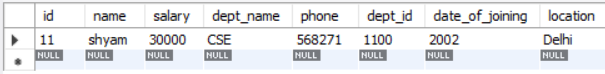
Command: select \* from emp where name LIKE 's%';

Output:



Command: select \* from emp where name LIKE '\_h%';

Output:

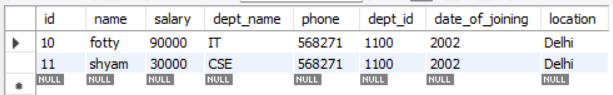


PROGRAM 19

**Q:** Use of IN, NOT IN operator

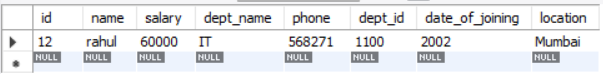
Command: select \* from emp where location IN ("delhi");

Output:



Command: select \* from emp where location NOT IN ("delhi");

Output:



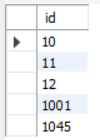
PROGRAM 20

**Q: Perform the following**

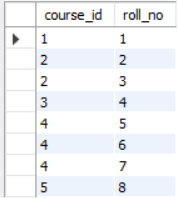
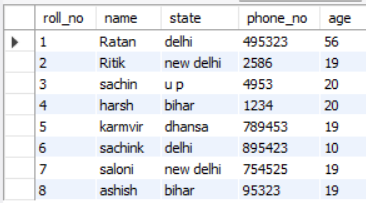
1)Demonstrate use of UNION

Command: SELECT id FROM emp UNION SELECT D\_name FROM Department;

Output:



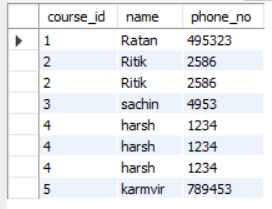
2)Apply joins: inner,left,right,full on the given tables

Command:

SELECT course.course\_id, student.name, student.phone\_no FROM course INNER JOIN student ON course.course\_id = student.roll\_no;

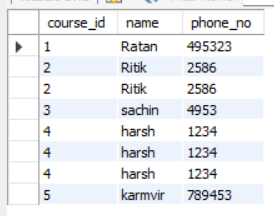
Output(Inner Join):



Command(Left Join):

SELECT course.course\_id, student.name, student.phone\_no FROM course LEFT JOIN student ON course.course\_id = student.roll\_no;

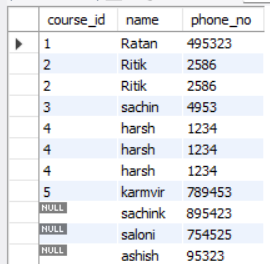
Output:



Command(Right Join):

SELECT course.course\_id, student.name, student.phone\_no FROM course RIGHT JOIN student ON course.course\_id = student.roll\_no;

Output:

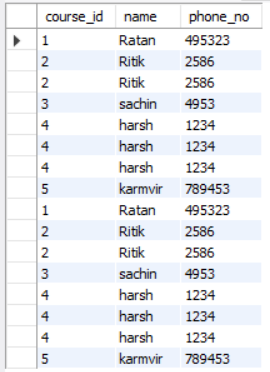


Command(FULL JOIN):

SELECT course.course\_id, student.name, student.phone\_no FROM course LEFT JOIN student ON course.course\_id = student.roll\_no

UNION ALL

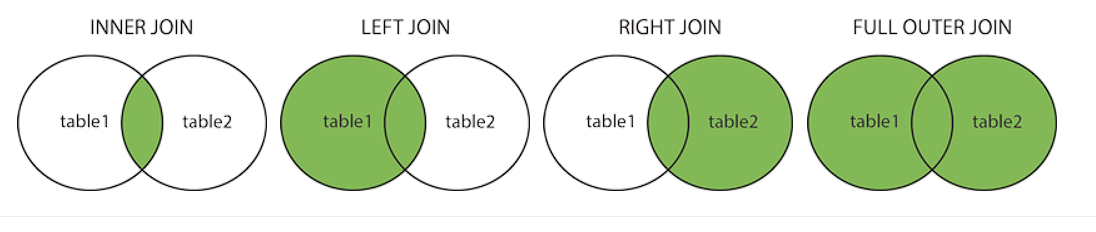
SELECT course.course\_id, student.name, student.phone\_no FROM course LEFT JOIN student ON course.course\_id = student.roll\_no;Output:



