LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

Department of Computer Science & Engineering

20CS56 - DATABASE MANAGEMENT SYSTEMS LAB

Name of the Student: KOTA SAGAR

Registered Number: 20761A0595

Branch & Section: CSE & B/Sec

Academic Year: 2021 –22

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

CERTIFICATE

This is to certify that this is a bonafide record of the practical work done by Mr./Ms. KOTA SAGAR, bearing Regd. Num.: 20761A0595 of B.Tech 3rd Semester, CSE Branch, B Section in the 20CS56 - DATABASE MANAGEMENT SYSTEMS LAB during the Academic Year: 2021 – 22.

No. of Experiments/Modules held: 14

No. of Experiments Done: ……..

Date: ….. / …… / 2022 **Signature of the Faculty**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DBMS - LAB

1) Create a table STUDENT with appropriate data types and perform the following queries. Attributes are Roll number, student name, date of birth, branch and year of study.

```
mysql> create table student(Roll_number int primary key,student_name
varchar(20),
->date_of_birth date,branch varchar(5),year_of_study int);
Query OK, 0 rows affected (0.46 sec)
```

a) Insert 5 to 10 rows in a table?

```
mysql> insert into
student(Roll_number,student_name,date_of_birth,branch,year_of_study)
values(222,"venkat","1991-09-26","cse",2020),
->(333,'siva','1990-04-10','AIDS',2021),(111,"srikanth","1990-03-
16","cse",2020),(444,'Rajani','1980-05-12','IT',2010),
->(555,'Sindhu','1993-03-26','ECE',2017),(666,'Nayana','1995-05-
05','AIML',2002);
Query OK, 5 rows affected (0.09 sec)

mysql> select * from student;
+-------------+--------------+---------------+--------+---------------+
| Roll_number | student_name | date_of_birth | branch | year_of_study |
+-------------+--------------+---------------+--------+---------------+
|         111| srikanth     | 1990-03-16    | cse    |          2020 |
|         222| venkat       | 1991-09-26    | cse    |          2020 |
|         333| siva         | 1990-04-10    | AIDS   |          2021 |
|         444| Rajani       | 1980-05-12    | IT     |         2010  |
|         555| Sindhu       | 1993-03-26    | ECE    |         2017  |
|         666| Nayana       | 1995-05-05    | AIML   |         2002  |
+-------------+--------------+---------------+--------+---------------+
```

---

b)List all the students of all branches

```
mysql> select student_name from student;
+--------------+
| student_name |
+--------------+
| srikanth     |
| venkat       |
| siva         |
| Rajani       |
| Sindhu       |
| Nayana       |
+--------------+
6 rows in set (0.00 sec)
```

---

c) list all student names start with 's'

```
mysql> select student_name from student
-> where student_name like 's%';
+--------------+
| student_name |
+--------------+
| srikanth     |
| siva         |
| Sindhu       |
```

```
+--------------+
3 rows in set (0.00 sec)
```

---

d) List student names whose name contains 's' as the third literal

```
mysql> select * from student
-> where student_name like '__s%';
Empty set (0.00 sec)
```

---

e) list student names whose contains two 's' any where

```
mysql> select student_name from student where student_name like '%s%s%';
Empty set (0.00 sec)
```

---

f) list of students whose branch is null

```
mysql> insert into
student(Roll_number,student_name,date_of_birth,branch,year_of_study)
-> values(777,'nandana','2003-04-28',null,2020);
Query OK, 1 row affected (0.07 sec)
mysql> select * from student;
+-------------+--------------+---------------+--------+---------------+
| Roll_number | student_name | date_of_birth | branch | year_of_study |
+-------------+--------------+---------------+--------+---------------+
|         111 | srikanth     | 1990-03-16    | cse    |          2020 |
|         222 | venkat       | 1991-09-26    | cse    |          2020 |
|         333 | siva         | 1990-04-10    | AIDS   |          2021 |
|         444 | Rajani       | 1980-05-12    | IT     |          2010 |
|         555 | Sindhu       | 1993-03-26    | ECE    |          2017 |
|         666 | Nayana       | 1995-05-05    | AIML   |          2002 |
|         777 | nandana      | 2003-04-28    | NULL   |          2020 |
+-------------+--------------+---------------+--------+---------------+
7 rows in set (0.01 sec)
mysql> select * from student
-> where branch is null;
+-------------+--------------+---------------+--------+---------------+
| Roll_number | student_name | date_of_birth | branch | year_of_study |
+-------------+--------------+---------------+--------+---------------+
|     777     | nandana      | 2003-04-28    | NULL   |    2020       |
+-------------+--------------+---------------+--------+---------------+
1 row in set (0.00 sec)
```

---

g) List students of CSE & ECE who born after 1980.

```
mysql> select * from student
-> where branch in ('cse','ECE') and date_of_birth > 1980;
+-------------+--------------+---------------+--------+---------------+
| Roll_number | student_name | date_of_birth | branch | year_of_study |
+-------------+--------------+---------------+--------+---------------+
|       111 | srikanth     | 1990-03-16    | cse    |      2020 |
|       222 | venkat       | 1991-09-26    | cse    |      2020 |
|       555 | Sindhu       | 1993-03-26    | ECE    |      2017 |
+-------------+--------------+---------------+--------+---------------+
3 rows in set, 1 warning (0.00 sec)
```

---

h) List all students in reverse order of their names

```
mysql> select * from student
-> order by student_name desc;
```

```
+-------------+--------------+---------------+--------+---------------+
| Roll_number | student_name | date_of_birth | branch | year_of_study |
+-------------+--------------+---------------+--------+---------------+
|         222 | venkat       | 1991-09-26    | cse    |    2020       |
|         111 | srikanth     | 1990-03-16    | cse    |    2020       |
|         333 | siva         | 1990-04-10    | AIDS   |    2021       |
|         555 | Sindhu       | 1993-03-26    | ECE    |    2017       |
|         444 | Rajani       | 1980-05-12    | IT     |    2010       |
|         666 | Nayana       | 1995-05-05    | AIML   |    2002       |
|         777 | nandana      | 2003-04-28    | NULL   |    2020       |
+-------------+--------------+---------------+--------+---------------+
7 rows in set (0.00 sec)
```

