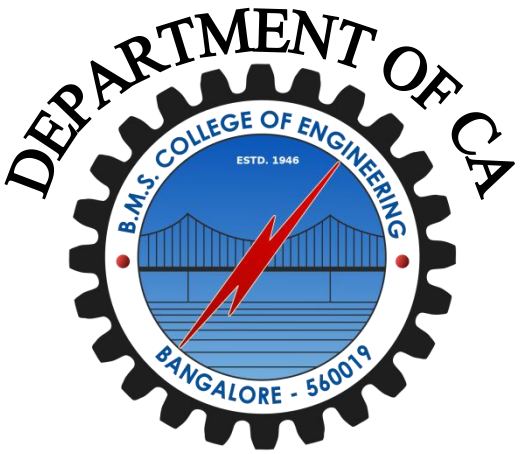
B.M.S. COLLEGE OF ENGINEERING

(Autonomous College, Affiliated to VTU)
Bull Temple Road, Bangalore-560019



Laboratory Certificate

This is to certify that Mr. Manjunath Pradeep Gaonkar (USN: 1BM23MC050) *has satisfactorily completed the integrated course of practical in* ***Database Management Systems (22MCA1PCDB)*** *Laboratory, first semester MCA course in this college during the year 2023-2024.*

Signature of Batch In-charge Signature of HOD

Student Name : Manjunath Pradeep Gaonkar
USN :1BM23MC050

Table of Contents

3.	The following tables are maintained by a book dealer. AUTHOR (author-id : int, name : string, city : string, country : string) PUBLISHER (publisher-id : int, name : string, city : string, country : string) CATALOG (book-id: int, title: string, author-id: int, pub-id: int, cat-id: int, year: int, price: int) CATEGORY (category-id: int, description: string) ORDER-DETAILS (order-no: int, book-id: int, quantity: int) i. Create the above tables by properly specifying the primary keys and the foreign keys. ii. Enter at least five tuples for each relation. iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000. iv. Find the author of the book which has maximum sales. v. Demonstrate how you increase the price of the books published by a specific publisher by 10%..	17-26
4.	Consider the following database for a banking enterprise BRANCH(branch-name : string, branch-city : string, assets : real) ACCOUNT(accno : int, branch-name : string, balance : real) DEPOSITOR(customer-name : string, accno : int) CUSTOMER(customer-name :string, customer-street: string, customer-city : string) LOAN(loan-number : int, branch-name:string, amount : real) BORROWER(customer-name : string, loan-number : int) i. Create above tables by properly specifying the primary keys and the foreign keys. ii. Enter at least five tuples for each relation. iii. Find all the customers who have at least two accounts at the main branch. iv. Find all the customers who have an account to all the branches located in a specific city. v. Demonstrate how u delete all account tuples at every branch located in a specific city. vi. Generate suitable reports. vii. Create suitable front end for querying and displaying the results.	27-35

5.	Consider the following database a Company. Write the queries in SQL. EMPLOYEE(fname, lastname, SSN, Bdate, city, Sex, Salary, Supervisor_SSN, DeptNo) DEPT(Dname, DeptNo, Mgr_SSN, Mgr_StartDate) DEPT_Locations(DeptNo, DLocation) PROJECT(Pname, Pnumber, Plocation, DeptNo) WORKS_ON(Essn, Pnumber, Hours) Dependent(Essn, Dependent_Name, Sex, Bdate, Relationship) i) Create the above tables by properly specifying the primary keys and the foreign keys ii) Enter at least five tuples for each relation iii) Retrieve the names of all employees who work in the department that has the employee with the highest salary among the employees. iv) Retrieve the names of all employees whose supervisor's supervisor has '8888' for SSN. v) Retrive the names of the employees who make atleast Rs. 5000 more than the employee who is paid the least in the company. vi) Create a view that has the dept name, manger name, and manager salary for every department. vii)Create a view that has project name, controlling department name, number of employees, and total hours worked per week on the project for each project. viii)Create a view that has emp name, supervisor name, and emp salary for each employee who works in the 'Research' Dept.	36-45

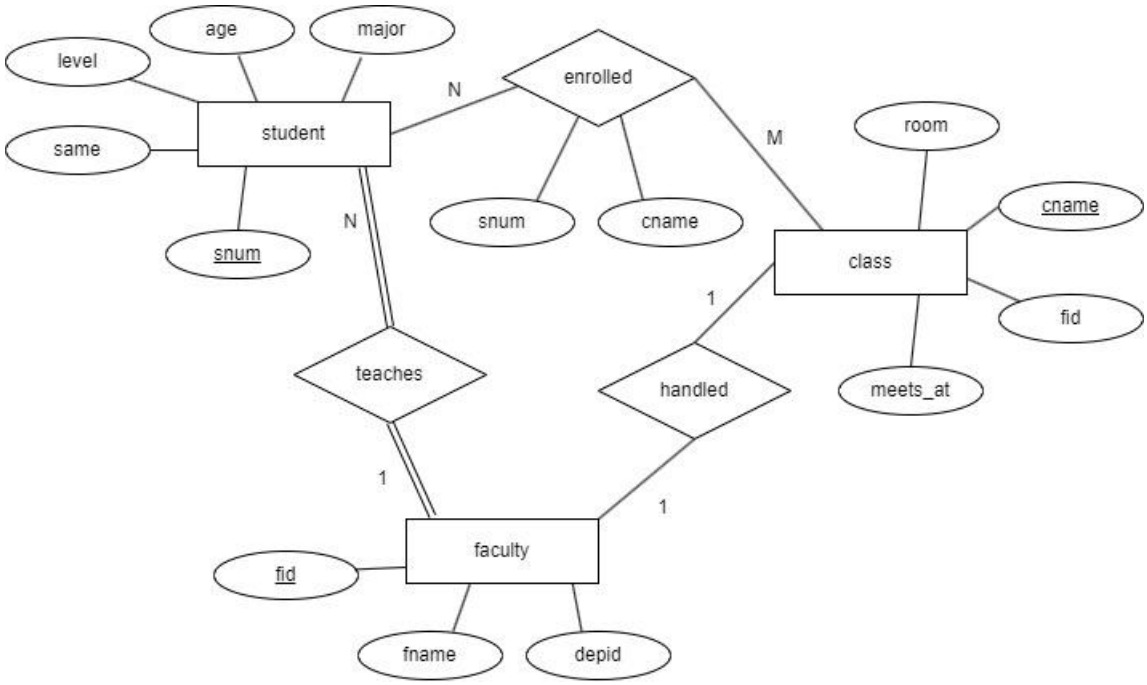
1. Consider the following relations:

- Student (snum: integer, sname: string, major: string, level: string, age: integer)
- Class (name: string, meets at: string, room: string, d: integer)
- Faculty (fid: integer, fname: string, deptid: integer)
- Enrolled (snum: integer, cname: string)

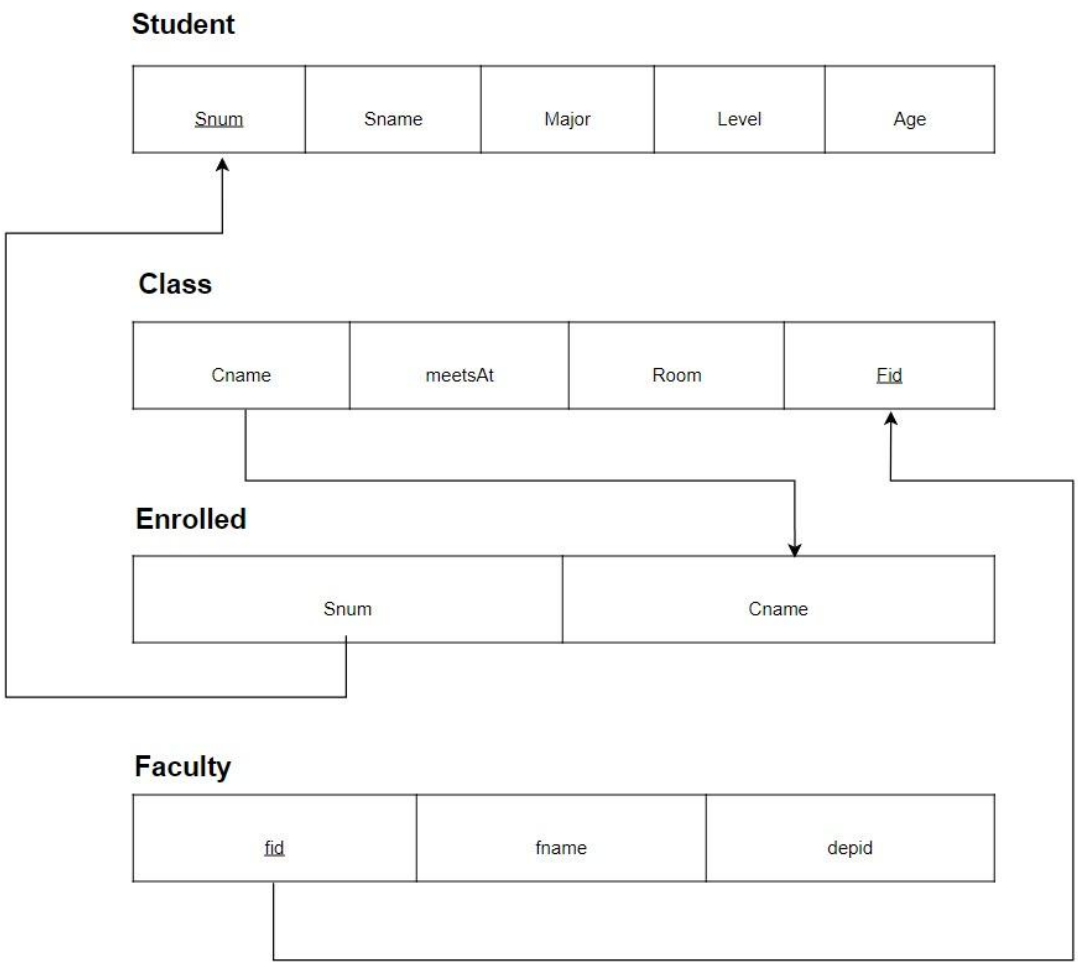
Write each of the following queries in SQL.

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith.
2. Find the names of all classes that either meet in room R128 or have five or more Studentsenrolled.
3. Find the names of all students who are enrolled in two classes that meet at the same time.
4. Find the names of faculty members who teach in every room in which some class is taught.
5. Find the names of faculty members for whom the combined enrollment of the courses that theyteach is less than five.