3)Perform all operations of emp and dept table

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

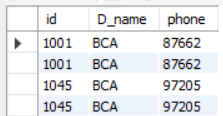
* (INNER) JOIN: Returns records that have matching values in both tables
* LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
* RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
* FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table



Command (Inner join ):

SELECT emp.id, department.D\_name, emp.phone FROM emp INNER JOIN department ON emp.dept\_name=department.D\_name;

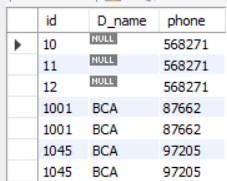
Output:



Command (Left join):

SELECT emp.id, department.D\_name, emp.phone FROM emp LEFT JOIN department ON emp.dept\_name=department.D\_name;

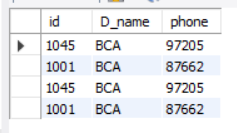
Output:



Command (Right join):

SELECT emp.id, department.D\_name, emp.phone FROM emp RIGHT JOIN department ON emp.dept\_name=department.D\_name;

Output:

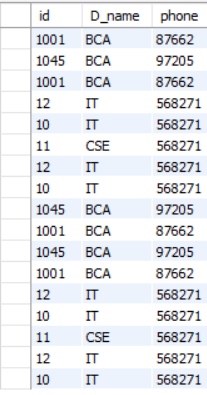


Command (Full Join-> Use Union all):

SELECT emp.id, department.D\_name, emp.phone FROM emp RIGHT JOIN department ON emp.dept\_name=department.D\_name

UNION ALL

SELECT emp.id, department.D\_name, emp.phone FROM emp RIGHT JOIN department ON emp.dept\_name=department.D\_name; Output:



4)Database Schema for a customer-sale scenario of a general store

a) Create the tables with the appropriate integrity constraints

Command:

create database GeneralStore;

use GeneralStore;

create table Customer

(

Cust\_id int Primary key,

Cust\_name varchar(30)

);

create table Item

(

Item\_id int Primary key,

Item\_name varchar(30),

Price float

);

create table Sale

(

bill\_no int Primary key,

bill\_date date,

cust\_id int,

item\_id int,

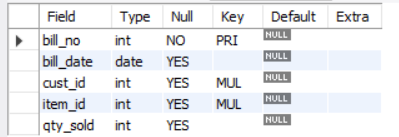
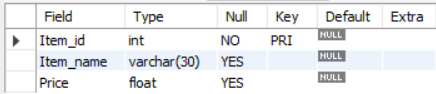
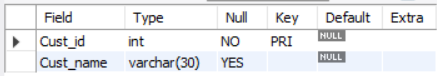
qty\_sold int,

foreign key(cust\_id) references customer(cust\_id),

foreign key(item\_id) references item(item\_id)

);

Output:

b) Insert around 10 records in each of the tables

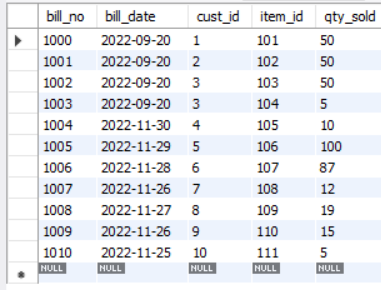
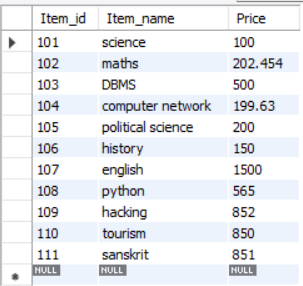
Command:

