Task-4

create database task-4

1.Department-table

CREATE TABLE Departments (

DepartmentID int AUTO\_INCREMENT NOT NULL,

DepartmentName varchar(100) NOT NULL,

PRIMARY KEY (DepartmentID)

);

INSERT INTO Departments (DepartmentID, DepartmentName, ManagerID) VALUES

(1, 'Human Resources', 1),

(2, 'Information Technology', 2),

(3, 'Finance', 3),

(4, 'Marketing', 4),

(5, 'Sales', 5),

(6, 'Operations', 6),

(7, 'Customer Support', 7),

(8, 'Legal', 8),

(9, 'Research and Development', 9),

(10, 'Public Relations', 10);

ALTER table departments

add COLUMN ManagerID INT

ADD CONSTRAINT fk\_manage

FOREIGN KEY (ManagerID) REFERENCES employees(EmployeeID)

UPDATE departments set ManagerID=1 WHERE DepartmentID=1;

=================================================================================================================

2.Employee-table

CREATE TABLE Employees (

EmployeeID int AUTO\_INCREMENT NOT NULL,

FirstName varchar (50) NOT NULL,

LastName varchar (50) NOT NULL,

Email varchar (255) NOT NULL UNIQUE,

Phone varchar (15) NOT NULL,

HireDate date NOT NULL,

DepartmentID int NOT NULL,

ManagerID int,

Salary decimal (10,2),

PRIMARY KEY (EmployeeID),

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ManagerID) REFERENCES Employees(EmployeeID) ON DELETE SET NULL ON UPDATE CASCADE

);

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

VALUES

(1,'Manisha', 'Rathore', 'manisha@gmail.com', '9876543210', '2022-06-15', 1, NULL, 60000.00),

(2,'Priya', 'Verma', 'priya@gmail.com', '9876543211', '2021-03-20', 2, 1, 75000.00),

(3,'Rahul', 'Kumar', 'rahul@gmail.com', '9876543212', '2020-09-10', 3, 2, 65000.00),

(4,'Anjali', 'Patel', 'anjali@gmail.com', '9876543213', '2019-01-22', 4, 1, 70000.00),

(5,'Ravi', 'Mehta', 'ravi@gmail.com', '9876543214', '2022-07-11', 5, 3, 80000.00),

(6,'Neha', 'Gupta', 'neha@gmail.com', '9876543215', '2020-05-30', 6, 2, 55000.00),

(7,'Suresh', 'Reddy', 'suresh@gmail.com', '9876543216', '2021-11-01', 7, 1, 72000.00),

(8,'Geeta', 'Bhat', 'geeta@gmail.com', '9876543217', '2018-10-03', 8, 4, 85000.00),

(9,'Vikram', 'Yadav', 'vikram@gmail.com', '9876543218', '2017-04-15', 9, 5, 95000.00),

(10,'Meera', 'Singh', 'meera@gmail.com', '9876543219', '2023-02-14', 10, 6, 55000.00);

UPDATE employees set ManagerID=4 WHERE EmployeeID=4;

==================================================================================================================

3.PerformanceReviews-table

CREATE TABLE PerformanceReviews(

ReviewID int AUTO\_INCREMENT NOT NULL,

EmployeeID int NOT NULL,

ReviewDate date NOT NULL,

PerformanceScore varchar (50) NOT NULL,

Comments varchar (255) NOT NULL,

PRIMARY KEY (ReviewID),

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

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

VALUES

(1,'1', '2023-12-15', 'Excellent', 'Manisha shows outstanding leadership and consistently exceeds targets.'),

(2,'2', '2023-12-15', 'Good', 'Priya meets her goals but could work on time management skills.'),

(3, '3','2023-12-15', 'Average', 'Rahul performs well, but needs to take more initiative in projects.'),

(4,'4', '2023-12-15', 'Excellent', 'Anjali consistently exceeds expectations and is a key team player.'),

(5,'5', '2023-12-15', 'Good', 'Ravi meets expectations but should improve communication with the team.'),

(6,'6', '2023-12-15', 'Excellent', 'Neha is proactive and always contributes to team discussions.'),

