Department of Information and Technology

Gauhati University Institute of Science & Technology

Gauhati University, GNB Nagar,Ghy-14

LABORATORY REPORT Database Management Systems(IT-311)

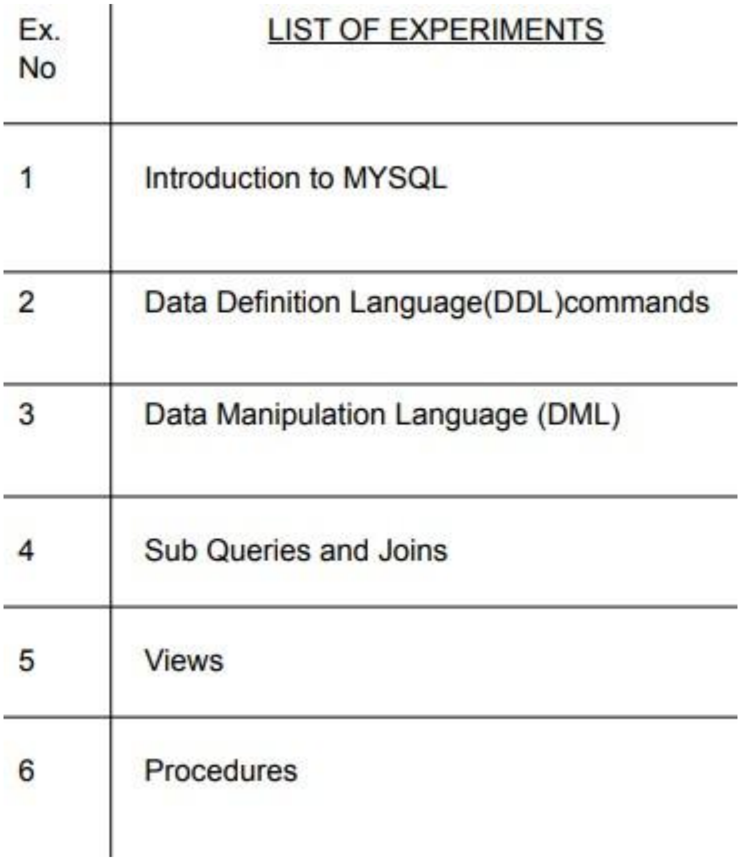
Name: Bhabajyoti Borah

Roll No. : 200102017

Branch: Information Technology

Semester: 5th

INDEX



EXPERIMENT NO: 1 Introduction to MYSQL

AIM: To study about Mysql database

OBJECTIVES:

THEORY:
MySQL is one of the most popular Open Source SQL database management systems. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, a Swedish company.

MySQL is becoming popular for many good reasons:
MySQL is released under an open-source license. So you have nothing to pay to use it.

MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

MySQL uses a standard form of the well-known SQL data language.

MySQL works on many operating systems and languages including PHP, PERL, C, C++, JAVA, etc.

MySQL works very quickly and works well even with large data sets.

MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table.
MySQL is customizable.

RESULT: The MySQL database is studied

EXPERIMENT NO:2 DATA DEFINITION LANGUAGE (DDL) COMMANDS

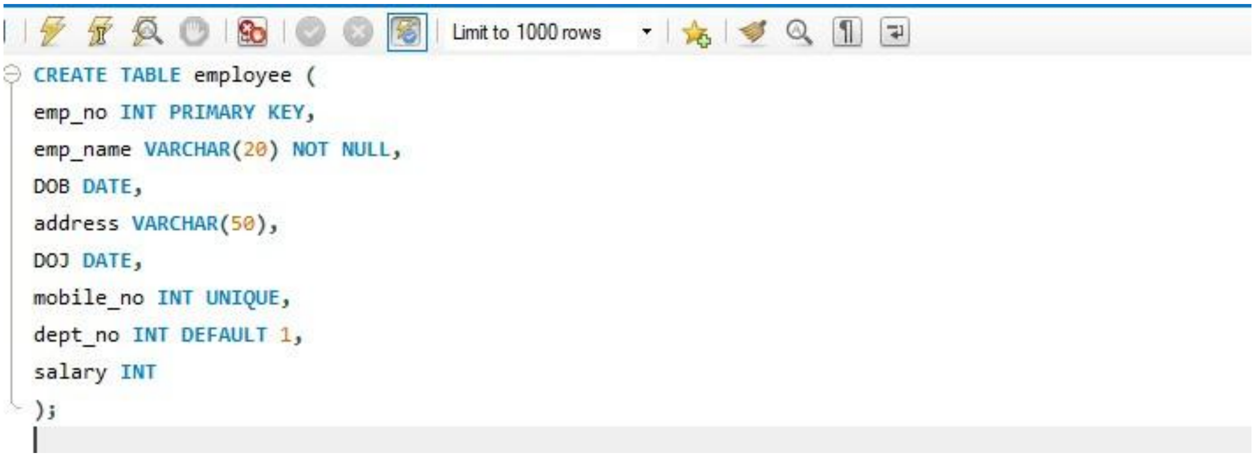
AIM: Consider the database for an organisation3. Write the queries for the following

(i) create the database
Ans: CREATE DATABASE organization5;

(ii) select the current database
Ans: USE organization5;

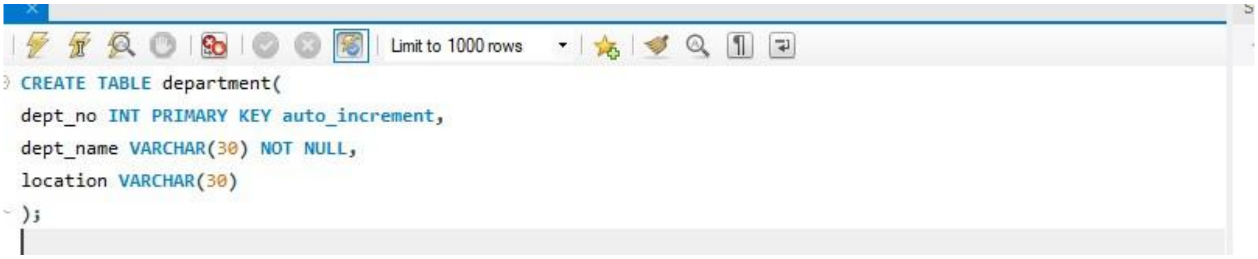
(iii)Create the following tables.
a. employee (emp_no, emp_name, DOB, address, doj, mobile_no,dept_no, salary).
b. department (dept_no, dept_name, location).

(iv)Include necessary constraints.
Ans:
(a) CREATE TABLE employee



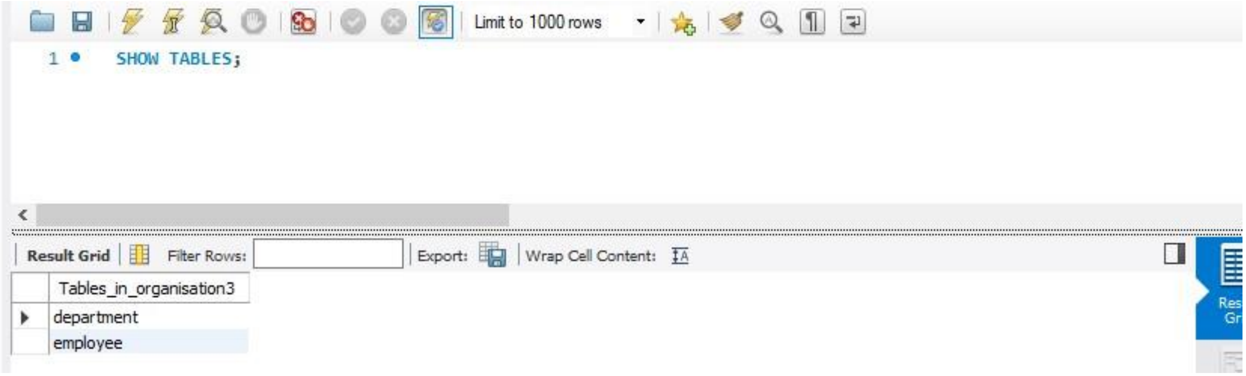
```
CREATE TABLE employee (
emp_no INT PRIMARY KEY,
emp_name VARCHAR(20) NOT NULL,
DOB DATE,
address VARCHAR(50),
DOJ DATE,
mobile_no INT UNIQUE,
dept_no INT DEFAULT 1,
salary INT
);
```

(b) CREATE TABLE department



```
CREATE TABLE department(
dept_no INT PRIMARY KEY auto_increment,
dept_name VARCHAR(30) NOT NULL,
location VARCHAR(30)
);
```

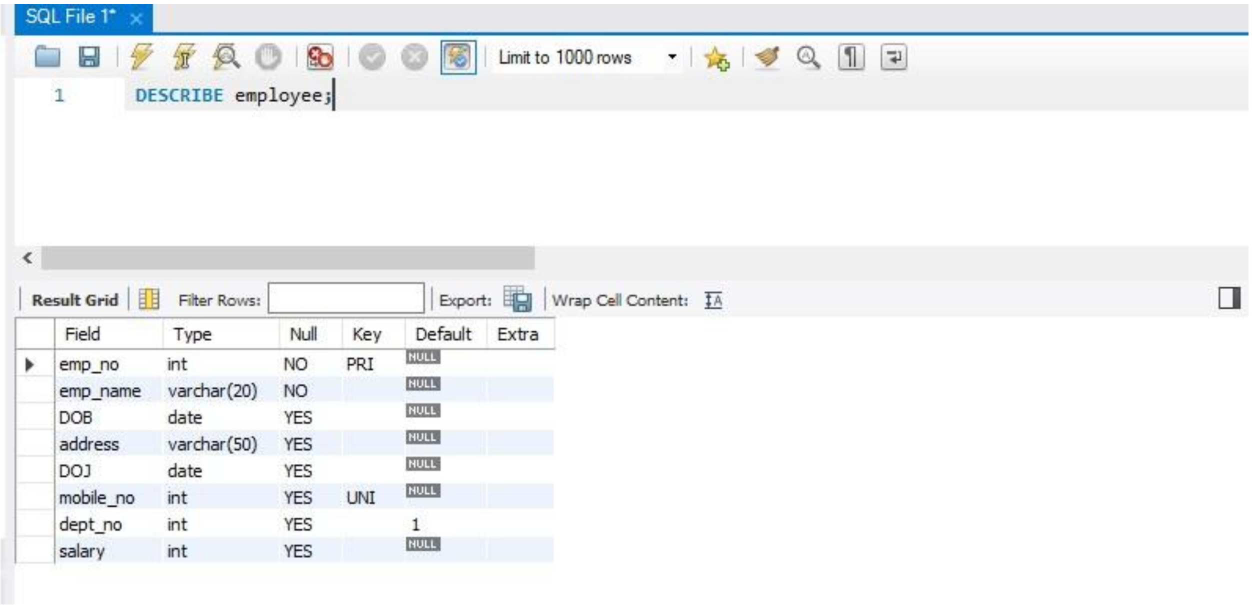
(v) List all the tables in the current database
Ans: SHOW TABLES;



```
SHOW TABLES;
```

Tables_in_organisation3
department
employee

(vi)Display the structure of the employee table
 Ans: DESCRIBE employee;



```
DESCRIBE employee;
```

Field	Type	Null	Key	Default	Extra
emp_no	int	NO	PRI	NULL	
emp_name	varchar(20)	NO		NULL	
DOB	date	YES		NULL	
address	varchar(50)	YES		NULL	
DOJ	date	YES		NULL	
mobile_no	int	YES	UNI	NULL	
dept_no	int	YES		1	
salary	int	YES		NULL	

(vii) Add a new column Designation to the employee table
 Ans: ALTER TABLE employee ADD Designation VARCHAR(20);

 (viii) Drop the column location from Dept table
Ans: ALTER TABLE department DROP location;

 (ix) Drop the tables
 Ans: DROP table department; DROP TABLE employee;

(x) Delete the database
 Ans: DROP DATABASE organization5

OBJECTIVES: To understand DDL commands.

THEORY:

Database Queries:

Before creating any tables, MySQL requires you to create a database by executing the CREATE DATABASE command

>Create a database

CREATE DATABASE

Delete a database DROP DATABASE

Select the database USE

List all databases SHOW databases;

Rename a database ALTER DATABASE RENAME

Table Queries:
>To Create a table CREATE TABLE (< fieldtype>() , …)
>List all tables in the current database SHOW tables;
>Show table format with column names and data types DESCRIBE
>Modify the structure of table
ALTER TABLE <tabler name> alter specifications>
ALTER TABLE <table name> DROP COLUMN< column name>
ALTER TABLE <table name> ADD COLUMN < column name > datatype>(<size>)
>Delete the table
DROP TABLE <table name>

Constraints:
>**Primary key** A PRIMARY KEY constraint for a table enforces the table to accept unique data for a specific column and this constraint create a unique index for accessing the table faster
>**UNIQUE** The UNIQUE constraint in Mysql does not allow to insert a duplicate value in a column.
>**NOT NULL** In Mysql NOT NULL constraint allows to specify that a column can not contain any NULL value
>**FOREIGN KEY A FOREIGN KE**Y in mysql creates a link between two tables by one specific column of both table. The specified column in one table must be a

>**PRIMARY KEY** and referred by the column of another table known as FOREIGN KEY.
>**CHECK** The CHECK constraint determines whether the value is valid or not from a logical expression.
>**DEFAULT** While inserting data into a table, if no value is supplied to a column, then the column gets the value set as DEFAULT

PROCEDURE:
(i) CREATE DATABASE command
(ii) USE DATABASE command
(iii) CREATE TABLE command
(iv) PRIMARY KEY,NOT NULL etc
(v) SHOW TABLES command
(vi) DESCRIBE TABLE command
(vii) ALTER TABLE command
(viii) ALTER TABLE command
(ix) DROP TABLE command
(x) DROP DATABASE command

RESULT: The DDL commands have been executed successfully**.**
OUTPUT:

Problems:
1. Consider the database for a college and design an ER diagram. Write the query for the following.

(i) Create the tables: Student (sid, sname, sex, dob,dno)

Department (dno, dname)

Faculty (F_id, fname, designation, salary,dno)

Faculty (F_id, fname, designation, salary,dno)

Course (cid, cname, credits,dno)

Register (sid,cid,sem)

Teaching (f_id,cid,sem)

Hostel(hid,hname,seats,)

(ii) Include the necessary constraints NOT NULL, DEFAULT, CHECK, and PRIMARY KEY, UNIQUE.

Ans.:

```
 CREATE TABLE Student(sid INT PRIMARY KEY, sname VARCHAR(20) NOT
NULL,sex VARCHAR(10), dob DATE, dno INT DEFAULT '1');

CREATE TABLE Department (dno INT PRIMARY KEY, dname VARCHAR(30) NOT
NULL);

CREATE TABLE Faculty (F_id INT PRIMARY KEY, fname VARCHAR(20) NOT NULL,
designation VARCHAR(20) DEFAULT 'Professor', salary INT, dno INT DEFAULT '1');

CREATE TABLE Course (cid CHAR(10) PRIMARY KEY, cname VARCHAR(30) NOT
NULL, credits INT, dno INT);

CREATE TABLE Register (sid INT PRIMARY KEY, cid CHAR(10) UNIQUE, sem INT
NOT NULL, CHECK (sem<=8) );

CREATE TABLE Teaching(f_id INT PRIMARY KEY, cid CHAR(10) UNIQUE, sem INT
NOT NULL, CHECK (sem<=8));
```

(iii) Create a database college.

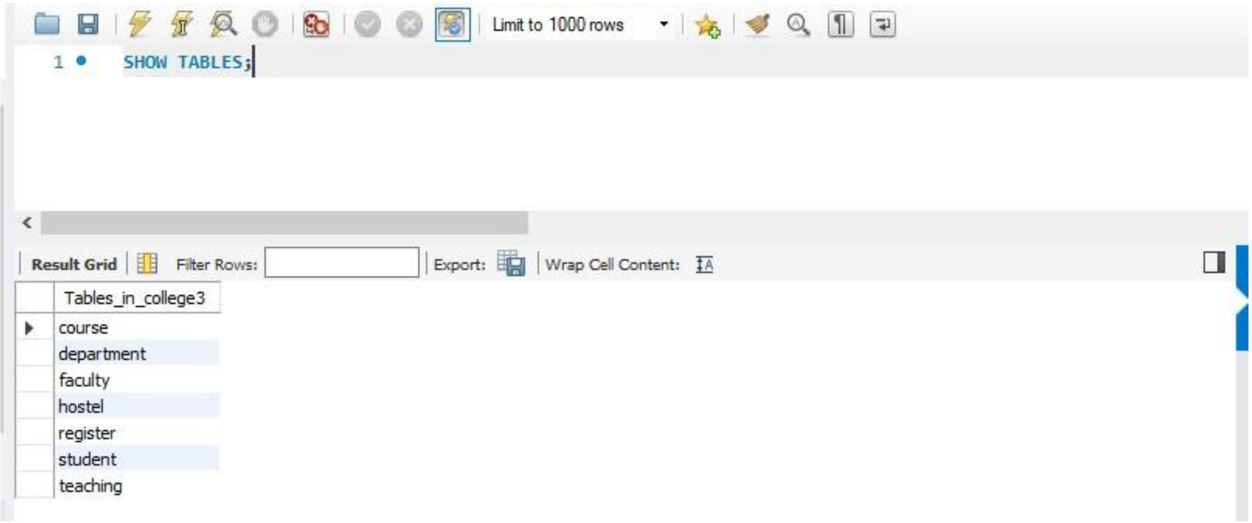
Ans.: `CREATE DATABASE college;`

(iv) Use college as the current database

Ans.: `USE college;`

(v) Display all the tables in college database

Ans.: `SHOW TABLES;`

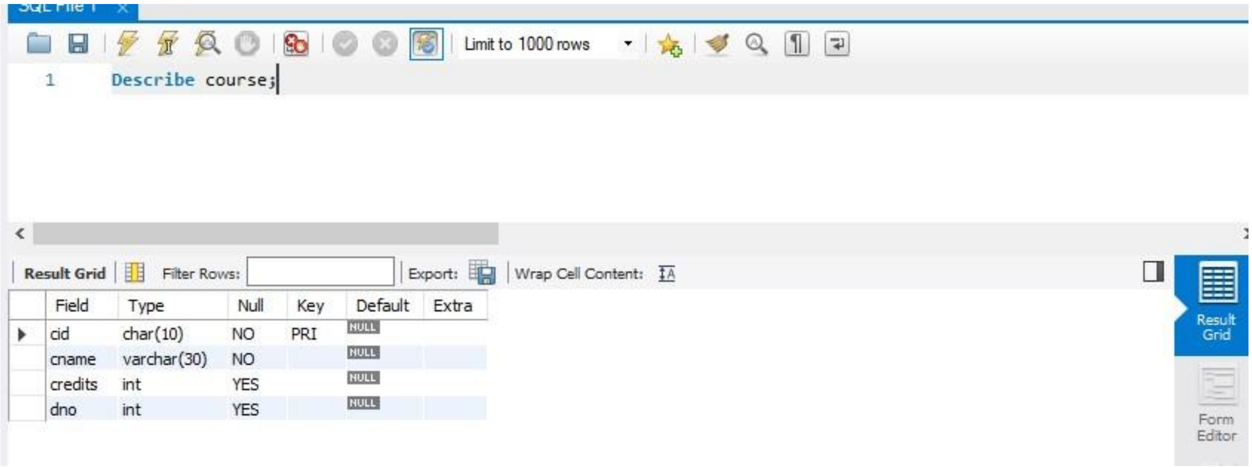


```
SHOW TABLES;
```

Tables_in_college3
course
department
faculty
hostel
register
student
teaching

(vi) Describe the structure of all tables.