---

i) Delete students of any branch whose name starts with 's'.

```
mysql> delete from student where student_name like "s%";
Query OK, 3 rows affected (0.06 sec)
mysql> select * from student;
+-------------+--------------+---------------+--------+---------------+
| Roll_number | student_name | date_of_birth | branch | year_of_study |
+-------------+--------------+---------------+--------+---------------+
|         222 | venkat       | 1991-09-26    | cse    |          2020 |
|         444 | Rajani       | 1980-05-12    | IT     |          2010 |
|         666 | Nayana       | 1995-05-05    | AIML   |          2002 |
|         777 | nandana      | 2003-04-28    | NULL   |          2020 |
+-------------+--------------+---------------+--------+---------------+
4 rows in set (0.00 sec)
To disable autocommit use
mysql> set autocommit=false;
Query OK, 0 rows affected (0.03 sec)
```

---

j) update the branch of cse students to ece

```
mysql> update student set branch='ece'
-> where branch='cse';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select * from student;
+-------------+--------------+---------------+--------+---------------+
| Roll_number | student_name | date_of_birth | branch | year_of_study |
+-------------+--------------+---------------+--------+---------------+
|         222 | venkat       | 1991-09-26    | ece    |    2020       |
|         444 | Rajani       | 1980-05-12    | IT     |    2010       |
|         666 | Nayana       | 1995-05-05    | AIML   |    2002       |
|         777 | nandana      | 2003-04-28    | NULL   |    2020       |
+-------------+--------------+---------------+--------+---------------+
4 rows in set (0.00 sec)
To create a savepoint we need to start the transtaction first.
mysql> start transaction;
Query OK, 0 rows affected
transaction-1
transaction-2
transaction-3
transaction-4
transaction-5
savepoint A;
transaction-6
transaction-7
transaction-8
if we rollback to A; then transaction-6,7,8 are removed.
```

k) display student name padded with * after the name of all the students.

```
mysql> select RPAD(student_name,30,"*") as Name
-> from student;
+--------------------------------+
| Name                           |
+--------------------------------+
| venkat************************ |
| Rajani************************ |
| Nayana************************ |
| nandana*********************** |
+--------------------------------+
4 rows in set (0.00 sec)
```

2) Create the following tables based on the above Schema Diagram with appropriate data types
and constraints and perform the following queries.
SAILORS (Saild, Salname, Rating, Age)
RESERVES (Sailid, boatid, Day)
BOATS (Boatid, Boat-name, Color)

TABLE CREATION:-

```
mysql>create table sailors(Sailid int primary key,Salname
varchar(20),Rating
int,Age int);

mysql>create table boats (Boatid int primary key,Boat_name
varchar(20),color
varchar(10));

mysql>create table reserves(Sailid int,Boatid int,day date,foreign
key(Sailid)
references sailors(Sailid), foreign key(Boatid) references
boats(Boatid));
```

```
mysql> show tables;
+----------------------+
| Tables_in_20761A0589 |
+----------------------+
| boats                |
| reserves             |
| sailors              |
+----------------------+
3 rows in set (0.01 sec)
mysql> desc reserves;
+--------+------+------+-----+---------+-------+
| Field  | Type | Null | Key | Default | Extra |
+--------+------+------+-----+---------+-------+
| Sailid | int  | YES  | MUL | NULL    |       |
| Boatid | int  | YES  | MUL | NULL    |       |
| day    | date | YES  |     | NULL    |       |
+--------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> desc sailors;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Sailid  | int         | NO   | PRI | NULL    |       |
| Salname | varchar(20) | YES  |     | NULL    |       |
| Rating  | int         | YES  |     | NULL    |       |
| Age     | int         | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc boats;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Boatid    | int         | NO   | PRI | NULL    |       |
| Boat_name | varchar(20) | YES  |     | NULL    |       |
| color     | varchar(10) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

INSERTING DATA :-

```
mysql> insert into sailors (Sailid,Salname,Rating,Age)
values(22,'Dustin',7,45), (29,'Brutus',1,33),(31,'Lubber',8,55),
(32,'Andy',8,25), (58,'Rusty',10,35), (64,'Horatio',7,35),
(71,'Zobra',10,16), (74,'Horatio',9,35), (85,'Art',3,25),
(95,'Bob',3,63.5);

mysql>insert into boats(Boatid,Boat_name,color)
values(101,'Interlake','Blue'),(102,'Interlake','Red'),(103,'Clipper','Gr
een'),(104,'Marine','Red');

mysql>insert into reserves(Sailid,Boatid,Day) values(22,101,'1998-10-
10'),(22,102,'1998-08-13'),(22,103,'1984-05-24'),(22,104,'1990-06-
13'),(31,102,'1997-02-13'),(31,103,'1998-06-11'),(31,104,'1998-12-
11'),(64,101,'1998-05-09'),(64,102,'1998-07-09'),(74,103,'1998-07-09');
```

---

b) Find the name of sailors who reserved boat number 3.

```
mysql> select s.Salname from sailors s,reserves r where s.Sailid=r.Sailid
and r.Boatid = 3;
mysql> show tables;
+----------------------+
| Tables_in_20761A0589 |
+----------------------+
| boats                |
| reserves             |
| sailors              |
+----------------------+
mysql> select * from boats;
+--------+-----------+-------+
| Boatid | Boat_name | color |
+--------+-----------+-------+
| 101    | Interlake | Blue  |
| 102    | Interlake | Red   |
| 103    | Clipper   | Green |
| 104    | Marine    | Red   |
+--------+-----------+-------+
```

```
mysql> select * from reserves;
+--------+--------+------------+
| Sailid | Boatid | day        |
+--------+--------+------------+
| 22     | 101    | 1998-10-10 |
| 22     | 102    | 1998-08-13 |
| 22     | 103    | 1984-05-24 |
| 22     | 104    | 1990-06-13 |
| 31     | 102    | 1997-02-13 |
| 31     | 103    | 1998-06-11 |
| 31     | 104    | 1998-12-11 |
| 64     | 101    | 1998-05-09 |
| 64     | 102    | 1998-07-09 |
| 74     | 103    | 1998-07-09 |
+--------+--------+------------+
10 rows in set (0.00 sec)

mysql> select * from sailors;
+--------+---------+--------+------+
| Sailid | Salname | Rating | Age  |
+--------+---------+--------+------+
|  22    | Dustin  | 7      | 45   |
|  29    | Brutus  | 1      | 33   |
|  31    | Lubber  | 8      | 55   |
|  32    | Andy    | 8      | 25   |
|  52    | Rusty   | 10     | 35   |
|  64    | Horatio | 7      | 35   |
|  71    | Zobra   | 10     | 16   |
|  74    | Horatio | 9      | 35   |
|  85    | Art     | 3      | 25   |
|  95    | Bob     | 3      | 64   |
+--------+---------+--------+------+
10 rows in set (0.00 sec)

mysql> select s.Salname from sailors s, reserves r where s.Sailid =
r.Sailid and r.Boatid = 103;
+---------+
| Salname |
+---------+
| Dustin  |
| Lubber  |
| Horatio |
+---------+
3 rows in set (0.00 sec)
```

---

c) Find the name of sailors who reserved green boat.

