

# ***DBS PROJECT REPORT***

## **PROJECT TITLE**

**Employee Management System**



## **Submitted to:**

SIR UMAR QASIM

## **Submitted by:**

Sadia Sher (2022-SE-07)

Rimsha Aamir(2022-SE-13)

**University of Engineering and Technology,  
New Campus, Lahore.**

# Employee MANAGEMENT SYSTEM

---

## Introduction

Database Table for Employee Management System-Employee management system is ever green project for real life.

This database schema is designed to manage employee information within an organization. It consists of several interconnected tables that store essential details about employees, their roles, salaries, departments, and addresses.

## Functionality

### 1. Data Manipulation

2. **Remove all data from the employee table:** Clears all records from the employee table, effectively resetting it.
3. **Delete specific employee:** Deletes a particular employee from the table based on their EmployeeID.
4. **Update employee:** Modifies the information of an employee, such as phone number, hire date, and department, based on their EmployeeID.
5. **Rename column name:** Renames a specific column in a table.
6. **Delete column:** Removes a specific column from a table.

### 2. Data Retrieval

1. **Retrieve employee with department and salary:** Fetches employee details along with their department name and salary.
2. **Total number of employees in each department:** Counts the total number of employees in each department.

3. **Average salary of employees:** Calculates the average salary of all employees.
4. **Maximum salary among employees:** Finds the highest salary among all employees.
5. **Maximum salary among employees with employee name:** Retrieves the employee with the highest salary along with their name.
6. **Minimum salary among employees:** Finds the lowest salary among all employees.
7. **Total salary expenditure by department:** Calculates the total salary expenditure for each department.
8. **Count of employees holding each role:** Counts the number of employees assigned to each role.
9. **Retrieve employee names grouped by roles:** Lists employee names along with their respective roles.
10. **Retrieve employee details along with their department names:** Displays employee details with their associated department names.
11. **Retrieve employee details along with their salary information:** Shows employee details along with their salary and effective date.
12. **Retrieve employee details along with their role names:** Displays employee details with their assigned role names.
13. **Retrieve employees with salaries greater than the average salary:** Lists employees with salaries higher than the average salary.
14. **Retrieve employees who are assigned a specific role:** Fetches employees assigned to a specific role.
15. **Retrieve departments with more than 5 employees:** Lists departments with more than five employees.
16. **Retrieve employees with the highest salary:** Displays employees with the highest salary.
17. **Retrieve employees in a specific department:** Lists employees belonging to a particular department.
18. **Retrieve employees without addresses:** Shows employees who do not have addresses listed.
19. **Top 2 employees with the highest salary:** Lists the top two employees with the highest salary.

### 3. Structural Modification

1. **Delete table from the database:** Removes a specific table and its associated data from the database.

## End Users

The end user of our system are the administration of the hotel that maintain records of employees.

# Tables Description

## Employees Table

This table serves as the central repository for employee data. It includes fields such as EmployeeID, FirstName, LastName, Email, PhoneNumber, HireDate, and DepartmentID. The EmployeeID is the primary key, uniquely identifying each employee. The DepartmentID field establishes a relationship with the Departments table through a foreign key constraint.

## Departments Table

Here, department-specific information is stored. Each department is identified by a unique DepartmentID and has a corresponding DepartmentName.

## Salaries Table

This table tracks the salary history of employees. It records SalaryID, EmployeeID, Salary, and EffectiveDate. The SalaryID is a primary key, while the EmployeeID establishes a relationship with the Employees table.

## Addresses Table

Employee addresses are stored in this table. It contains AddressID, EmployeeID, Address, City, State, and ZipCode fields. The AddressID is the primary key, and the EmployeeID establishes a relationship with the Employees table..

