DATABASE MANAGEMENT SYSTEM - CSA0593

ASSIGNMENT 2

N.MOKSHA SAI

192372374

QUESTION:

Employee Leave Management Model tables for employees, leave applications, and leave balances. Write stored procedures to apply for leave and approve or reject leave requests. Implement triggers to automatically update leave balances when leave is taken. Write SQL queries to analyze leave trends by department and employee.

ANSWER:

CONCEPTUAL MODEL[E.R DIAGRAM]:



LOGICAL MODEL[ E.R.DIAGRAM]:



PHYSICAL MODEL[E.R.DIAGRAM]:



SQL STATEMENTS :

Here are the SQL statements and conclusion for the topic:

SQL Statements:

Database Schema:

mysql

CREATE DATABASE EmployeeLeaveManagement;

USE EmployeeLeaveManagement;

CREATE TABLE Employees (

EmployeeID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Department VARCHAR(100),

LeaveBalance INT

);

CREATE TABLE LeaveApplications (

ApplicationID INT AUTO\_INCREMENT PRIMARY KEY,

EmployeeID INT,

LeaveType VARCHAR(50),

StartDate DATE,

EndDate DATE,

Status VARCHAR(50),

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

CREATE TABLE LeaveBalances (

BalanceID INT AUTO\_INCREMENT PRIMARY KEY,

EmployeeID INT,

LeaveType VARCHAR(50),

Balance INT,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

Stored Procedures:

mysql

DELIMITER //

CREATE PROCEDURE sp\_ApplyForLeave(

IN employeeID INT,

IN leaveType VARCHAR(50),

IN startDate DATE,

IN endDate DATE

)

BEGIN

INSERT INTO LeaveApplications (EmployeeID, LeaveType, StartDate, EndDate, Status)

VALUES (employeeID, leaveType, startDate, endDate, 'Pending');

END //

CREATE PROCEDURE sp\_ApproveLeave(

IN applicationID INT

)

BEGIN

UPDATE LeaveApplications

SET Status = 'Approved'

WHERE ApplicationID = applicationID;

DECLARE employeeID INT;

SELECT EmployeeID INTO employeeID FROM LeaveApplications WHERE ApplicationID = applicationID;

DECLARE leaveBalance INT;

SELECT Balance INTO leaveBalance FROM LeaveBalances WHERE EmployeeID = employeeID AND LeaveType = (SELECT LeaveType FROM LeaveApplications WHERE ApplicationID = applicationID);

UPDATE LeaveBalances

SET Balance = Balance - 1

WHERE EmployeeID = employeeID AND LeaveType = (SELECT LeaveType FROM LeaveApplications WHERE ApplicationID = applicationID);

END //

CREATE PROCEDURE sp\_RejectLeave(

IN applicationID INT

)

BEGIN

UPDATE LeaveApplications

SET Status = 'Rejected'

WHERE ApplicationID = applicationID;

END //

DELIMITER;

Triggers:

mysql

DELIMITER //

CREATE TRIGGER tr\_UpdateLeaveBalance

AFTER UPDATE ON LeaveApplications

FOR EACH ROW

BEGIN

IF NEW.Status = 'Approved' THEN

UPDATE LeaveBalances

SET Balance = Balance - 1

WHERE EmployeeID = NEW.EmployeeID AND LeaveType = NEW.LeaveType;

END IF;

END //

DELIMITER;

SQL Queries:

mysql

-- Leave Trends by Department

SELECT

Department,

COUNT(\*) AS TotalLeaves

FROM

Employees

JOIN LeaveApplications ON Employees.EmployeeID = LeaveApplications.EmployeeID

GROUP BY

Department;

-- Leave Trends by Employee

SELECT

EmployeeID,

FirstName,

LastName,

COUNT(\*) AS TotalLeaves

FROM

Employees

JOIN LeaveApplications ON Employees.EmployeeID = LeaveApplications.EmployeeID

GROUP BY

EmployeeID, FirstName, LastName;

-- Leave Balance Report

SELECT

EmployeeID,

FirstName,

LastName,

LeaveType,

Balance

FROM

Employees

JOIN LeaveBalances ON Employees.EmployeeID = LeaveBalances.EmployeeID;

Conclusion:

This database design provides a comprehensive foundation for managing employee leaves, leave applications, and leave balances. The stored procedures simplify leave application, approval, and rejection processes, while the triggers ensure data consistency and accuracy. The SQL queries enable analysis of leave trends by department and employee.

Best Practices:

1. Regularly backup the database.

2. Use secure passwords and authentication.

3. Implement data validation and error handling.

4. Optimize queries for performance.

5. Use indexing for efficient data retrieval.

Future Enhancements:

1. Integrate with HR management software.

2. Implement automated email notifications.

3. Develop a web-based interface for leave management.

4. Add support for multiple leave types.

5. Integrate with payroll software.