ER Diagram



Schema Diagram



QUERIES

➢ **Creating relations**

CREATE TABLE Student (snum INTEGER PRIMARY KEY, sname VARCHAR(50), majorVARCHAR(50), level1 VARCHAR(2), age INTEGER);

CREATE TABLE Faculty (fid INTEGER PRIMARY KEY, fname VARCHAR(50), deptidINTEGER);

CREATE TABLE Class (name VARCHAR(50) primary key, meets_at VARCHAR(50), roomVARCHAR(50),fid INTEGER, FOREIGN KEY (fid) REFERENCES Faculty(fid));

CREATE TABLE Enrolled (snum INTEGER,cname VARCHAR(50), FOREIGN KEY (snum) REFERENCES Student(snum), FOREIGN KEY (cname) REFERENCES Class(name));

➢ **Inserting values**

INSERT INTO Student VALUES

(1, 'Shashank', 'Computer', 'SR', 20),

(2, 'Ganesh', 'Biology', 'SR', 22),

(3, 'Rajesh', 'Mathematics', 'JR', 21),

(4, 'Karthik', 'History', 'JR', 20),

(5, 'Pavan', 'Physics', 'SO', 19);

INSERT INTO Faculty VALUES

(1, 'Prof. Harshith', 101),

(2, 'Prof. Krishna', 102),

(3, 'Prof. Jagannath', 103),

(4, 'Prof. Punith', 104),

(5, 'Prof. Veena', 105);

```
INSERT INTO Class VALUES
('COM101',  'Monday  11:00AM',
'R128', 1),
('BIO202', 'Tuesday 2:00PM', 'R120', 2),
('MATH303', 'Wednesday 10:00AM', 'R126', 3),
('HIST201', 'Thrusday 1:00PM', 'R128', 4),
('PHYS101', 'Tuesday 9:00AM', 'R131', 5),
('COM102', 'Wednesday 11:00AM', 'R128', 1),
('PHYS103', 'Wednesday 11:00AM', 'R120', 5),
('PHYS104', 'Thursday 11:00AM', 'R128', 5),
('PHYS105', 'Friday 11:00AM', 'R126', 5),
('PHYS106', 'Saturday 11:00AM', 'R131', 5);
```

```
INSERT  INTO
Enrolled VALUES
(1, 'COM101'),
(2, 'BIO202'),
(3, 'MATH303'),
(4, 'HIST201'),
(5, 'PHYS101'),
(1, 'BIO202'),
(3, 'COM101'),
(4, 'MATH303'),
(2, 'COM102'),
(2, 'PHYS103');
```

➢ **Displaying tables**

SELECT * FROM Student;

```
mysql> select * from student;
+------+----------+-------------+--------+------+
| snum | sname    | major       | level1 | age  |
+------+----------+-------------+--------+------+
|    1 | Shashank | Computer    | SR     |   20 |
|    2 | Ganesh   | Biology     | SR     |   22 |
|    3 | Rajesh   | Mathematics | JR     |   21 |
|    4 | Karthik  | History     | JR     |   20 |
|    5 | Pavan    | Physics     | SO     |   19 |
+------+----------+-------------+--------+------+
5 rows in set (0.00 sec)
```

SELECT * FROM Faculty;

```
mysql> SELECT * FROM Faculty;
+-----+------------------+---------+
| fid | fname            | deptid |
+-----+------------------+---------+
|   1 | Prof. Harshith   |     101 |
|   2 | Prof. Krishna    |     102 |
|   3 | Prof. Jagannath  |     103 |
|   4 | Prof. Punith     |     104 |
|   5 | Prof. Veena      |     105 |
+-----+------------------+---------+
5 rows in set (0.00 sec)
```

SELECT * FROM Class;

```
mysql> SELECT * FROM Class;
+---------+------------------+------+------+
| name    | meets_at         | room | fid  |
+---------+------------------+------+------+
| COM101  | Monday 11:00AM   | R128 |    1 |
| BIO202  | Tuesday 2:00PM   | R120 |    2 |
| MATH303 | Wednesday 10:00AM| R126 |    3 |
| HIST201 | Thrusday 1:00PM  | R128 |    4 |
| PHYS101 | Tuesday 9:00AM   | R131 |    5 |
| COM102  | Wednesday 11:00AM| R128 |    1 |
| PHYS103 | Wednesday 11:00AM| R120 |    5 |
| PHYS104 | Thursday 11:00AM | R128 |    5 |
| PHYS105 | Friday 11:00AM   | R126 |    5 |
| PHYS106 | Saturday 11:00AM | R131 |    5 |
+---------+------------------+------+------+
10 rows in set (0.00 sec)
```

SELECT * FROM faculty;

```
mysql> SELECT * FROM faculty;
+-----+-----------------+--------+
| fid | fname           | deptid |
+-----+-----------------+--------+
|   1 | Prof. Harshith  |    101 |
|   2 | Prof. Krishna   |    102 |
|   3 | Prof. Jagannath |    103 |
|   4 | Prof. Punith    |    104 |
|   5 | Prof. Veena     |    105 |
+-----+-----------------+--------+
5 rows in set (0.00 sec)
```

➢ **Queries**

1. Find the names of all Juniors who are enrolled in a class taught by Prof. Harshith.

```
SELECT DISTINCT s.sname FROM
Student sJOIN Enrolled e ON
s.snum =e.snum
JOIN Class c ON
e.cname = c.nameJOIN
Faculty f ON c.fid =f.fid
WHERE s.level1 = 'JR' AND f.fname = 'Prof. Harshith';
```

```
+--------+
| sname  |
+--------+
| Rajesh |
+--------+
1 row in set (0.00 sec)
```

2. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.

```
SELECT DISTINCT name FROM
ClassWHERE room = 'R128'
OR (SELECT COUNT(*) FROM Enrolled
WHERE Enrolled.cname = Class.name) >= 5;
```

```
+---------+
| name    |
+---------+
| COM101  |
| HIST201 |
| COM102  |
| PHYS104 |
+---------+
4 rows in set (0.00 sec)
```

3. Find the names of all students who are enrolled in two classes that meet at the same time.

```
SELECT DISTINCT s.sname FROM Student s
WHERE s.snum IN (SELECT e1.snum FROM enrolled e1,enrolled e2,class
c1,class c2WHERE e1.snum=e2.snum AND e1.cname<>e2.cname AND
e1.cname=c1.name ANDe2.cname=c2.nameAND c1.meets_at=c2.meets_at);
```

```
+--------+
| sname  |
+--------+
| Ganesh |
+--------+
1 row in set (0.00 sec)
```

4. Find the names of faculty members who teach in every room in which some class is taught.

```
SELECT DISTINCT fname
FROM FacultyWHERE NOT
EXISTS (SELECT room FROM
ClassWHERE NOT EXISTS
( SELECT * FROM Class c2
WHERE c2.fid = Faculty.fid AND c2.room = Class.room ));
```

```
+-------------+
| fname       |
+-------------+
| Prof. Veena |
+-------------+
1 row in set (0.00 sec)
```

5. Find the names of faculty members for whom the combined enrollment of the courses that they teachisless than five.

```
SELECT fname FROM Faculty
WHERE fid IN ( SELECT c.fid FROM Class c
JOIN Enrolled e ON c.name = e.cname GROUP BY c.fid HAVING COUNT(*) < 5);
```

```
+-----------------+
| fname           |
+-----------------+
| Prof. Harshith  |
| Prof. Krishna   |
| Prof. Jagannath |
| Prof. Punith    |
| Prof. Veena     |
+-----------------+
5 rows in set (0.00 sec)
```

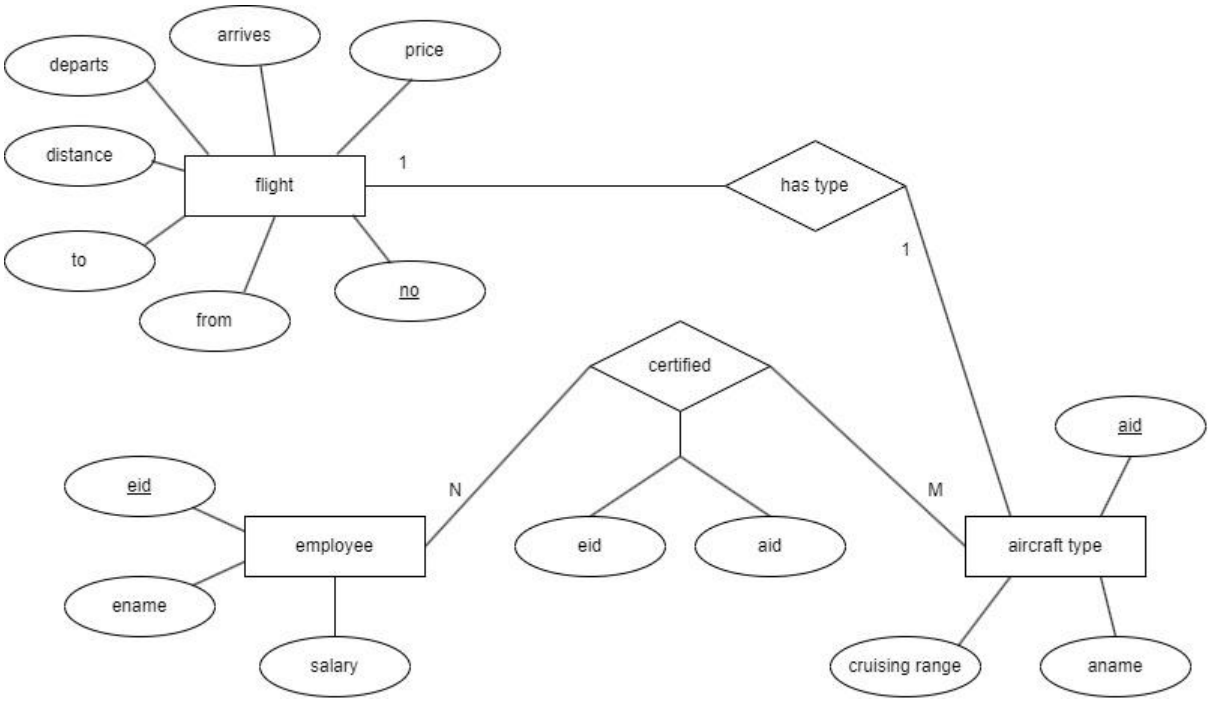
2. **The following relations keep track of airline flight information:**

• Flights (no: integer, from: string, to: string, distance: integer, Departs: time, arrives: time, price: real)
• Aircraft (aid: integer, aname: string, cruisingrange: integer)
• Certified (eid: integer, aid: integer)
• Employees (eid: integer, ename: string, salary: integer)

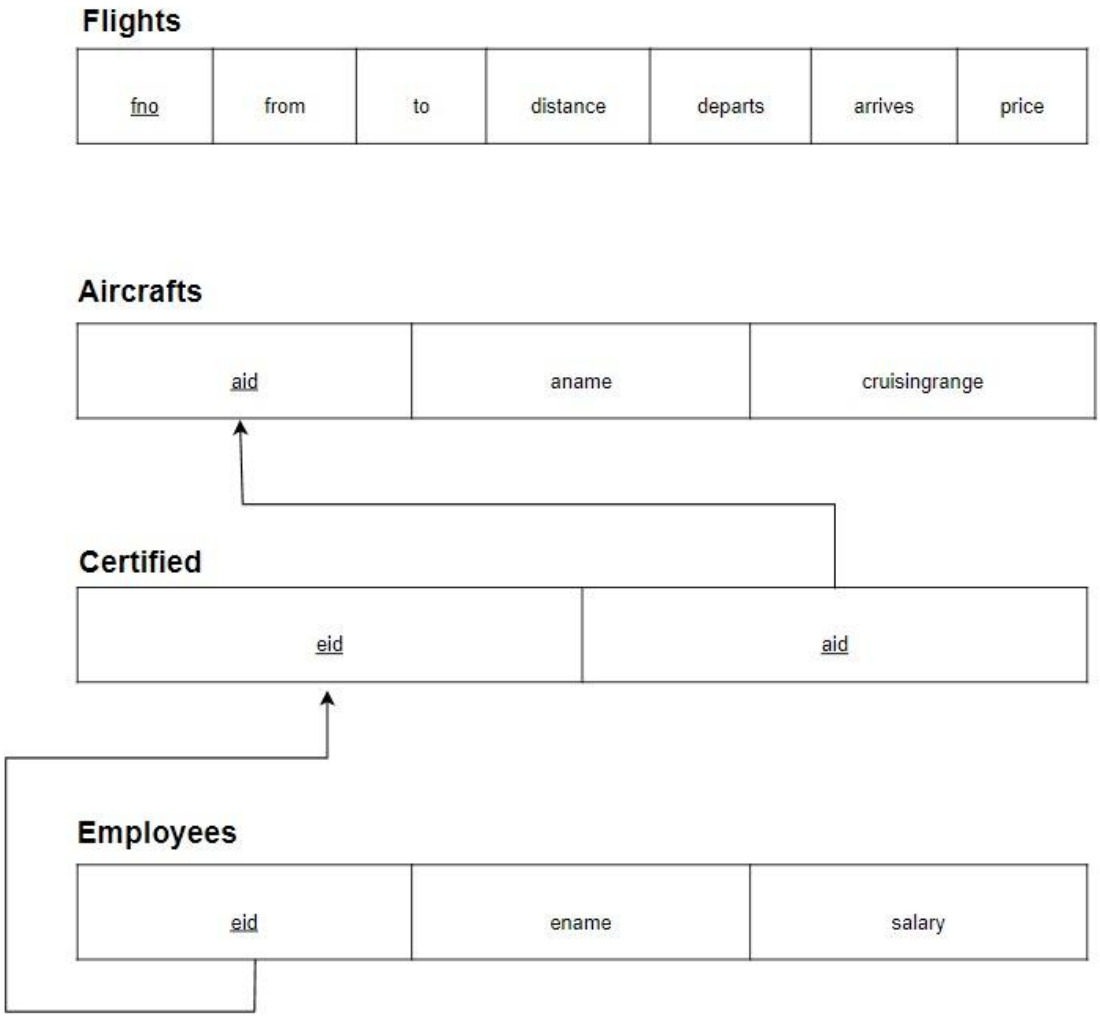
Write each of the following queries in SQL.

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
2. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
4. For all aircraft with cruisingrange over 1000 Kms, .find the name of the aircraft and the average salary of all pilots certified for this aircraft.
5. Find the names of pilots certified for some Boeing aircraft. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

ER Diagram



Schema Diagram



QUERIES

➢ **Creating relations**

```
CREATE TABLE Flights (
  flight_number INTEGER PRIMARY KEY,
  deploc VARCHAR(15),
  arrloc VARCHAR(15),
  distance INTEGER,
  deptime TIME,
  arrtime TIME,
  price VARCHAR(6));

CREATE TABLE Aircraft (
  aid INTEGER PRIMARY KEY,
  aname TEXT,
  cruising_range INTEGER);


CREATE TABLE Employees (
  eid INTEGER PRIMARY KEY,
  ename TEXT,
  salary INTEGER);

CREATE TABLE Certified (
  eid INTEGER,
  aid INTEGER,
  FOREIGN KEY (eid) REFERENCES Employees(eid),
  FOREIGN KEY (aid) REFERENCES Aircraft(aid),
  PRIMARY KEY (eid, aid));
```

➢ Inserting values

```
INSERT INTO Flights VALUES
  (1,'Bangalore','Mangalore',360,'10:45:00','12:00:00',10000),
  (2,'Bangalore','Delhi',5000,'12:15:00','04:30:00',25000),
  (3,'Bangalore','Mumbai',3500,'02:15:00','05:25:00',30000),
  (4,'Delhi','Mumbai',4500,'10:15:00','12:05:00',35000),
  (5,'Delhi','Frankfurt',18000,'07:15:00','05:30:00',90000),
  (6,'Bangalore','Frankfurt',19500,'10:00:00','07:45:00',95000),
  (7,'Bangalore','Frankfurt',17000,'12:00:00','06:30:00',99000);
```

```
INSERT INTO aircraft values
  (123,'Airbus',1000),
  (302,'Boeing',5000),
  (306,'Jet01',5000),
  (378,'Airbus380',8000),
  (456,'Aircraft',500),
  (789,'Aircraft02',800),
  (951,'Aircraft03',01000);

INSERT INTO employees VALUES
  (1,'Dinesh',30000),
  (2,'Shikhar',85000),
  (3,'Rahane',50000),
  (4,'Abhishek',45000),
  (5,'Viraj',90000),
  (6,'Om',75000),
  (7,'Rakesh',100000);

INSERT INTO certified VALUES
  (1,123),
  (2,123),
  (1,302),
  (5,302),
  (7,302),
  (1,306),
  (2,306),
  (1,378),
  (2,378),
  (4,378),
  (6,456),
  (3,456),
  (5,789),
  (6,789),
  (3,951),
  (1,951),
  (1,789);
```

➢ **Displaying tables :**

SELECT * FROM Flights;

flight_number	deploc	arrloc	distance	deptime	arrtime	price
1	Bangalore	Mangalore	360	10:45:00	12:00:00	10000
2	Bangalore	Delhi	5000	12:15:00	04:30:00	25000
3	Bangalore	Mumbai	3500	02:15:00	05:25:00	30000
4	Delhi	Mumbai	4500	10:15:00	12:05:00	35000
5	Delhi	Frankfurt	18000	07:15:00	05:30:00	90000
6	Bangalore	Frankfurt	19500	10:00:00	07:45:00	95000
7	Bangalore	Frankfurt	17000	12:00:00	06:30:00	99000

7 rows in set (0.00 sec)

SELECT * FROM Aircraft;

aid	aname	cruising_range
123	Airbus	1000
302	Boeing	5000
306	Jet01	5000
378	Airbus380	8000
456	Aircraft	500
789	Aircraft02	800
951	Aircraft03	1000

7 rows in set (0.00 sec)

SELECT * FROM Employees;

eid	ename	salary
1	Dinesh	30000
2	Shikhar	85000
3	Rahane	50000
4	Abhishek	45000
5	Viraj	90000
6	Om	75000
7	Rakesh	100000

7 rows in set (0.00 sec)

SELECT * FROM certified;

```
+-----+-----+
| eid | aid |
+-----+-----+
|   1 | 123 |
|   1 | 302 |
|   1 | 306 |
|   1 | 378 |
|   1 | 789 |
|   1 | 951 |
|   2 | 123 |
|   2 | 306 |
|   2 | 378 |
|   3 | 456 |
|   3 | 951 |
|   4 | 378 |
|   5 | 302 |
|   5 | 789 |
|   6 | 456 |
|   6 | 789 |
|   7 | 302 |
+-----+-----+
17 rows in set (0.00 sec)
```

➢ **Queries**

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.

```
SELECT DISTINCT a.aname FROM aircraft a,certified c,employees e
WHERE a.aid=c.aid AND c.eid=e.eid AND NOT EXISTS
(SELECT * FROM employees e1
WHERE e1.eid=e.eid AND e1.salary<80000);
```

```
+------------+
| aname      |
+------------+
| Airbus     |
| Jet01      |
| Airbus380  |
| Boeing     |
| Aircraft02 |
+------------+
5 rows in set (0.00 sec)
```

2. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

```
SELECT c.eid, MAX(cruising_range) FROM certified c,aircraft a
WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>3;
```

```
+-----+--------------------+
| eid | MAX(cruising_range) |
+-----+--------------------+
|   1 |               8000 |
+-----+--------------------+
1 row in set (0.00 sec)
```

3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.

```
SELECT DISTINCT e.ename FROM Employees e
WHERE e.salary<
(SELECT MIN(f.price) FROM Flights f
WHERE f.deploc='Bangalore' AND f.arrloc='Frankfurt');
```

```
+----------+
| ename    |
+----------+
| Dinesh   |
| Shikhar  |
| Rahane   |
| Abhishek |
| Viraj    |
| Om       |
+----------+
6 rows in set (0.00 sec)
```

4. For all aircraft with cruisingrange over 1000 Kms, .find the name of the aircraft and the average salary of all pilots certified for this aircraft.

SELECT a.aid,a.aname,AVG(e.salary) FROM aircraft a,certified c,employees e
WHERE a.aid=c.aid AND c.eid=e.eid AND a.cruising_range>1000 GROUP BY
a.aid,a.aname;

```
+-----+-----------+---------------+
| aid | aname     | AVG(e.salary) |
+-----+-----------+---------------+
| 302 | Boeing    |    73333.3333 |
| 306 | Jet01     |    57500.0000 |
| 378 | Airbus380 |    53333.3333 |
+-----+-----------+---------------+
3 rows in set (0.00 sec)
```

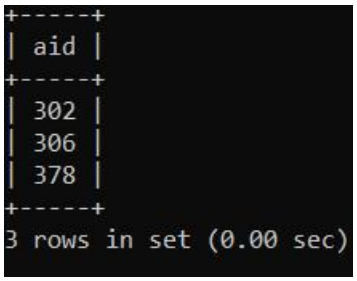
5. Find the names of pilots certified for some Boeing aircraft.

SELECT distinct e.ename FROM employees e,aircraft a,certified c
WHERE e.eid=c.eid AND c.aid=a.aid AND a.aname='Boeing';

```
+--------+
| ename  |
+--------+
| Dinesh |
| Viraj  |
| Rakesh |
+--------+
3 rows in set (0.00 sec)
```

6. Find the aid's of all aircraft that can be used on routes from Bengaluru to New Delhi.

```
SELECT a.aid FROM aircraft a
WHERE a.cruising_range>=(SELECT MIN(f.distance) FROM Flights f
WHERE f.deploc='Bangalore' AND f.arrloc='Delhi');
```



