```
create database assignment5;
use assignment5;
CREATE TABLE Department (
   DeptID INT PRIMARY KEY,
   DepartmentName VARCHAR(50),
   Location VARCHAR(50)
);

CREATE TABLE Employee (
   EmpID INT PRIMARY KEY,
   EmpName VARCHAR(50),
   DeptID INT,
   Salary DECIMAL(10,2),
   Job VARCHAR(50),
   City VARCHAR(50),
   JoiningDate DATE,
   FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
);

-- Project Table
CREATE TABLE Project (
   ProjectID INT PRIMARY KEY,
   ProjectName VARCHAR(50),
   DeptID INT,
   Budget DECIMAL(15,2),
   StartDate DATE,
   EndDate DATE,
   FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
);

INSERT INTO Department (DeptID, DepartmentName, Location)
VALUES
(10, 'Sales', 'Mumbai'),
(20, 'HR', 'Delhi'),
(30, 'IT', 'Pune'),
(40, 'Finance', 'Mumbai'),
(50, 'Marketing', 'Bangalore');

INSERT INTO Employee (EmpID, EmpName, DeptID, Salary, Job, City,
JoiningDate)
VALUES
(101, 'Rahul', 10, 55000, 'Salesperson', 'Mumbai', '2022-01-15'),
(102, 'Priya', 10, 60000, 'Salesperson', 'Pune', '2023-03-10'),
(103, 'Amit', 20, 48000, 'HR Executive', 'Delhi', '2021-07-20'),
(104, 'Neha', 30, 75000, 'Developer', 'Mumbai', '2022-11-05'),
(105, 'Sanjay', 40, 65000, 'Accountant', 'Mumbai', '2022-03-25');

INSERT INTO Project (ProjectID, ProjectName, DeptID, Budget, StartDate, EndDate)
VALUES
(201, 'Sales Expansion', 10, 500000, '2023-01-01', '2023-12-31'),
```

(202, 'HR Onboarding', 20, 200000, '2023-02-01', '2023-08-31'),
(203, 'Website Upgrade', 30, 300000, '2023-03-01', '2023-09-30'),
(204, 'Annual Audit', 40, 150000, '2023-04-01', '2023-10-31'),

1.(205, 'Marketing Campaign', 50, 250000, '2023-05-01', '2023-11-30');
Write a Relational Algebra expression to perform Cartesian Product (Cross Join) between Emp and Dept tables.

>>SELECT *FROM Employee CROSS JOIN Department;

2. Write a SQL query to perform an **INNER JOIN** between Emp and Dept tables displaying EmpName and DeptName.

>>SELECT E.EmpName, D.DepartmentName FROM Employee E INNER JOIN Department D ON E.DeptID = D.DeptID;

3. Write a SQL query to perform a **LEFT OUTER JOIN** between Emp and Dept tables to display all employees along with their department information, including employees without a department.

>>SELECT E.EmpID, E.EmpName, E.DeptID, E.Salary, E.Job, E.City, E.JoiningDate, D.DepartmentName FROM Employee E LEFT JOIN Department D ON E.DeptID = D.DeptID;

4. Write a SQL query to perform a **RIGHT OUTER JOIN** between Emp and Dept tables.

>>SELECT E.EmpID, E.EmpName, E.DeptID, E.Salary, E.Job, E.City, E.JoiningDate, D.DepartmentName FROM Emp E RIGHT JOIN Department D ON E.DeptID = D.DeptID;

5. Write a SQL query to perform a **NATURAL JOIN** between Emp and Dept.
>>SELECT EmpID, EmpName, DeptID, Salary, Job, City, JoiningDate, DepartmentName FROM Employee NATURAL JOIN Department;

6.Write a SQL query to perform a **CROSS JOIN** between Emp and Project table.

>>SELECT E.EmpID, E.EmpName, E.DeptID, E.Salary, E.Job, E.City, E.JoiningDate,P.ProjectID, P.ProjectName, P.DeptID AS ProjectDeptID, P.Budget, P.StartDate, P.EndDate

FROM Employee E

CROSS JOIN Project P;

7.Write a SQL query to **create a new table EmpBackup with the same structure as Emp but no data**.

>>CREATE TABLE EmpBackup LIKE Employee;

8.Write a SQL query to **copy all data from Emp into EmpBackup table**.

>>INSERT INTO EmpBackup SELECT * FROM Employee;

9. Write a SQL query to **create a new table ProjectArchive with the same structure and data as Project**.

>>CREATE TABLE ProjectArchive AS SELECT * FROM Project;

10.Write a SQL query to create an **AUTO_INCREMENT sequence for EmpID in the Emp table** during table creation.

```
>>CREATE TABLE Emp (

EmpID INT NOT NULL AUTO_INCREMENT,

EmpName VARCHAR(50),

DeptID INT,

Salary DECIMAL(10,2),

Job VARCHAR(50),

City VARCHAR(50),

JoiningDate DATE,

PRIMARY KEY (EmpID),

FOREIGN KEY (DeptID) REFERENCES Department(DeptID));
```

11.Write a SQL query using a **subquery** to find all employees whose salary is greater than the **average salary of all employees**.

```
>>SELECT *FROM Employee

WHERE Salary > (

   SELECT AVG(Salary)

   FROM Employee
```

);

12.Write a SQL query using a **subquery in WHERE clause** to find employees working in departments located in 'Mumbai'.

```
>>SELECT *FROM Employee

WHERE DeptID IN (

SELECT DeptID

FROM Department

WHERE city = 'Mumbai'

);
```

13.Write a SQL query to display all departments where the number of employees is greater than **5**, using a subquery.

```
>>SELECT *FROM Department

WHERE DeptID IN (

 SELECT DeptID

  FROM Employee

GROUP BY DeptID

 HAVING COUNT(*) > 5

);
```

14. Write a subqueries for find out Max salary of employee.

```
>>SELECT *

FROM Employee

WHERE Salary = (

    SELECT MAX(Salary)

    FROM Emp

);
```

15.Write a self join query for each table.

```
SELECT E1.EmpName AS Employee1, E2.EmpName AS Employee2, E1.DeptID

FROM Employee E1

JOIN Employee E2 ON E1.DeptID = E2.DeptID AND E1.EmpID <> E2.EmpID;
```