(7, '7','2023-12-15', 'Above Average', 'Suresh performs solidly but needs to take on more challenging tasks.'),

(8,'8', '2023-12-15', 'Good', 'Geeta meets deadlines but needs to work better in collaborative settings.'),

(9, '9','2023-12-15', 'Excellent', 'Vikram is an exceptional performer with strong problem-solving skills.'),

(10, '10','2023-12-15', 'Average', 'Meera has room for improvement in prioritizing tasks and meeting deadlines.');

=================================================================================================================

4.Payroll-table

CREATE TABLE Payroll(

PayrollID int AUTO\_INCREMENT NOT NULL,

EmployeeID int NOT NULL,

PaymentDate date NOT NULL,

Amount decimal (10,2),

PaymentMethod varchar (50) NOT NULL,

PRIMARY KEY (PayrollID),

FOREIGN KEY (EmployeeID) REFERENCES employees (EmployeeID)

)

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

VALUES

(1,'1', '2023-01-31', 60000.00, 'Bank Transfer'),

(2,'2', '2023-01-31', 75000.00, 'Cheque'),

(3,'3', '2023-01-31', 65000.00, 'Bank Transfer'),

(4,'4','2023-01-31', 70000.00, 'Cash'),

(5, '5','2023-01-31', 80000.00, 'Bank Transfer'),

(6,'6', '2023-01-31', 55000.00, 'Cheque'),

(7, '7','2023-01-31', 72000.00, 'Bank Transfer'),

(8,'8', '2023-01-31', 85000.00, 'Cash'),

(9,'9', '2023-01-31', 95000.00, 'Bank Transfer'),

(10,'10', '2023-01-31', 55000.00, 'Cheque');

==============================================================================================================

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

SELECT FirstName,LastName,Email,Phone,HireDate

FROM employees

where HireDate >'2023-01-01'

==============================================================================================================

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

SELECT d.DepartmentName, SUM(p.Amount) AS TotalPayrollAmount

FROM Departments d

JOIN Employees e ON d.DepartmentID = e.DepartmentID

JOIN Payroll p ON e.EmployeeID = p.EmployeeID

GROUP BY d.DepartmentName;

=============================================================================================================

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

SELECT EmployeeID, FirstName, LastName, DepartmentID, HireDate, Salary

FROM Employees

WHERE ManagerID IS NULL;

========================================================================================================

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

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

FROM Employees e

JOIN Departments d ON e.DepartmentID = d.DepartmentID

WHERE e.Salary = (

SELECT MAX(Salary)

FROM Employees

WHERE DepartmentID = e.DepartmentID

);

===============================================================================================================

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

SELECT e.EmployeeID, e.FirstName, e.LastName, pr.ReviewDate, pr.PerformanceScore, pr.Comments

FROM Employees e

JOIN PerformanceReviews pr ON e.EmployeeID = pr.EmployeeID

WHERE pr.ReviewDate = (

SELECT MAX(ReviewDate)

FROM PerformanceReviews

WHERE EmployeeID = e.EmployeeID

);

================================================================================================================

6. Count the number of employees in each department.

SELECT d.DepartmentName, COUNT(e.EmployeeID) AS NumberOfEmployees

FROM Employees e

JOIN Departments d ON e.DepartmentID = d.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.

SELECT e.EmployeeID, e.FirstName, e.LastName, pr.PerformanceScore, pr.Comments

FROM Employees e

JOIN PerformanceReviews pr ON e.EmployeeID = pr.EmployeeID

WHERE pr.PerformanceScore = 'Excellent';

================================================================================================

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

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

FROM Employees e

JOIN Departments d ON e.DepartmentID = d.DepartmentID

ORDER BY e.Salary DESC

LIMIT 5;

=================================================================================================

9. Showdetails of all employees who report directly to a specific manager (e.g.,

ManagerID = 101)

SELECT EmployeeID, FirstName, LastName, Email, Phone, HireDate, DepartmentID, ManagerID, Salary

FROM Employees

WHERE ManagerID = 101;