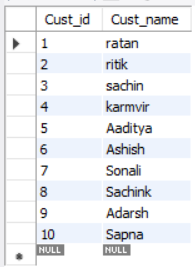
insert into sale values (1000,'2022-09-20',1,101,50),(1001,'2022-09-20',2,102,50),(1002,'2022-09-20',3,103,50),(1003,'2022-09-20',3,104,5),(1004,'2022-11-30',4,105,10),(1005,'2022-11-29',5,106,100),(1006,'2022-11-28',6,107,87),(1007,'2022-11-26',7,108,12),(1008,'2022-11-27',8,109,19),(1009,'2022-11-26',9,110,15),(1010,'2022-11-25',10,111,5);

insert into item values (101,"science",100),(102,"maths",202.454),(103,"DBMS",500),(104,"computer network",199.63),(105,"political science",200),(106,"history",150),(107,"english",1500),(108,"python",565),(109,"hacking",852),(110,"tourism",850),(111,"sanskrit",851);

insert into customer values (1,"ratan"),(2,"ritik"),(3,"sachin"),(4,"karmvir"),(5,"Aaditya"),(6,"Ashish"),(7,"Sonali"),(8,"Sachink"),(9,"Adarsh"),(10,"Sapna");

Output:



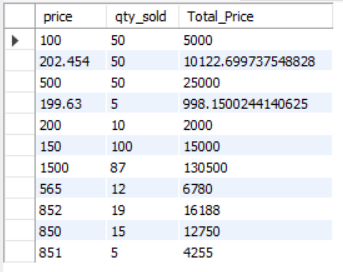
c) List the total Bill details with the quantity sold, price of the item and

the final amount

Command:

SELECT price, qty\_sold, (price\*qty\_sold) Total\_Price FROM item, sale WHERE item.Item\_id=sale.Item\_id;

Output:



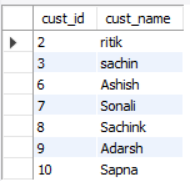
d) List the details of the customer who have bought a product which

has a price>200

Command:

SELECT customer.cust\_id, Customer.cust\_name FROM Customer, Item, sale WHERE Item.Price>200 AND customer.cust\_id=sale.cust\_id AND Item.Item\_id=sale.item\_id;

Output:



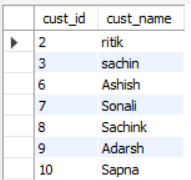
e) Give a count of how many products have been bought by each

customer

Command:

SELECT cust\_id, count(Item\_id) FROM sale GROUP BY cust\_id;

Output:

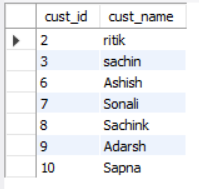


f) Give a list of products bought by a customer having cust\_id as 5

Command:

SELECT i.Item\_Name FROM item i, sale s WHERE s.cust\_id=5 AND i.item\_id-s.item\_id;

Output:



g) List all the bills for the current date with the customer names and

item numbers

Command:

SELECT Item.item\_id,Item.item\_name FROM item, sale WHERE Item.item\_id=sale.item\_id AND sale.bill\_date=curdate();

Output:



PROGRAM 21

**Q:**

1)Demonstrate the application of AUTO\_INCREMENT in MySql

Command:

CREATE TABLE Persons (

Personid int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int

);

INSERT INTO Persons (FirstName,LastName)

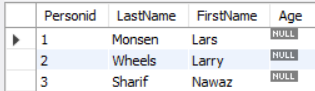
VALUES ('Lars','Monsen');

INSERT INTO Persons (FirstName,LastName)

VALUES ('Larry','Wheels'),('Nawaz','Sharif');

SELECT \* FROM Persons;

Output:



2)Execute following functions for mySql

Strings Command:

SELECT CHAR\_LENGTH("SQL Lab");

SELECT CONCAT("SQL ","Lab ","is ","fun ");

SELECT UCASE("sql lab");

SELECT LCASE("SQL LAB");

SELECT RTRIM("SQL Lab ");

SELECT LTRIM(" SQL Lab");

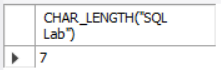
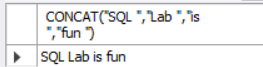
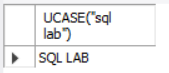
SELECT REVERSE("SQL Lab");

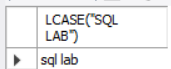
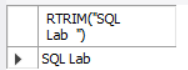
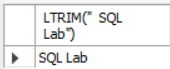
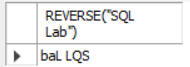
SELECT SUBSTR("SQL Lab",3);

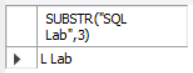
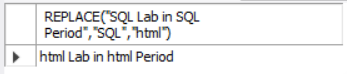
SELECT STRCMP("SQL","sql");