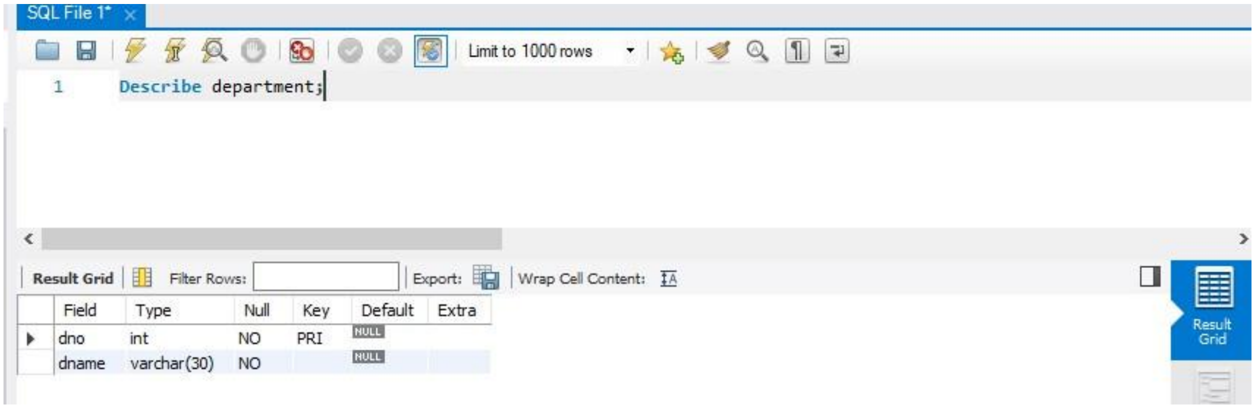
Ans.: Describe course;



```
Describe course;
```

Field	Type	Null	Key	Default	Extra
cid	char(10)	NO	PRI	NULL	
cname	varchar(30)	NO		NULL	
credits	int	YES		NULL	
dno	int	YES		NULL	

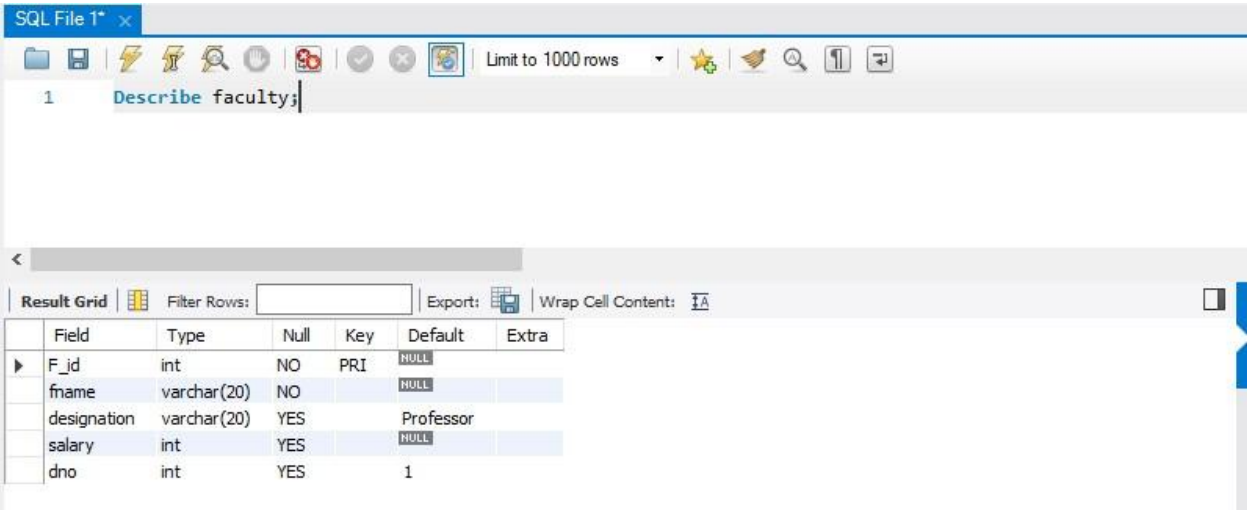
Describe department;



```
Describe department;
```

Field	Type	Null	Key	Default	Extra
dno	int	NO	PRI	NULL	
dname	varchar(30)	NO		NULL	

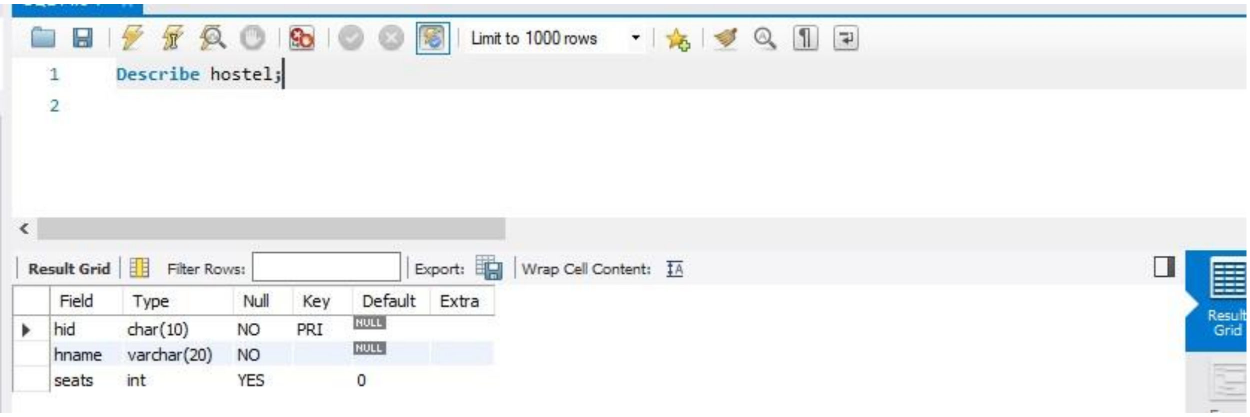
Describe faculty;



```
Describe faculty;
```

Field	Type	Null	Key	Default	Extra
F_id	int	NO	PRI	NULL	
fname	varchar(20)	NO		NULL	
designation	varchar(20)	YES		Professor	
salary	int	YES		NULL	
dno	int	YES		1	

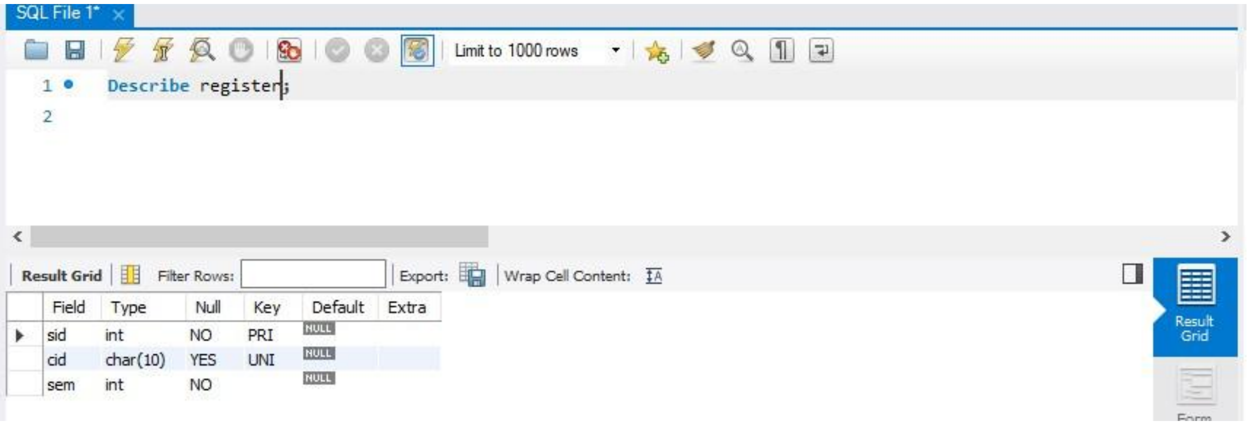
Describe hostel;



```
Describe hostel;
```

Field	Type	Null	Key	Default	Extra
hid	char(10)	NO	PRI	NULL	
hname	varchar(20)	NO		NULL	
seats	int	YES		0	

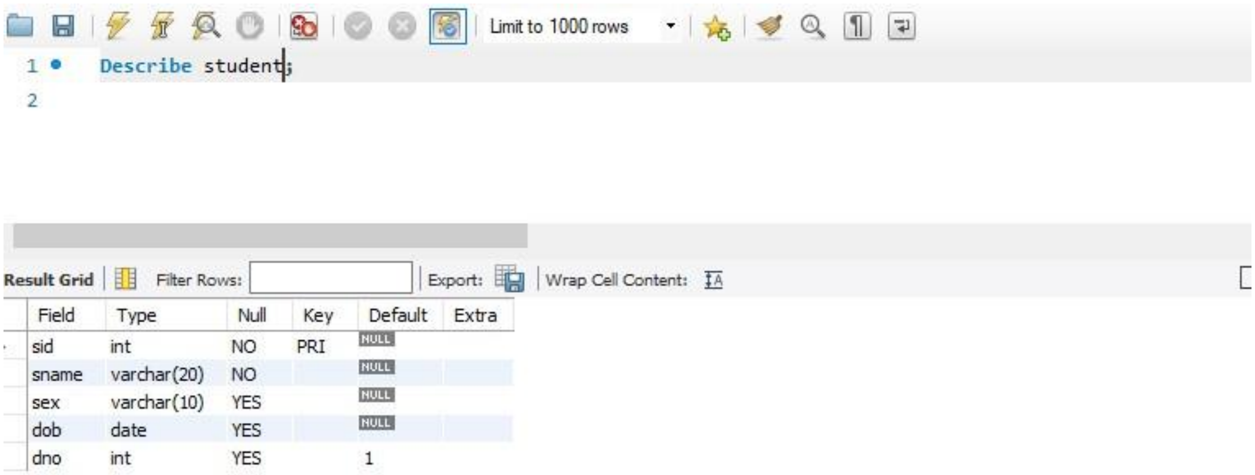
Describe register;



```
Describe register;
```

Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
cid	char(10)	YES	UNI	NULL	
sem	int	NO		NULL	

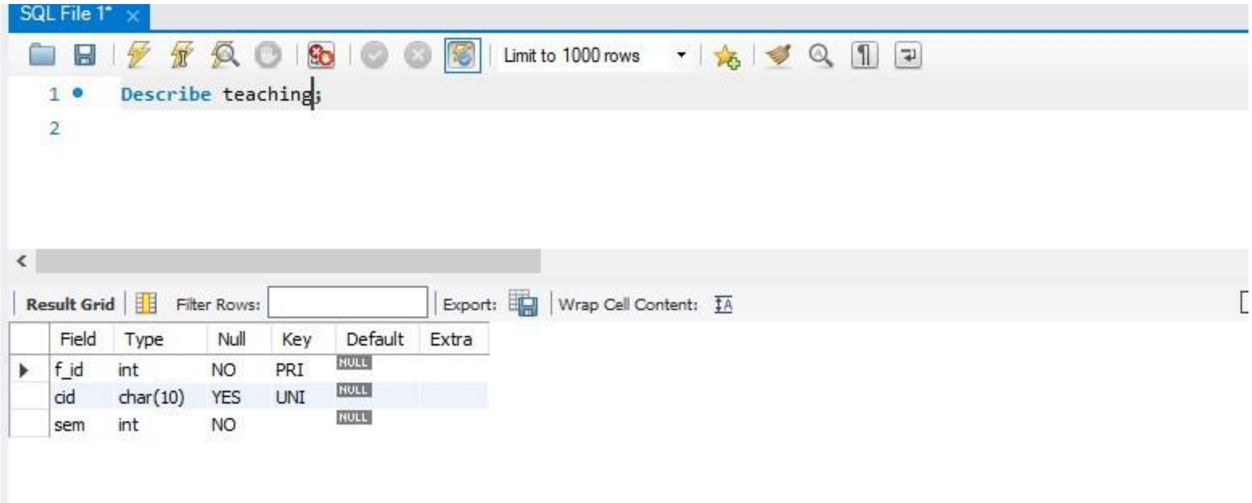
Describe student;



```
Describe student;
```

Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
sname	varchar(20)	NO		NULL	
sex	varchar(10)	YES		NULL	
dob	date	YES		NULL	
dno	int	YES		1	

Describe teaching;



```
Describe teaching;
```

Field	Type	Null	Key	Default	Extra
f_id	int	NO	PRI	NULL	
cid	char(10)	YES	UNI	NULL	
sem	int	NO		NULL	

(vii) Modify the student table to add a new field ‘grade’.
Ans. ALTER TABLE student ADD grade CHAR(2);

2. Consider the database for a banking enterprise.
Write the queries for the below questions. (i) Create the following tables

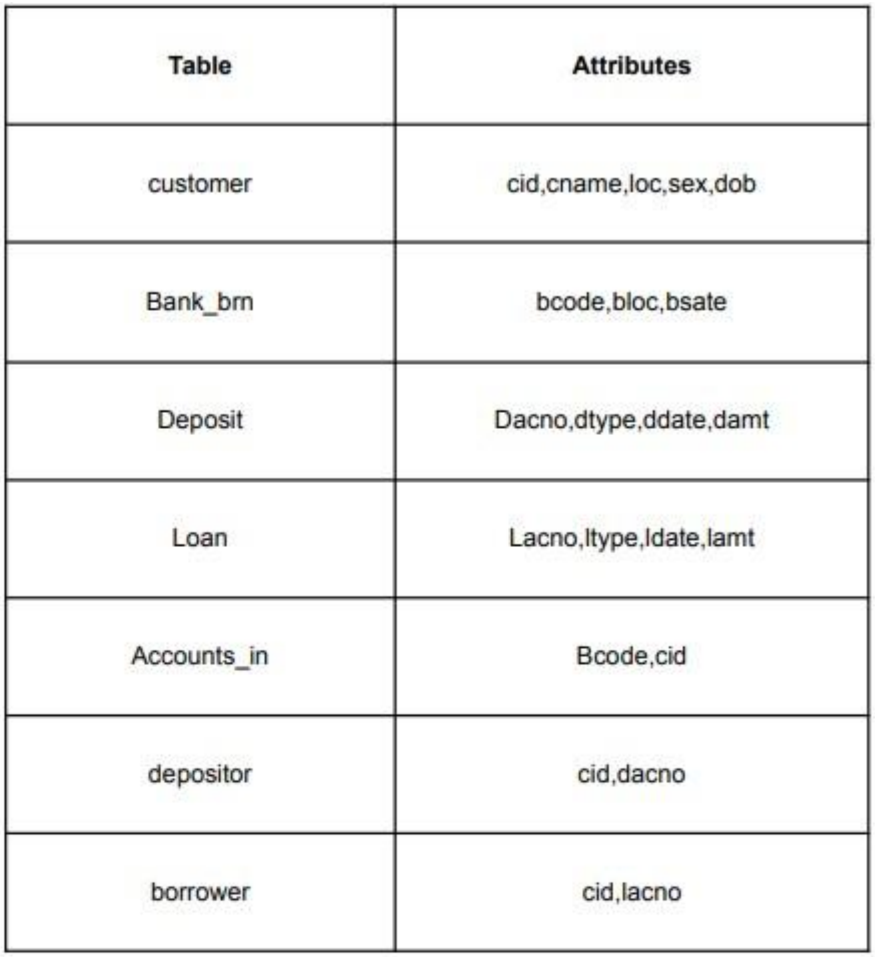


Table	Attributes
customer	cid,cname,loc,sex,dob
Bank_brn	bcode,bloc,bsate
Deposit	Dacno,dtype,ddate,damt
Loan	Lacno,ltype,ldate,lamt
Accounts_in	Bcode,cid
depositor	cid,dacno
borrower	cid,lacno

(ii). Include necessary constraints.

Ans:

CREATE TABLE customer (cid INT PRIMARY KEY, cname VARCHAR(20) NOT NULL, loc VARCHAR(30), sex VARCHAR(10), dob DATE);

CREATE TABLE Bank_brn (bcode VARCHAR(20) PRIMARY KEY, bloc VARCHAR(30), bstate VARCHAR(20));

CREATE TABLE Deposit (Dacno INT PRIMARY KEY, dtype VARCHAR(20), ddate DATE, damt DECIMAL(9,2) NOT NULL);

CREATE TABLE Loan (Lacno INT PRIMARY KEY, ltype VARCHAR(20), ldate DATE, lamt DECIMAL(10,2) NOT NULL);

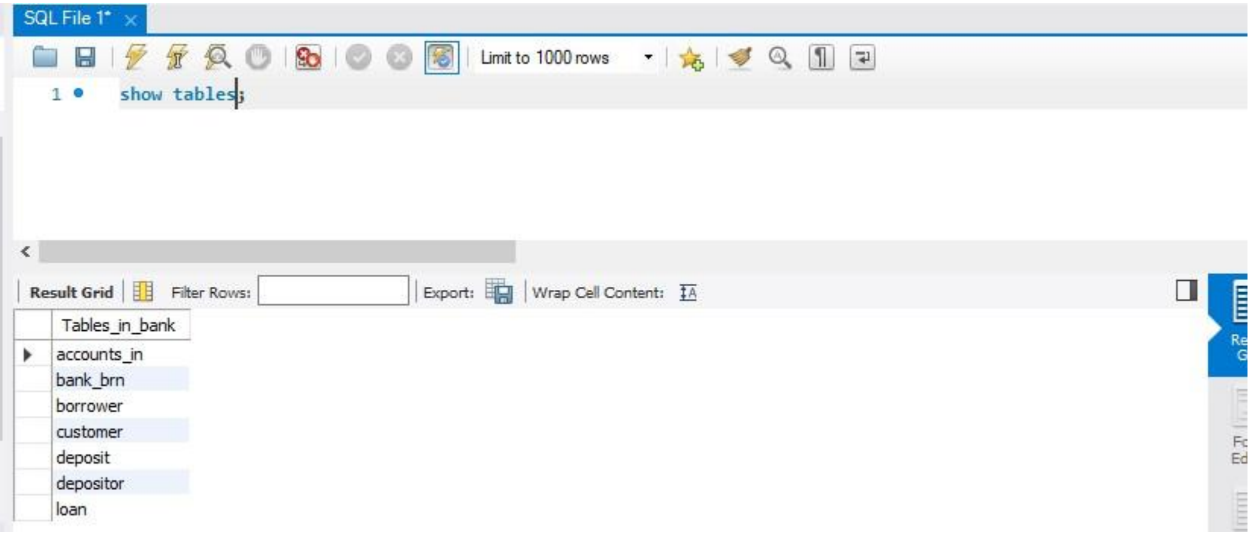
CREATE TABLE Accounts_in (Bcode VARCHAR(20) PRIMARY KEY, cid INT NOT NULL);

CREATE TABLE depositor (cid INT PRIMARY KEY, dacno INT UNIQUE);

CREATE TABLE borrower (cid INT PRIMARY KEY, lacno INT UNIQUE);

(iii). Tables are created under the database ‘bank’.
Ans: CREATE DATABASE bank;
USE bank;

