Q)Construct an ER diagram for a travel agency

Main Entities and Their Attributes

1. **Customer**
    - Customer_ID (PK)
    - Name
    - Email
    - Phone
    - Address
2. **Package**
    - Package_ID (PK)
    - Destination
    - Description
    - Duration
    - Price
3. **Booking**
    - Booking_ID (PK)
    - Booking_Date
    - Total_Amount
    - Customer_ID (FK)
    - Package_ID (FK)
4. **Payment**
    - Payment_ID (PK)
    - Payment_Date
    - Amount
    - Payment_Method
    - Booking_ID (FK)
5. **Travel_Agent**
    - Agent_ID (PK)
    - Name
    - Email
    - Phone
6. **Feedback**
    - Feedback_ID (PK)
    - Comments
    - Rating
    - Customer_ID (FK)
    - Package_ID (FK)

⚯ Relationships

- A **Customer** can make **many Bookings** → 1-to-many
- A **Package** can be booked in **many Bookings** → 1-to-many
- A **Booking** can have **one Payment**
- A **Travel Agent** can manage **many Bookings** (optional) → 1-to-many
- A **Customer** can leave **Feedback** for multiple **Packages** → many-to-many

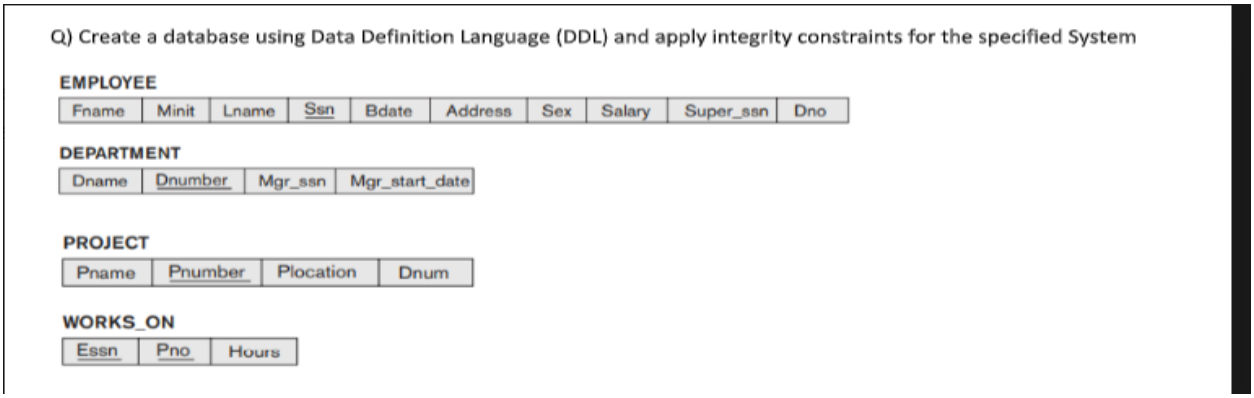
Q) Construct an ER diagram for a Library management system

Main Entities and Attributes

1. **Book**
    - Book_ID (PK)
    - Title
    - Author
    - Publisher
    - ISBN
    - Genre
    - Year_Published
    - Quantity
2. **Member**
    - Member_ID (PK)
    - Name
    - Email
    - Phone
    - Address
    - Membership_Date
3. **Loan**
    - Loan_ID (PK)
    - Issue_Date
    - Due_Date
    - Return_Date
    - Book_ID (FK)
    - Member_ID (FK)
4. **Librarian**
    - Librarian_ID (PK)
    - Name
    - Email
    - Phone
5. **Fine**
    - Fine_ID (PK)
    - Amount
    - Date_Issued
    - Loan_ID (FK)

⚯ Relationships

- A **Member** can borrow multiple **Books** via **Loans** → 1-to-many
- A **Book** can be issued in many **Loans** → 1-to-many
- A **Loan** can incur **one Fine** (optional) → 1-to-1 (optional)
- A **Librarian** can manage many **Loans** → 1-to-many




```
CREATE TABLE EMPLOYEE (
    Fname VARCHAR(30) NOT NULL,
    Minit CHAR(1),
    Lname VARCHAR(30) NOT NULL,
    Ssn CHAR(9) PRIMARY KEY,
    Bdate DATE,
    Address VARCHAR(100),
    Sex CHAR(1),
    Salary DECIMAL(10, 2),
    Super_ssn CHAR(9),
```

```
    Dno INT,
    FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
    FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
);

-- DEPARTMENT Table
CREATE TABLE DEPARTMENT (
    Dname VARCHAR(50) NOT NULL,
    Dnumber INT PRIMARY KEY,
    Mgr_ssn CHAR(9),
    Mgr_start_date DATE,
    FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
);

-- PROJECT Table
CREATE TABLE PROJECT (
    Pname VARCHAR(50) NOT NULL,
    Pnumber INT PRIMARY KEY,
    Plocation VARCHAR(50),
    Dnum INT,
    FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber)
);

-- WORKS_ON Table
CREATE TABLE WORKS_ON (
    Essn CHAR(9),
    Pno INT,
```

```
    Hours DECIMAL(5,2),
    PRIMARY KEY (Essn, Pno),
    FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
    FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber)
);
```

Q) Create a database using Data Definition Language (DDL) and apply integrity constraints for the specified System

Company(Company_id, Name, Address)

Customer(Customer_id, Name, Address, phone, Insurance_company)

Car(Car_Number, Car_Model, Owner_id)

Accidents (Accident_id, Car_Number, Location, date, time)

```
-- COMPANY Table
CREATE TABLE Company (
    Company_id INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(200)
);

-- CUSTOMER Table
CREATE TABLE Customer (
    Customer_id INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(200),
    Phone VARCHAR(15),
    Insurance_company INT,
    FOREIGN KEY (Insurance_company) REFERENCES Company(Company_id)
);
```

```
-- CAR Table
CREATE TABLE Car (
    Car_Number VARCHAR(20) PRIMARY KEY,
    Car_Model VARCHAR(50),
    Owner_id INT,
    FOREIGN KEY (Owner_id) REFERENCES Customer(Customer_id)
);

-- ACCIDENTS Table
CREATE TABLE Accidents (
    Accident_id INT PRIMARY KEY,
    Car_Number VARCHAR(20),
    Location VARCHAR(100),
    Date DATE,
    Time TIME,
    FOREIGN KEY (Car_Number) REFERENCES Car(Car_Number)
);
```

Q) Create and insert four rows in the following relations. Write a query to modify the salary of each employee by incrementing with 20%

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

```
CREATE TABLE DEPARTMENT (
    Dname VARCHAR(50) NOT NULL,
```

```
    Dnumber INT PRIMARY KEY,
    Mgr_ssn CHAR(9),
    Mgr_start_date DATE
);

CREATE TABLE EMPLOYEE (
    Fname VARCHAR(30) NOT NULL,
    Minit CHAR(1),
    Lname VARCHAR(30) NOT NULL,
    Ssn CHAR(9) PRIMARY KEY,
    Bdate DATE,
    Address VARCHAR(100),
    Sex CHAR(1),
    Salary DECIMAL(10, 2),
    Super_ssn CHAR(9),
    Dno INT,
    FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
    FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
);
-- Insert into DEPARTMENT
INSERT INTO DEPARTMENT VALUES
('HR', 1, '123456789', '2020-01-01'),
('IT', 2, '987654321', '2021-02-15'),
('Finance', 3, '456123789', '2019-07-01'),
('Marketing', 4, '789456123', '2022-06-30');

-- Insert into EMPLOYEE
```

```
INSERT INTO EMPLOYEE VALUES
('John', 'A', 'Doe', '123456789', '1990-05-15', '123 Elm St', 'M', 50000, NULL, 1),
('Jane', 'B', 'Smith', '987654321', '1988-09-23', '456 Oak St', 'F', 60000, '123456789', 2),
('Alice', 'C', 'Brown', '456123789', '1992-11-30', '789 Pine St', 'F', 55000, '987654321', 3),
('Bob', 'D', 'White', '789456123', '1985-04-10', '321 Maple St', 'M', 52000, '456123789', 4);
UPDATE EMPLOYEE
SET Salary = Salary * 1.2;
```

Q) Create and insert four rows in the following relations. Write a query to remove all the projects belonging to any one department.