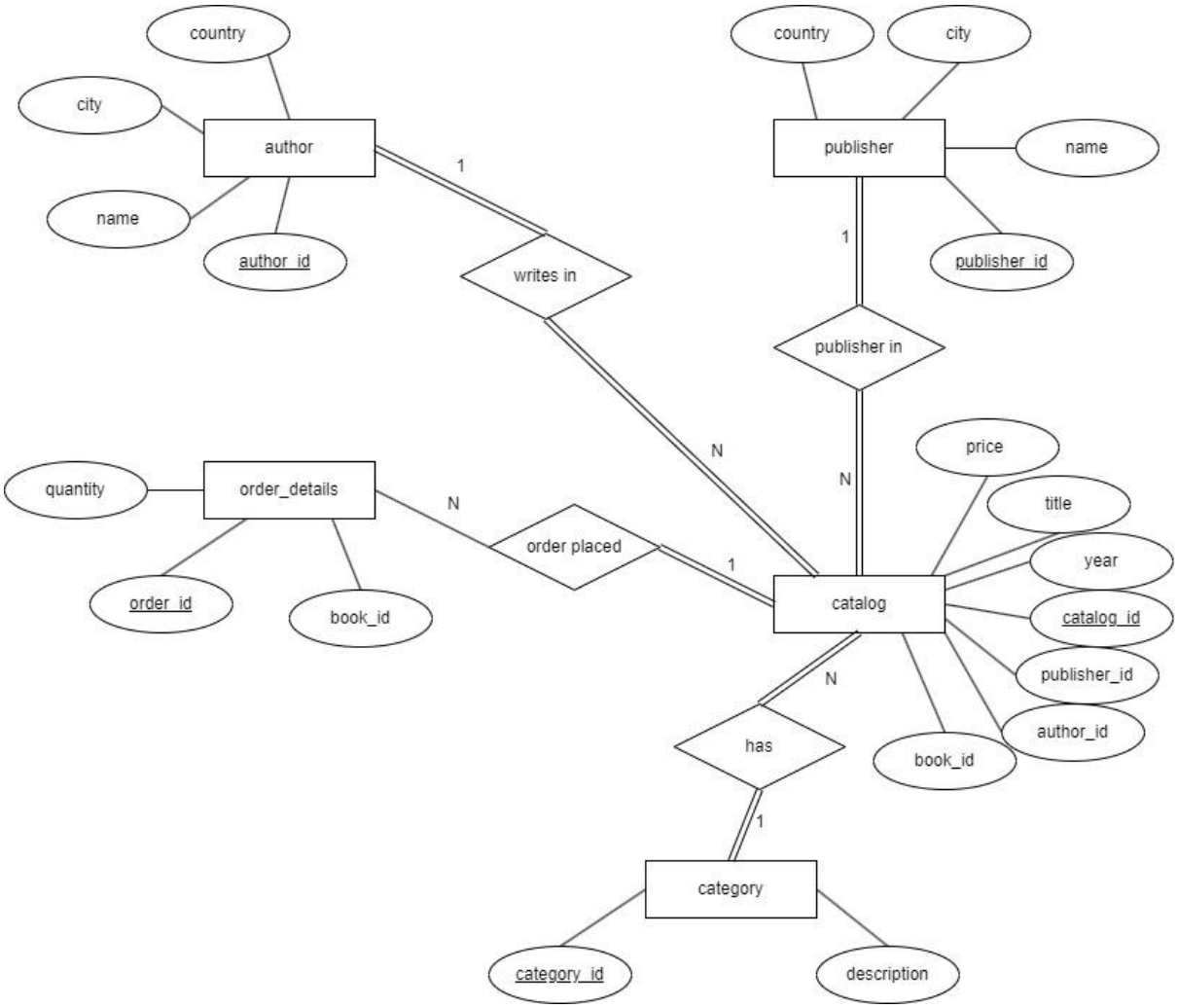
```
+-----+
| aid |
+-----+
| 302 |
| 306 |
| 378 |
+-----+
3 rows in set (0.00 sec)
```

3. The following tables are maintained by a book dealer.

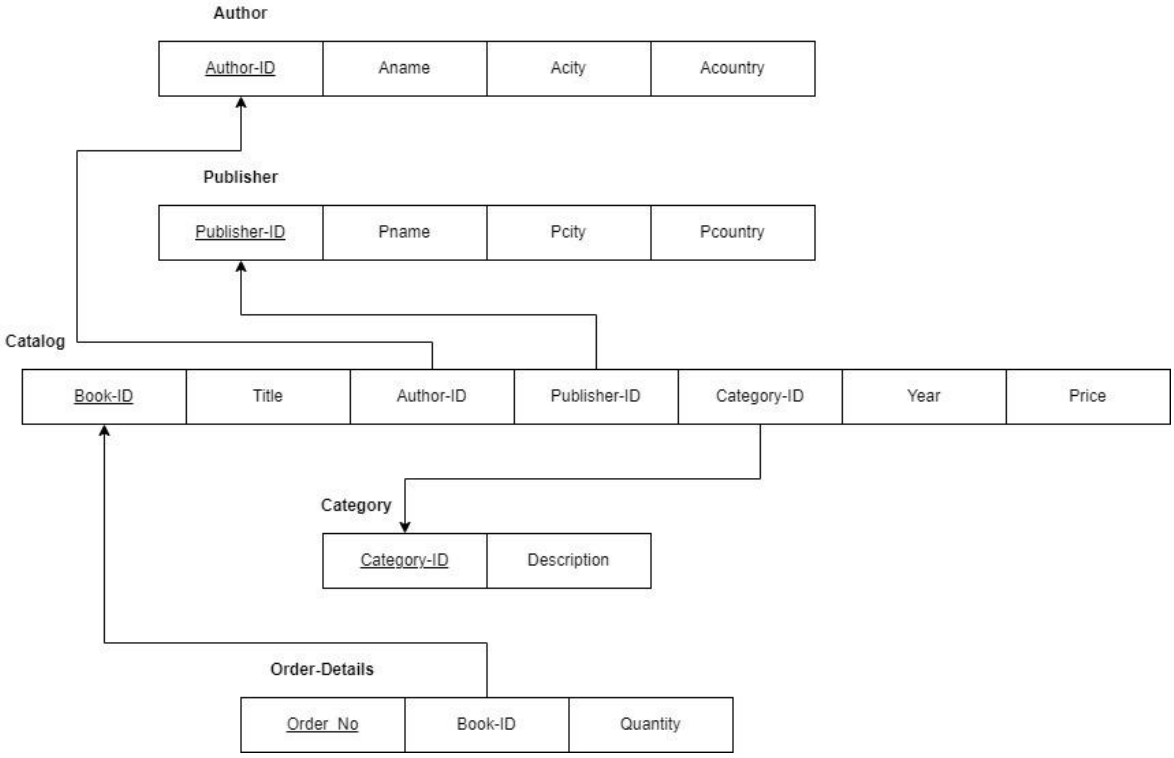
AUTHOR (author-id:int, name:string, city:string, country:string)
PUBLISHER (publisher-id:int, name:string, city:string, country:string)
CATALOG (book-id:int, title:string, author-id:int, publisher-id:int, category-id:int, year:int, price:int)
CATEGORY (category-id:int, description:string)
ORDER-DETAILS (order-no:int, book-id:int, quantity:int)

i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
iv. Find the author of the book which has maximum sales.
v. Demonstrate how you increase the price of books published by a specific publisher by 10%.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

ER Diagram:


country
city
author
name
author_id
1
writes in
country
city
publisher
name
1
publisher_id
publisher in
N
N
price
title
year
catalog_id
publisher_id
author_id
book_id
quantity
order_details
N
order placed
1
order_id
book_id
catalog
N
has
1
category
category_id
description

Schema Diagram:



QUERIES

➢ **Creating relations**

```
CREATE TABLE AUTHOR (
  author_id INT PRIMARY KEY,
  name VARCHAR(10),
  city VARCHAR(10),
  country VARCHAR(10));


CREATE TABLE PUBLISHER (
  publisher_id INT PRIMARY KEY,
  name VARCHAR(10),
  city VARCHAR(10),
  country VARCHAR(10));


CREATE TABLE CATEGORY (
  category_id INT PRIMARY KEY,
  description VARCHAR(10));

CREATE TABLE CATALOG (
  book_id INT PRIMARY KEY,
  title VARCHAR(30),
  author_id INT,
  publisher_id INT,
  category_id INT,
  year INT,
  price INT,
  FOREIGN KEY (author_id) REFERENCES AUTHOR(author_id),
  FOREIGN KEY (publisher_id) REFERENCES PUBLISHER(publisher_id),
  FOREIGN KEY (category_id) REFERENCES CATEGORY(category_id));


CREATE TABLE ORDER_DETAILS (
  order_no INT PRIMARY KEY,
  book_id INT,
  quantity INT,
  FOREIGN KEY (book_id) REFERENCES CATALOG(book_id));
```

➢ **Inserting values**

```
INSERT INTO AUTHOR VALUES
  (1, 'Stephen King', 'Bangor', 'USA'),
  (2, 'J.K. Rowling', 'Edinburgh', 'UK'),
  (3, 'Agatha Christie', 'Torquay', 'UK'),
  (4, 'Dan Brown', 'Exeter', 'USA'),
  (5, 'Harper Lee', 'Monroeville', 'USA');


INSERT INTO PUBLISHER VALUES
  (1, 'Penguin Random House', 'New York', 'USA'),
  (2, 'HarperCollins', 'London', 'UK'),
  (3, 'Hachette Livre', 'Paris', 'France'),
  (4, 'Simon & Schuster', 'New York', 'USA'),
  (5, 'Macmillan Publishers', 'London', 'UK');


INSERT INTO CATEGORY VALUES
  (1, 'Fiction'),
  (2, 'Mystery'),
  (3, 'Thriller'),
  (4, 'Non-Fiction'),
  (5, 'Science Fiction');

INSERT INTO catalogue1 VALUES
  (4001,'HP and Goblet Of Fire',1001,2001,3001,2002,600),
  (4002,'HP and Order Of Phoenix',1001,2002,3001,2005,650),
  (4003,'Two States',1002,2004,3001,2009,65),
  (4004,'3 Mistakes of my life',1002,2004,3001,2007,55),
  (4005,'Da Vinci Code',1004,2003,3001,2004,450),
  (4006,'Angels and Demons',1004,2003,3001,2003,350),
  (4007,'Artificial Intelligence',1003,2002,3002,1970,500);

INSERT INTO orderdetails1 VALUES
  (5001,4001,5),
  (5002,4002,7),
  (5003,4003,15),
  (5004,4004,11),
  (5005,4005,9),
  (5006,4006,8),
  (5007,4007,2),
  (5008,4004,3) ;
```

➢ **Displaying tables :**

SELECT * FROM AUTHOR;

author_id	name	city	country
1	Stephen King	Bangor	USA
2	J.K. Rowling	Edinburgh	UK
3	Agatha Christie	Torquay	UK
4	Dan Brown	Exeter	USA
5	Harper Lee	Monroeville	USA

SELECT * FROM PUBLISHER;

publisher_id	name	city	country
1	Penguin Random House	New York	USA
2	HarperCollins	London	UK
3	Hachette Livre	Paris	France
4	Simon & Schuster	New York	USA
5	Macmillan Publishers	London	UK

SELECT * FROM CATEGORY;

category_id	description
1	Fiction
2	Mystery
3	Thriller
4	Non-Fiction
5	Science Fiction

SELECT * FROM CATALOG;

book_id	title	author_id	publisher_id	category_id	year	price
1	The Shining	1	1	1	1977	20
2	Harry Potter and the Philosopher's Stone	2	2	1	1997	25
3	Murder on the Orient Express	3	3	2	1934	15
4	The Da Vinci Code	4	4	3	2003	30
5	To Kill a Mockingbird	5	5	1	1960	18

SELECT * FROM ORDER_DETAILS;

order_no	book_id	quantity
1	1	10
2	2	15
3	3	8
4	4	20
5	5	12

➢ **Queries**

iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.

```
SELECT * FROM author1
    WHERE author1_id IN
    (SELECT author1_id FROM catalogue1 WHERE
    year>2000 AND price>
    (SELECT AVG(price) FROM catalogue1)
    GROUP BY author1_id HAVING COUNT(*)>1);
```

```
+------------+--------------+--------------+------------------+
| author1_id | author1_name | author1_city | author1_country |
+------------+--------------+--------------+------------------+
|       1001 | JK Rowling   | London       | England          |
+------------+--------------+--------------+------------------+
1 row in set (0.00 sec)
```

iv. Find the author of the book which has maximum sales.

```
SELECT A.name, SUM(OD.quantity) AS total_sales
FROM AUTHOR A
JOIN CATALOG C ON A.author_id = C.author_id
JOIN ORDER_DETAILS OD ON C.book_id = OD.book_id
GROUP BY A.name
ORDER BY total_sales DESC
LIMIT 1;
```

```
+-----------+-------------+
| name      | total_sales |
+-----------+-------------+
| Dan Brown |          20 |
+-----------+-------------+
```

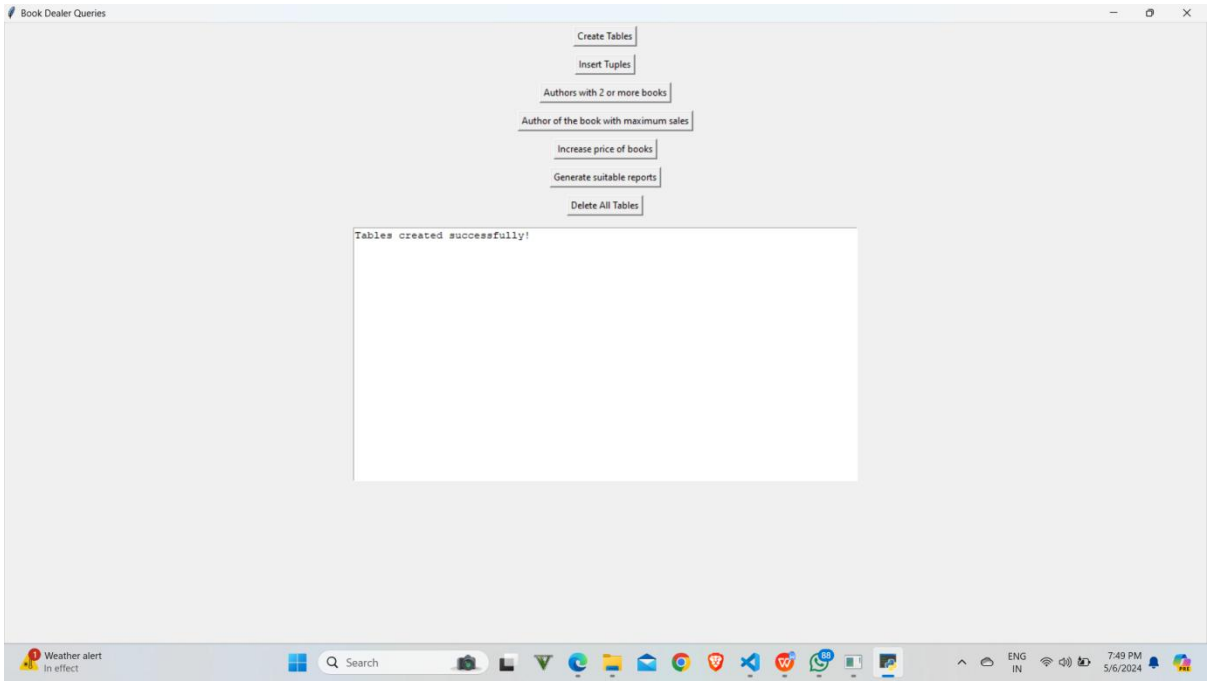
v. Demonstrate how you increase the price of books published by a specific publisher by 10%.

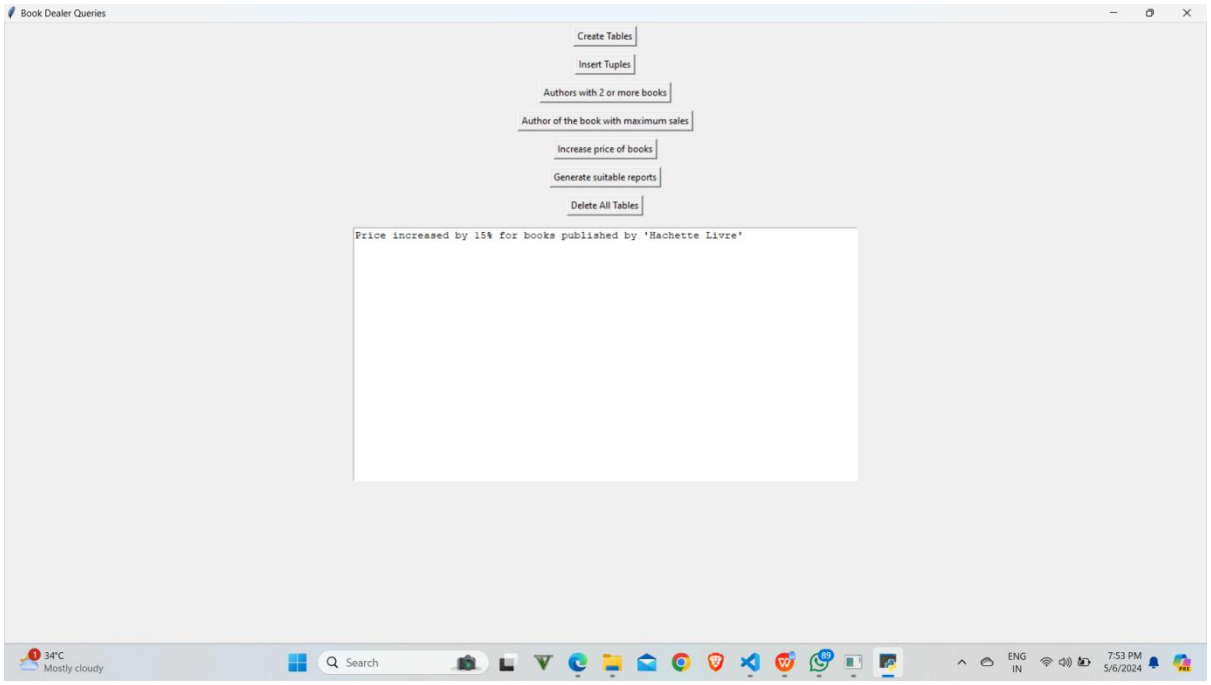
```
UPDATE CATALOG
SET price = price * 1.10
WHERE publisher_id = 4;
```

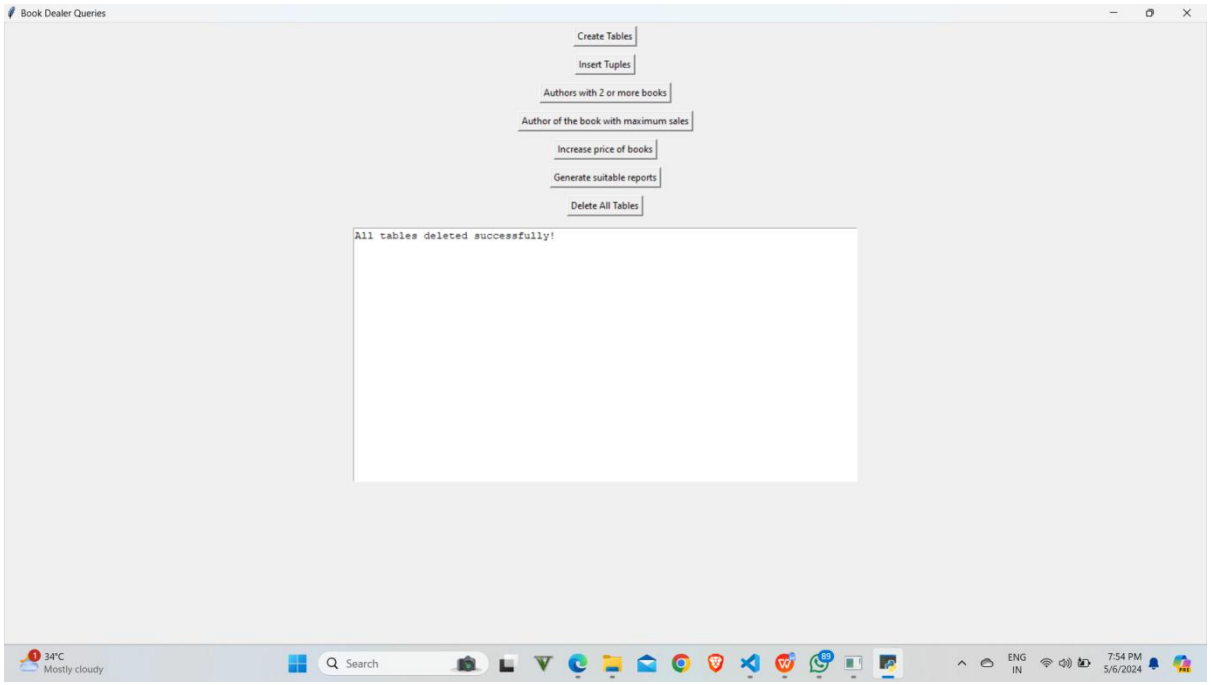
```
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT * FROM CATALOG;
+---------+-----------------------------------------+-----------+--------------+-------------+------+-------
| book_id | title                                   | author_id | publisher_id | category_id | year | price
+---------+-----------------------------------------+-----------+--------------+-------------+------+-------
|       1 | The Shining                             |         1 |            1 |           1 | 1977 |    20
|       2 | Harry Potter and the Philosopher's Stone |         2 |            2 |           1 | 1997 |    25
|       3 | Murder on the Orient Express            |         3 |            3 |           2 | 1934 |    15
|       4 | The Da Vinci Code                       |         4 |            4 |           3 | 2003 |    33
|       5 | To Kill a Mockingbird                   |         5 |            5 |           1 | 1960 |    18
+---------+-----------------------------------------+-----------+--------------+-------------+------+-------
```

vii. Create suitable front end for querying and displaying the results.



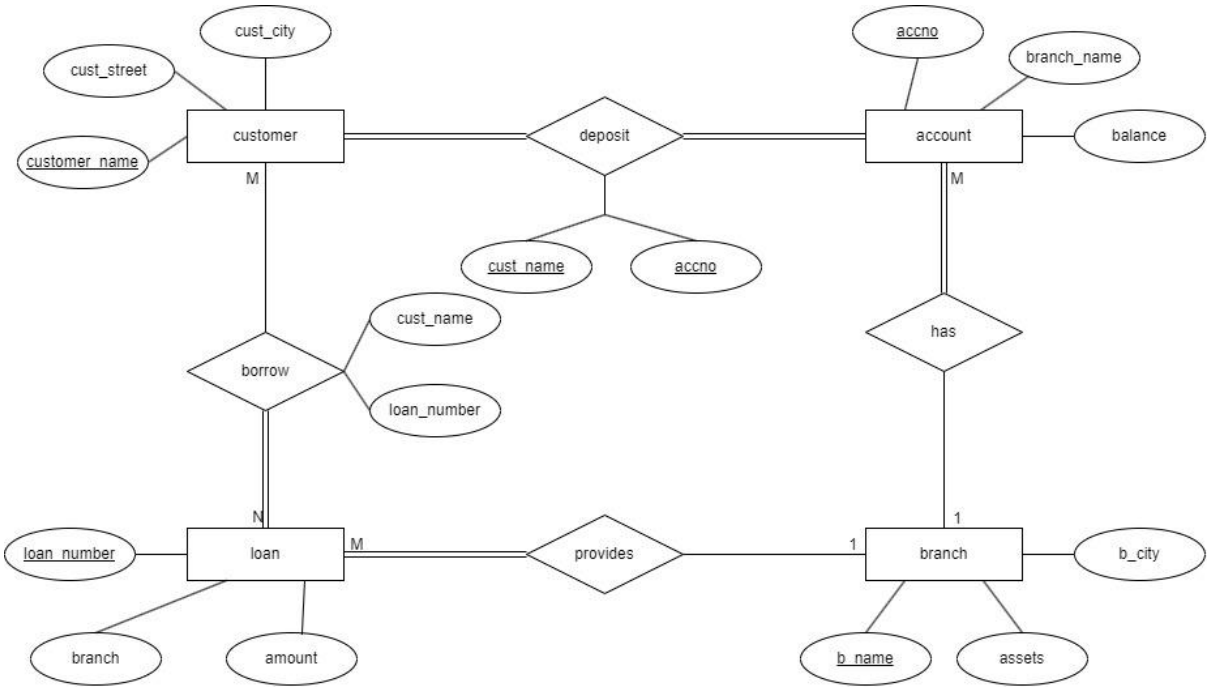

Book Dealer Queries
Create Tables
Insert Tuples
Authors with 2 or more books
Author of the book with maximum sales
Increase price of books
Generate suitable reports
Delete All Tables
Price increased by 15% for books published by 'Hachette Livre'
34°C
Mostly cloudy
Search
ENG
IN
7:53 PM
5/6/2024


Book Dealer Queries
Create Tables
Insert Tuples
Authors with 2 or more books
Author of the book with maximum sales
Increase price of books
Generate suitable reports
Delete All Tables
All tables deleted successfully!
34°C
Mostly cloudy
Search
ENG
IN
7:54 PM
5/6/2024

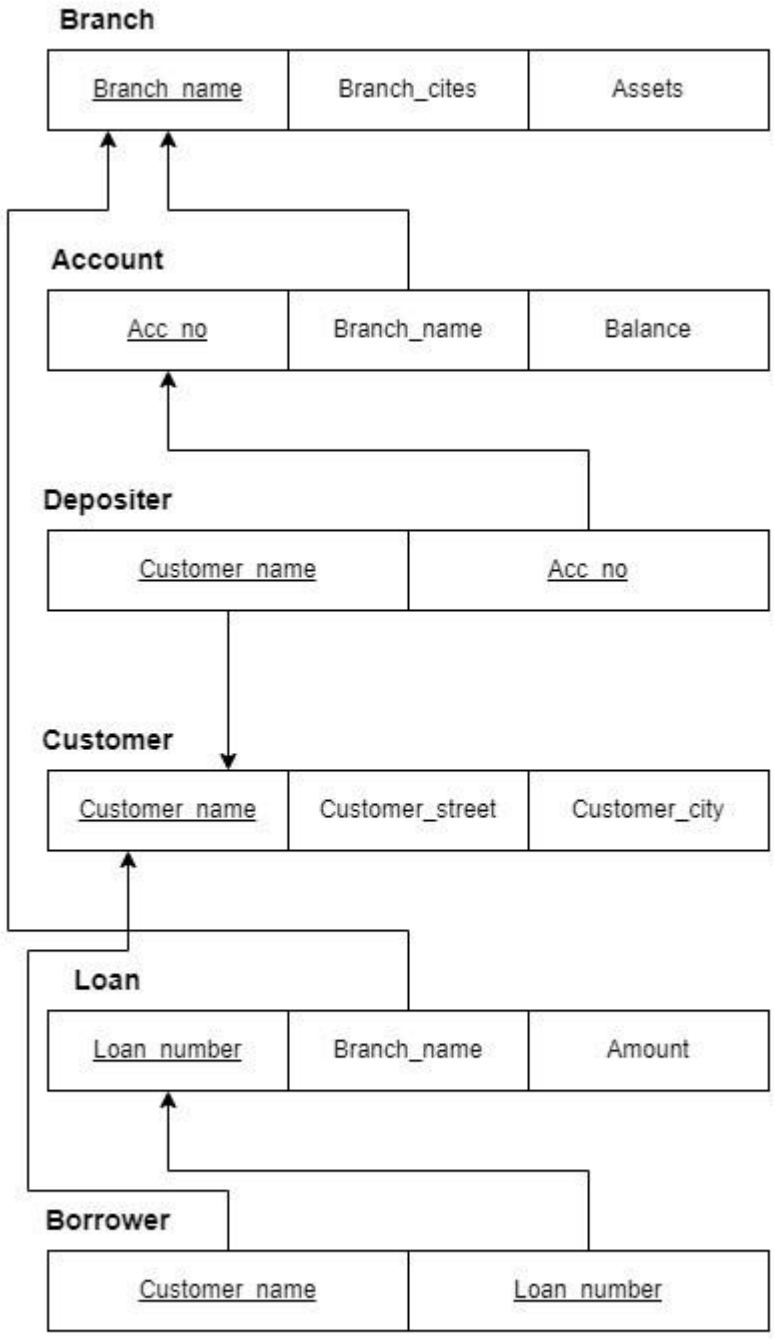
4. Consider the following database for a banking enterprise

BRANCH (branch-name:string, branch-city:string, assets:real)
ACCOUNT (accno:int, branch-name:string, balance:real)
DEPOSITOR (customer-name:string, accno:int)
CUSTOMER (customer-name:string, customer-street:string, customer-city:string)
LOAN (loan-number:int, branch-name:string, amount:real)
BORROWER (customer-name:string, loan-number:int)
i. Create the above tables by properly specifying the primary keys and the foreign keys
ii. Enter at least five tuples for each relation
iii. Find all the customers who have at least two accounts at the *Main* branch.
iv. Find all the customers who have an account at *all* the branches located in a specific city.
v. Demonstrate how you delete all account tuples at every branch located in a specific city.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

ER Diagram:



Schema Diagram:



QUERIES

➢ **Creating relations**

```
CREATE TABLE branch(
  branch_name varchar(10) primary key,
  branch_city varchar(10),
  assets int);
```

```
CREATE TABLE account(
  accno int primary key,
  branch_name varchar(10),
  balance int,
  foreign key(branch_name) references branch(branch_name)on delete cascade);
```

```
CREATE TABLE customer(
  customer_name varchar(15),
  customer_street varchar(10),
  customer_city varchar(10),
  primary key(customer_name));
```

```
CREATE TABLE loan(
  loan_number int primary key,
  branch_name varchar(10),
  amount int,
  foreign key(branch_name) references branch(branch_name));

CREATE TABLE borrower(
  customer_name varchar(10),
  loan_number int,
  primary key(customer_name,loan_number),
  foreign key(customer_name) references customer(customer_name),
  foreign key(loan_number) references loan(loan_number));

CREATE TABLE depositor(
  customer_name varchar(10),
  accno int,
  primary key(customer_name,accno),
  foreign key(customer_name) references customer(customer_name),
  foreign key(accno) references accoumt(accno));
```

➢ **Inserting values**

```
INSERT INTO branch VALUES
  ("Herohalli","bengaluru",10000),
  ("Chamrajpete","bengaluru",25000),
  ("Bantwal","Dakshina Kannada",30000),
  ("Narasimharajapura"," Chikmagalur",40000),
  (" Nanjangudu","Mysore",40000);

INSERT INTO account values
  (10001,"Herohalli",4000),
  (10002,"Chamrajpete",5000),
  (10003,"Bantwal",6000),
  (10004,"Narasimharajapura",7000),
  (10005," Nanjangudu",5000);

INSERT INTO  customer VALUES
  ('sachin',' Dadar','mumbai'),
  ('Rohit','Nagpur','Maharashtra'),
  ('Rahul','Kannanur','Mangaluru'),
  ('Shreyas ','Chembur','mumbai');


INSERT INTO depositor VALUES
  ('sachin',10001),
  ('Rohit',10002),
  ('Rahul',10003),
  ('Shreyas',10004),
  ('virat',10005);
```

➢ **Displaying tables :**

SELECT * FROM branch;

branch_name	branch_city	assets
Herohalli	bengaluru	10000
Chamrajpet	bengaluru	25000
Bantwal	Dakshina K	30000
Narasimhar	Chikmagal	40000
Nanjangud	Mysore	40000

SELECT * FROM account;

accno	branch_name	balance
10001	Herohalli	4000
10002	Chamrajpet	5000
10003	Bantwal	6000
10004	Narasimhar	7000
10005	Nanjangud	5000

SELECT * FROM customer;

customer_name	customer_street	customer_city
sachin	Dadar	mumbai
Rohit	Nagpur	Maharashtr
Rahul	Kannanur	Mangaluru
Shreyas	Chembur	mumbai

SELECT * FROM loan;

loan_number	branch_name	amount
10	Herohalli	10000
11	Chamrajpet	10000
12	Bantwal	13000
13	Narasimhar	25000
14	Nanjangudu	30000

SELECT * FROM borrower;

customer_name	loan_number
Rahul	13
Rohit	12
sachin	11
Shreyas	14
virat	10

SELECT * FROM depositor;

customer_name	accno
Rahul	10003
Rohit	10002
sachin	10001
Shreyas	10004
virat	10005

➢ **Queries**

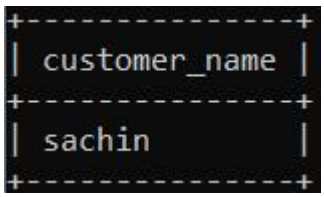
iii. Find all the customers who have at least two accounts at the Main branch.

```
SELECT customer_name
FROM depositor d,account a
WHERE d.accno=a.accno
AND a.branch_name='Main'
GROUP BY d.customer_name
HAVING COUNT(d.customer_name)>=2;
```

Empty set (0.00 sec)

iv. Find all the customers who have an account at all the branches located in a specific city.

```
SELECT d.customer_name
FROM account a,branch b,depositor d
WHERE b.branch_name=a.branch_name AND
a.accno=d.accno AND
b.branch_city='bengaluru'
GROUP BY d.customer_name
HAVING COUNT(distinct b.branch_name)=(
SELECT COUNT(branch_name)
FROM branch
WHERE branch_city='bengaluru');
```



```
+---------------+
| customer_name |
+---------------+
| sachin        |
+---------------+
```

v. Demonstrate how you delete all account tuples at every branch located in a specific city.

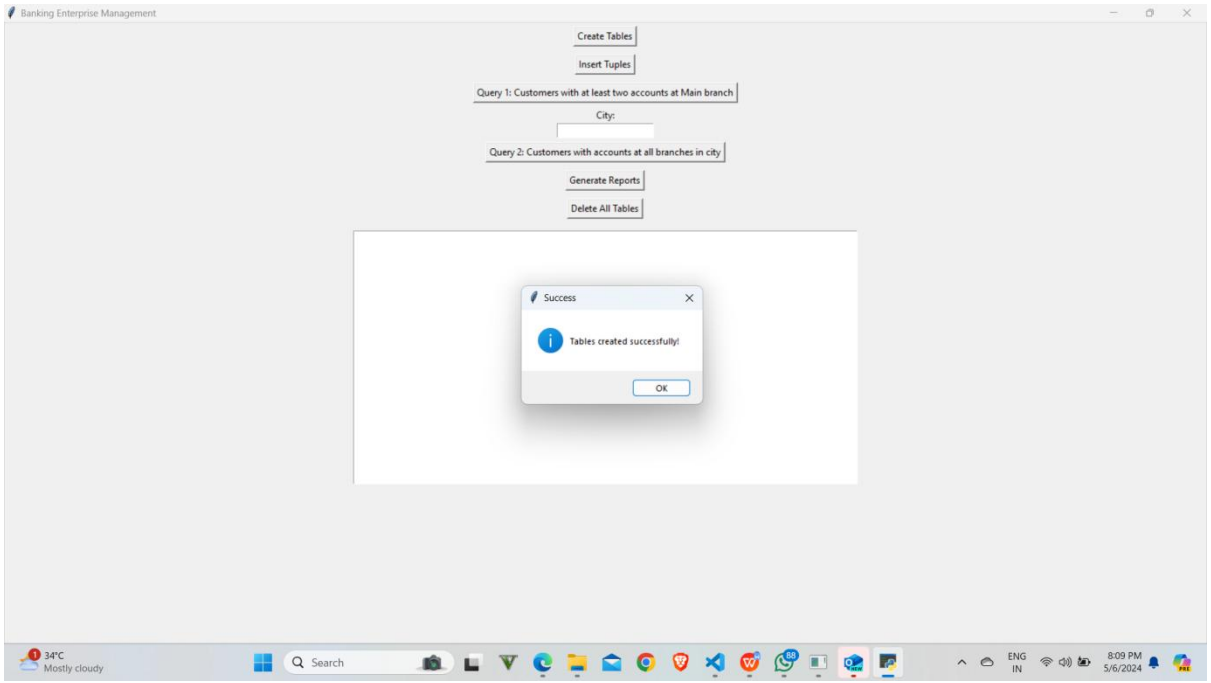
DELETE FROM account WHERE branch_name IN(SELECT branch_name FROM branch WHERE branch_city='Mysore');

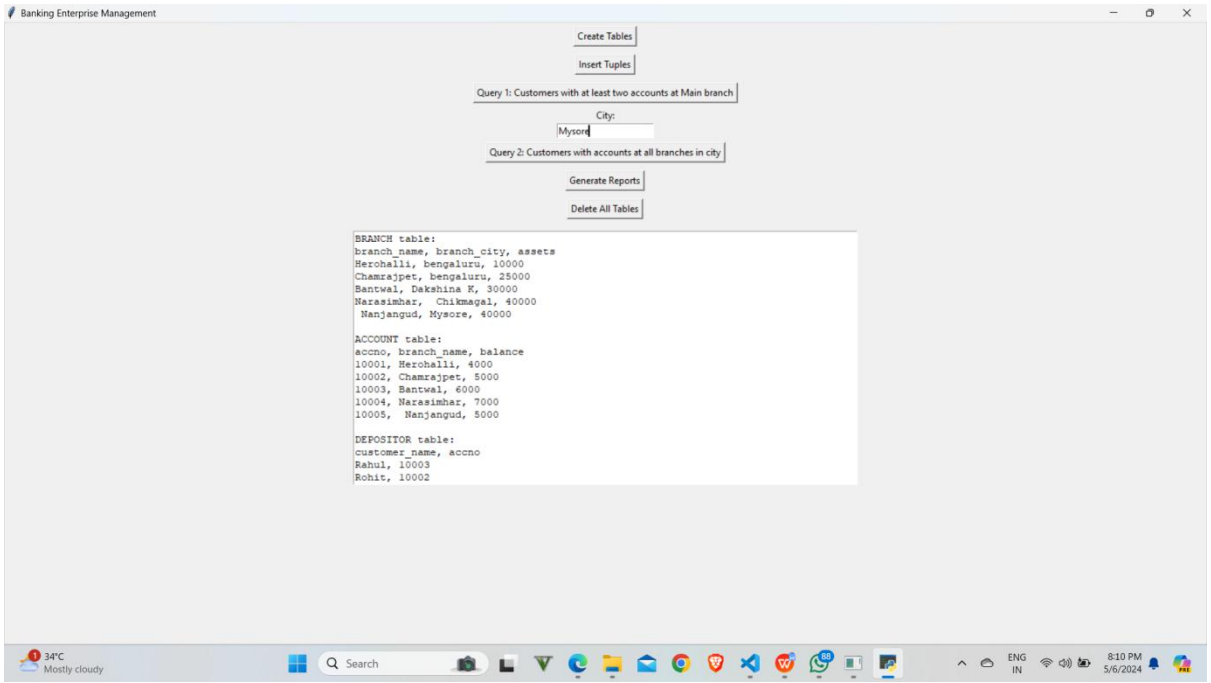
Query OK, 1 row affected (0.04 sec)

mysql>SELECT * FROM account;

```
mysql> select * from account;
+-------+-------------+---------+
| accno | branch_name | balance |
+-------+-------------+---------+
| 10001 | Herohalli   |    4000 |
| 10002 | Chamrajpet  |    5000 |
| 10003 | Bantwal     |    6000 |
| 10004 | Narasimhar  |    7000 |
| 10005 |  Nanjangud  |    5000 |
+-------+-------------+---------+
```

vii. Create suitable front end for querying and displaying the results.



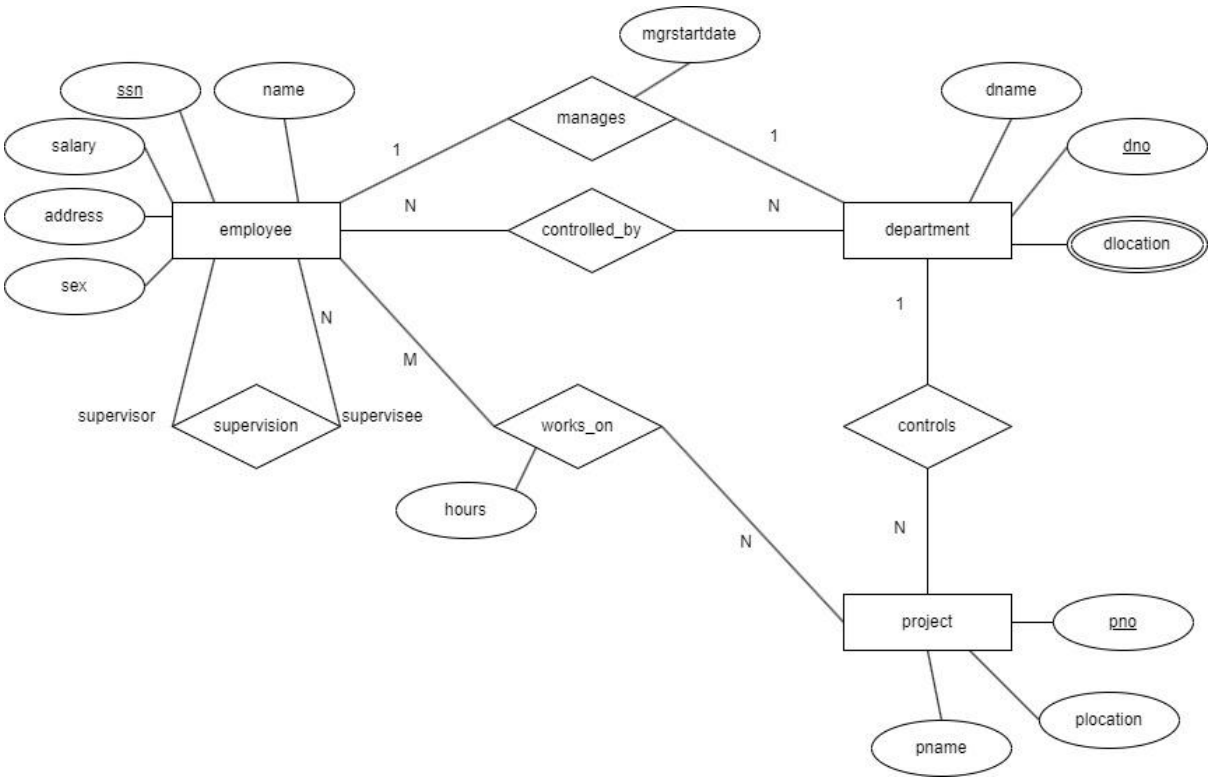


5. Consider the following database a Company. Write the queries in SQL.

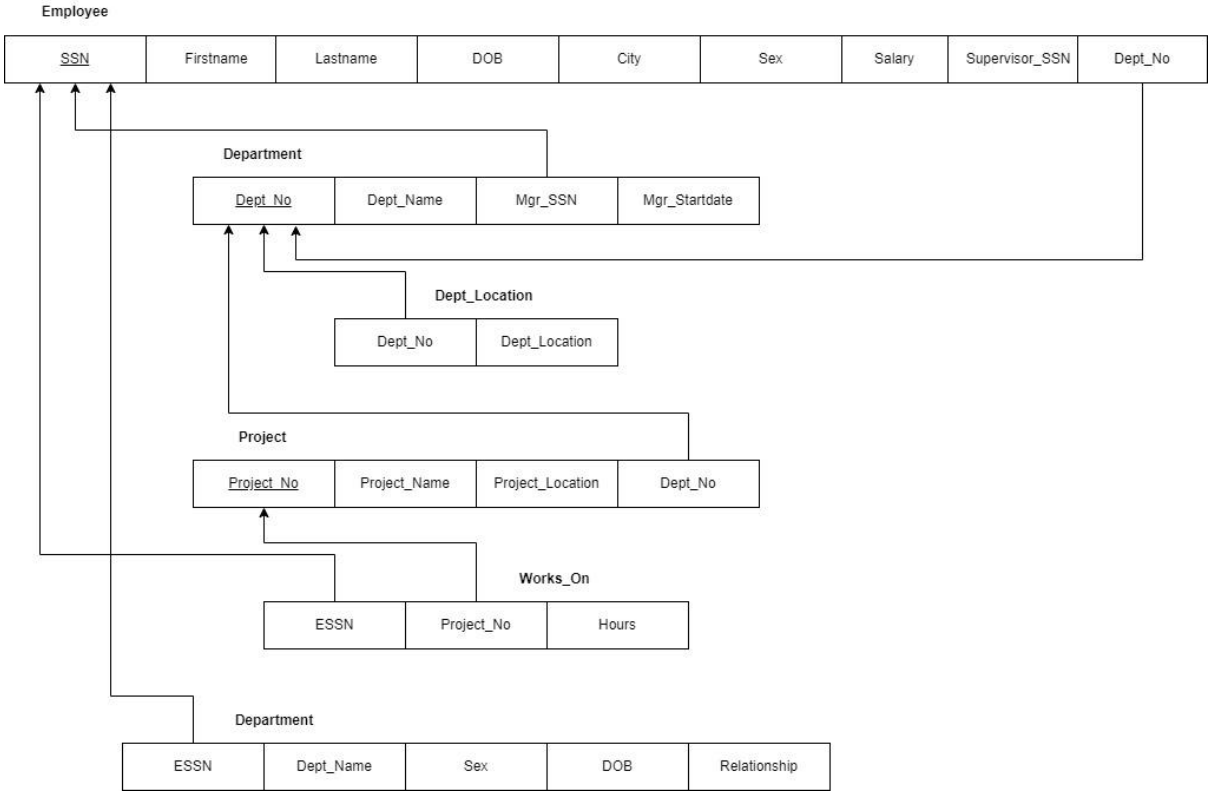
EMPLOYEE(fname, lastname, SSN, Bdate, city, Sex, Salary, Supervisor_SSN, DeptNo)
DEPT(Dname, DeptNo, Mgr_SSN, Mgr_StartDate)
DEPT_Locations(DeptNo, DLocation)
PROJECT(Pname, Pnumber, Plocation, DeptNo)
WORKS_ON(Essn, Pnumber, Hours)
Dependent(Essn, Dependent_Name, Sex, Bdate, Relationship)

i) Create the above tables by properly specifying the primary keys and the foreign keys
ii) Enter at least five tuples for each relation
iii) Retrieve the names of all employees who work in the department that has the employee with the highest salary among the employees.
iv) Retrieve the names of all employees whose supervisor's supervisor has '8888' for SSN.
v) Retrive the names of the employees who make atleast Rs. 5000 more than the employee who is paid the least in the company.
vi) Create a view that has the dept name, manger name, and manager salary for every department.
vii)Create a view that has project name, controlling department name, number of employees, and total hours worked per week on the project for each project.
viii)Create a view that has emp name, supervisor name, and emp salary for each employee who works in the 'Research' Dept.