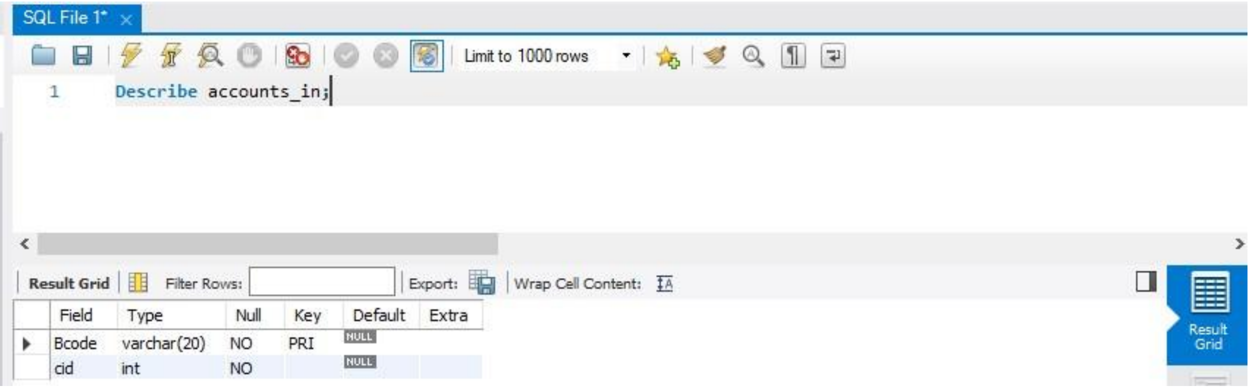
(iv). Display all the tables in bank database.
Ans:



```
show tables;
```

Tables_in_bank
accounts_in
bank_brn
borrower
customer
deposit
depositor
loan

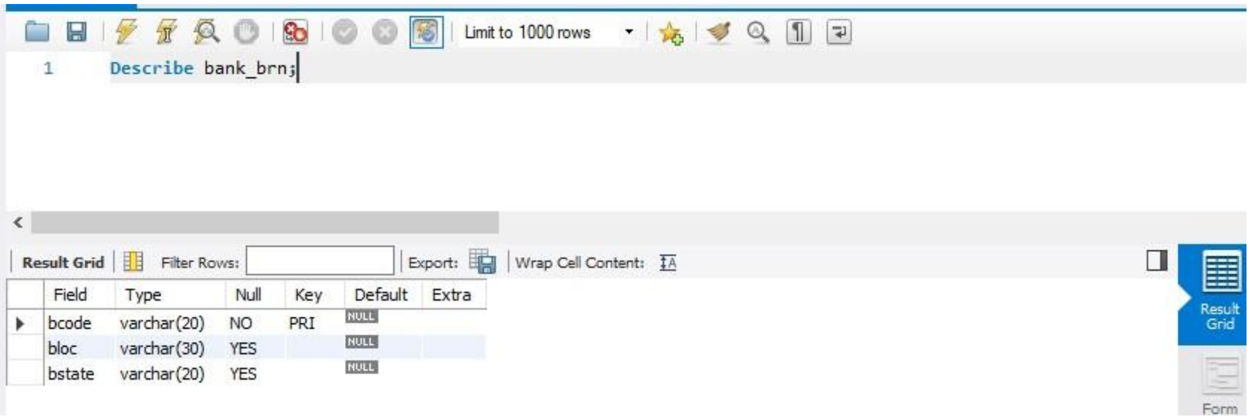
(v). Describe the structure of all tables.
Ans:
Describe accounts_in;



```
Describe accounts_in;
```

Field	Type	Null	Key	Default	Extra
Bcode	varchar(20)	NO	PRI	NULL	
cid	int	NO		NULL	

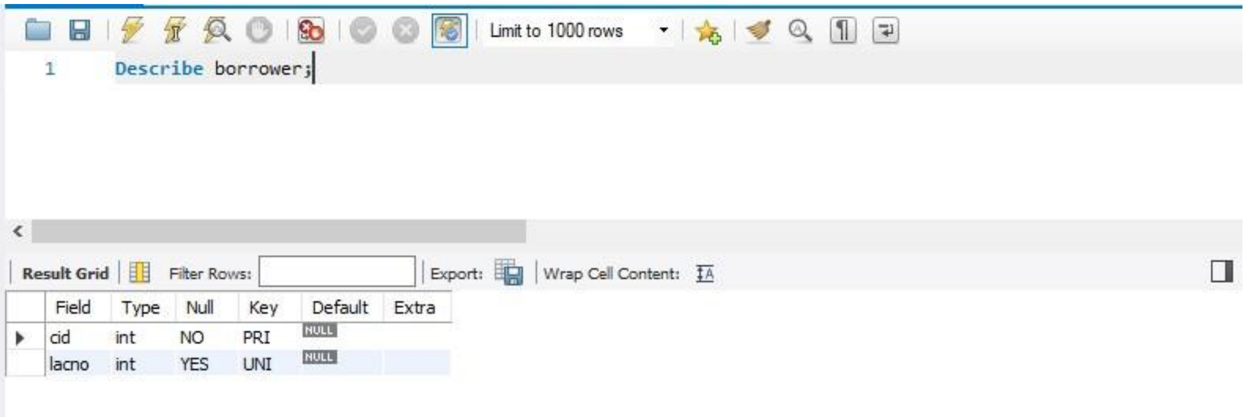
Describe bank_brn;



```
Describe bank_brn;
```

Field	Type	Null	Key	Default	Extra
bcode	varchar(20)	NO	PRI	NULL	
bloc	varchar(30)	YES		NULL	
bstate	varchar(20)	YES		NULL	

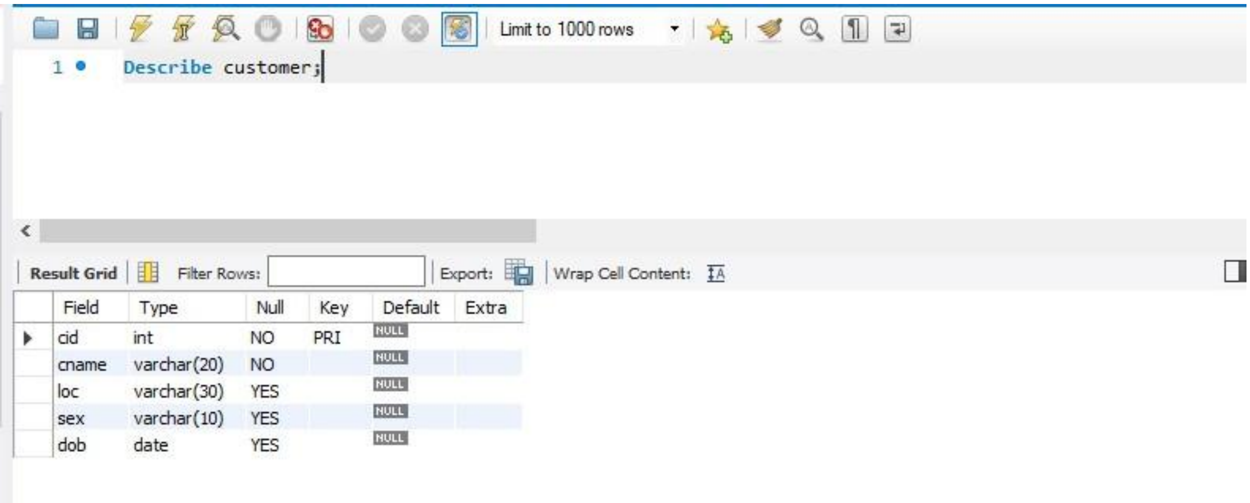
Describe borrower;



```
Describe borrower;
```

Field	Type	Null	Key	Default	Extra
cid	int	NO	PRI	NULL	
lacno	int	YES	UNI	NULL	

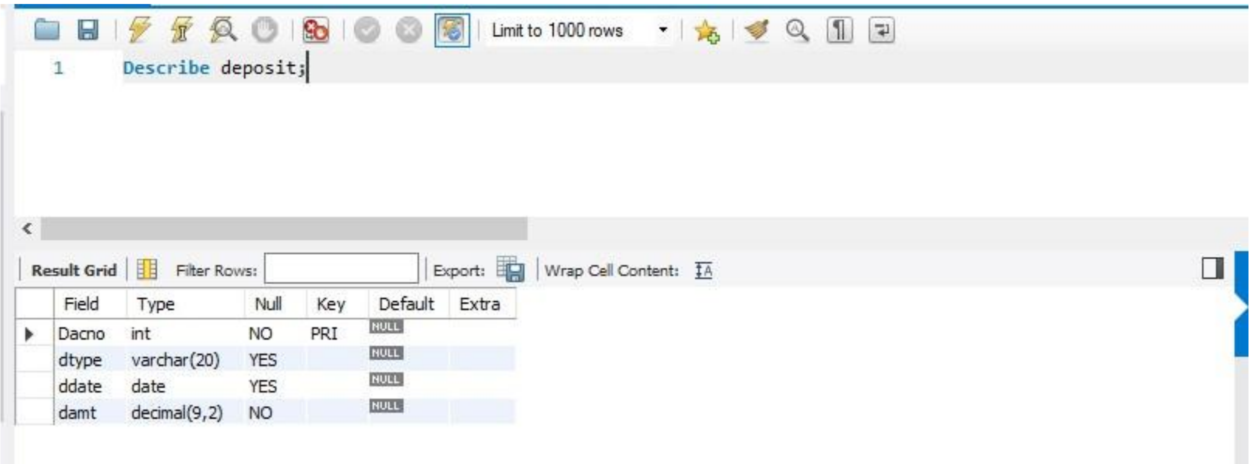
Describe customer;



```
Describe customer;
```

Field	Type	Null	Key	Default	Extra
cid	int	NO	PRI	NULL	
cname	varchar(20)	NO		NULL	
loc	varchar(30)	YES		NULL	
sex	varchar(10)	YES		NULL	
dob	date	YES		NULL	

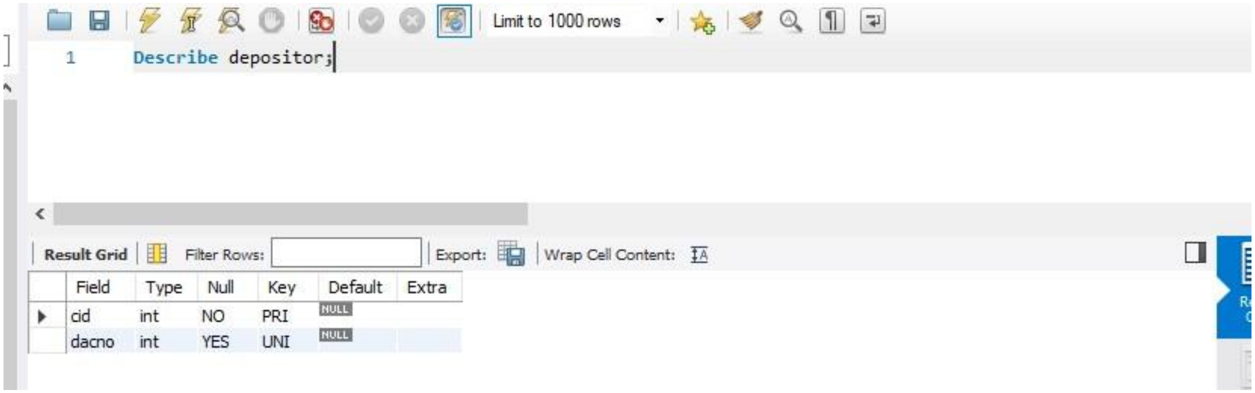
Describe deposit;



```
Describe deposit;
```

	Field	Type	Null	Key	Default	Extra
▶	Dacno	int	NO	PRI	NULL	
	dtype	varchar(20)	YES		NULL	
	ddate	date	YES		NULL	
	damt	decimal(9,2)	NO		NULL	

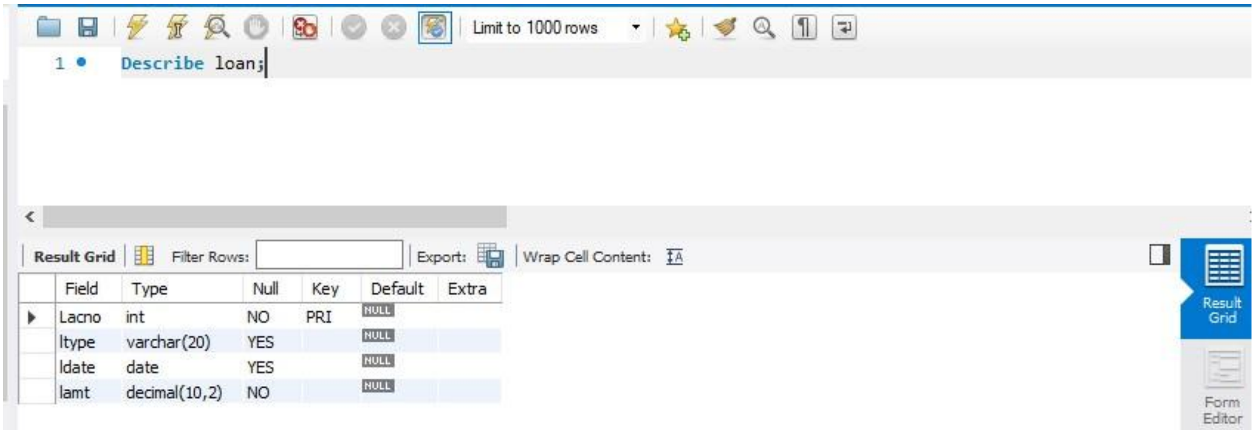
Describe depositor;



```
Describe depositor;
```

	Field	Type	Null	Key	Default	Extra
▶	cid	int	NO	PRI	NULL	
	dacno	int	YES	UNI	NULL	

Describe loan;



```
Describe loan;
```

	Field	Type	Null	Key	Default	Extra
▶	Lacno	int	NO	PRI	NULL	
	ltype	varchar(20)	YES		NULL	
	ldate	date	YES		NULL	
	lamt	decimal(10,2)	NO		NULL	

(vi). Delete tables

Ans:

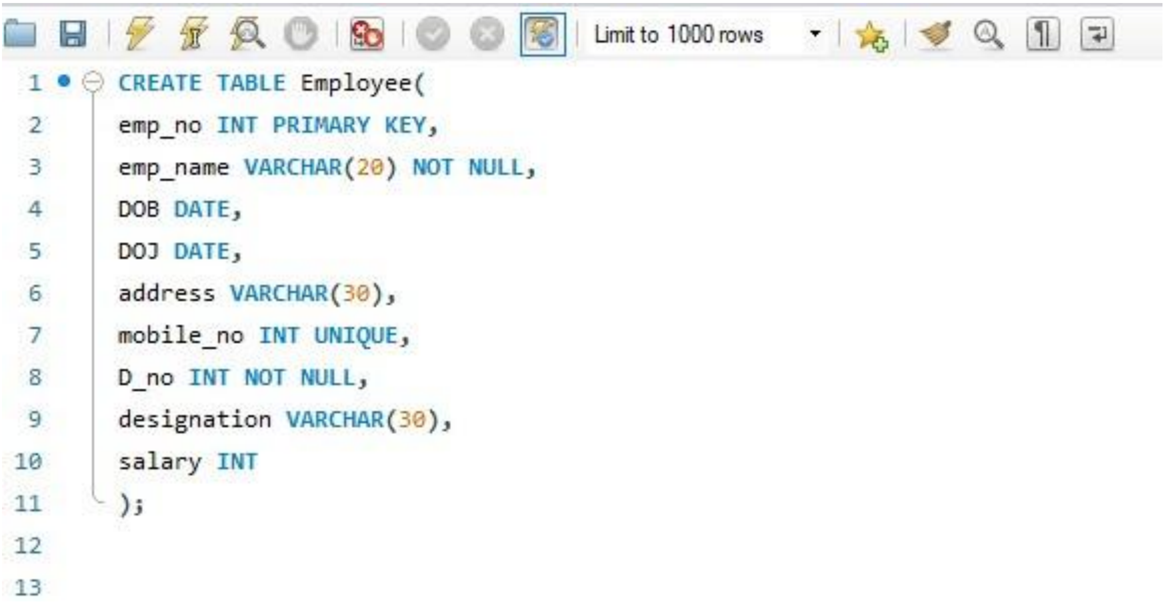
DROP TABLE Bank_brn;
DROP TABLE Deposit;
DROP TABLE customer;
DROP TABLE Loan;
DROP TABLE Accounts_in;
DROP TABLE depositor;

```
DROP TABLE borrower;
```

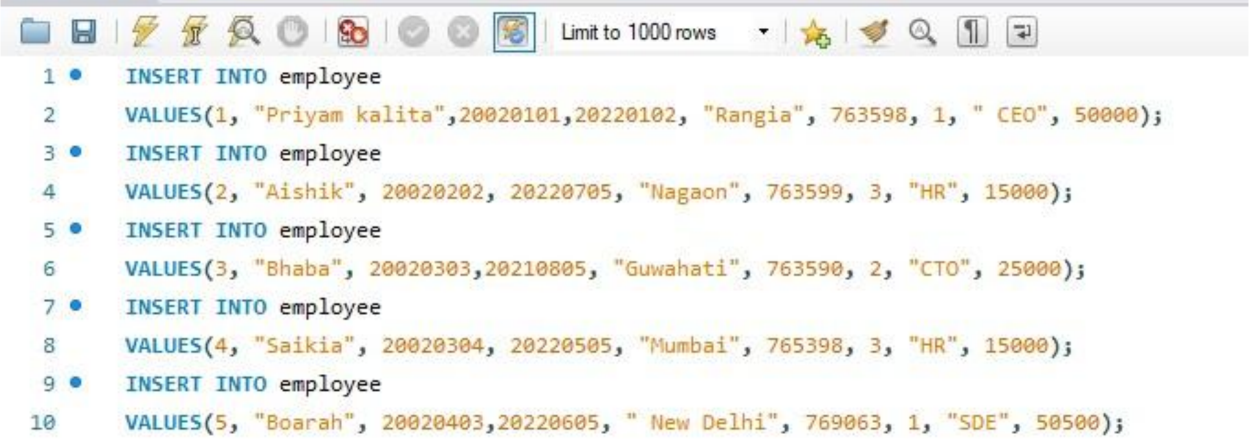
EXPERIMENT NO 3 : DATA MANIPULATION LANGUAGE(DML)

(i) Add 5 rows in the employee and dept tables.