SELECT REPLACE("SQL Lab in SQL Period","SQL","html");

Output:

Numeric Functions Command:

SELECT ABS(-245.563);

SELECT CEIL(2.7);

SELECT FLOOR(2.3);

SELECT GREATEST(3,2,4,5,6);

SELECT LEAST(3,2,4,5,6);

SELECT LOG(2);

SELECT LOG10(2);

SELECT 12.1 DIV 3;

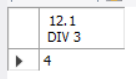
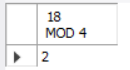
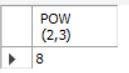
SELECT 18 MOD 4;

SELECT POW (2,3);

SELECT SQRT(9);

SELECT TRUNCATE(3.144444,3);

Output:

PROGRAM 22

**Q:** To create and update views

i) from single table

Command:

CREATE VIEW viewDept\_id AS

SELECT Dept\_id

FROM department

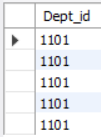
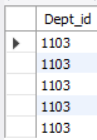
WHERE Dept\_id = 1100;

UPDATE viewDeptIds

SET Dept\_id = 1103

WHERE Dept\_id = 1101;

Output:

ii) from multiple tables

Command:

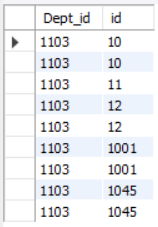
CREATE VIEW Deptid\_Empid AS

SELECT department.Dept\_id, emp.id

FROM department,emp

WHERE department.D\_name = emp.dept\_name;

Output:



PROGRAM 23

**Q:** To perform various Date functions in MySQL

Command:

SELECT CURRENT\_DATE();

SELECT sysdate();

SELECT DATE("2019-10-15 09:34:21");

SELECT MONTH("2022-10-15 09:34:21") AS MONTH;

SELECT YEAR("2022-10-15 09:34:21") AS YEAR;

SELECT QUARTER("2022-10-15 09:34:21") AS QUARTER;

SELECT WEEK("2022-10-15 09:34:21") AS WEEK;

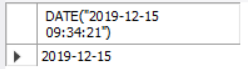
SELECT EXTRACT( MONTH FROM "2022-10-15 09:34:21") AS DATE;

SELECT YEARWEEK("2022-10-15 09:34:21") AS YEAR\_WEEK;

SELECT WEEKOFYEAR("2022-10-15 09:34:21") AS YEAR\_WEEK;

SELECT DAY('2023/01/11 09:08') AS DayOfMonth;

Output:

PROGRAM 24

**Q:** **For a given database with following attributes:**

**EMPLOYEE ID, FIRST NAME, LAST NAME EMAIL PHONE NUMBER, HIRE\_DATE, JOB\_ID, SALARY, MANAGER\_ID, DEPARTMENT\_ID**

****

1)write a SQL query to find those employees whose salary matches the lowest salary of any of the departments. Return first name, last name and department ID.

Command:

select First\_name,Last\_name,Department\_ID from emp where salary=(select min(Salary) from emp) group by Department\_ID;

Output:

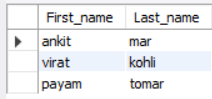


2)write a SQL query to find those employees who receive a higher salary than the employee with ID 163. Return first name, last name.

Command:

select First\_name,Last\_name from emp where salary > (select salary from emp where Employee\_id=163);

Output:



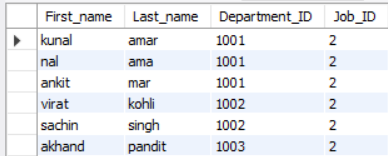
3) write a SQL query to find out which employees have the same designation as the

employee whose ID is 169. Return first name, last name, department ID and job ID.

Command:

select First\_name,Last\_name,Department\_ID,Job\_ID from emp where Job\_ID=(select Job\_ID from emp where Employee\_ID=169);

Output:



4) write a SQL query to find those employees who report to that manager whose first name

is 'Payam'. Return first name, last name, employee ID and salary.

Command:

select First\_name,Last\_name,Employee\_ID,Salary from emp where First\_name=(select First\_name from emp where Manager\_ID=31 && First\_name='payam');

Output:



PROGRAM 2

**Q:**

Command:

Output:

PROGRAM 2

**Q:**

Command:

Output:

PROGRAM 2

**Q:**

Command:

Output: