**Internship Task - RDBMS and SQL Task #4**

**SQL case-based assignment with a scenario and 10 queries for an HR Management Application. Scenario You are managing a database for an HR management system. The system tracks employees, their departments, salaries, and performance reviews. The database includes the following tables:**

CREATE DATABASE HR\_Management;

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(100) ,

LastName VARCHAR(100) ,

Email VARCHAR(100) ,

Phone VARCHAR(20),

HireDate DATE ,

ManagerID INT,

Salary DECIMAL(10, 2) ,

FOREIGN KEY (ManagerID) REFERENCES Employees(EmployeeID)

);

ALTER TABLE Employees

ADD DepartmentID INT;

ALTER TABLE Employees

ADD CONSTRAINT FK\_Employees\_DepartmentID FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, Phone, HireDate, ManagerID, Salary)

VALUES

(1, 'Amit', 'Sharma', 'amit.sharma@example.com', '9876543210', '2020-01-15', NULL, 60000),

(2, 'Priya', 'Singh', 'priya.singh@example.com', '9876543211', '2019-06-10', 1, 55000),

(3, 'Rahul', 'Verma', 'rahul.verma@example.com', '9876543212', '2021-03-20', 1, 50000),

(4, 'Sneha', 'Patel', 'sneha.patel@example.com', '9876543213', '2018-11-25', 2, 70000),

(5, 'Vikas', 'Gupta', 'vikas.gupta@example.com', '9876543214', '2022-05-14', 2, 45000),

(6, 'Anjali', 'Rao', 'anjali.rao@example.com', '9876543215', '2020-09-30', NULL, 65000),

(7, 'Deepak', 'Yadav', 'deepak.yadav@example.com', '9876543216', '2021-07-19', 3, 48000),

(8, 'Kavita', 'Mehta', 'kavita.mehta@example.com', '9876543217', '2023-02-11', 6, 52000),

(9, 'Manoj', 'Desai', 'manoj.desai@example.com', '9876543218', '2020-12-01', 6, 50000),

(10, 'Neha', 'Jain', 'neha.jain@example.com', '9876543219', '2019-04-18', 4, 60000),

(11, 'Amit', 'Sharma', 'amit.sharma@example.com', '9876543210', '2020-01-15', NULL, 60000), -- Duplicate

(12, 'Priya', 'Singh', 'priya.singh@example.com', '9876543211', '2019-06-10', 1, 55000), -- Duplicate

(13, 'Sneha', 'Patel', 'sneha.patel@example.com', '9876543213', '2018-11-25', 2, 70000), -- Duplicate

(14, 'Deepak', 'Yadav', 'deepak.yadav@example.com', '9876543216', '2021-07-19', 3, 48000), -- Duplicate

(15, 'Kavita', 'Mehta', 'kavita.mehta@example.com', '9876543217', '2023-02-11', 6, 52000); -- Duplicate

UPDATE Employees SET DepartmentID = 1 WHERE EmployeeID =1

UPDATE Employees SET DepartmentID = 3 WHERE EmployeeID =2

UPDATE Employees SET DepartmentID = 5 WHERE EmployeeID =3

UPDATE Employees SET DepartmentID = 7 WHERE EmployeeID =4

UPDATE Employees SET DepartmentID = 8 WHERE EmployeeID =5

UPDATE Employees SET DepartmentID = 9 WHERE EmployeeID =6

UPDATE Employees SET DepartmentID = 10 WHERE EmployeeID =7

UPDATE Employees SET DepartmentID = 6 WHERE EmployeeID =8

UPDATE Employees SET DepartmentID = 4 WHERE EmployeeID =9

UPDATE Employees SET DepartmentID = 2 WHERE EmployeeID =10

UPDATE Employees SET DepartmentID = 1 WHERE EmployeeID =11

UPDATE Employees SET DepartmentID = 4 WHERE EmployeeID =12

UPDATE Employees SET DepartmentID = 8 WHERE EmployeeID =13

UPDATE Employees SET DepartmentID = 9 WHERE EmployeeID =14

UPDATE Employees SET DepartmentID = 7 WHERE EmployeeID =15

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100) NOT NULL,

ManagerID INT,

FOREIGN KEY (ManagerID) REFERENCES Employees(EmployeeID)

);

INSERT INTO Departments (DepartmentID, DepartmentName, ManagerID)

VALUES

(1, 'HR', 1),

(2, 'Finance', 2),

(3, 'IT', 3),

(4, 'Sales', 4),

(5, 'Operations', 6),

(6, 'Marketing', 1), -- Duplicate ManagerID

(7, 'Logistics', 2), -- Duplicate ManagerID

(8, 'R&D', 3), -- Duplicate ManagerID

(9, 'Legal', 4), -- Duplicate ManagerID

(10, 'Support', 6); -- Duplicate ManagerID

CREATE TABLE PerformanceReviews (

ReviewID INT PRIMARY KEY,

EmployeeID INT,

ReviewDate DATE ,

PerformanceScore VARCHAR(20) ,

Comments VARCHAR(20),

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

INSERT INTO PerformanceReviews (ReviewID, EmployeeID, ReviewDate, PerformanceScore, Comments)