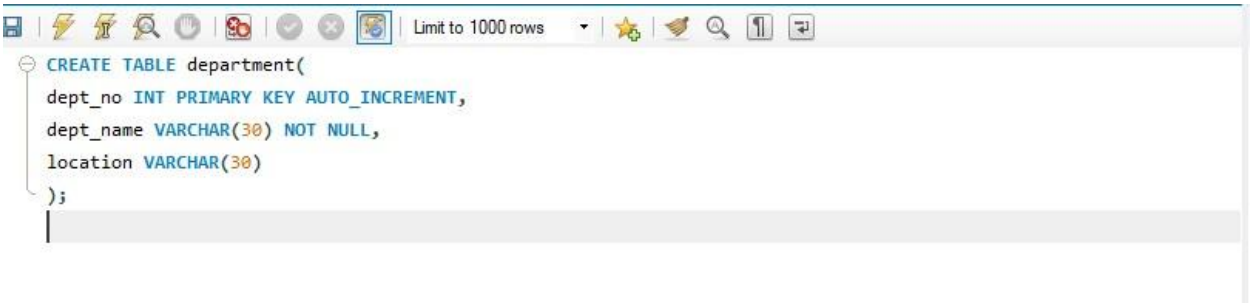
Ans:



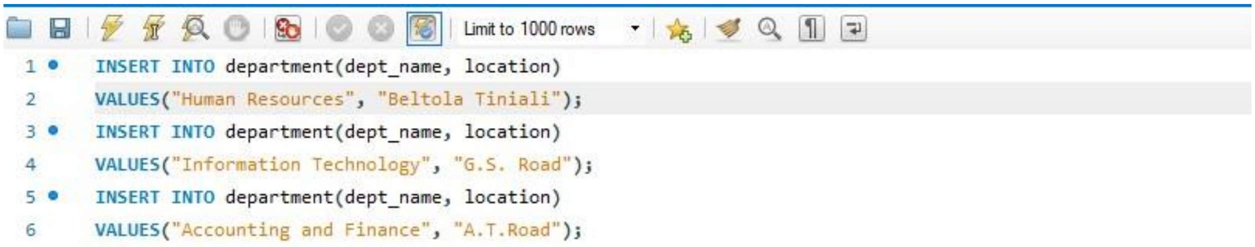
```
CREATE TABLE Employee(
emp_no INT PRIMARY KEY,
emp_name VARCHAR(20) NOT NULL,
DOB DATE,
DOJ DATE,
address VARCHAR(30),
mobile_no INT UNIQUE,
D_no INT NOT NULL,
designation VARCHAR(30),
salary INT
);
```



```
INSERT INTO employee
VALUES(1, "Priyam kalita",20020101,20220102, "Rangia", 763598, 1, " CEO", 50000);
INSERT INTO employee
VALUES(2, "Aishik", 20020202, 20220705, "Nagaon", 763599, 3, "HR", 15000);
INSERT INTO employee
VALUES(3, "Bhaba", 20020303,20210805, "Guwahati", 763590, 2, "CTO", 25000);
INSERT INTO employee
VALUES(4, "Saikia", 20020304, 20220505, "Mumbai", 765398, 3, "HR", 15000);
INSERT INTO employee
VALUES(5, "Boarah", 20020403,20220605, " New Delhi", 769063, 1, "SDE", 50500);
```



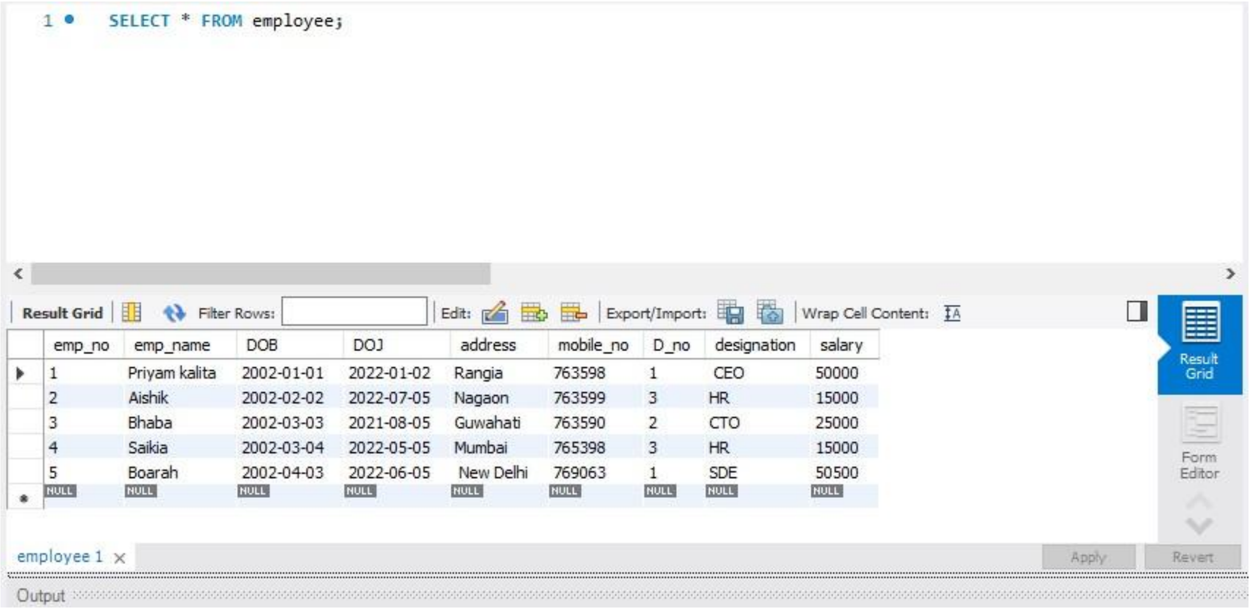
```
CREATE TABLE department(
dept_no INT PRIMARY KEY AUTO_INCREMENT,
dept_name VARCHAR(30) NOT NULL,
location VARCHAR(30)
);
```



```
INSERT INTO department(dept_name, location)
VALUES("Human Resources", "Beltola Tiniali");
INSERT INTO department(dept_name, location)
VALUES("Information Technology", "G.S. Road");
INSERT INTO department(dept_name, location)
VALUES("Accounting and Finance", "A.T.Road");
```

(ii) Display all the records from the above tables.

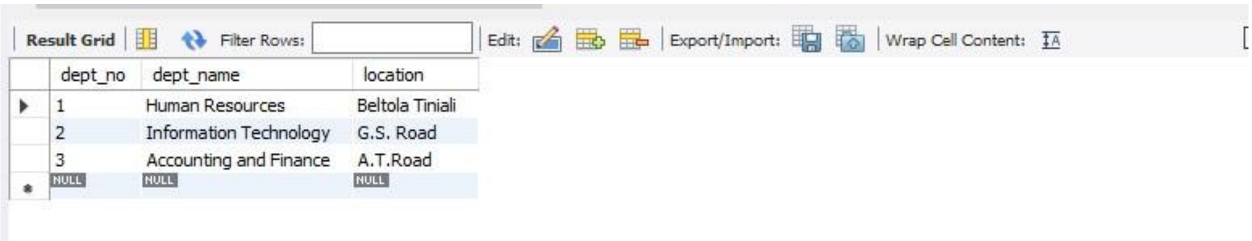
Ans: SELECT * FROM employee;



```
SELECT * FROM employee;
```

emp_no	emp_name	DOB	DOJ	address	mobile_no	D_no	designation	salary
1	Priyam kalita	2002-01-01	2022-01-02	Rangia	763598	1	CEO	50000
2	Aishik	2002-02-02	2022-07-05	Nagaon	763599	3	HR	15000
3	Bhaba	2002-03-03	2021-08-05	Guwahati	763590	2	CTO	25000
4	Saikia	2002-03-04	2022-05-05	Mumbai	765398	3	HR	15000
5	Boarah	2002-04-03	2022-06-05	New Delhi	769063	1	SDE	50500
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

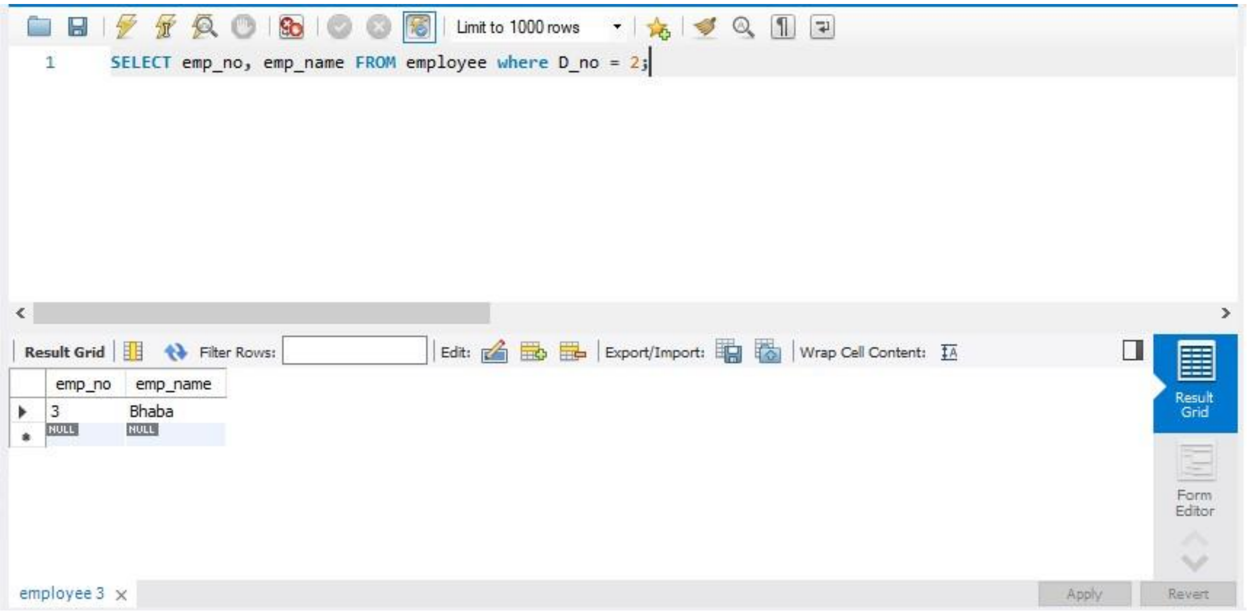
SELECT * FROM department;



dept_no	dept_name	location
1	Human Resources	Beltola Tiniali
2	Information Technology	G.S. Road
3	Accounting and Finance	A.T.Road
NULL	NULL	NULL

(iii) Display the empno and name of all the employees from department no 2.

Ans: SELECT emp_no, emp_name FROM employee where D_no = 2;

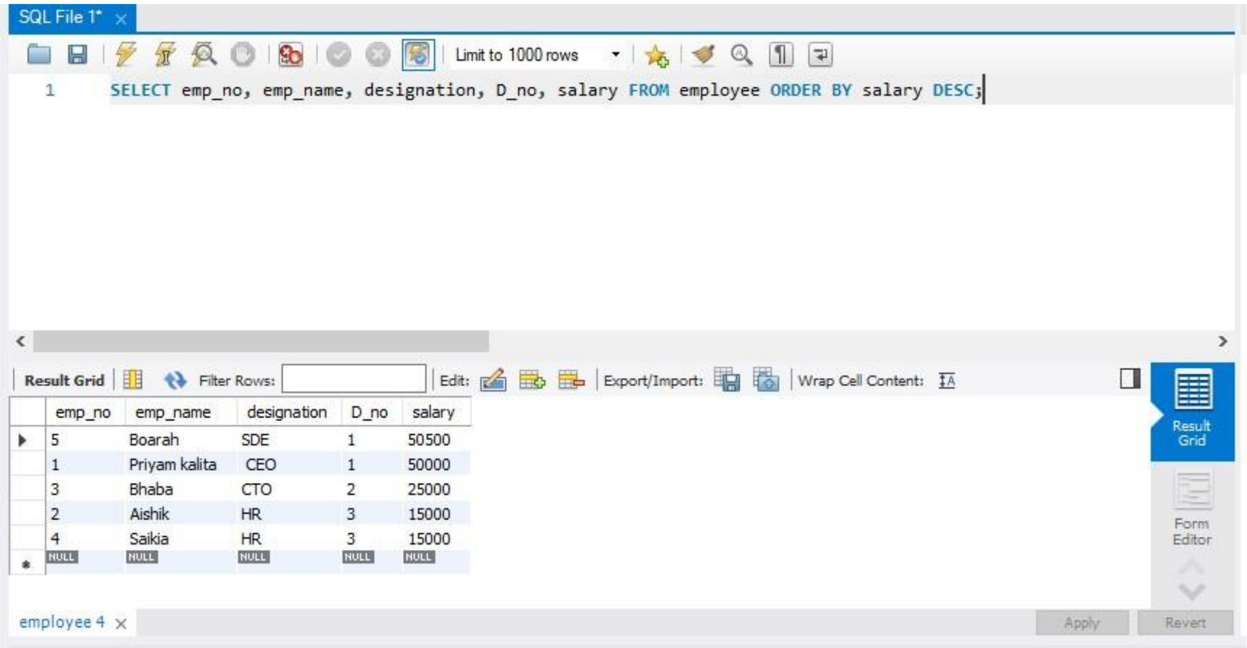


```
SELECT emp_no, emp_name FROM employee where D_no = 2;
```

	emp_no	emp_name
▸	3	Bhaba
*	NULL	NULL

(iv) Display empno,name,designation,dept no and salary in the descending order of salary.

Ans: SELECT emp_no, emp_name, designation, D_no, salary FROM employee ORDER BY salary DESC;

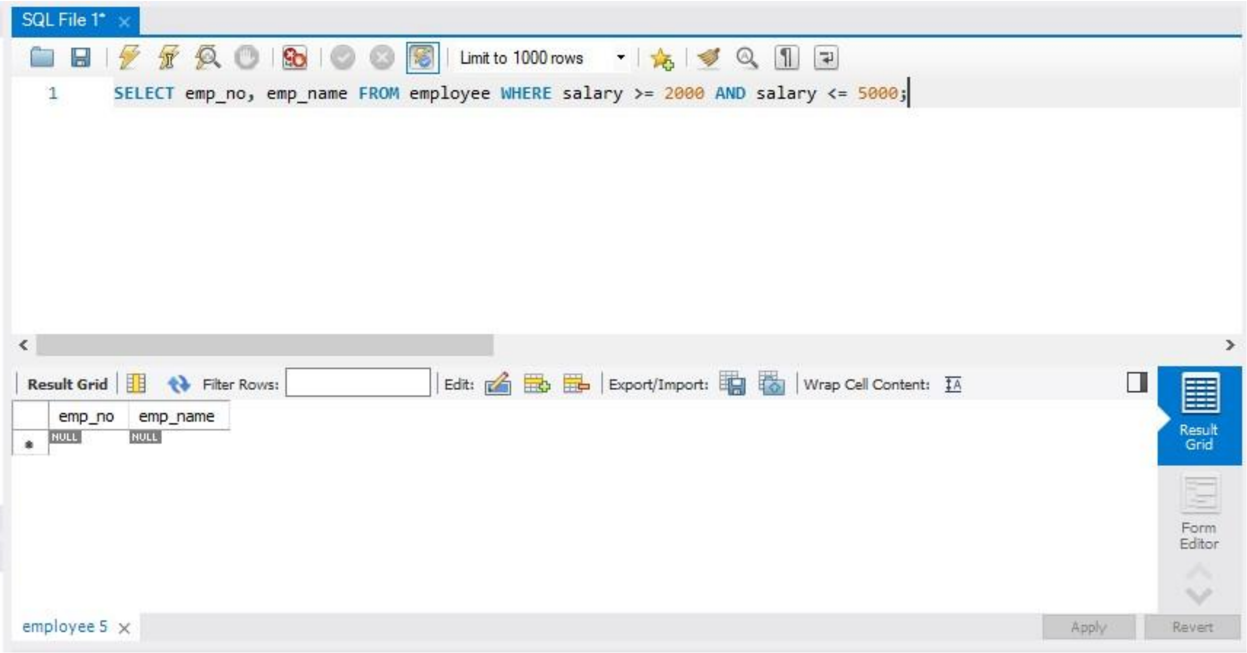


```
SELECT emp_no, emp_name, designation, D_no, salary FROM employee ORDER BY salary DESC;
```

	emp_no	emp_name	designation	D_no	salary
▸	5	Boarah	SDE	1	50500
	1	Priyam kalita	CEO	1	50000
	3	Bhaba	CTO	2	25000
	2	Aishik	HR	3	15000
	4	Saikia	HR	3	15000
*	NULL	NULL	NULL	NULL	NULL

(v) Display the empno and name of all employees whose salary is between 2000 and 5000.

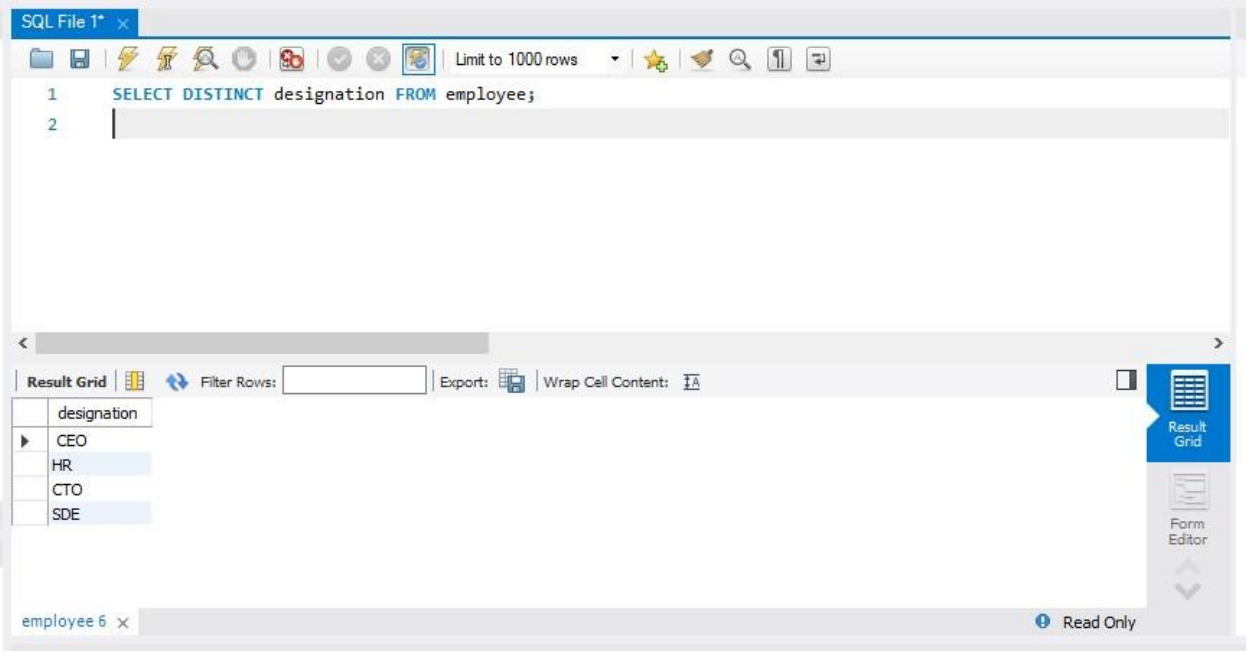
Ans: SELECT emp_no, emp_name FROM employee WHERE salary >= 2000 AND salary <= 5000;



```sql
SELECT emp_no, emp_name FROM employee WHERE salary >= 2000 AND salary <= 5000;
```

emp_no	emp_name
NULL	NULL

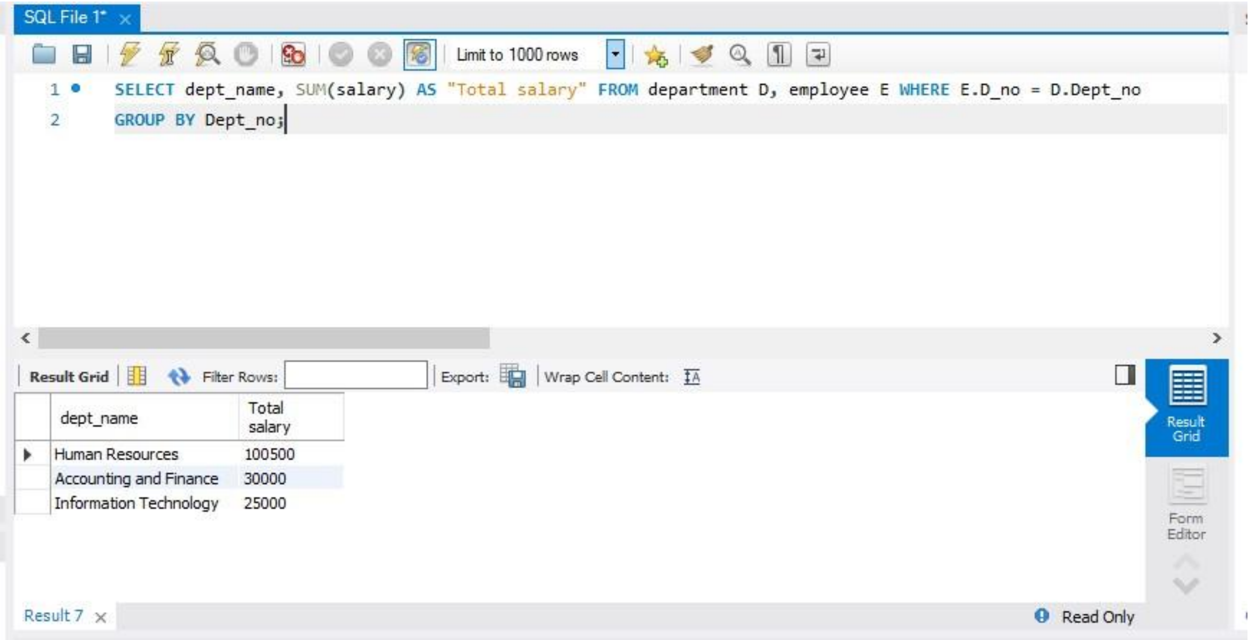
(vi) Display all designations without duplicate values.
Ans: SELECT DISTINCT designation FROM employee;



```sql
SELECT DISTINCT designation FROM employee;
```