```
mysql> select s.Salname from sailors s, reserves r,boats b where s.Sailid
= r.Sailid and r.Boatid = b.Boatid and b.color = 'Green';
+---------+
| Salname |
+---------+
| Dustin  |
| Lubber  |
| Horatio |
+---------+
3 rows in set (0.00 sec)
```

d) Find the color of boats reserved by Dustin

```
mysql> select color from boats inner join reserves on reserves.Boatid =
boats.Boatid inner join sailors on sailors.Sailid = reserves.Sailid where
Salname='Dustin';
+-------+
| color |
+-------+
| Blue  |
| Red   |
| Green |
| Red   |
+-------+
4 rows in set (0.00 sec)
```

---

e) Find the names of the sailors who have reserved atleast one boat.

```
mysql> select s.Salname
-> from sailors s, reserves r
-> where s.Sailid = r.Sailid;
+---------+
| Salname |
+---------+
| Dustin  |
| Dustin  |
| Dustin  |
| Dustin  |
| Lubber  |
| Lubber  |
| Lubber  |
| Horatio |
| Horatio |
| Horatio |
+---------+
10 rows in set (0.00 sec)
```

---

f) Find the allsailid of sailors who have a rating of 10 or have reserved boated 104.

```
mysql> select Salname from sailors inner join reserves on sailors.Sailid
= reserves.Sailid where Rating = 10 or Boatid = 104;
+---------+
| Salname |
+---------+
| Dustin  |
| Lubber  |
+---------+
2 rows in set (0.00 sec)
```

---

g) Find the Sailid's of sailors with age over 20 who have not registered a red boat.

```
mysql> select distinct s.Sailid from sailors s,boats b,reserves r where
s.Sailid = r.Sailid and r.Boatid = b.Boatid and s.Age > 20 and b.color !=
'Red';
+--------+
| Sailid |
+--------+
|     22 |
|     64 |
|     31 |
|     74 |
```

```
+--------+
4 rows in set (0.04 sec)
```

---

h) Find the names of sailors who have reserved a red or green boat.

```
mysql> select s.Salname from sailors s, boats b, reserves r where
s.Sailid = r.Sailid and r.Boatid = b.Boatid and (b.color = 'Red' or
b.color = 'Green');
+---------+
| Salname |
+---------+
| Dustin  |
| Lubber  |
| Horatio |
| Dustin  |
| Lubber  |
| Horatio |
| Dustin  |
| Lubber  |
+---------+
8 rows in set (0.00 sec)
```

---

i) Find sailors whose rating is better than some sailor called Salvador.

```
mysql> select Salname from sailors where Rating > (select Rating from
sailors where Salname='Dustin');
+---------+
| Salname |
+---------+
| Lubber  |
| Andy    |
| Rusty   |
| Zobra   |
| Horatio |
+---------+
5 rows in set (0.00 sec)
```

---

j) Find the names of sailors who are older than the oldest sailor with a rating of 10.

```
mysql> select Salname from sailors
-> where Age > (select max(Age) from sailors where Rating = 10);
+---------+
| Salname |
+---------+
| Dustin  |
| Lubber  |
| Bob     |
+---------+
3 rows in set (0.04 sec)
```

3) Schema Diagram for the rest of the SQL and PLSQL Programs.
Create the following tables based on the above Schema Diagram with appropriate data types and constraints.

EMPLOYEE (Fname, Mname, Lname, SSN, Bdate, Address, Gender, Salary, SuperSSN,Dno)
DEPARTMENT (Dnumber, Dname, MgrSSN, Mgrstartdate)
DEPENDENT (ESSN, Dependent_Name, Gender, Bdate, Relationship)

TABLE CREATION :-

DEPARTMENT TABLE:-

```
CREATE TABLE DEPARTMENT (DNO VARCHAR(20) PRIMARY KEY, DNAME
VARCHAR(20),MGRSTARTDATE DATE);
```

EMPLOYEE TABLE:

```
CREATE TABLE EMPLOYEE (FNAME VARCHAR(20), MNAME VARCHAR(20), LNAME
VARCHAR(20), SSN VARCHAR (20) PRIMARY KEY, DOB DATE, ADDRESS VARCHAR
(20), GENDER VARCHAR(10),  SALARY INTEGER, SUPERSSN VARCHAR(20)
REFERENCES EMPLOYEE (SSN), DNO VARCHAR(20) REFERENCES DEPARTMENT (DNO));
```

NOTE: Once DEPARTMENT and EMPLOYEE tables are created we must alter department table to add foreign constraint MGRSSN using sql command

```
ALTER TABLE DEPARTMENT
ADD MGRSSN VARCHAR(20) REFERENCES EMPLOYEE (SSN);
```

DEPENDENT TABLE:

```
CREATE TABLE DEPENDENT (ESSN VARCHAR(20) REFERENCES EMPLOYEE(SSN),
DEPENDENTNAME VARCHAR(20), GENDER VARCHAR(20), DOB DATE, RELATIONSHIP
VARCHAR(20));
```

DLOCATION TABLE:

```
CREATE TABLE DLOCATION(DLOC VARCHAR(20),DNO VARCHAR(20) REFERENCES
DEPARTMENT(DNO), PRIMARY KEY(DNO, DLOC));
```

PROJECT TABLE:

```
CREATE TABLE PROJECT (PNAME VARCHAR(20),PNO INTEGER PRIMARY KEY,
PLOCATION VARCHAR(20), DNO VARCHAR(20) REFERENCES DEPARTMENT (DNO));
```

WORKS_ON TABLE:

```
CREATE TABLE WORKS_ON(ESSN VARCHAR(20) REFERENCES EMPLOYEE(SSN),PNO
INTEGER REFERENCES PROJECT(PNO), PRIMARY KEY (ESSN, PNO), HOURS INTEGER);
```

a)Insert 5 to 10 rows into all the tables.

INSERT DATA INTO EMPLOYEE:

```
INSERT INTO EMPLOYEE (FNAME , MNAME , LNAME , SSN , DOB , ADDRESS ,
GENDER ,  SALARY , SUPERSSN , DNO ) VALUES
('John','B','Smith','123456789','1965-02-09','731
Fondren','M',30000,'333445555',5),
('Franklin','T','Wong','333445555','1955-12-08','638
Voss','M',40000,'888665555',5),
('Alicia','J','Zelaya','999887777','1968-01-19','3321
Castle','F',25000,'987654321',4),('Jennifer','S','Wallance','987654321','
1941-06-20','291
Berry','F',43000,'888665555',4),('Ramesh','K','Narayana','666884444','196
```

```
2-09-15','975 Fire
Oak','M',38000,'333445555',5),('Joyce','A','English','453453453','1972-
07-31','5631
Rice','F',25000,'333445555',5),('Ahmad','V','Jabbar','987987987','1969-
03-22','980
Dallas','M',25000,'987987987',4),('James','E','Brog','888665555','1937-
10-10','450 Stone','M',55000,'NULL',1);

INSERT INTO DEPARTMENT:
---------------------------------------------------------------
INSERT INTO DEPARTMENT (DNO, DNAME,MGRSTARTDATE,MGRSSN) VALUES
('5','Research','1988-05-22','333445555'),('4','Administration','1995-01-
01','987654321'),('1','Headquarters','1981-06-19','888665555');

INSERT INTO DEPENDENT:
---------------------------------------------------------------
INSERT INTO DEPENDENT (ESSN,DEPENDENTNAME,GENDER,DOB, RELATIONSHIP)
VALUES('333445555','Alice','F','1986-04-
05','Daughter'),('333445555','Theodore','M','1983-10-
25','Son'),('333445555','Joy','F','1958-05-
03','Spouse'),('987654321','Abner','M','1942-02-
28','Spouse'),('123456789','Michael','M','1988-01-
04','Son'),('123456789','Elizabeth','F','1967-05-05','Spouse');

INSERT INTO DLOCATION:
---------------------------------------------------------------
INSERT INTO DLOCATION(DLOC,DNO) VALUES
('Houstan','1'),
('Stafford','4'),
('Bellaire','5'),
('Sugarland','5'),
('Houstan','5');

INSERT INTO PROJECT:
---------------------------------------------------------------
INSERT INTO PROJECT (PNAME,PNO,  PLOCATION, DNO) VALUES
('ProductX',1,'Bellaire','5'),
('ProductY',2,'Sugarland','5'),
('ProductZ',3,'Houstan','5'),
('Computerization',10,'Stafford','4'),
('Reorganization',20,'Houstan','1'),
('Newbenefits',30,'Stafford','4');

INSERT INTO WORKS_ON:
---------------------------------------------------------------
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES('123456789',1,32),
('123456789',2,47),
('666884444',3,40),
('453453453',1,20),
('333445555',2,20),
('333445555',1,10),
('333445555',3,10),
('333445555',10,10),
('999887777',20,10),
('999887777',30,30),
('987987987',10,10),
('987987987',11,35),
('987654321',30,5),
```

```
('987654321',31,20),
('888665555',20,15);
```

B)Display all employees' names along with their department names.

```
mysql> CREATE VIEW RESULTB AS (SELECT FNAME,LNAME,DNAME FROM
EMPLOYEE,DEPARTMENT WHERE EMPLOYEE.DNO=DEPARTMENT.DNO);
Query OK, 0 rows affected (0.06 sec)
mysql> SELECT * FROM RESULTB;
+----------+----------+----------------+
| FNAME    | LNAME    | DNAME          |
+----------+----------+----------------+
| John     | Smith    | Research       |
| Franklin | Wong     | Research       |
| Joyce    | English  | Research       |
| Ramesh   | Narayana | Research       |
| James    | Brog     | Headquarters   |
| Jennifer | Wallance | Administration |
| Ahmad    | Jabbar   | Administration |
| Alicia   | Zelaya   | Administration |
+----------+----------+----------------+
8 rows in set (0.02 sec)
```

C)Display all employees' names along with their dependent details

```
mysql> CREATE VIEW RESULTC AS (SELECT FNAME,LNAME,DEPENDENTNAME FROM
EMPLOYEE,DEPENDENT WHERE DEPENDENT.ESSN=EMPLOYEE.SSN);
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM RESULTC;
+----------+----------+---------------+
| FNAME    | LNAME    | DEPENDENTNAME |
+----------+----------+---------------+
| Franklin | Wong     | Alice         |
| Franklin | Wong     | Theodore      |
| Franklin | Wong     | Joy           |
| Jennifer | Wallance | Abner         |
| John     | Smith    | Michael       |
| John     | Smith    | Elizabeth     |
+----------+----------+---------------+
6 rows in set (0.00 sec)
```

D)Display name and address of all employees who work for Research department.

```
mysql> CREATE VIEW RESULTD AS (SELECT FNAME,LNAME,ADDRESS FROM
EMPLOYEE,DEPARTMENT WHERE EMPLOYEE.DNO=DEPARTMENT.DNO AND
DNAME='RESEARCH');
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM RESULTD;
+----------+----------+--------------+
| FNAME    | LNAME    | ADDRESS      |
+----------+----------+--------------+
| John     | Smith    | 731 Fondren  |
| Franklin | Wong     | 638 Voss     |
| Joyce    | English  | 5631 Rice    |
| Ramesh   | Narayana | 975 Fire Oak |
+----------+----------+--------------+
4 rows in set (0.00 sec)
```

E)List the names of all employees with two or more dependents

```
mysql> CREATE VIEW RESULTE AS (SELECT FNAME,LNAME FROM EMPLOYEE
WHERE(SELECT COUNT(*) FROM DEPENDENT WHERE SSN=ESSN)>=2);
Query OK, 0 rows affected (0.06 sec)
mysql> SELECT * FROM RESULTE;
+----------+-------+
| FNAME    | LNAME |
+----------+-------+
| John     | Smith |
| Franklin | Wong  |
+----------+-------+
2 rows in set (0.00 sec)
```

F)List the names of employee who have no dependents.

```
mysql> CREATE VIEW RESULTF AS (SELECT FNAME,LNAME FROM EMPLOYEE WHERE NOT
EXISTS(SELECT * FROM DEPENDENT WHERE SSN=ESSN));
Query OK, 0 rows affected (0.06 sec)
mysql> SELECT * FROM RESULTF;
+--------+----------+
| FNAME  | LNAME    |
+--------+----------+
| Joyce  | English  |
| Ramesh | Narayana |
| James  | Brog     |
| Ahmad  | Jabbar   |
| Alicia | Zelaya   |
+--------+----------+
5 rows in set (0.01 sec)
```

G)List the names of employees who have at least one dependent.

```
mysql> CREATE VIEW RESULTG AS (SELECT FNAME,LNAME FROM EMPLOYEE WHERE
EXISTS (SELECT * FROM DEPENDENT WHERE SSN=ESSN) AND EXISTS (SELECT * FROM
DEPARTMENT WHERE SSN=MGRSSN));
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM RESULTG;
+----------+----------+
| FNAME    | LNAME    |
+----------+----------+
| Franklin | Wong     |
| Jennifer | Wallance |
+----------+----------+
2 rows in set (0.00 sec)
```

H)List the names of the employees along with names of their supervisors using aliases.

```
mysql> CREATE VIEW RESULTH AS (SELECT E1.FNAME,E1.LNAME,E2.FNAME AS
SUPERVISOR FROM EMPLOYEE E1,EMPLOYEE E2 WHERE E2.SSN=E1.SUPERSSN);
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM RESULTH;
+----------+----------+------------+
| FNAME    | LNAME    | SUPERVISOR |
+----------+----------+------------+
| John     | Smith    | Franklin   |
| Franklin | Wong     | James      |
| Joyce    | English  | Franklin   |
| Ramesh   | Narayana | Franklin   |
| Jennifer | Wallance | James      |
| Ahmad    | Jabbar   | Ahmad      |
```

```
| Alicia   | Zelaya   | Jennifer   |
+----------+----------+------------+
7 rows in set (0.00 sec)
```

I)Display name of the department and name of manager for all the departments.

```
mysql> CREATE VIEW RESULTI AS (SELECT DNAME,FNAME FROM EMPLOYEE
E,DEPARTMENT D WHERE E.SSN=D.MGRSSN);
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM RESULTI;
+----------------+----------+
| DNAME          | FNAME    |
+----------------+----------+
| Headquarters   | James    |
| Administration | Jennifer |
| Research       | Franklin |
+----------------+----------+
3 rows in set (0.00 sec)
```

J)Display the name of each employee who has a dependent with the same first name and gender as the employee.

```
mysql> CREATE VIEW RESULTJ AS (SELECT E.FNAME,E.LNAME FROM EMPLOYEE AS E
WHERE E.SSN IN (SELECT ESSN FROM EMPLOYEE,DEPENDENT WHERE
FNAME=DEPENDENTNAME AND EMPLOYEE.GENDER=DEPENDENT.GENDER));
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM RESULTJ;
Empty set (0.00 sec)
```

4**) Create the following tables based on the above Schema Diagram with appropriate data**
typesand constraints in addition to the tables in Experiment 2.

```
DEPT_LOCATIONS (Dnumber, Dloaction)
PROJECT (Pname, Pnumber, Plocation,
Dnum) WORKS_ON (ESSN, Pno, Hours)
```

TABLE CREATION :-

```
DLOCATION TABLE:
CREATE TABLE DLOCATION(DLOC VARCHAR(20),DNO VARCHAR(20) REFERENCES
DEPARTMENT(DNO), PRIMARY KEY(DNO, DLOC));

PROJECT TABLE:
CREATE TABLE PROJECT (PNAME VARCHAR(20),PNO INTEGER PRIMARY KEY,
PLOCATION VARCHAR(20), DNO VARCHAR(20) REFERENCES DEPARTMENT (DNO));

WORKS_ON TABLE:
CREATE TABLE WORKS_ON(ESSN VARCHAR(20) REFERENCES EMPLOYEE(SSN),PNO
INTEGER REFERENCES PROJECT(PNO), PRIMARY KEY (ESSN, PNO), HOURS INTEGER);
```

a) Insert 5 to 10 rows into all the tables.

INSERT INTO DLOCATION:

```
INSERT INTO DLOCATION(DLOC,DNO) VALUES
('Houstan','1'),
('Stafford','4'),
('Bellaire','5'),
('Sugarland','5'),
('Houstan','5');
```

INSERT INTO PROJECT:

```
INSERT INTO PROJECT (PNAME,PNO,  PLOCATION, DNO) VALUES
('ProductX',1,'Bellaire','5'),
('ProductY',2,'Sugarland','5'),
('ProductZ',3,'Houstan','5'),
('Computerization',10,'Stafford','4'),
('Reorganization',20,'Houstan','1'),
('Newbenefits',30,'Stafford','4');
```

INSERT INTO WORKS_ON:

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS)
VALUES('123456789',1,32),('123456789',2,47),
('666884444',3,40),
('453453453',1,20),
('333445555',2,20),
('333445555',1,10),
('333445555',3,10),
('333445555',10,10),
('999887777',20,10),
('999887777',30,30),
('987987987',10,10),
('987987987',11,35),
('987654321',30,5),
('987654321',31,20),
('888665555',20,15);
```

B)Find the names of the employees who work on all the projects controlled by the department Research.

```
mysql> CREATE VIEW RESULT4B AS (SELECT DISTINCT E.FNAME,E.LNAME FROM
EMPLOYEE E,DEPARTMENT D, PROJECT P,WORKS_ON W WHERE D.DNAME='RESEARCH'
AND D.DNO=P.DNO AND W.ESSN=E.SSN AND P.PNO=W.PNO);
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM RESULT4B;
+----------+----------+
| FNAME    | LNAME    |
+----------+----------+
| John     | Smith    |
| Franklin | Wong     |
| Joyce    | English  |
| Ramesh   | Narayana |
+----------+----------+
4 rows in set (0.00 sec)
```

C)List the project number, name and no. Of employees who work on that project for all the projects.

```
mysql> CREATE VIEW RESULT4C AS (SELECT P.PNO,P.PNAME,COUNT(W.ESSN) FROM
PROJECT P,WORKS_ON W WHERE P.PNO=W.PNO GROUP BY P.PNO,P.PNAME);
Query OK, 0 rows affected (0.09 sec)
mysql> SELECT * FROM RESULT4C;
+-----+-----------------+---------------+
| PNO | PNAME           | COUNT(W.ESSN) |
+-----+-----------------+---------------+
|   1 | ProductX        |             3 |
|   2 | ProductY        |             2 |
|   3 | ProductZ        |             2 |
```

```
|  10 | Computerization |             2 |
|  20 | Reorganization  |             2 |
|  30 | Newbenefits     |             2 |
+-----+-----------------+---------------+
6 rows in set (0.01 sec)
```

D)List the names of all the projects controlled by the departments department wise.

```
mysql> CREATE VIEW RESULT4D AS (SELECT P.PNAME,D.DNAME FROM PROJECT
P,DEPARTMENT D WHERE P.DNO=D.DNO);
Query OK, 0 rows affected (0.06 sec)
mysql> SELECT * FROM RESULT4D;
+-----------------+----------------+
| PNAME           | DNAME          |
+-----------------+----------------+
| ProductX        | Research       |
| ProductY        | Research       |
| ProductZ        | Research       |
| Computerization | Administration |
| Reorganization  | Headquarters   |
| Newbenefits     | Administration |
+-----------------+----------------+
6 rows in set (0.01 sec)
```

E)Retrieve the names of employees who work on all projects that John works on.

```
mysql> CREATE VIEW RESULT4E AS (SELECT DISTINCT E.FNAME,E.LNAME FROM
EMPLOYEE E,WORKS_ON W WHERE E.SSN=W.ESSN AND W.PNO IN (SELECT W.PNO FROM
EMPLOYEE E,WORKS_ON W WHERE E.SSN=W.ESSN AND E.FNAME='JOHN'));
Query OK, 0 rows affected (0.10 sec)
mysql> SELECT * FROM RESULT4E;
+----------+---------+
| FNAME    | LNAME   |
+----------+---------+
| John     | Smith   |
| Franklin | Wong    |
| Joyce    | English |
+----------+---------+
3 rows in set (0.00 sec)
```

F)List the project numbers for projects that involve an employee either as worker or as a manager of the department that controls the project.

```
mysql> CREATE VIEW RESULT4F AS (SELECT DISTINCT P.PNO FROM PROJECT
P,DEPARTMENT D,EMPLOYEE E WHERE  D.MGRSSN=E.SSN AND D.DNO=P.DNO)
UNION(SELECT DISTINCT P.PNO FROM EMPLOYEE E,PROJECT P,WORKS_ON W WHERE
E.SSN=W.ESSN AND P.PNO=W.PNO);
Query OK, 0 rows affected (0.10 sec)
mysql> SELECT *FROM RESULT4F;
+-----+
| PNO |
+-----+
|  20 |
|  10 |
|  30 |
|   1 |
|   2 |
|   3 |
+-----+
```

G)List the names of all employees in one department who work more than 10 hours on one specific project.

```
mysql> CREATE VIEW RESULT4G AS (SELECT E.FNAME,E.LNAME FROM EMPLOYEE
E,PROJECT P,DEPARTMENT D,WORKS_ON W WHERE D.DNO=E.DNO AND
D.DNAME='RESEARCH' AND D.DNO=P.DNO AND P.PNAME='PRODUCTX' AND P.PNO=W.PNO
AND W.ESSN=E.SSN AND W.HOURS>10);
Query OK, 0 rows affected (0.11 sec)
mysql> SELECT * FROM RESULT4G;
+-------+---------+
| FNAME | LNAME   |
+-------+---------+
| John  | Smith   |
| Joyce | English |
+-------+---------+
2 rows in set (0.00 sec)
```

H)For each project, list the project name and total hours (by all employees) spent on that project.

```
mysql> CREATE VIEW RESULT4H AS (SELECT P.PNAME,SUM(W.HOURS) FROM PROJECT
P,WORKS_ON W WHERE P.PNO=W.PNO GROUP BY P.PNAME,P.PNO);
Query OK, 0 rows affected (0.04 sec)
mysql> SELECT * FROM RESULT4H;
+-----------------+--------------+
| PNAME           | SUM(W.HOURS) |
+-----------------+--------------+
| Computerization |           20 |
| Newbenefits     |           35 |
| ProductX        |           62 |
| ProductY        |           67 |
| ProductZ        |           50 |
| Reorganization  |           25 |
+-----------------+--------------+
6 rows in set (0.00 sec)
```

I)Retrieve the names of all employees who work on every project.

```
mysql> CREATE VIEW RESULT4I AS (SELECT E.FNAME FROM EMPLOYEE E WHERE
E.SSN IN (SELECT W.ESSN FROM WORKS_ON W WHERE W.PNO=ALL(SELECT PNO FROM
PROJECT)));
Query OK, 0 rows affected (0.06 sec)
mysql> SELECT * FROM RESULT4I;
Empty set (0.00 sec)
```

J)Retrieve the names of all employees who do not work on any project.

```
mysql> CREATE VIEW RESULT4J AS (SELECT E.FNAME,E.LNAME FROM EMPLOYEE E
WHERE E.SSN NOT IN (SELECT W.ESSN FROM WORKS_ON W));
Query OK, 0 rows affected (0.08 sec)
mysql> SELECT * FROM RESULT4J;
Empty set (0.01 sec)
```

5) Create a view that has project name, controlling department name, number of employees
andtotal hours worked on the project for each project with more than one employee working on it.

TABLE CREATION :-

```
mysql> CREATE VIEW PROJECT_VIEW(PNAME,DNAME,NOOFEMP,NOOFHRS) AS SELECT
P.PNAME,D.DNAME,COUNT(W.ESSN),SUM(W.HOURS) FROM PROJECT P,DEPARTMENT
D,WORKS_ON W WHERE P.DNO=D.DNO AND P.PNO=W.PNO GROUP BY
W.PNO,P.PNAME,D.DNAME;
Query OK, 0 rows affected (0.05 sec)
mysql> SELECT * FROM PROJECT_VIEW;
+-----------------+----------------+---------+---------+
| PNAME           | DNAME          | NOOFEMP | NOOFHRS |
+-----------------+----------------+---------+---------+
| ProductX        | Research       |       3 |      62 |
| ProductY        | Research       |       2 |      67 |
| ProductZ        | Research       |       2 |      50 |
| Computerization | Administration |       2 |      20 |
| Reorganization  | Headquarters   |       2 |      25 |
| Newbenefits     | Administration |       2 |      35 |
+-----------------+----------------+---------+---------+
6 rows in set (0.00 sec)
```

A)List the projects that are controlled by one department from this view.

```
mysql> CREATE VIEW RESULT5A AS (SELECT PNAME FROM PROJECT_VIEW WHERE
DNAME='RESEARCH');
Query OK, 0 rows affected (0.09 sec)
mysql> SELECT * FROM RESULT5A;
+----------+
| PNAME    |
+----------+
| ProductZ |
| ProductX |
| ProductY |
+----------+
3 rows in set (0.00 sec)
```

B)List the managers of the controlling departments for all the projects.

```
mysql> CREATE VIEW RESULT5B AS (SELECT E.FNAME,E.LNAME,P.DNAME,P.PNAME
FROM EMPLOYEE E, PROJECT_VIEW P,DEPARTMENT D WHERE E.SSN=D.MGRSSN AND
D.DNAME=P.DNAME);
Query OK, 0 rows affected (0.09 sec)
mysql> SELECT * FROM RESULT5B;
+----------+----------+----------------+-----------------+
| FNAME    | LNAME    | DNAME          | PNAME           |
+----------+----------+----------------+-----------------+
| James    | Brog     | Headquarters   | Reorganization  |
| Jennifer | Wallance | Administration | Newbenefits     |
| Jennifer | Wallance | Administration | Computerization |
| Franklin | Wong     | Research       | ProductY        |
| Franklin | Wong     | Research       | ProductZ        |
| Franklin | Wong     | Research       | ProductX        |
+----------+----------+----------------+-----------------+
6 rows in set (0.00 sec)
```

C)Demonstrate one update operation on this view.

IF PARENT TABLE HAS ANY CONSTRAINTS VIEW TABLE IS NOT UPDATED ;

D)List the Location of the controlling departments for all the projects.

```
mysql> CREATE VIEW RESULT5D AS (SELECT P.PNAME,PV.DNAME,D.DLOC FROM
PROJECT_VIEW PV,DLOCATION D,PROJECT P WHERE P.DNO=D.DNO AND
P.PNAME=PV.PNAME);
Query OK, 0 rows affected (0.09 sec)
mysql> SELECT * FROM RESULT5D;
+-----------------+----------------+-----------+
| PNAME           | DNAME          | DLOC      |
+-----------------+----------------+-----------+
| ProductX        | Research       | Bellaire  |
| ProductX        | Research       | Houstan   |
| ProductX        | Research       | Sugarland |
| ProductY        | Research       | Bellaire  |
| ProductY        | Research       | Houstan   |
| ProductY        | Research       | Sugarland |
| ProductZ        | Research       | Bellaire  |
| ProductZ        | Research       | Houstan   |
| ProductZ        | Research       | Sugarland |
| Computerization | Administration | Stafford  |
| Reorganization  | Headquarters   | Houstan   |
| Newbenefits     | Administration | Stafford  |
+-----------------+----------------+-----------+
12 rows in set (0.01 sec)
```

E)Retrieve the data from the view.

```
mysql> SELECT * FROM PROJECT_VIEW;
+-----------------+----------------+---------+---------+
| PNAME           | DNAME          | NOOFEMP | NOOFHRS |
+-----------------+----------------+---------+---------+
| ProductX        | Research       |       3 |      62 |
| ProductY        | Research       |       2 |      67 |
| ProductZ        | Research       |       2 |      50 |
| Computerization | Administration |       2 |      20 |
| Reorganization  | Headquarters   |       2 |      25 |
| Newbenefits     | Administration |       2 |      35 |
+-----------------+----------------+---------+---------+
6 rows in set (0.01 sec)
```

6) Create a view emp from employee such that it contains only emp_noemp_name and department.

```
mysql> CREATE VIEW EMP_VIEW AS (SELECT E.SSN,E.FNAME,E.LNAME,D.DNAME FROM
EMPLOYEE E,DEPARTMENT D);
Query OK, 0 rows affected (0.11 sec)

mysql> SELECT * FROM EMP_VIEW;
+-----------+----------+----------+----------------+
| SSN       | FNAME    | LNAME    | DNAME          |
+-----------+----------+----------+----------------+
| 123456789 | John     | Smith    | Headquarters   |
| 123456789 | John     | Smith    | Administration |
| 123456789 | John     | Smith    | Research       |
| 333445555 | Franklin | Wong     | Headquarters   |
| 333445555 | Franklin | Wong     | Administration |
| 333445555 | Franklin | Wong     | Research       |
| 453453453 | Joyce    | English  | Headquarters   |
```

```
| 453453453 | Joyce    | English  | Administration |
| 453453453 | Joyce    | English  | Research       |
| 666884444 | Ramesh   | Narayana | Headquarters   |
| 666884444 | Ramesh   | Narayana | Administration |
| 666884444 | Ramesh   | Narayana | Research       |
| 888665555 | James    | Brog     | Headquarters   |
| 888665555 | James    | Brog     | Administration |
| 888665555 | James    | Brog     | Research       |
| 987654321 | Jennifer | Wallance | Headquarters   |
| 987654321 | Jennifer | Wallance | Administration |
| 987654321 | Jennifer | Wallance | Research       |
| 987987987 | Ahmad    | Jabbar   | Headquarters   |
| 987987987 | Ahmad    | Jabbar   | Administration |
| 987987987 | Ahmad    | Jabbar   | Research       |
| 999887777 | Alicia   | Zelaya   | Headquarters   |
| 999887777 | Alicia   | Zelaya   | Administration |
| 999887777 | Alicia   | Zelaya   | Research       |
+-----------+----------+----------+----------------+
24 rows in set (0.00 sec)
```

---

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

```
mysql> CREATE VIEW DEPT_VIEW AS (SELECT DNO,DLOC FROM DLOCATION);
Query OK, 0 rows affected (0.09 sec)

mysql> SELECT * FROM DEPT_VIEW;
+-----+-----------+
| DNO | DLOC      |
+-----+-----------+
| 1   | Houstan   |
| 4   | Stafford  |
| 5   | Bellaire  |
| 5   | Houstan   |
| 5   | Sugarland |
+-----+-----------+
5 rows in set (0.00 sec)
```

