-- Create Employee Table

CREATE TABLE Employee (

Emp\_id INT AUTO\_INCREMENT PRIMARY KEY,

Dept\_id INT,

Emp\_fname VARCHAR(50),

Emp\_lname VARCHAR(50),

Emp\_Position VARCHAR(50),

Emp\_salary DECIMAL(10, 2),

Emp\_JoinDate DATE,

FOREIGN KEY (Dept\_id) REFERENCES Dept(Dept\_id) ON DELETE CASCADE

);

-- Create Dept Table

CREATE TABLE Dept (

Dept\_id INT PRIMARY KEY,

Dept\_name VARCHAR(50),

Dept\_location VARCHAR(50)

);

-- Create Project Table

CREATE TABLE Project (

Proj\_id INT AUTO\_INCREMENT PRIMARY KEY,

Dept\_id INT,

Proj\_Name VARCHAR(100),

Proj\_Location VARCHAR(50),

Proj\_cost DECIMAL(12, 2),

Proj\_year YEAR,

FOREIGN KEY (Dept\_id) REFERENCES Dept(Dept\_id) ON DELETE CASCADE

);

CREATE VIEW EmployeeView AS

SELECT Emp\_id, Emp\_fname, Emp\_lname, Emp\_Position FROM Employee;

CREATE INDEX idx\_emp\_fname ON Employee(Emp\_fname);

CREATE INDEX idx\_proj\_location\_year ON Project(Proj\_Location, Proj\_year);

1. **Display all Employee details with Department 'Computer' and 'IT' and Employee first name starting with 'p' or 'h'.**

SELECT \* FROM Employee

JOIN Dept ON Employee.Dept\_id = Dept.Dept\_id

WHERE Dept\_name IN ('Computer', 'IT')

AND (Emp\_fname LIKE 'p%' OR Emp\_fname LIKE 'h%');

1. **List the number of different Employee Positions.**

SELECT COUNT(DISTINCT Emp\_Position) AS NumberOfPositions FROM Employee;

1. **Give 10% increase in Salary of Employees who joined before 1985.**

UPDATE Employee

SET Emp\_salary = Emp\_salary \* 1.10

WHERE YEAR(Emp\_JoinDate) < 1985;

7. **Find the project with the maximum cost and the average of all Project costs.**

SELECT Proj\_Name, MAX(Proj\_cost) AS MaxCost FROM Project;

SELECT AVG(Proj\_cost) AS AverageCost FROM Project;

STUDENT SCHEMA

-- Create PlacementDrive Table

CREATE TABLE PlacementDrive (

Drive\_id INT AUTO\_INCREMENT PRIMARY KEY,

Pcompany\_name VARCHAR(100),

package DECIMAL(10, 2),

location VARCHAR(50)

);

-- Create Training Table

CREATE TABLE Training (

T\_id INT AUTO\_INCREMENT PRIMARY KEY,

Tcompany\_name VARCHAR(100),

T\_Fee DECIMAL(10, 2),

T\_year YEAR

);

-- Create Student Table

CREATE TABLE Student (

s\_id INT AUTO\_INCREMENT PRIMARY KEY,

Drive\_id INT,

T\_id INT,

s\_name VARCHAR(100),

CGPA DECIMAL(3, 2),

s\_branch VARCHAR(50),

s\_dob DATE,

FOREIGN KEY (Drive\_id) REFERENCES PlacementDrive(Drive\_id) ON DELETE CASCADE,

FOREIGN KEY (T\_id) REFERENCES Training(T\_id) ON DELETE CASCADE

);

6. **Find the student names who joined training on 1-1-2019 and on 1-1-2021.**

SELECT s\_name FROM Student

JOIN Training ON Student.T\_id = Training.T\_id

WHERE (T\_year = 2019 OR T\_year = 2021) AND s\_dob = '2019-01-01';

7. **Find the student name with the highest CGPA and names of students with a CGPA between 7 and 9.**

-- Student with the highest CGPA

SELECT s\_name FROM Student WHERE CGPA = (SELECT MAX(CGPA) FROM Student);

-- Students with CGPA between 7 and 9

SELECT s\_name FROM Student WHERE CGPA BETWEEN 7 AND 9;

**Display all students' names with T\_id in decreasing order of T\_Fee.**

SELECT s\_name, Student.T\_id

FROM Student

JOIN Training ON Student.T\_id = Training.T\_id

ORDER BY T\_Fee DESC;