designation
CEO
HR
CTO
SDE

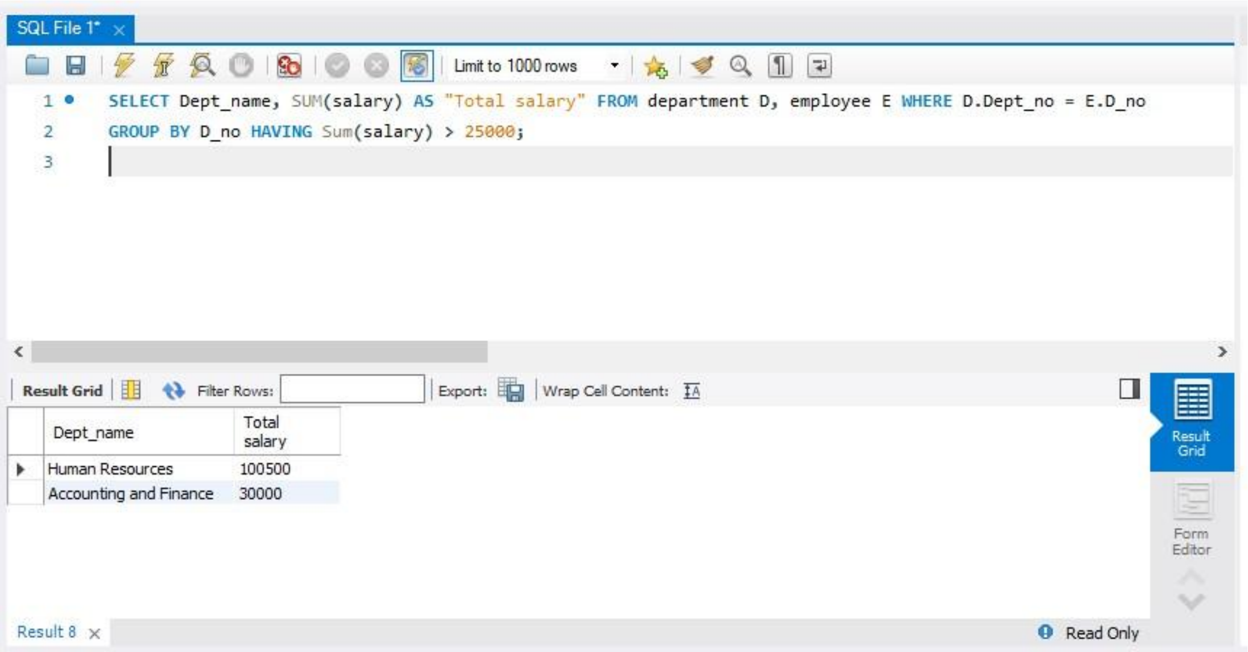
(vii) Display the dept name and total salary of employees of each department.
Ans: SELECT dept_name, SUM(salary) AS "Total salary" FROM department D, employee E WHERE E.D_no = D.Dept_no GROUP BY Dept_no;



```
SELECT dept_name, SUM(salary) AS "Total salary" FROM department D, employee E WHERE E.D_no = D.Dept_no
GROUP BY Dept_no;
```

dept_name	Total salary
Human Resources	100500
Accounting and Finance	30000
Information Technology	25000

(xi) Select the department that has total salary paid for its employees more than 25000.
Ans: SELECT Dept_name, SUM(salary) AS "Total salary" FROM department D, employee E WHERE D.Dept_no = E.D_no GROUP BY D_no HAVING Sum(salary) > 25000;



```
SELECT Dept_name, SUM(salary) AS "Total salary" FROM department D, employee E WHERE D.Dept_no = E.D_no
GROUP BY D_no HAVING Sum(salary) > 25000;

```

Dept_name	Total salary
Human Resources	100500
Accounting and Finance	30000

OBJECTIVES:

- To understand how to insert, update and delete data from within a table.
- To learn how to retrieve data from a table using the SELECT statement.

THEORY:

1. INSERT INSERT INTO tablename VALUES (value1, value2, ..., valuen).

2. UPDATE

Update <table> SET <field1> = <value1> = <value1> AND<field2>= <value2> WHERE <conditions>

3.DELETE

DELETE FROM <table> WHERE <condition>

4.SELECT

(a) Retrieve from all columns

SELECT * FROM<table>

(b) Retrieve from selected columns

SELECT <column 1> <column 2> FROM<table>

(c) Retrieve unique values

SELECT DISTINCT <column name> FROm <table>

(d) REtrueve data satisfying a given con dition

SELECT <columns> FROM <tables>WHERE <condition>

PROCEDURE:

(i) Use insert command

(ii) Use Select command

(iii) Use Select command with where condition

(iv)Use Select command with order by clause

(v) Use Select command with operators

(vi)Use Select command with DISTINCT keyword

(vii) Use Select command with group by clause

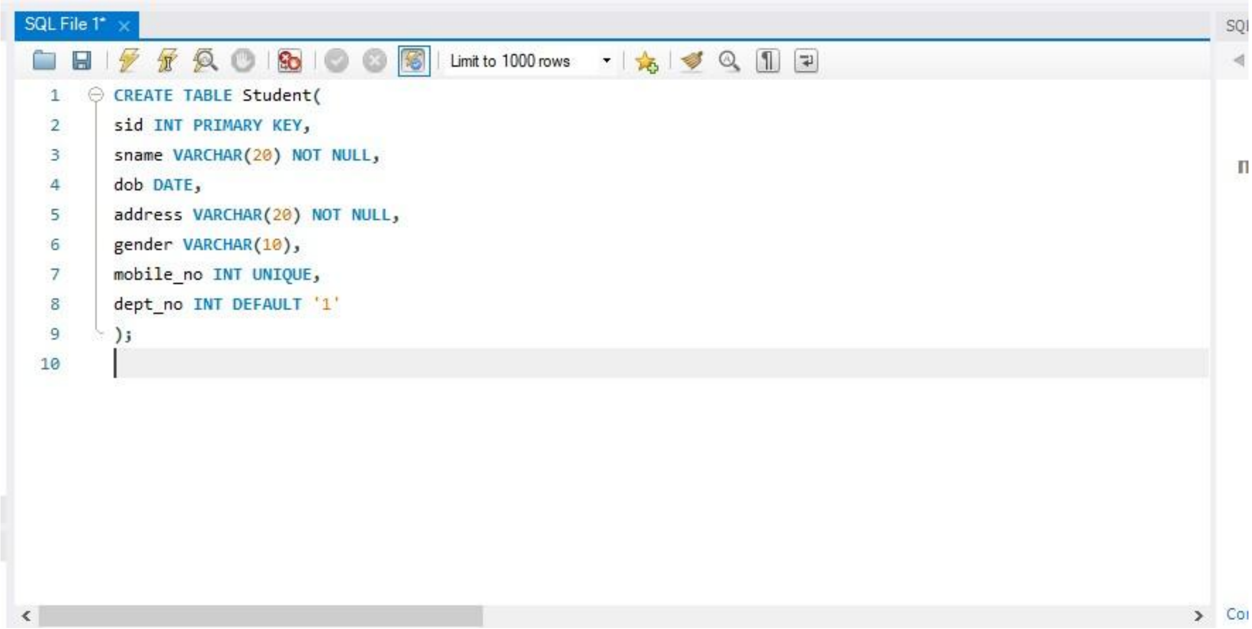
(viii) Use Update command

(ix)Use Update command

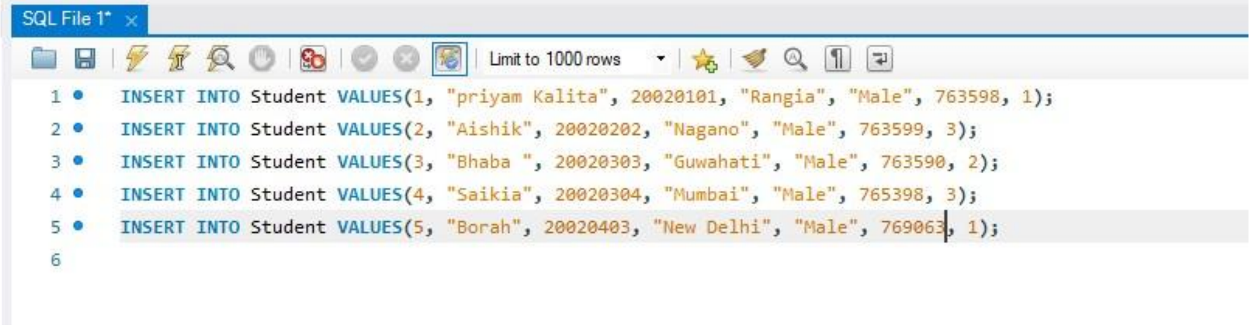
(x) Use Delete command

(xi)Use select command with group by and having clause

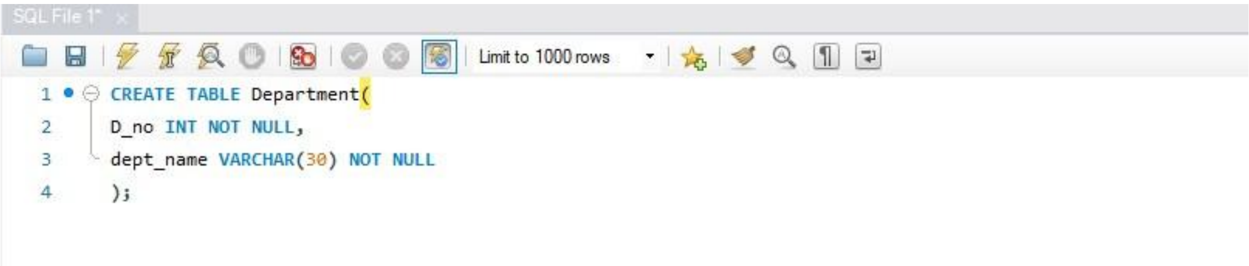
Problems: 1. Consider the database for a college. Write the query for the following. (i) Insert at least 5 tuples into each table.



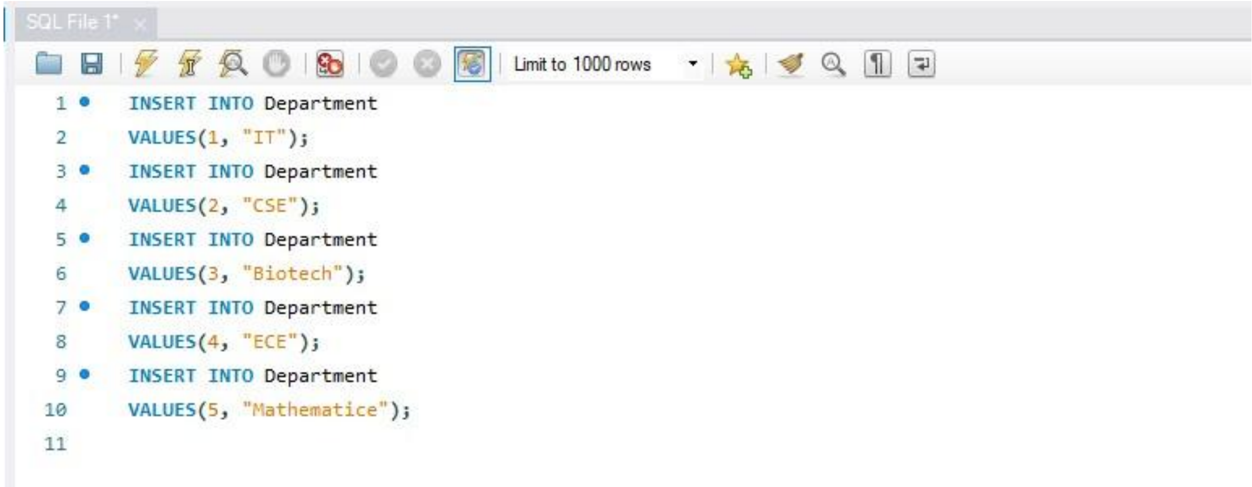
```sql
CREATE TABLE Student(
sid INT PRIMARY KEY,
sname VARCHAR(20) NOT NULL,
dob DATE,
address VARCHAR(20) NOT NULL,
gender VARCHAR(10),
mobile_no INT UNIQUE,
dept_no INT DEFAULT '1'
);
```



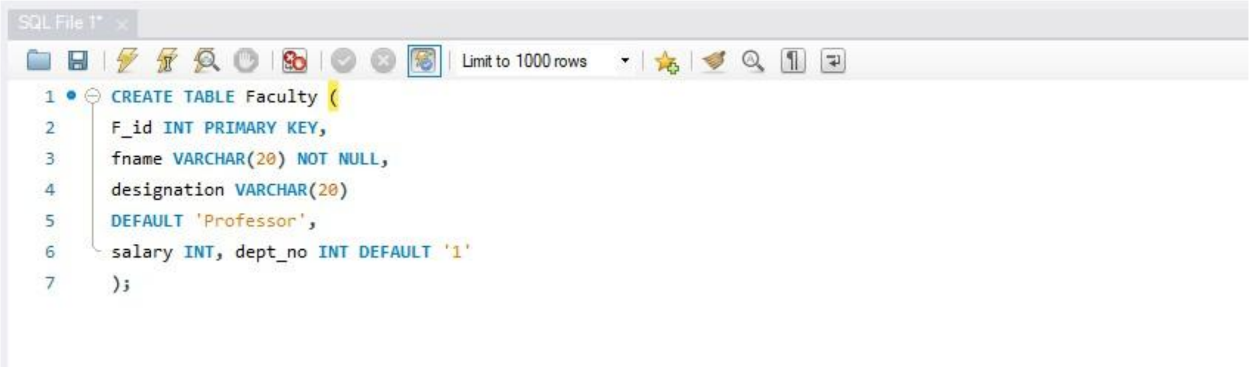
```sql
INSERT INTO Student VALUES(1, "priyam Kalita", 20020101, "Rangia", "Male", 763598, 1);
INSERT INTO Student VALUES(2, "Aishik", 20020202, "Nagano", "Male", 763599, 3);
INSERT INTO Student VALUES(3, "Bhaba ", 20020303, "Guwahati", "Male", 763590, 2);
INSERT INTO Student VALUES(4, "Saikia", 20020304, "Mumbai", "Male", 765398, 3);
INSERT INTO Student VALUES(5, "Borah", 20020403, "New Delhi", "Male", 769063, 1);
```



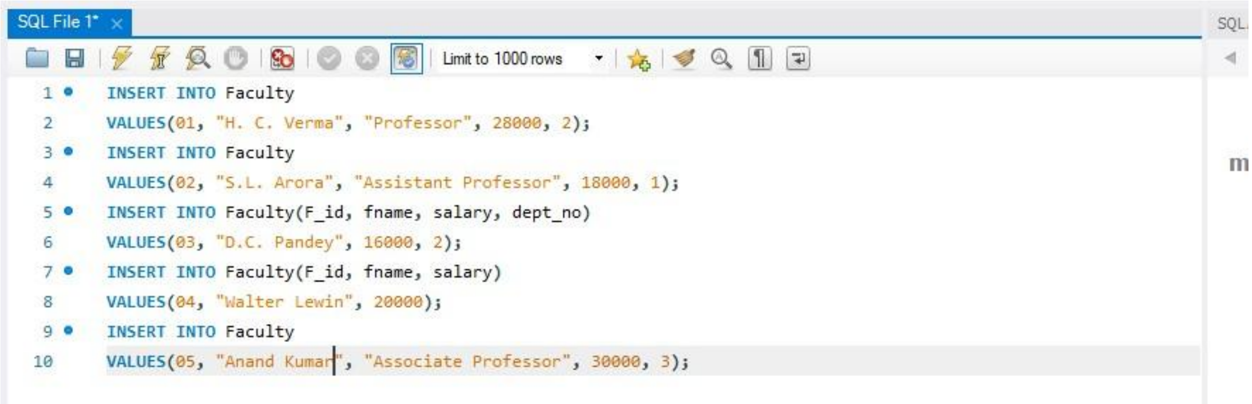
```sql
CREATE TABLE Department(
D_no INT NOT NULL,
dept_name VARCHAR(30) NOT NULL
);
```



```sql
INSERT INTO Department
VALUES(1, "IT");
INSERT INTO Department
VALUES(2, "CSE");
INSERT INTO Department
VALUES(3, "Biotech");
INSERT INTO Department
VALUES(4, "ECE");
INSERT INTO Department
VALUES(5, "Mathematice");
```



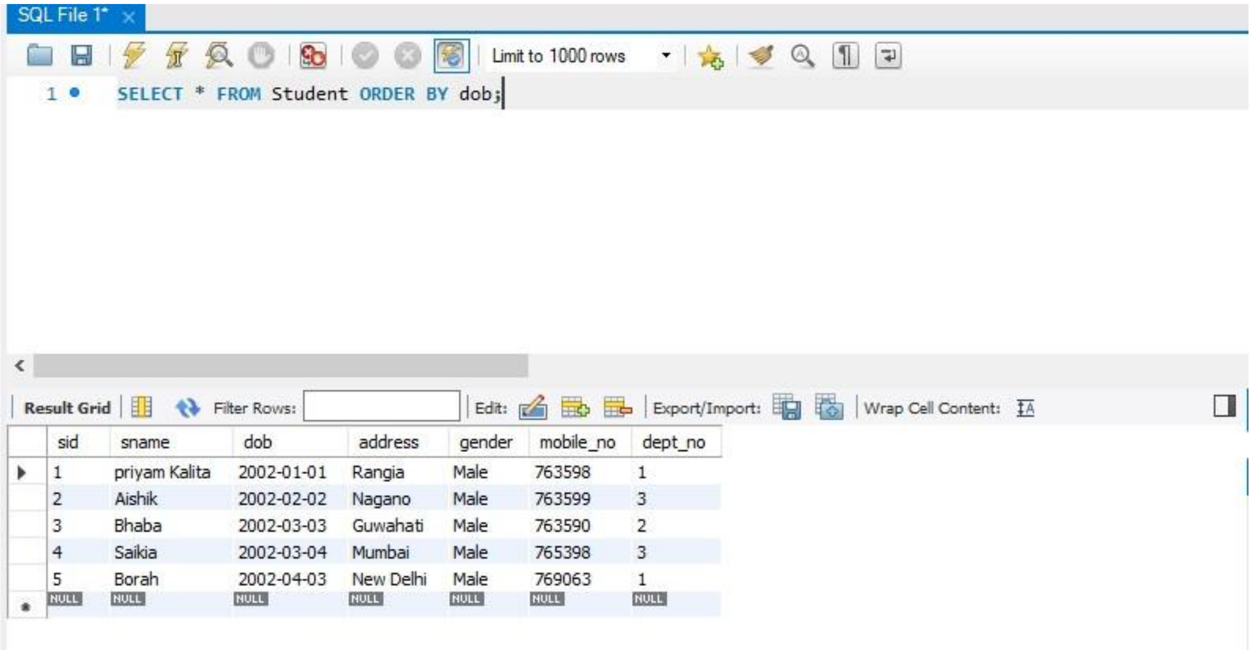
```sql
CREATE TABLE Faculty (
F_id INT PRIMARY KEY,
fname VARCHAR(20) NOT NULL,
designation VARCHAR(20)
DEFAULT 'Professor',
salary INT, dept_no INT DEFAULT '1'
);
```



```sql
INSERT INTO Faculty
VALUES(01, "H. C. Verma", "Professor", 28000, 2);
INSERT INTO Faculty
VALUES(02, "S.L. Arora", "Assistant Professor", 18000, 1);
INSERT INTO Faculty(F_id, fname, salary, dept_no)
VALUES(03, "D.C. Pandey", 16000, 2);
INSERT INTO Faculty(F_id, fname, salary)
VALUES(04, "Walter Lewin", 20000);
INSERT INTO Faculty
VALUES(05, "Anand Kumar", "Associate Professor", 30000, 3);
```

(ii) List the details of students in the ascending order of date of birth

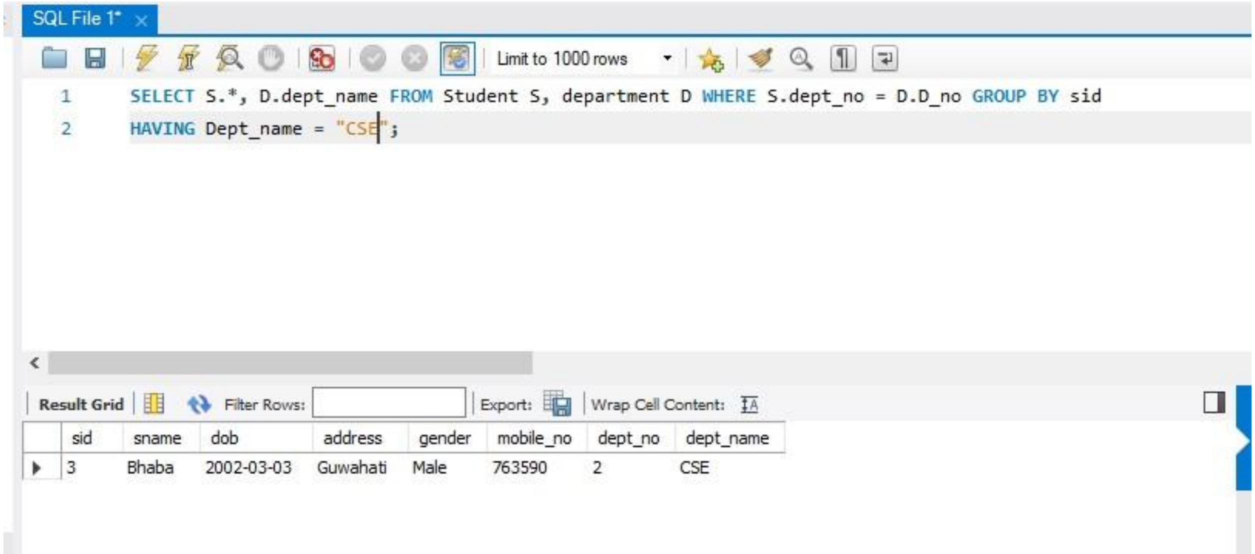
Ans: SELECT * FROM Student ORDER BY dob;



```sql
SELECT * FROM Student ORDER BY dob;
```

sid	sname	dob	address	gender	mobile_no	dept_no
1	priyam Kalita	2002-01-01	Rangia	Male	763598	1
2	Aishik	2002-02-02	Nagano	Male	763599	3
3	Bhaba	2002-03-03	Guwahati	Male	763590	2
4	Saikia	2002-03-04	Mumbai	Male	765398	3
5	Borah	2002-04-03	New Delhi	Male	769063	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

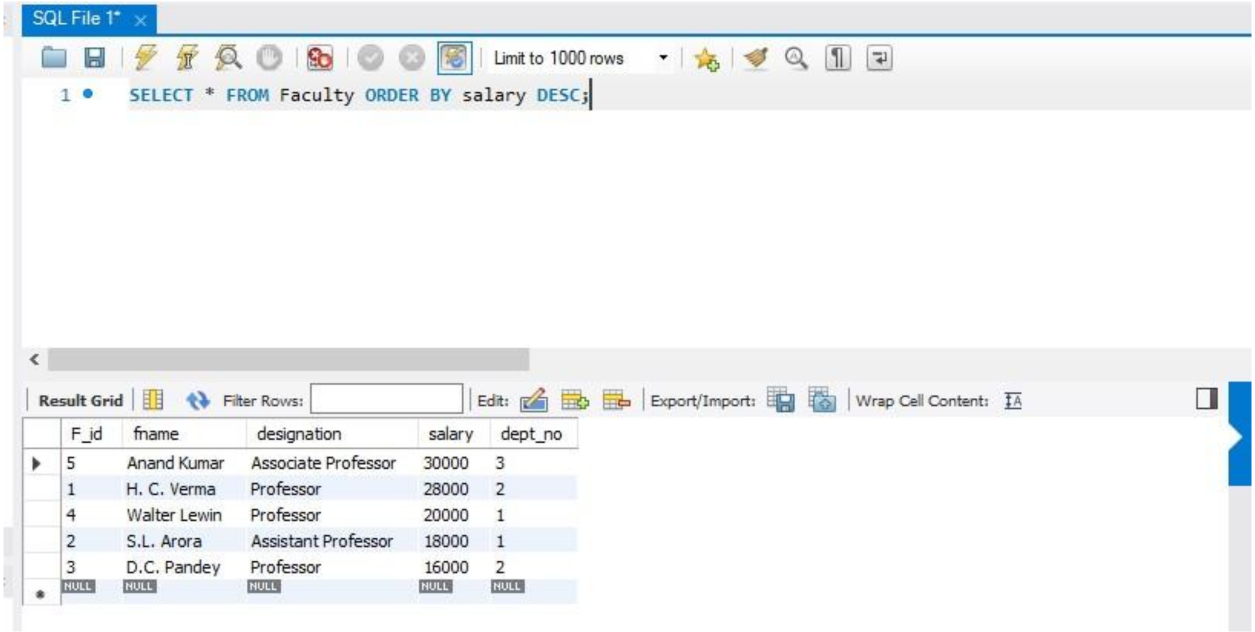
(iii) Display the details of students from computer department Ans: SELECT S.*, D.dept_name FROM Student S, department D WHERE S.dept_no = D.D_no GROUP BY sid HAVING Dept_name = "CSE";



```sql
SELECT S.*, D.dept_name FROM Student S, department D WHERE S.dept_no = D.D_no GROUP BY sid
HAVING Dept_name = "CSE";
```

sid	sname	dob	address	gender	mobile_no	dept_no	dept_name
3	Bhaba	2002-03-03	Guwahati	Male	763590	2	CSE

(iv) List the faculties in the descending order of salary
Ans: SELECT * FROM Faculty ORDER BY salary DESC;

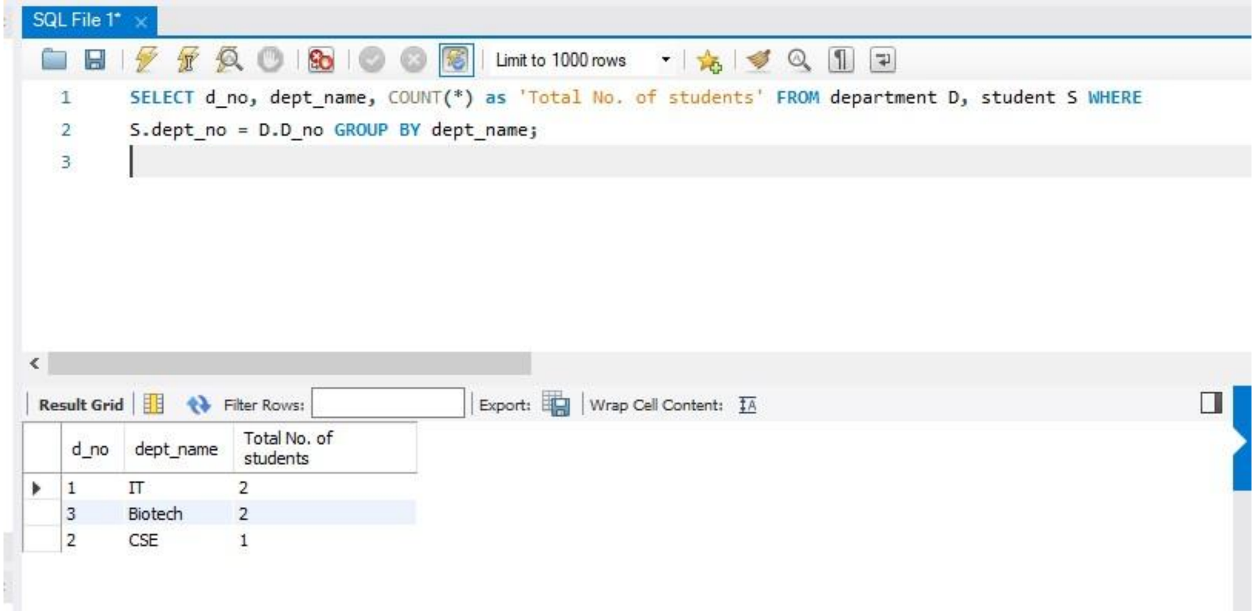


```sql
SELECT * FROM Faculty ORDER BY salary DESC;
```

F_id	fname	designation	salary	dept_no
5	Anand Kumar	Associate Professor	30000	3
1	H. C. Verma	Professor	28000	2
4	Walter Lewin	Professor	20000	1
2	S.L. Arora	Assistant Professor	18000	1
3	D.C. Pandey	Professor	16000	2
NULL	NULL	NULL	NULL	NULL

(v) Display the total number of students in each department

Ans: SELECT d_no, dept_name, COUNT(*) as 'Total No. of students' FROM department D, student S WHERE S.dept_no = D.D_no GROUP BY dept_name;

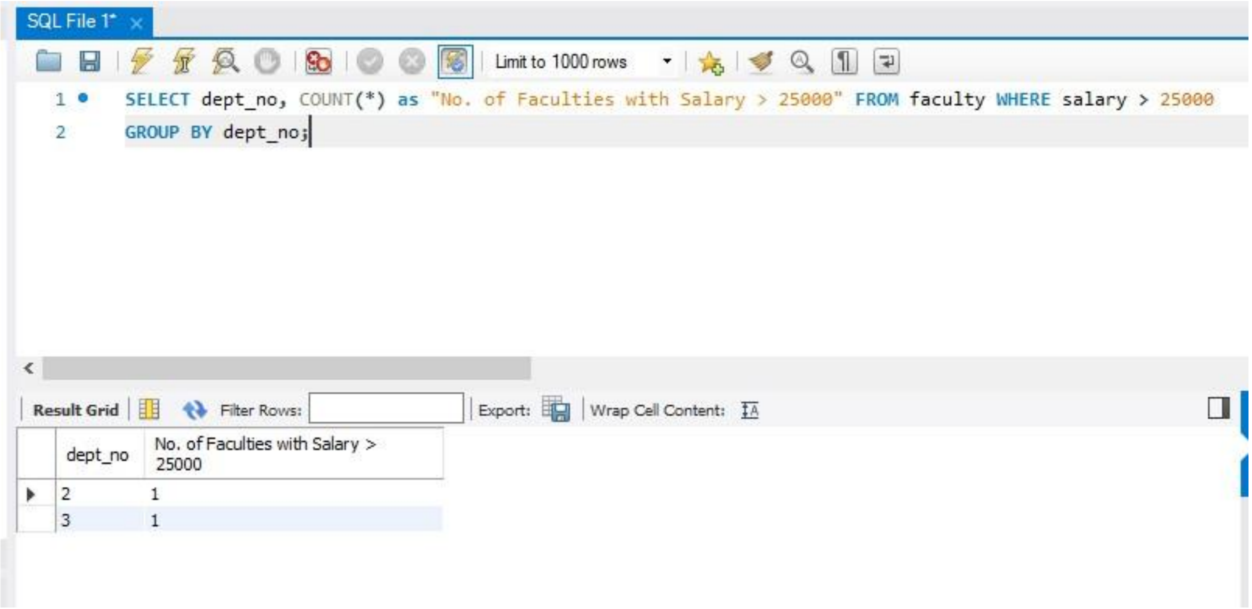


```sql
SELECT d_no, dept_name, COUNT(*) as 'Total No. of students' FROM department D, student S WHERE
S.dept_no = D.D_no GROUP BY dept_name;
```

d_no	dept_name	Total No. of students
1	IT	2
3	Biotech	2
2	CSE	1

(vi) Display the total number of faculties in each department with salary greater than 25000.

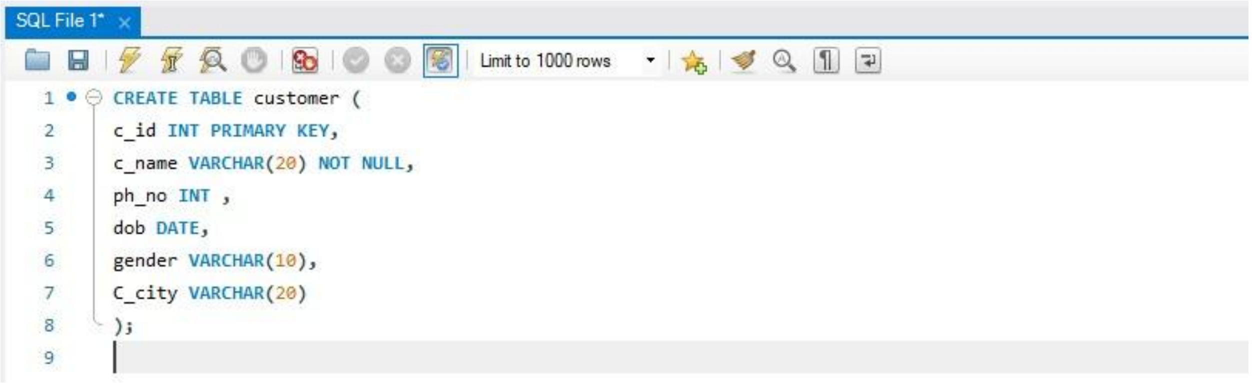
Ans: SELECT dept_no, COUNT(*) as "No. of Faculties with Salary > 25000" FROM faculty WHERE salary > 25000 GROUP BY dept_no;



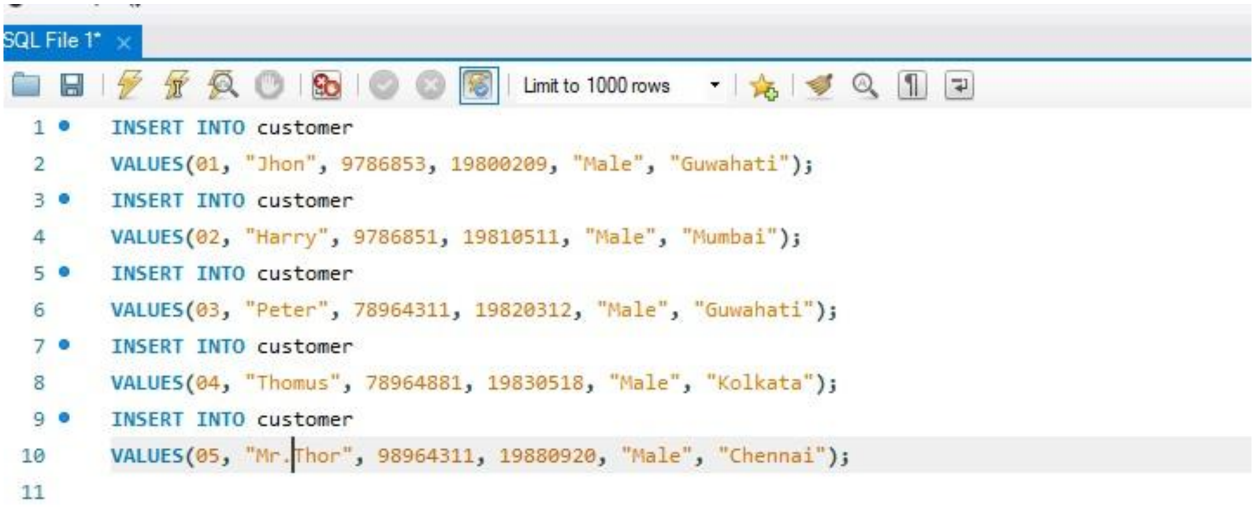
```
SELECT dept_no, COUNT(*) as "No. of Faculties with Salary > 25000" FROM faculty WHERE salary > 25000
GROUP BY dept_no;
```

dept_no	No. of Faculties with Salary > 25000
2	1
3	1

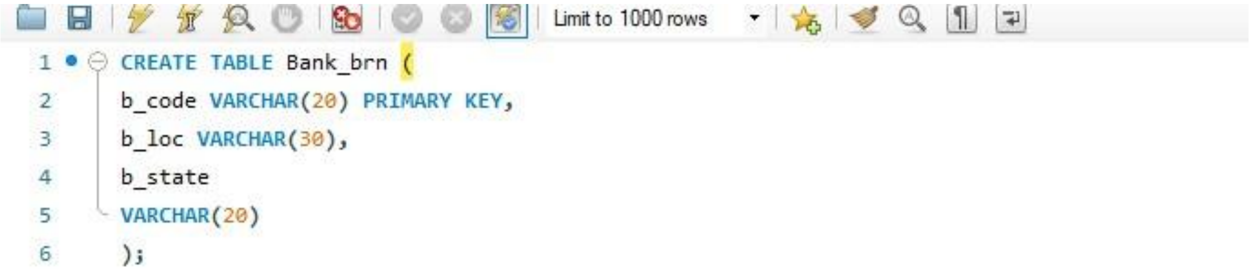
2. Consider the database for a banking enterprise.
Write the queries for the below questions.
(i) Insert at least 5 tuples in each table



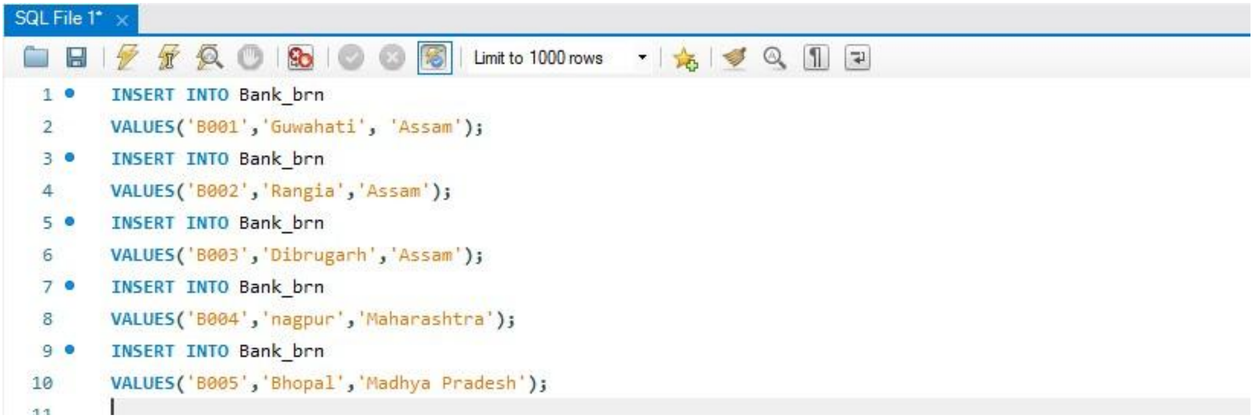
```
CREATE TABLE customer (
c_id INT PRIMARY KEY,
c_name VARCHAR(20) NOT NULL,
ph_no INT ,
dob DATE,
gender VARCHAR(10),
C_city VARCHAR(20)
);
```



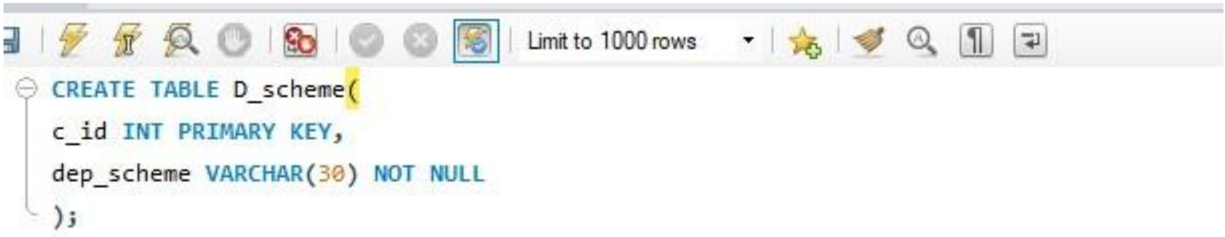
```sql
INSERT INTO customer
VALUES(01, "Jhon", 9786853, 19800209, "Male", "Guwahati");
INSERT INTO customer
VALUES(02, "Harry", 9786851, 19810511, "Male", "Mumbai");
INSERT INTO customer
VALUES(03, "Peter", 78964311, 19820312, "Male", "Guwahati");
INSERT INTO customer
VALUES(04, "Thomus", 78964881, 19830518, "Male", "Kolkata");
INSERT INTO customer
VALUES(05, "Mr.Thor", 98964311, 19880920, "Male", "Chennai");
```



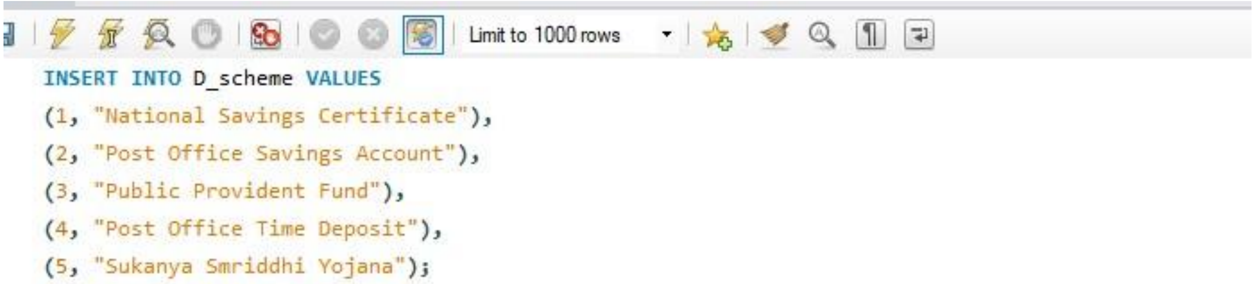
```sql
CREATE TABLE Bank_brn (
b_code VARCHAR(20) PRIMARY KEY,
b_loc VARCHAR(30),
b_state
VARCHAR(20)
 );
```



```sql
INSERT INTO Bank_brn
VALUES('B001','Guwahati', 'Assam');
INSERT INTO Bank_brn
VALUES('B002','Rangia','Assam');
INSERT INTO Bank_brn
VALUES('B003','Dibrugarh','Assam');
INSERT INTO Bank_brn
VALUES('B004','nagpur','Maharashtra');
INSERT INTO Bank_brn
VALUES('B005','Bhopal','Madhya Pradesh');
```



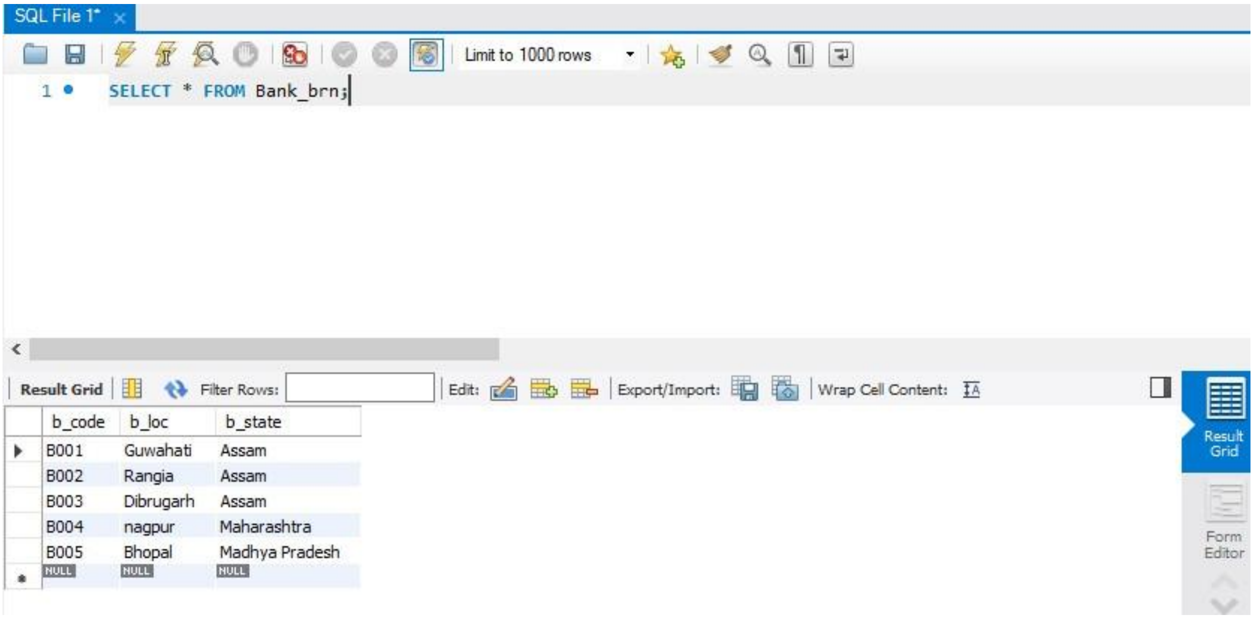
```sql
CREATE TABLE D_scheme(
c_id INT PRIMARY KEY,
dep_scheme VARCHAR(30) NOT NULL
);
```



```
INSERT INTO D_scheme VALUES
(1, "National Savings Certificate"),
(2, "Post Office Savings Account"),
(3, "Public Provident Fund"),
(4, "Post Office Time Deposit"),
(5, "Sukanya Smriddhi Yojana");
```

(ii) Display the branch details

Ans: SELECT * FROM Bank_brn;

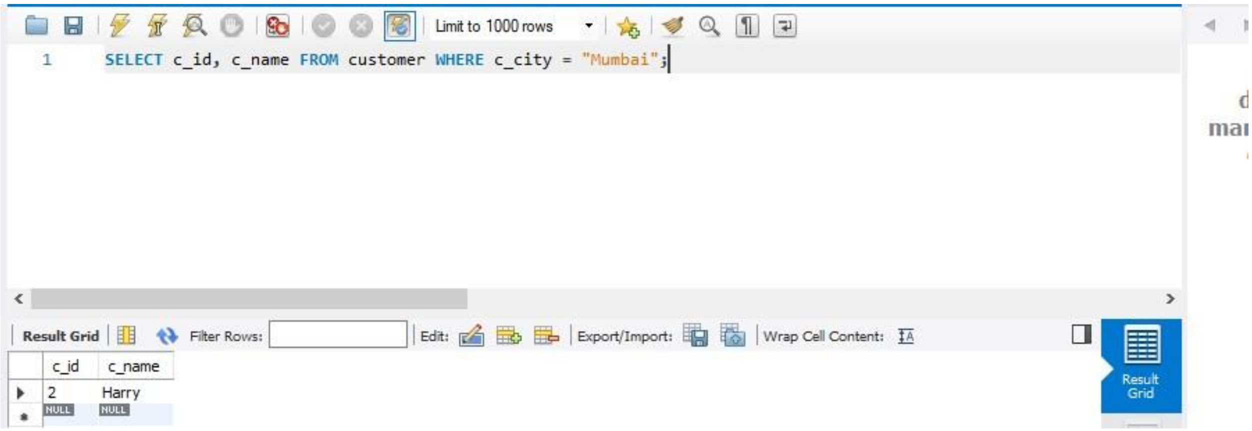


```
SELECT * FROM Bank_brn;
```

b_code	b_loc	b_state
B001	Guwahati	Assam
B002	Rangia	Assam
B003	Dibrugarh	Assam
B004	nagpur	Maharashtra
B005	Bhopal	Madhya Pradesh
NULL	NULL	NULL

(iii)List the customers of 'Mumbai' city

Ans: SELECT c_id, c_name FROM customer WHERE c_city = "Mumbai";

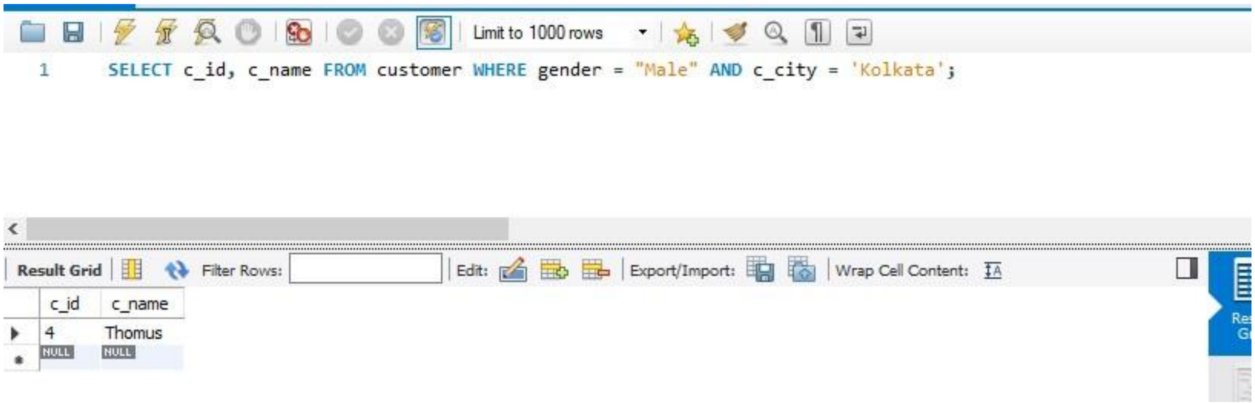


```
SELECT c_id, c_name FROM customer WHERE c_city = "Mumbai";
```

c_id	c_name
2	Harry
NULL	NULL

(iv) List the male customers of 'Kolkata'city

Ans: SELECT c_id, c_name FROM customer WHERE gender = "Male" AND c_city = 'Kolkata'

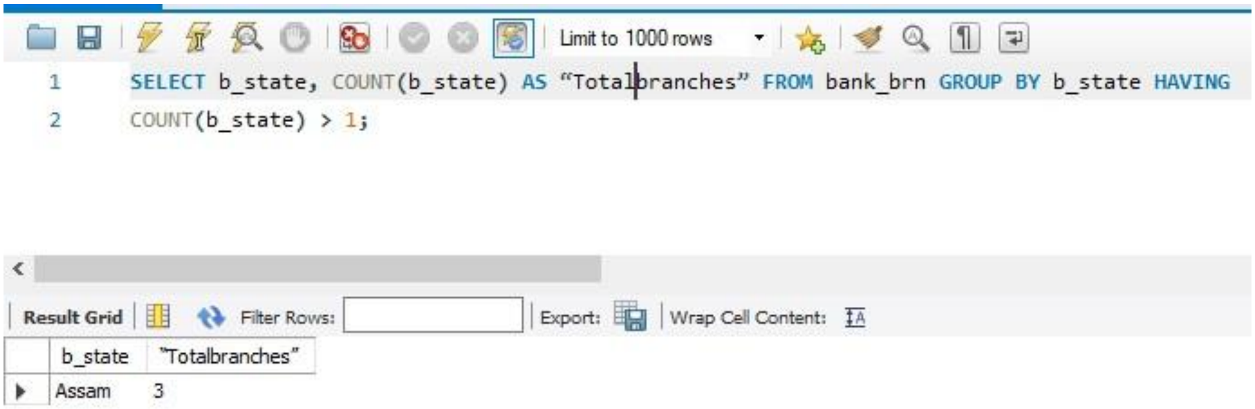


```
SELECT c_id, c_name FROM customer WHERE gender = "Male" AND c_city = 'Kolkata';
```

c_id	c_name
4	Thomus
NULL	NULL

(v) List the state having more than one branch.

Ans: SELECT b_state, COUNT(b_state) AS “Total no. of branches” FROM bank_brn GROUP BY b_state HAVING COUNT(b_state) > 1;

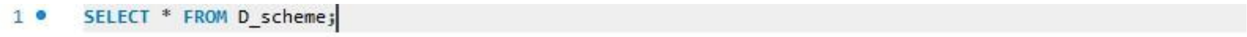


```
SELECT b_state, COUNT(b_state) AS “Totalbranches” FROM bank_brn GROUP BY b_state HAVING
COUNT(b_state) > 1;
```

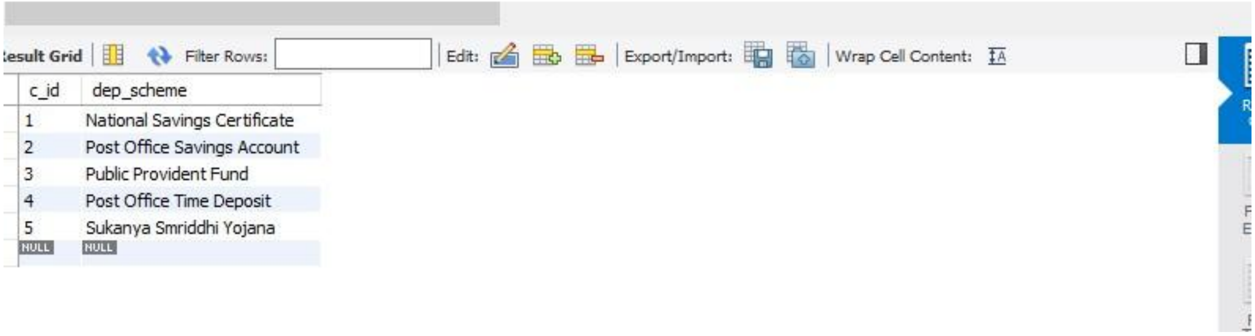
b_state	“Totalbranches”
Assam	3

(vi) List the deposit schemes provided by the bank to the customers

Ans: SELECT * FROM D_scheme;



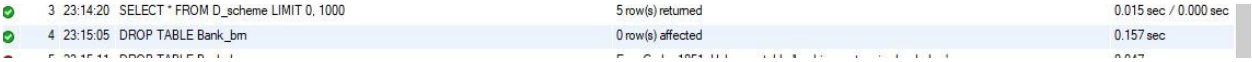
```
SELECT * FROM D_scheme;
```



c_id	dep_scheme
1	National Savings Certificate
2	Post Office Savings Account
3	Public Provident Fund
4	Post Office Time Deposit
5	Sukanya Smriddhi Yojana
NULL	NULL

(vii) Delete the entire content of any table

Ans: DROP TABLE Bank_brn;



```
3  23:14:20  SELECT * FROM D_scheme LIMIT 0, 1000    5 row(s) returned     0.015 sec / 0.000 sec
4  23:15:05  DROP TABLE Bank_brn                     0 row(s) affected     0.157 sec
```

EXPERIMENT NO 4 Sub Queries and Joins

AIM: Consider the database for the organization and Write the queries for the following

(i) Display the empno, name, and salaries for employees whose average salary is higher than the average salary of the organization.

(ii) Display the details of employees whose salary is equal to the minimum salary of organisation.

(iii) Display all the employees whose designation is same as that of 'Arun'

(iv) display the empno and name of employees who earn more than any Employee in dept 1.

(v) Display the empno, name, departments that the departments are same in both the emp and dept

(vi)Display the employee details by implementing left inner join (vii) Display employee details by implementing a right outer join

OBJECTIVES: To understand sub queries and join in Mysql.

THEORY:

NESTED QUERIES:

A sub query is a query within a query. These sub queries can reside in the WHERE clause, the FROM clause, or the SELECT clause. The first query in the SQL statement is known as the outer query. The query inside the SQL statement is known as the inner query. The inner query is executed first. The output of an inner query is used as the input for the outer query. The entire SQL statement is sometimes referred to as a nested query.

JOINS: MySQL JOINS are used to retrieve data from multiple tables. A MySQL JOIN is performed whenever two or more tables are joined in a SQL statement.

There are different types of MySQL joins:

- MySQL INNER JOIN (or sometimes called simple join) MySQL LEFT
- OUTER JOIN (or sometimes called LEFT JOIN)
- MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)

INNER JOIN (simple join): MySQL INNER JOINS return all rows from multiple tables where the join condition is met. Syntax: Select columns from table1 Inner join table2 On table1.column=table2.column;

INNER JOIN (simple join): MySQL INNER JOINS return all rows from multiple tables where the join condition is met.

Syntax: Select columns from table1 Inner join table2 On table1.column=table2.column;

LEFT OUTER JOIN

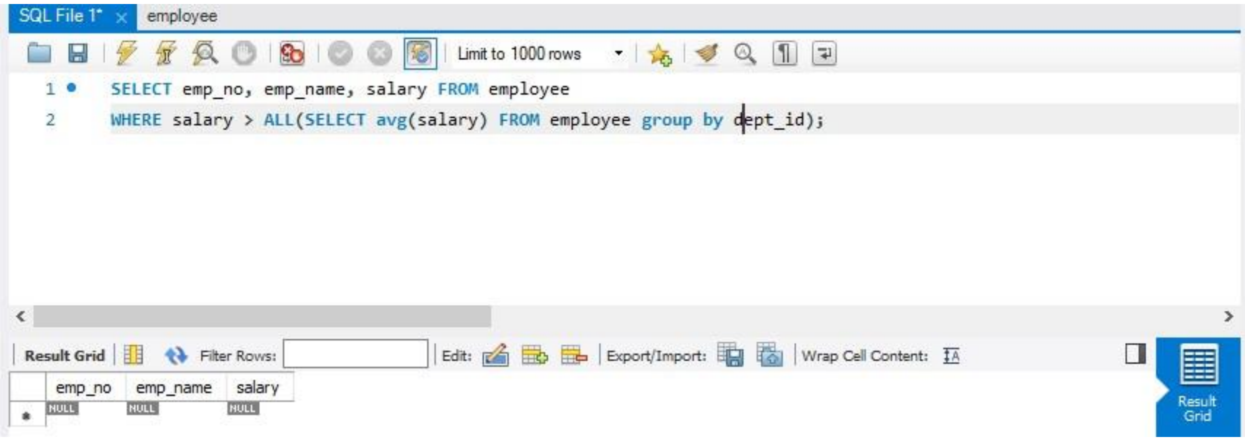
Another type of join is called a MySQL LEFT OUTER JOIN. This type of join returns all rows from the LEFT-hand table specified in the ON condition and only those rows from the other table where the joined fields are equal.