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

```
-- PROJECT table
CREATE TABLE PROJECT (
    Pname VARCHAR(50) NOT NULL,
    Pnumber INT PRIMARY KEY,
    Plocation VARCHAR(100),
    Dnum INT
);

-- WORKS_ON table
CREATE TABLE WORKS_ON (
    Essn CHAR(9),
```

```
    Pno INT,
    Hours DECIMAL(5,2),
    PRIMARY KEY (Essn, Pno),
    FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber)
);
-- Insert into PROJECT
INSERT INTO PROJECT VALUES
('Project A', 101, 'New York', 1),
('Project B', 102, 'Chicago', 2),
('Project C', 103, 'San Francisco', 1),
('Project D', 104, 'Houston', 3);

-- Insert into WORKS_ON
INSERT INTO WORKS_ON VALUES
('123456789', 101, 20.5),
('987654321', 102, 15.0),
('456123789', 103, 18.0),
('789456123', 104, 25.0);
DELETE FROM PROJECT
WHERE Dnum = 1;
```

Q) Create and insert four rows in the following relations. Write a query to change the address of all the customers with the name beginning with letter "A".

Company(Company_id, Name, Address)

Customer(Customer_id, Name, Address, phone, Insurance_company)

```
-- COMPANY table
CREATE TABLE Company (
```

```
    Company_id INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(200)
);

-- CUSTOMER table
CREATE TABLE Customer (
    Customer_id INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(200),
    Phone VARCHAR(15),
    Insurance_company INT,
    FOREIGN KEY (Insurance_company) REFERENCES Company(Company_id)
);
-- Insert into COMPANY
INSERT INTO Company VALUES
(1, 'SafeGuard Insurance', '101 Market St'),
(2, 'SecureLife Corp', '202 River Rd'),
(3, 'Aegis Insurance', '303 Hilltop Blvd'),
(4, 'TrustShield Inc.', '404 Valley View');

-- Insert into CUSTOMER
INSERT INTO Customer VALUES
(101, 'Alice Johnson', '12 Apple St', '1234567890', 1),
(102, 'Bob Smith', '34 Berry Blvd', '2345678901', 2),
(103, 'Andrew Lee', '56 Cherry Ln', '3456789012', 3),
(104, 'Clara Davis', '78 Date Dr', '4567890123', 4);
```

```
UPDATE Customer
SET Address = 'Updated Address'
WHERE Name LIKE 'A%';
```

Q) Create and insert four rows in the following relations. Write a query to delete all cars owned by a single owner.

Car(Car_Number, Car_Model, Owner_id)

Accidents (Accident_id, Car_Number, Location, date, time)

```
-- CAR table
CREATE TABLE Car (
    Car_Number VARCHAR(15) PRIMARY KEY,
    Car_Model VARCHAR(50),
    Owner_id INT
);

-- ACCIDENTS table
CREATE TABLE Accidents (
    Accident_id INT PRIMARY KEY,
    Car_Number VARCHAR(15),
    Location VARCHAR(100),
    date DATE,
    time TIME,
    FOREIGN KEY (Car_Number) REFERENCES Car(Car_Number)
);
-- Insert into CAR
INSERT INTO Car VALUES
```

('MH12AB1234', 'Toyota Corolla', 1),

('MH12AB5678', 'Honda Civic', 2),

('MH12CD9012', 'Ford Focus', 1),

('MH12EF3456', 'Hyundai Elantra', 3);

-- Insert into ACCIDENTS

INSERT INTO Accidents VALUES

(101, 'MH12AB1234', 'Mumbai', '2024-01-10', '10:00:00'),

(102, 'MH12AB5678', 'Pune', '2024-02-12', '12:30:00'),

(103, 'MH12CD9012', 'Nashik', '2024-03-15', '08:45:00'),

(104, 'MH12EF3456', 'Nagpur', '2024-04-01', '14:15:00');

DELETE FROM Car

WHERE Owner_id = 1;

Q) Create and insert four rows in the following relations. Write a query to find average salary of all employees

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

-- DEPARTMENT table

CREATE TABLE Department (

Dname VARCHAR(50),

Dnumber INT PRIMARY KEY,

Mgr_ssn CHAR(9),

Mgr_start_date DATE

);

-- EMPLOYEE table

```
CREATE TABLE Employee (
   Fname VARCHAR(50),
   Minit CHAR(1),
   Lname VARCHAR(50),
   Ssn CHAR(9) PRIMARY KEY,
   Bdate DATE,
   Address VARCHAR(200),
   Sex CHAR(1),
   Salary DECIMAL(10, 2),
   Super_ssn CHAR(9),
   Dno INT,
   FOREIGN KEY (Dno) REFERENCES Department(Dnumber),
   FOREIGN KEY (Super_ssn) REFERENCES Employee(Ssn)
);
-- Insert into DEPARTMENT
INSERT INTO Department VALUES
('HR', 1, '123456789', '2022-01-01'),
('IT', 2, '987654321', '2022-02-01'),
('Finance', 3, '112233445', '2022-03-01'),
('Marketing', 4, '556677889', '2022-04-01');

-- Insert into EMPLOYEE
INSERT INTO Employee VALUES
('John', 'A', 'Doe', '111111111', '1990-01-01', '123 Elm St', 'M', 50000, NULL, 1),
('Jane', 'B', 'Smith', '222222222', '1985-02-02', '456 Oak St', 'F', 60000, '111111111', 2),
('Alice', 'C', 'Brown', '333333333', '1992-03-03', '789 Pine St', 'F', 55000, '111111111', 3),
('Bob', 'D', 'Johnson', '444444444', '1988-04-04', '321 Maple St', 'M', 65000, '222222222', 4);
```

```
SELECT AVG(Salary) AS Average_Salary
FROM Employee;
```

Q) Create and insert four rows in the following relations. Write a query to count the number of project belonging to each department.

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

```
-- PROJECT table
CREATE TABLE Project (
   Pname VARCHAR(50),
   Pnumber INT PRIMARY KEY,
   Plocation VARCHAR(100),
   Dnum INT
);

-- WORKS_ON table
CREATE TABLE Works_On (
   Essn CHAR(9),
   Pno INT,
   Hours DECIMAL(5, 2),
   FOREIGN KEY (Pno) REFERENCES Project(Pnumber)
);
-- Insert into PROJECT
INSERT INTO Project VALUES
('AI Research', 101, 'New York', 1),
('Web Dev', 102, 'Chicago', 2),
```

```
('App Dev', 103, 'Seattle', 1),
('Security Audit', 104, 'Boston', 3);

-- Insert into WORKS_ON
INSERT INTO Works_On VALUES
('111111111', 101, 20),
('222222222', 102, 25),
('333333333', 103, 30),
('444444444', 104, 15);

SELECT Dnum, COUNT(*) AS Project_Count
FROM Project
GROUP BY Dnum;
```

Q) Create and insert four rows in the following relations. Write a query to find number of customers in each insurance company

Company(Company_id, Name, Address)

Customer(Customer_id, Name, Address, phone, Insurance_company)

```
-- COMPANY table
CREATE TABLE Company (
    Company_id INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(200)
);

-- CUSTOMER table
CREATE TABLE Customer (
```

```
    Customer_id INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(200),
    Phone VARCHAR(15),
    Insurance_company VARCHAR(100)
);
-- Insert into COMPANY
INSERT INTO Company VALUES
(1, 'SafeLife Insurance', 'New York'),
(2, 'HealthFirst Co.', 'Boston'),
(3, 'SecureCare', 'Chicago'),
(4, 'LifeShield', 'Seattle');

-- Insert into CUSTOMER
INSERT INTO Customer VALUES
(101, 'Alice Johnson', 'NY', '1234567890', 'SafeLife Insurance'),
(102, 'Bob Smith', 'LA', '2345678901', 'HealthFirst Co.'),
(103, 'Amy Adams', 'TX', '3456789012', 'SafeLife Insurance'),
(104, 'Daniel Grey', 'CA', '4567890123', 'SecureCare');

SELECT Insurance_company, COUNT(*) AS Customer_Count
FROM Customer
GROUP BY Insurance_company;
```

Q) Create and insert four rows in the following relations. Write a query to perform Equi Join.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

```
-- DEPARTMENT table
CREATE TABLE DEPARTMENT (
    Dname VARCHAR(50),
    Dnumber INT PRIMARY KEY,
    Mgr_ssn VARCHAR(9),
    Mgr_start_date DATE
);

-- EMPLOYEE table
CREATE TABLE EMPLOYEE (
    Fname VARCHAR(50),
    Minit CHAR(1),
    Lname VARCHAR(50),
    Ssn VARCHAR(9) PRIMARY KEY,
    Bdate DATE,
    Address VARCHAR(200),
    Sex CHAR(1),
    Salary DECIMAL(10, 2),
    Super_ssn VARCHAR(9),
    Dno INT,
    FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
```

```
);
-- Insert into DEPARTMENT
INSERT INTO DEPARTMENT VALUES
('HR', 1, '123456789', '2015-01-01'),
('IT', 2, '987654321', '2018-03-15'),
('Finance', 3, '555666777', '2019-07-01'),
('Marketing', 4, '888999000', '2020-11-11');

-- Insert into EMPLOYEE
INSERT INTO EMPLOYEE VALUES
('Alice', 'A', 'Brown', '111111111', '1990-01-01', 'NY', 'F', 60000, NULL, 1),
('Bob', 'B', 'Smith', '222222222', '1985-06-15', 'LA', 'M', 75000, '111111111', 2),
('Cathy', 'C', 'Jones', '333333333', '1992-09-10', 'TX', 'F', 65000, '222222222', 3),
('David', 'D', 'Lee', '444444444', '1988-12-22', 'CA', 'M', 70000, '333333333', 2);
SELECT
   E.Fname, E.Lname, E.Ssn, D.Dname, D.Dnumber
FROM
   EMPLOYEE E, DEPARTMENT D
WHERE
   E.Dno = D.Dnumber;
```

Q) Create and insert four rows in the following relations. Write a query to perform Natural Join.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

```
CREATE TABLE EMPLOYEE (
   Fname VARCHAR(20),
   Minit CHAR(1),
   Lname VARCHAR(20),
   Ssn CHAR(9) PRIMARY KEY,
   Bdate DATE,
   Address VARCHAR(50),
   Sex CHAR(1),
   Salary DECIMAL(10,2),
   Super_ssn CHAR(9),
   Dno INT
);

CREATE TABLE DEPARTMENT (
   Dname VARCHAR(20),
   Dnumber INT PRIMARY KEY,
   Mgr_ssn CHAR(9),
   Mgr_start_date DATE
);
-- Inserting into EMPLOYEE
INSERT INTO EMPLOYEE VALUES
('John', 'A', 'Doe', '123456789', '1990-05-14', 'Pune', 'M', 50000, NULL, 1),
('Jane', 'B', 'Smith', '987654321', '1992-07-20', 'Mumbai', 'F', 60000, '123456789', 2),
('Alex', 'C', 'Brown', '456789123', '1988-03-30', 'Delhi', 'M', 55000, '987654321', 1),
('Sara', 'D', 'Lee', '789123456', '1995-11-11', 'Nagpur', 'F', 48000, '123456789', 3);

-- Inserting into DEPARTMENT
```

```
INSERT INTO DEPARTMENT VALUES
('HR', 1, '123456789', '2020-01-01'),
('IT', 2, '987654321', '2021-02-15'),
('Finance', 3, '789123456', '2019-06-20'),
('Admin', 4, '456789123', '2022-09-05');
-- Alter DEPARTMENT table column for Natural Join (only if necessary)
ALTER TABLE DEPARTMENT RENAME COLUMN Dnumber TO Dno;

-- Natural Join Query
SELECT *
FROM EMPLOYEE
NATURAL JOIN DEPARTMENT;
```