VALUES

(1, 1, '2023-01-10', 'Excellent', 'Great work!'),

(2, 2, '2023-01-15', 'Good', 'Consistent performance.'),

(3, 3, '2023-01-20', 'Average', 'Needs improvement.'),

(4, 4, '2023-01-25', 'Excellent', 'Exceptional leadership.'),

(5, 5, '2023-02-01', 'Good', 'Solid contribution.'),

(6, 6, '2023-02-05', 'Excellent', 'Great work!'), -- Duplicate Score

(7, 7, '2023-02-10', 'Good', 'Consistent performance.'), -- Duplicate Score

(8, 8, '2023-02-15', 'Average', 'Needs improvement.'), -- Duplicate Score

(9, 9, '2023-02-20', 'Poor', 'Requires close monitoring.'),

(10, 10, '2023-02-25', 'Good', 'Improving steadily.'),

(11, 11, '2023-01-10', 'Excellent', 'Great work!'), -- Duplicate

(12, 12, '2023-01-15', 'Good', 'Consistent performance.'), -- Duplicate

(13, 13, '2023-01-20', 'Average', 'Needs improvement.'), -- Duplicate

(14, 14, '2023-02-10', 'Good', 'Consistent performance.'), -- Duplicate

(15, 15, '2023-02-15', 'Average', 'Needs improvement.'); -- Duplicate

CREATE TABLE Payroll (

PayrollID INT PRIMARY KEY,

EmployeeID INT,

PaymentDate DATE ,

Amount DECIMAL(10, 2),

PaymentMethod VARCHAR(20) ,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

INSERT INTO Payroll (PayrollID, EmployeeID, PaymentDate, Amount, PaymentMethod)

VALUES

(1, 1, '2023-03-01', 60000, 'Bank Transfer'),

(2, 2, '2023-03-01', 55000, 'Check'),

(3, 3, '2023-03-01', 50000, 'Bank Transfer'),

(4, 4, '2023-03-01', 70000, 'Bank Transfer'),

(5, 5, '2023-03-01', 45000, 'Check'),

(6, 6, '2023-03-01', 65000, 'Bank Transfer'), -- Duplicate PaymentMethod

(7, 7, '2023-03-01', 48000, 'Bank Transfer'), -- Duplicate PaymentMethod

(8, 8, '2023-03-01', 52000, 'Check'), -- Duplicate PaymentMethod

(9, 9, '2023-03-01', 50000, 'Bank Transfer'), -- Duplicate PaymentMethod

(10, 10, '2023-03-01', 60000, 'Check'), -- Duplicate PaymentMethod

(11, 11, '2023-03-01', 60000, 'Bank Transfer'), -- Duplicate

(12, 12, '2023-03-01', 55000, 'Check'), -- Duplicate

(13, 13, '2023-03-01', 70000, 'Bank Transfer'), -- Duplicate

(14, 14, '2023-03-01', 48000, 'Bank Transfer'), -- Duplicate

(15, 15, '2023-03-01', 52000, 'Check'); -- Duplicate

**#Assignment Queries#**

**1. Retrieve the names and contact details of employees hired after January 1, 2023.**

Ans=

SELECT EmployeeID,FirstName,LastName, phone FROM employees where HireDate>'2020-12-01'

**2. Find the total payroll amount paid to each department.**

Ans=

SELECT d.DepartmentName, sum(p.Amount) as payrollAmount from departments as d left join employees as e on d.DepartmentID=e.DepartmentID left join payroll as p on e.EmployeeID=p.EmployeeID group by d.DepartmentName

**3. List all employees who have not been assigned a manager.**

Ans=

SELECT e.EmployeeID,e.FirstName,e.LastName FROM employees as e inner join employees as m on e.EmployeeID=m.EmployeeID WHERE e.ManagerID is null GROUP by e.EmployeeID

**4. Retrieve the highest salary in each department along with the employee’s name.**

Ans=

SELECT d.DepartmentName, e.FirstName, e.LastName, max(e.salary)as highest\_salary from departments as d LEFT join employees as e on d.DepartmentID=e.DepartmentID GROUP by d.DepartmentName

**5. Find the most recent performance review for each employee.**

Ans=

select e.EmployeeID, e.FirstName, e.LastName, max(p.ReviewDate)as reviewDate, p.comments as review from employees as e left join performancereviews as p on e.EmployeeID=p.EmployeeID GROUP by e.EmployeeID

**6. Count the number of employees in each department.**

Ans=

SELECT d.DepartmentName, COUNT(e.EmployeeID) as noOfEmployee from departments as d left join employees as e on d.DepartmentID=e.DepartmentID GROUP by d.DepartmentName

**7. List all employees who have received a performance score of "Excellent." Identify the most frequently used payment method in payroll.**

Ans=

SELECT e.EmployeeID, e.FirstName, e.LastName,

(SELECT PaymentMethod

FROM Payroll

GROUP BY PaymentMethod

ORDER BY COUNT(PaymentMethod) DESC

LIMIT 1) AS MostFrequentPaymentMethod

FROM Employees e

JOIN PerformanceReviews p ON e.EmployeeID = p.EmployeeID

WHERE p.PerformanceScore = 'Excellent';

**8. Retrieve the top 5 highest-paid employees along with their departments.**

Ans=

SELECT e.EmployeeID, e.FirstName, e.LastName, d.DepartmentName

FROM employees as e left join departments as d on e.DepartmentID=d.DepartmentID GROUP by e.EmployeeID

ORDER by MAX(e.Salary) desc limit 5

**9. Show details of all employees who report directly to a specific manager (e.g.,ManagerID = 101)**

Ans=

SELECT EmployeeID, FirstName, LastName, Email, Phone, HireDate, ManagerID, Salary, DepartmentID FROM employees where ManagerID = 6 GROUP by EmployeeID