**HR & Payroll Management System**

**--CREATE DATABASE**

CREATE TABLE **Employees** (

Emp\_ID INT PRIMARY KEY IDENTITY(1,1),

FirstName NVARCHAR(50) NOT NULL,

LastName NVARCHAR(50) NOT NULL,

Email NVARCHAR(100) UNIQUE NOT NULL,

PhoneNumber NVARCHAR(20) NULL,

DepartmentID INT NOT NULL,

HireDate DATE DEFAULT GETDATE(),

Salary DECIMAL(10,2) DEFAULT 50000,

Status NVARCHAR(20) DEFAULT 'Active',

CONSTRAINT FK\_Department FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

CREATE TABLE **Departments** (

DepartmentID INT PRIMARY KEY IDENTITY(1,1),

DepartmentName NVARCHAR(50) UNIQUE NOT NULL

);

CREATE TABLE **Payroll** (

PayrollID INT PRIMARY KEY IDENTITY(1,1),

Emp\_ID INT NOT NULL,

PayDate DATE DEFAULT GETDATE(),

BaseSalary DECIMAL(10,2) NOT NULL,

Bonus DECIMAL(10,2) DEFAULT 0,

TotalSalary AS (BaseSalary + Bonus), -- Computed column

CONSTRAINT FK\_Payroll\_Emp FOREIGN KEY (Emp\_ID) REFERENCES Employees(Emp\_ID)

);

SHOW DATABASES

**--Stored Procedure**

CREATE PROCEDURE ProcessPayroll

AS

BEGIN

DECLARE @EmpID INT, @BaseSalary DECIMAL(10,2), @Bonus DECIMAL(10,2);

DECLARE payroll\_cursor CURSOR FOR

SELECT Emp\_ID, Salary FROM Employees WHERE Status = 'Active';

BEGIN TRANSACTION;

BEGIN TRY

OPEN payroll\_cursor;

FETCH NEXT FROM payroll\_cursor INTO @EmpID, @BaseSalary;

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @Bonus = @BaseSalary \* 0.05;

INSERT INTO Payroll (Emp\_ID, BaseSalary, Bonus)

VALUES (@EmpID, @BaseSalary, @Bonus);

FETCH NEXT FROM payroll\_cursor INTO @EmpID, @BaseSalary;

END;

COMMIT TRANSACTION;

CLOSE payroll\_cursor;

DEALLOCATE payroll\_cursor;

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION;

PRINT 'Error processing payroll!';

END CATCH;

END;

**--Triggers**

CREATE TRIGGER trg\_InsertPayroll

ON Employees

AFTER INSERT

AS

BEGIN

INSERT INTO Payroll (Emp\_ID, BaseSalary)

SELECT Emp\_ID, Salary FROM inserted;

END;

CREATE TRIGGER trg\_FireEmployee

ON Employees

AFTER UPDATE

AS

BEGIN

UPDATE Employees

SET Status = 'Inactive'

WHERE Emp\_ID IN (SELECT Emp\_ID FROM inserted WHERE Salary = 0);

END;

**--Querying and Data Analysis**

SELECT e.Emp\_ID, e.FirstName, e.LastName, e.Email, d.DepartmentName, e.Status,

p.PayDate, p.BaseSalary, p.Bonus, p.TotalSalary

FROM Employees e

JOIN Departments d ON e.DepartmentID = d.DepartmentID

LEFT JOIN Payroll p ON e.Emp\_ID = p.Emp\_ID

ORDER BY p.PayDate DESC;

SELECT d.DepartmentName, SUM(p.TotalSalary) AS TotalExpense

FROM Payroll p

JOIN Employees e ON p.Emp\_ID = e.Emp\_ID

JOIN Departments d ON e.DepartmentID = d.DepartmentID

GROUP BY d.DepartmentName;

SELECT TOP 5 e.FirstName, e.LastName, p.Bonus, p.TotalSalary

FROM Payroll p

JOIN Employees e ON p.Emp\_ID = e.Emp\_ID

ORDER BY p.Bonus DESC;

**--Optimization Techniques**

CREATE INDEX idx\_Employees\_LastName ON Employees (LastName);

CREATE INDEX idx\_Payroll\_PayDate ON Payroll (PayDate);

CREATE PARTITION FUNCTION pf\_PayDateRange (DATE)

AS RANGE RIGHT FOR VALUES ('2024-01-01', '2025-01-01');

CREATE PARTITION SCHEME ps\_PayDateScheme

AS PARTITION pf\_PayDateRange ALL TO ([PRIMARY]);