Q) Create and insert four rows in the following relations. Write a query to perform Left Outer Join.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

```
CREATE TABLE EMPLOYEE (
    Fname VARCHAR(20),
    Minit CHAR(1),
    Lname VARCHAR(20),
    Ssn CHAR(9) PRIMARY KEY,
    Bdate DATE,
    Address VARCHAR(50),
    Sex CHAR(1),
    Salary DECIMAL(10,2),
    Super_ssn CHAR(9),
```

```
    Dno INT
);

CREATE TABLE DEPARTMENT (
    Dname VARCHAR(20),
    Dnumber INT PRIMARY KEY,
    Mgr_ssn CHAR(9),
    Mgr_start_date DATE
);
-- Inserting into EMPLOYEE
INSERT INTO EMPLOYEE VALUES
('John', 'A', 'Doe', '123456789', '1990-05-14', 'Pune', 'M', 50000, NULL, 1),
('Jane', 'B', 'Smith', '987654321', '1992-07-20', 'Mumbai', 'F', 60000, '123456789', 2),
('Alex', 'C', 'Brown', '456789123', '1988-03-30', 'Delhi', 'M', 55000, '987654321', 3),
('Sara', 'D', 'Lee', '789123456', '1995-11-11', 'Nagpur', 'F', 48000, '123456789', NULL);  -- No department

-- Inserting into DEPARTMENT
INSERT INTO DEPARTMENT VALUES
('HR', 1, '123456789', '2020-01-01'),
('IT', 2, '987654321', '2021-02-15'),
('Finance', 3, '456789123', '2019-06-20'),
('Admin', 4, '789123456', '2022-09-05');

SELECT *
FROM EMPLOYEE E
LEFT OUTER JOIN DEPARTMENT D
ON E.Dno = D.Dnumber;
```

Q) Create and insert four rows in the following relations. Write a query to perform Right Outer Join.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

```
CREATE TABLE EMPLOYEE (
   Fname VARCHAR(20),
   Minit CHAR(1),
   Lname VARCHAR(20),
   Ssn CHAR(9) PRIMARY KEY,
   Bdate DATE,
   Address VARCHAR(50),
   Sex CHAR(1),
   Salary DECIMAL(10,2),
   Super_ssn CHAR(9),
   Dno INT
);

CREATE TABLE DEPARTMENT (
   Dname VARCHAR(20),
   Dnumber INT PRIMARY KEY,
   Mgr_ssn CHAR(9),
   Mgr_start_date DATE
);
-- Inserting into EMPLOYEE
INSERT INTO EMPLOYEE VALUES
('John', 'A', 'Doe', '123456789', '1990-05-14', 'Pune', 'M', 50000, NULL, 1),
```

```
('Jane', 'B', 'Smith', '987654321', '1992-07-20', 'Mumbai', 'F', 60000, '123456789', 2),
('Alex', 'C', 'Brown', '456789123', '1988-03-30', 'Delhi', 'M', 55000, '987654321', 3),
('Sara', 'D', 'Lee', '789123456', '1995-11-11', 'Nagpur', 'F', 48000, '123456789', NULL);  -- No department

-- Inserting into DEPARTMENT
INSERT INTO DEPARTMENT VALUES
('HR', 1, '123456789', '2020-01-01'),
('IT', 2, '987654321', '2021-02-15'),
('Finance', 3, '456789123', '2019-06-20'),
('Admin', 4, '789123456', '2022-09-05');
SELECT *
FROM EMPLOYEE E
RIGHT OUTER JOIN DEPARTMENT D
ON E.Dno = D.Dnumber;
```