## Role Table

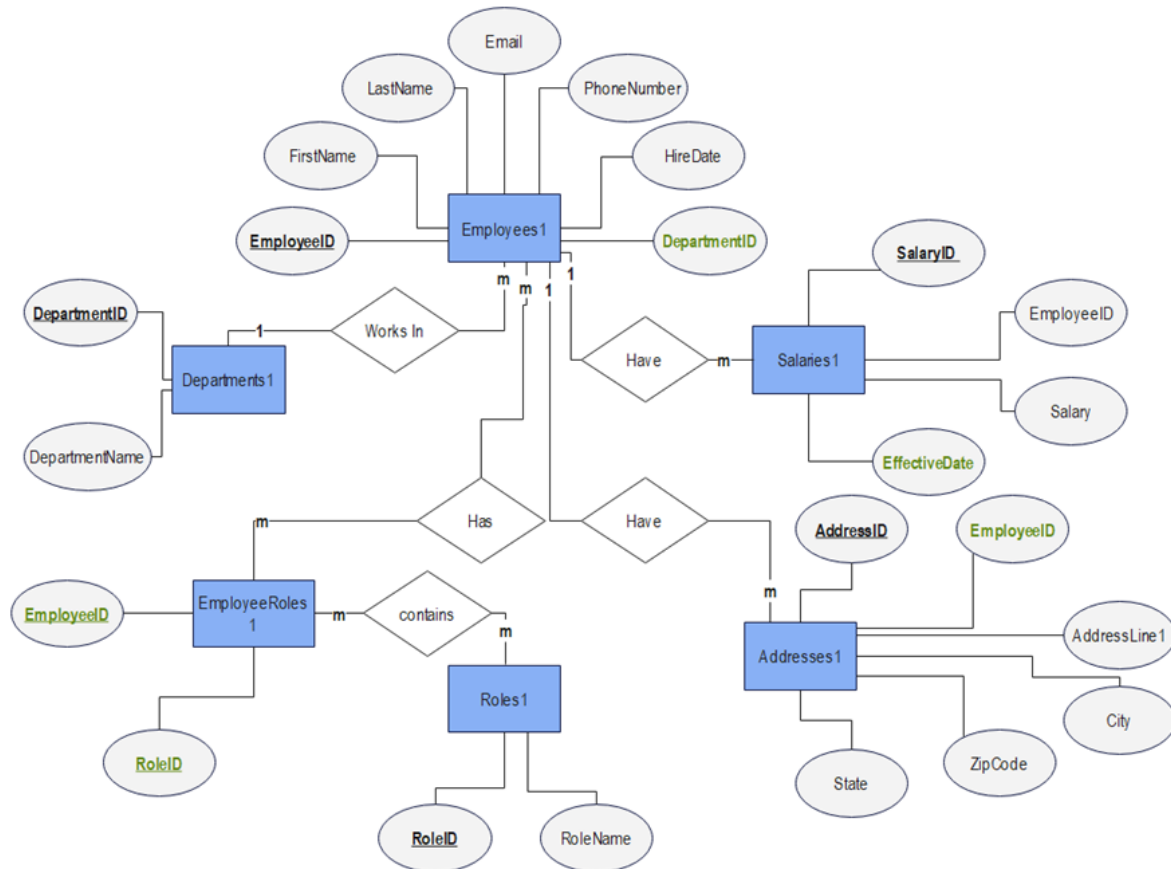
This table defines various roles within the organization. Each role is identified by a unique RoleID and has a corresponding RoleName..