---

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

```
mysql> CREATE VIEW MANAGER AS (SELECT FNAME,LNAME FROM EMPLOYEE WHERE
SSN=SUPERSSN);
Query OK, 0 rows affected (0.05 sec)

mysql> SELECT * FROM MANAGER;
+-------+--------+
| FNAME | LNAME  |
+-------+--------+
| Ahmad | Jabbar |
+-------+--------+
1 row in set (0.00 sec)
```

---

9)Write a procedure to check whether the given number is Armstrong or not.

```
mysql> delimiter //
mysql> create procedure arms(in n int)
    -> begin
    -> declare m int;
    -> declare sum int;
    -> declare temp int;
```

```
    -> declare len int;
    -> set temp=n;
    -> set sum=0;
    -> set len=char_length(n);
    -> while n>0 do
    -> set m=mod(n,10);
    -> set sum=sum+pow(m,len);
    -> set n=n div 10;
    -> end while;
    -> select if(sum=temp,'armstromg','not a armstrong');
    -> end
    -> //
mysql>delimiter ;
mysql>call arms(153);
+-------------------------------------------+
| if(sum=temp,'armstromg','not a armstrong') |
+-------------------------------------------+
| armstromg                                 |
+-------------------------------------------+
1 row in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
```

---

10)Write a procedure which accept the account number of a customer and retrieve the balance.

```
mysql> create table customer(acc int,name varchar(20),bal int);
Query OK, 0 rows affected (0.49 sec)

mysql> insert into customer(acc,name,bal)
values(1,'sagar',1050),(2,'ram',150),(3,'bhim',100),(4,'srk',105),(5,'sir
',175);
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from customer;
+------+-------+------+
| acc  | name  | bal  |
+------+-------+------+
|    1 | sagar | 1050 |
|    2 | ram   |  150 |
|    3 | bhim  |  100 |
|    4 | srk   |  105 |
|    5 | sir   |  175 |
+------+-------+------+
5 rows in set (0.00 sec)

mysql> delimiter //
mysql> create procedure tab(in ac int)
    -> begin
    -> select bal from customer where acc=ac;
    -> end
    -> //
mysql> delimiter ;
mysql> call tab(1);
+------+
| bal  |
+------+
| 1050 |
+------+
```

```
1 row in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
```

---

11)Write a procedure which accepts the student number and displays the department in which he belongs to.

```
mysql> delimiter //
mysql> create procedure stud(in a int)
    -> begin
    -> select branch from student where no=a;
    -> end
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call stud(4);
+--------+
| branch |
+--------+
| AIDS   |
+--------+
1 row in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
```

12) Create a cursor to modify the salary of all employees belonging to 'Research' department by 150%.

```
mysql>delimiter //
->create procedure emp_sal_update(in dept varchar(20))
->begin
->declare flag int default 0;
->declare s int default 0;
->declare update_cur cursor for select SALARY from EMPLOYEE,DEPARTMENT
where EMPLOYEE.DNO=DEPARTMENT.DNO and DEPARTMENT.DNAME=dept;
->declare continue handler for not found set flag=1;
->open update_cur;
->getRec: LOOP
->fetch update_cur into s;
->if flag=1 then
->LEAVE getRec;
->end if;
->update EMPLOYEE,DEPARTMENT set SALARY=SALARY+(s*150/100) where
EMPLOYEE.DNO=DEPARTMENT.DNO and DEPARTMENT.DNAME=dept;
->END LOOP getRec;
->close update_cur;
->end
->//
mysql>delimiter ;
```

13) Consider the college database. Retrieve all students who have registered for a specific course and store their details into another table using Cursors.

```
mysql> create table student(sno int primary key,sname varchar(20),
    -> dob date,course varchar(5),year int);
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> insert into
    -> student(sno,sname,dob,course,year) values(222,"venu","1991-09-
26","cse",2020),
    -> (333,'siva','1990-04-10','AIDS',2021),(111,"sagar","1990-03-
16","cse",2020),(444,'Ramu','1980-05-12','IT',2010),
    -> (555,'niteesh','1993-03-26','ECE',2017),(666,'joel','1995-05-
05','AIML',2002);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0
mysql> select * from student;
+-----+---------+------------+--------+------+
| sno | sname   | dob        | course | year |
+-----+---------+------------+--------+------+
| 111 | sagar   | 1990-03-16 | cse    | 2020 |
| 222 | venu    | 1991-09-26 | cse    | 2020 |
| 333 | siva    | 1990-04-10 | AIDS   | 2021 |
| 444 | Ramu    | 1980-05-12 | IT     | 2010 |
| 555 | niteesh | 1993-03-26 | ECE    | 2017 |
| 666 | joel    | 1995-05-05 | AIML   | 2002 |
+-----+---------+------------+--------+------+
6 rows in set (0.00 sec)

mysql> create table temp_student(stdno int,stdname varchar(20),stdcourse
varchar(20));
Query OK, 0 rows affected (0.00 sec)

mysql>delimiter //
mysql>create procedure getStudents(in x varchar(10))
->begin
->declare flag int default 0;
->declare stdno int;
->declare stdname varchar(20);
->declare stdcourse varchar(10);
->declare get_cur cursor for select sno,sname,course from student where
course=x;
->declare continue handler for not found set flag=1;
->open get_cur;
->getRec: LOOP
->fetch get_cur into stdno,stdname,stdcourse;
->if flag=1 then
->LEAVE getRec;
->end if;
->insert into temp_student values(stdno,stdname,stdcourse);
->END LOOP getRec;
->close get_cur;
->end
->//
mysql>delimiter ;
mysql> call getStudents("cse");
Query OK, 0 rows affected, 1 warning (0.00 sec)
mysql> select * from temp_student;
+-------+---------+-----------+
| stdno | stdname | stdcourse |
+-------+---------+-----------+
|   111 | sagar   | cse       |
|   222 | venu    | cse       |
+-------+---------+-----------+
2 rows in set (0.00 sec)
```

14) Write an update trigger on Account table. The system should keep track of the records that are being updated.

```
mysql>delimiter //
->CREATE TRIGGER ACCUPDATE BEFORE UPDATE ON Account FOR EACH ROW
->BEGIN
->DECLARE emsg varchar(250);
->SET emsg="NEW BALANCE CANNOT BE LESSTHAN OLD BALANCE";
->IF new.balance<old.balance THEN
->SIGNAL SQLSTATE '45000'
->SET MESSAGE_TEXT=emsg;
->END IF;
->END
->//
mysql>delimiter ;
```