ER Diagram:


mgrstartdate
ssn
name
manages
dname
salary
1
1
dno
address
employee
N
controlled_by
N
department
dlocation
sex
1
N
M
supervisor
supervision
supervisee
works_on
controls
hours
N
N
project
pno
plocation
pname

Schema Diagram:



QUERIES

➢ **Creating relations**

```
CREATE TABLE EMPLOYEE (
   SSN CHAR(9) PRIMARY KEY,
   Fname VARCHAR(20),
   Lname VARCHAR(20),
   Bdate DATE,
   City VARCHAR(20),
   Sex CHAR,
   Salary DECIMAL(10,2),
   Supervisor_SSN CHAR(9),
   DeptNo INT,
   FOREIGN KEY (Supervisor_SSN) REFERENCES EMPLOYEE(SSN),
   FOREIGN KEY (DeptNo) REFERENCES DEPT(DeptNo));

CREATE TABLE DEPT (
   DeptNo INT PRIMARY KEY,
   Dname VARCHAR(20),
   Mgr_SSN CHAR(9),
   Mgr_StartDate DATE,
   FOREIGN KEY (Mgr_SSN) REFERENCES EMPLOYEE(SSN));

CREATE TABLE DEPT_Locations (
   DeptNo INT,
   DLocation VARCHAR(20),
   PRIMARY KEY (DeptNo, DLocation),
   FOREIGN KEY (DeptNo) REFERENCES DEPT(DeptNo));

CREATE TABLE PROJECT (
   Pnumber INT PRIMARY KEY,
   Pname VARCHAR(20),
   Plocation VARCHAR(20),
   DeptNo INT,
   FOREIGN KEY (DeptNo) REFERENCES DEPT(DeptNo));

CREATE TABLE WORKS_ON (
   Essn CHAR(9),
   Pnumber INT,
   Hours DECIMAL(5,2),
   PRIMARY KEY (Essn, Pnumber),
   FOREIGN KEY (Essn) REFERENCES EMPLOYEE(SSN),
   FOREIGN KEY (Pnumber) REFERENCES PROJECT(Pnumber));
```

```
CREATE TABLE Dependent (
   Essn CHAR(9),
   Dependent_Name VARCHAR(20),
   Sex CHAR,
   Bdate DATE,
   Relationship VARCHAR(20),
   PRIMARY KEY (Essn, Dependent_Name),
   FOREIGN KEY (Essn) REFERENCES EMPLOYEE(SSN));
```

- **Inserting values**

```
INSERT INTO EMPLOYEE VALUES
   ('123456789', 'Virat', 'Kohli', '1990-01-01', 'Dehli', 'M', 60000, NULL, 1),
   ('234567890', 'Smrithi', 'Mandanna', '1995-05-05', 'Mumbai', 'F', 55000, '123456789', 1),
   ('345678901', 'Shubhman', 'Gill', '1985-10-10', 'Gujrat', 'M', 70000, '123456789', 2),
   ('456789012', 'Shreyanka', 'Patil', '1988-03-15', 'Bangaluru', 'F', 65000, '345678901', 2),
   ('567890123', 'David', 'Warner', '1992-07-20', 'Pune', 'M', 62000, '345678901', 3);

INSERT INTO DEPT VALUES
   (1, 'HR', '123456789', '2000-01-01'),
   (2, 'Finance', '345678901', '2001-01-01'),
   (3, 'Research', '567890123', '2002-01-01');


INSERT INTO DEPT_Locations VALUES
  (1, 'Dehli'),
  (2, 'Mumbais'),
  (3, 'Gujrat');


INSERT INTO PROJECT VALUES
  (1, 'Project A', 'Dehli', 1),
  (2, 'Project B', 'Mumbai', 2),
  (3, 'Project C', 'Gujrat', 3);




INSERT INTO WORKS_ON VALUES
   ('234567890', 1, 40),
   ('345678901', 1, 30),
   ('345678901', 2, 35),
   ('456789012', 2, 25),
   ('567890123', 3, 45);
```

```
INSERT INTO Dependent VALUES
   ('234567890', 'Renuka', 'F', '2010-01-01', 'Daughter'),
   ('345678901', 'Devraj', 'M', '2012-05-05', 'Son'),
   ('345678901', 'Rohit', 'M', '2014-07-07', 'Son'),
   ('456789012', 'Richa', 'F', '2016-10-10', 'Daughter'),
   ('567890123', 'Pooja', 'F', '2018-12-12', 'Daughter');
```

- **Displaying tables**

SELECT * FROM EMPLOYEE;

SSN	Fname	Lname	Bdate	City	Sex	Salary	Supervisor_SSN	DeptNo
123456789	Virat	Kohli	1990-01-01	Dehli	M	60000.00	NULL	1
234567890	Smrithi	Mandanna	1995-05-05	Mumbai	F	55000.00	123456789	1
345678901	Shubhman	Gill	1985-10-10	Gujrat	M	70000.00	123456789	2
456789012	Shreyanka	Patil	1988-03-15	Bangaluru	F	65000.00	345678901	2
567890123	David	Warner	1992-07-20	Pune	M	62000.00	345678901	3

SELECT * FROM DEPT;

DeptNo	Dname	Mgr_SSN	Mgr_StartDate
1	HR	123456789	2000-01-01
2	Finance	345678901	2001-01-01
3	Research	567890123	2002-01-01

SELECT * FROM DEPT_Locations;

DeptNo	DLocation
1	Dehli
2	Mumbais
3	Gujrat

SELECT * FROM PROJECT;

Pnumber	Pname	Plocation	DeptNo
1	Project A	Dehli	1
2	Project B	Mumbai	2
3	Project C	Gujrat	3

SELECT * FROM WORKS_ON;

Essn	Pnumber	Hours
234567890	1	40.00
345678901	1	30.00
345678901	2	35.00
456789012	2	25.00
567890123	3	45.00

SELECT * FROM Dependent;

Essn	Dependent_Name	Sex	Bdate	Relationship
234567890	Renuka	F	2010-01-01	Daughter
345678901	Devraj	M	2012-05-05	Son
345678901	Rohit	M	2014-07-07	Son
456789012	Richa	F	2016-10-10	Daughter
567890123	Pooja	F	2018-12-12	Daughter

➢ **Queries**

iii) Retrieve the names of all employees who work in the department that has the employee with the highest salary among the employees.

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E
WHERE E.DeptNo = (SELECT DeptNo    FROM EMPLOYEE
   WHERE Salary = (SELECT MAX(Salary) FROM EMPLOYEE));
```

Fname	Lname
Shubhman	Gill
Shreyanka	Patil

iv) Retrieve the names of all employees whose supervisor's supervisor has '8888' for SSN.

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E
JOIN EMPLOYEE S ON E.Supervisor_SSN = S.SSN
JOIN EMPLOYEE SS ON S.Supervisor_SSN = SS.SSN
WHERE SS.SSN = '8888';
```

Empty set (0.00 sec)

v) Retrive the names of the employees who make atleast Rs. 5000 more than the employee who is paid the least in the company.

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E
WHERE E.Salary >= (
   SELECT MIN(Salary) + 5000
   FROM EMPLOYEE);
```

Fname	Lname
Virat	Kohli
Shubhman	Gill
Shreyanka	Patil
David	Warner

vi) Create a view that has the dept name, manger name, and manager salary for every department.

```
CREATE VIEW DeptManagerInfo AS
SELECT D.Dname, E.Fname AS Manager_Fname, E.Lname AS Manager_Lname, E.Salary
AS Manager_Salary
FROM DEPT D
JOIN EMPLOYEE E ON D.Mgr_SSN = E.SSN;
```

Query OK, 0 rows affected (0.01 sec)

```
SELECT * FROM  DeptManagerInfo;
```

Dname	Manager_Fname	Manager_Lname	Manager_Salary
HR	Virat	Kohli	60000.00
Finance	Shubhman	Gill	70000.00
Research	David	Warner	62000.00

vii)Create a view that has project name, controlling department name, number of employees, and total hours worked per week on the project for each project.

```
CREATE VIEW ProjectSummary AS
SELECT P.Pname, D.Dname AS Controlling_Dept, COUNT(W.Essn) AS Num_Employees,
SUM(W.Hours) AS Total_Hours_Per_Week
FROM PROJECT P
JOIN DEPT D ON P.DeptNo = D.DeptNo
JOIN WORKS_ON W ON P.Pnumber = W.Pnumber
GROUP BY P.Pname, D.Dname;
```

Query OK, 0 rows affected (0.01 sec)

SELECT * FROM ProjectSummary;

Pname	Controlling_Dept	Num_Employees	Total_Hours_Per_Week
Project A	HR	2	70.00
Project B	Finance	2	60.00
Project C	Research	1	45.00

viii)Create a view that has emp name, supervisor name, and emp salary for each employee who works in the ‘Research’ Dept.

```
CREATE VIEW ResearchEmployeeInfo AS
SELECT E.Fname AS Emp_Fname, E.Lname AS Emp_Lname, S.Fname AS
Supervisor_Fname, S.Lname AS Supervisor_Lname, E.Salary AS Emp_Salary
FROM EMPLOYEE E
JOIN DEPT D ON E.DeptNo = D.DeptNo
JOIN EMPLOYEE S ON E.Supervisor_SSN = S.SSN
WHERE D.Dname = 'Research';
```

Query OK, 0 rows affected (0.01 sec)

SELECT * FROM ResearchEmployeeInfo;

Emp_Fname	Emp_Lname	Supervisor_Fname	Supervisor_Lname	Emp_Salary
David	Warner	Shubhman	Gill	62000.00