Syntax:

Select columns from table 1
Right join table2
On table1.column=table2.column;

CODE AND OUTPUT:

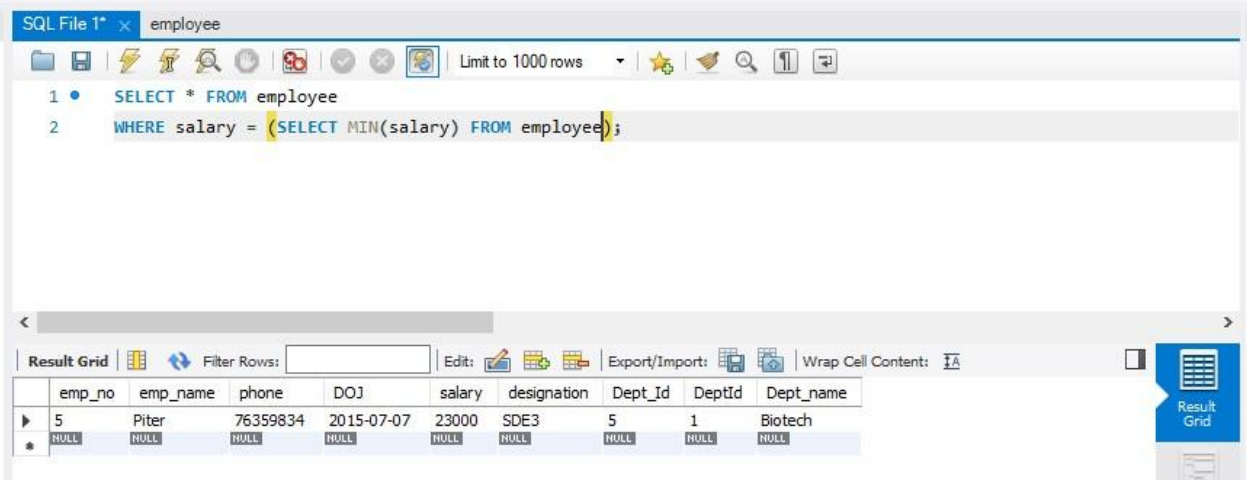
(1)



```sql
SELECT emp_no, emp_name, salary FROM employee
WHERE salary > ALL(SELECT avg(salary) FROM employee group by dept_id);
```

```
  emp_no  emp_name  salary
* NULL    NULL      NULL
```

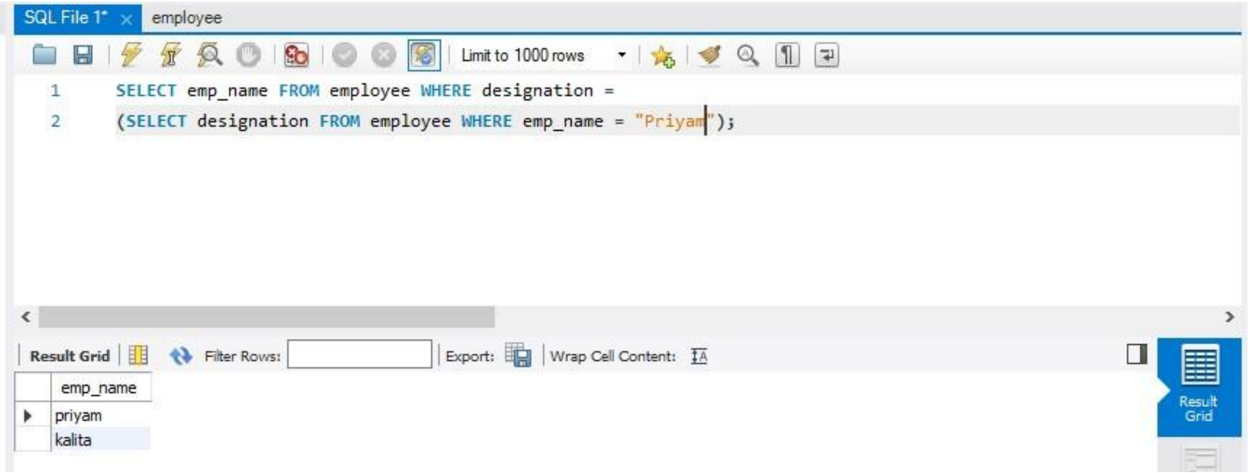
(2)



```sql
SELECT * FROM employee
WHERE salary = (SELECT MIN(salary) FROM employee);
```

	emp_no	emp_name	phone	DOJ	salary	designation	Dept_Id	DeptId	Dept_name
▶	5	Piter	76359834	2015-07-07	23000	SDE3	5	1	Biotech
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

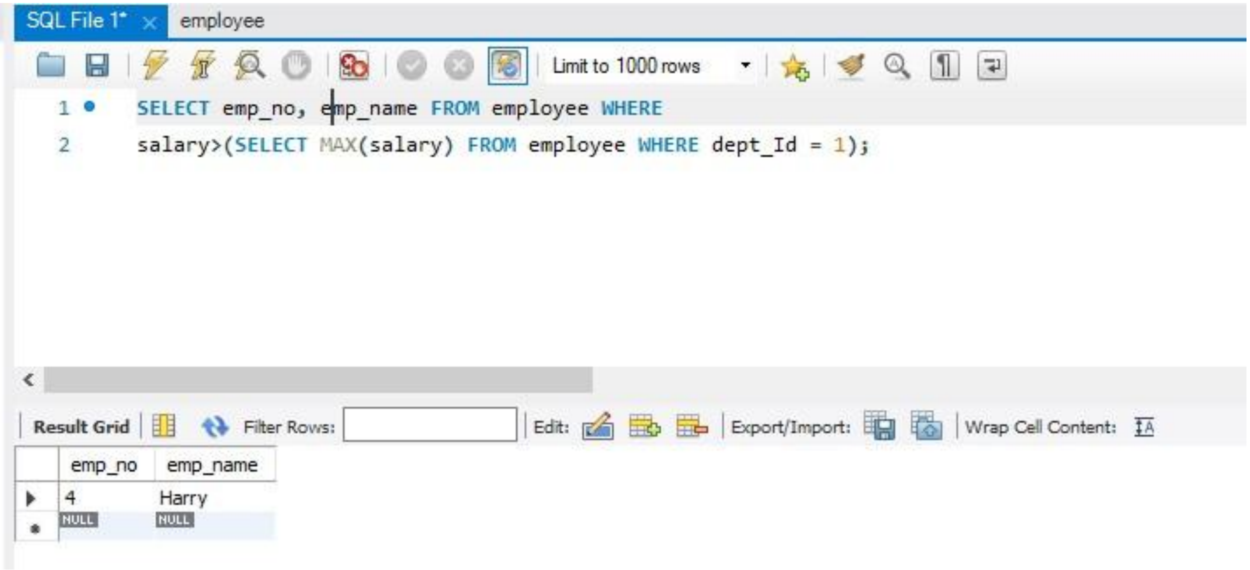
(3)



```sql
SELECT emp_name FROM employee WHERE designation =
(SELECT designation FROM employee WHERE emp_name = "Priyam");
```

	emp_name
▶	priyam
	kalita

(4)



```sql
SELECT emp_no, emp_name FROM employee WHERE
salary>(SELECT MAX(salary) FROM employee WHERE dept_Id = 1);
```

emp_no	emp_name
4	Harry
NULL	NULL

RESULT: The nested queries and joins are executed successfully.

EXPERIMENT NO 5 : VIEWS

AIM: Write the queries for the following.

(i) Create a view emp from employee such that it contains only emp_no and emp_name and department.

(ii) Create a view dept from department with only dept_no and location.

(iii) Create a view that contains the details of employees who are managers only.

(iv) drop the views. OBJECTIVES: To understand views in Mysql

OBJECTIVES: To understand views in Mysql

THEORY:
A view is the tailored presentation of data contained in one or more table and can also be said as restricted view to the data in the tables. A view is a “virtual table” or a “stored query” which takes the output of a query and treats it as a table. The table upon which a view is created is called as base table. A view is a logical table based on a table or another view. A view contains no data of its own but is like a window through which data from tables can be viewed or changed. The tables on which a view is based are called base tables. The view is stored as a SELECT statement in the data dictionary

Advantages of a view:
(a). Additional level of table security.
(b). Hides data complexity.
(c). Simplifies the usage by combining multiple tables into a single table

Creating and dropping view:

Syntax:

```
Create or replace view view_name AS SELECT
        column_name(s) FROM table_name WHERE condition;
Drop view <view name> ;
```

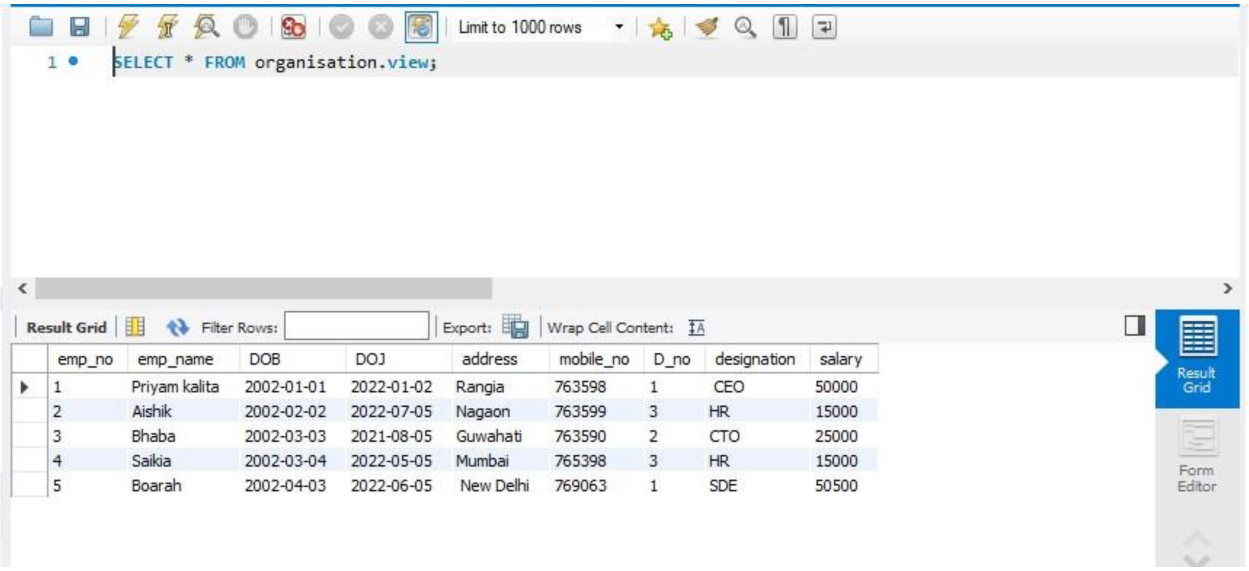
PROCEDURE:
(1) Create the employee table
(2) Create the view

(3) display the content of view

Result: Thus the views are created successfully.

Problems:

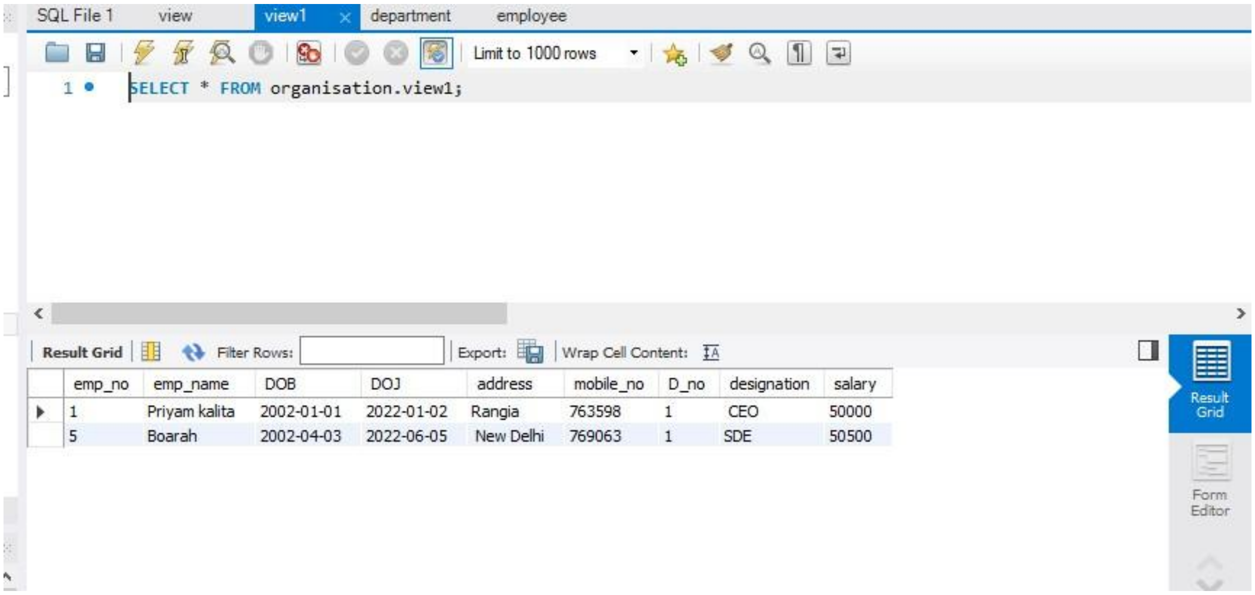
(i) View



```
SELECT * FROM organisation.view;
```

emp_no	emp_name	DOB	DOJ	address	mobile_no	D_no	designation	salary
1	Priyam kalita	2002-01-01	2022-01-02	Rangia	763598	1	CEO	50000
2	Aishik	2002-02-02	2022-07-05	Nagaon	763599	3	HR	15000
3	Bhaba	2002-03-03	2021-08-05	Guwahati	763590	2	CTO	25000
4	Saikia	2002-03-04	2022-05-05	Mumbai	765398	3	HR	15000
5	Boarah	2002-04-03	2022-06-05	New Delhi	769063	1	SDE	50500

(ii) View1



```
SELECT * FROM organisation.view1;
```

emp_no	emp_name	DOB	DOJ	address	mobile_no	D_no	designation	salary
1	Priyam kalita	2002-01-01	2022-01-02	Rangia	763598	1	CEO	50000
5	Boarah	2002-04-03	2022-06-05	New Delhi	769063	1	SDE	50500

```
(v) DROP VIEW view;
    DROP VIEW view1;
```

EXPERIMENT NO:6 PROCEDURE

AIM:

Write a procedure which increases the salary of an employee. It accepts an employee number and salary increase amount. It uses the employee number to find the current salary from the EMPLOYEE table and update the salary.

OBJECTIVES

To understand the procedure in Mysql
THEORY

PROCEDURE:
In MySQL, a procedure is a stored program that you can pass parameters into. It does not return a value like a function does.
Syntax

Create procedure name (parameter data type, parameter data type...)

Begin
Declaration section
Executable _section

End;

Procedure name

The name to assign to this procedure in MySQL.
Parameter

DBMS LAB MANUAL

When creating a procedure, there are three types of parameters that can be declared:

1. IN - The parameter can be referenced by the procedure. The value of the parameter cannot be overwritten by the procedure.

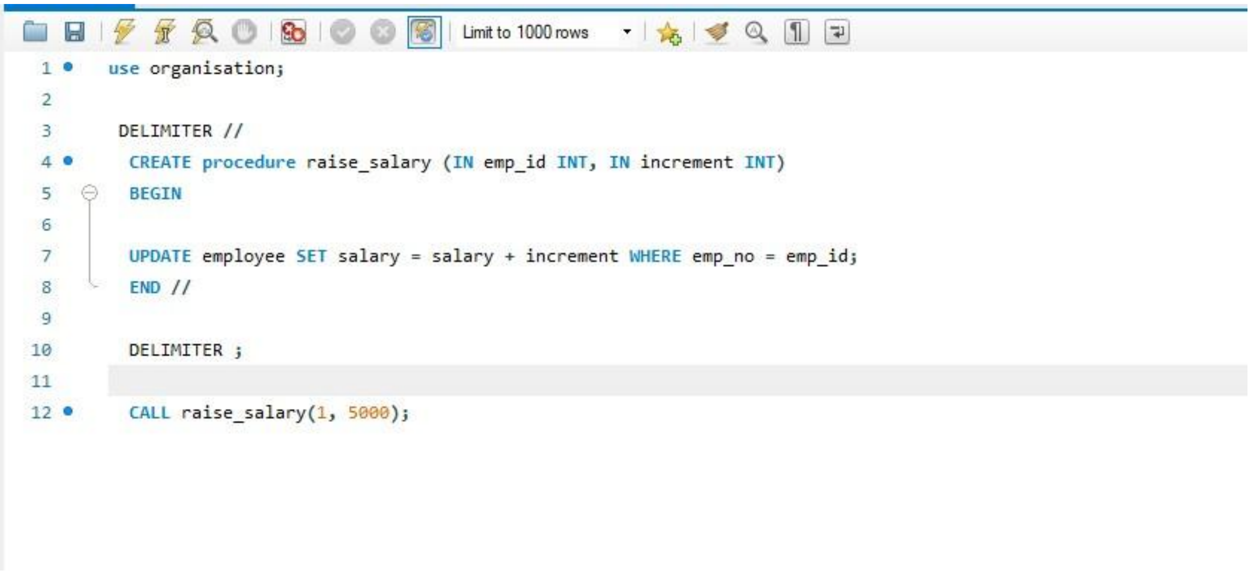
2. OUT - The parameter cannot be referenced by the procedure, but the value of the parameter can be overwritten by the procedure.

3. IN OUT - The parameter can be referenced by the procedure and the value of the parameter can be overwritten by the procedure.

Declaration section
The place in the procedure where you declare local variables.

Executable section
The place in the procedure where you enter the code for the procedure.

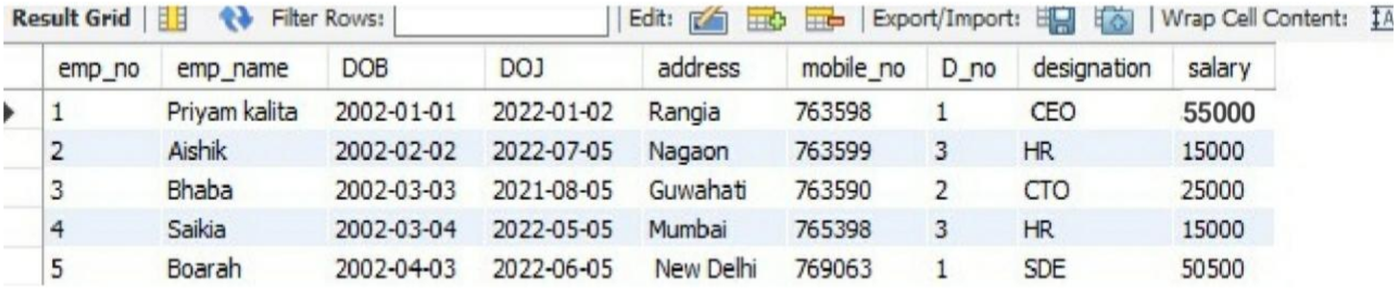
PROCEDURE:



```
use organisation;

 DELIMITER //
  CREATE procedure raise_salary (IN emp_id INT, IN increment INT)
  BEGIN

  UPDATE employee SET salary = salary + increment WHERE emp_no = emp_id;
  END //

  DELIMITER ;

  CALL raise_salary(1, 5000);
```

OUTPUT:

Create a procedure that changed the salary of emp_no 1 to 55000 from 50000



emp_no	emp_name	DOB	DOJ	address	mobile_no	D_no	designation	salary
1	Priyam kalita	2002-01-01	2022-01-02	Rangia	763598	1	CEO	55000
2	Aishik	2002-02-02	2022-07-05	Nagaon	763599	3	HR	15000
3	Bhaba	2002-03-03	2021-08-05	Guwahati	763590	2	CTO	25000
4	Saikia	2002-03-04	2022-05-05	Mumbai	765398	3	HR	15000
5	Boarah	2002-04-03	2022-06-05	New Delhi	769063	1	SDE	50500