## EmployeeRoles Table

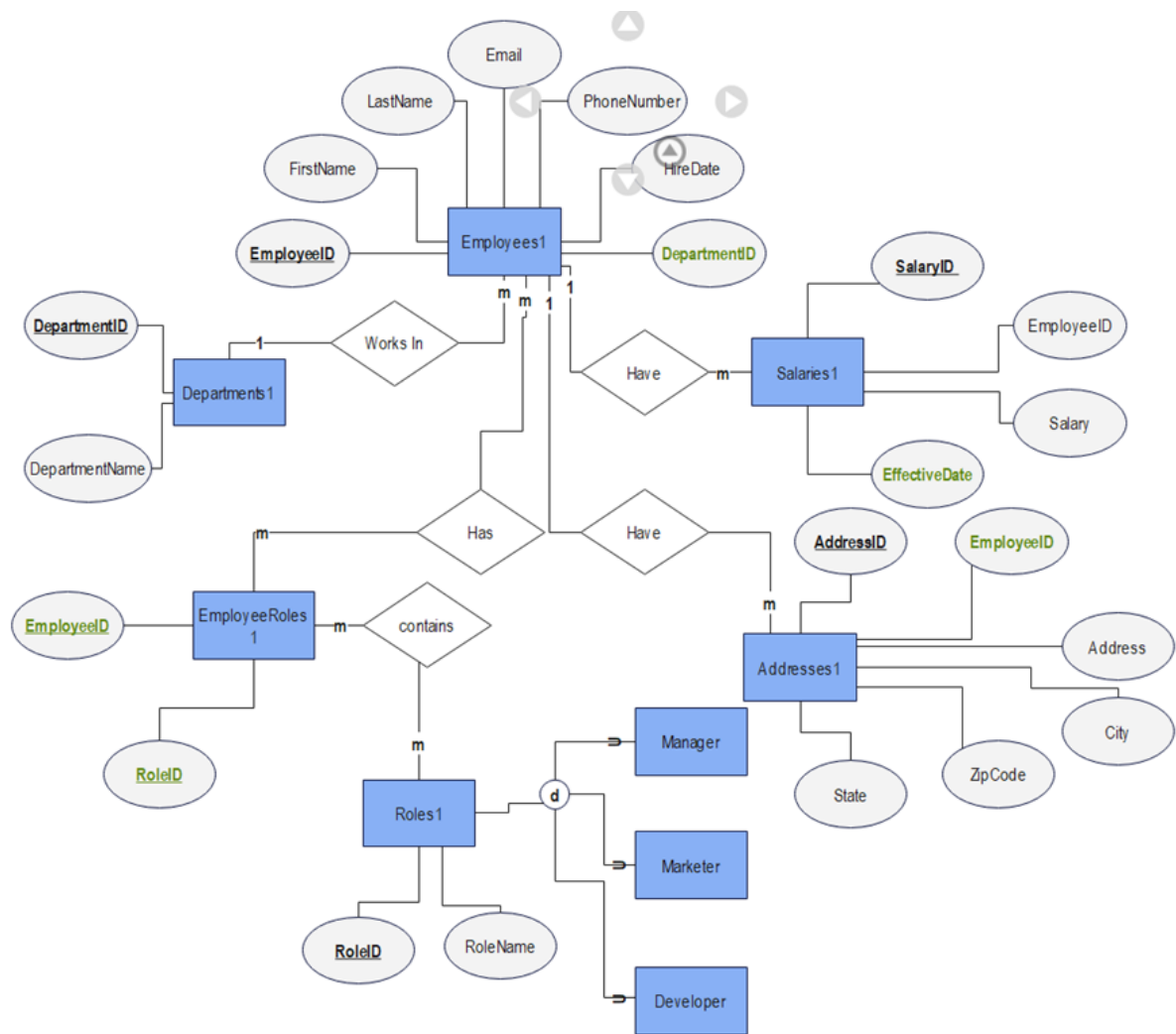
This table establishes the relationship between employees and their roles. It contains EmployeeID and RoleID fields, forming a composite primary key. Both fields are foreign keys referencing the Employees and Roles tables, respectively.

# Relational Model

## ER Diagram



# EER Diagram



# Conceptual Level

Database normalization is the process of restructuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. Generally, if a database is normalized until third normal form, then it is considered to be normalized. We tried to normalize the database until third normal form.

## EMPLOYEE1

1. **Primary Key:** EmployeeID
2. **Foreign Key:** DepartmentID
3. **Domains:** Unique Primary Key
4. **Integrity Checks:** not null
5. **Functional Dependences:** EmployeeID  $\twoheadrightarrow$  FirstName, LastName, Email, PhoneNumber, HireDate

## DEPARTMENT1

1. **Primary Key:** DepartmentID
2. **Foreign Key:** No
3. **Domains:** Unique Primary Key
4. **Integrity Checks:** not null
5. **Functional Dependences:** DepartmentID  $\twoheadrightarrow$  DepartmentName

## SALARIES1

1. **Primary Key:** SalaryID
2. **Foreign Key:** EmployeeID
3. **Domains:** Unique Primary Key
4. **Integrity Checks:** not null
5. **Functional Dependences:** SalaryID  $\twoheadrightarrow$  Salary, EffectiveDate

## ADDRESSES1

1. **Primary Key:** AddressID
2. **Foreign Key:** EmployeeID
3. **Domains:** Unique Primary Key
4. **Integrity Checks:** not null
5. **Functional Dependences:** AddressID  $\twoheadrightarrow$  Address, City, State, ZipCode

## ROLES

1. **Primary Key:** RoleID
2. **Foreign Key:** No
3. **Domains:** Unique Primary Key
4. **Integrity Checks:** not null
5. **Functional Dependences:** RoleID  $\twoheadrightarrow$  RoleName

## EMPLOYEEROLES1

1. **Primary Key:** (EmployeeID, RoleID)

2. **Foreign Key:** EmployeeID, RoleID
3. **Domains:** Unique Primary Key
4. **Integrity Checks:** not null
5. **Functional Dependences:** Null

## Internal View

- View to Display Employee Details with Department Names
- View to Display Employee Details with Role Names
- View to Display Employees with Salaries
- View to Display Employees with Department and Role Names
- View to Display Employees with Salaries and Roles
- View to Display Employees with Department and Address Information

## Triggers

- Insert a default role for each newly inserted employee
- Delete employee info
- Trigger to log salary changes

**THE END** 😊