EMPLOYEES MANAGEMENT SYSTEM

**Overview:**

**Project Title**

Employees Management System

**Description**

This system enables seamless handling of employee records, salary details, and departmental data through well-structured SQL queries.

**Scope**

The system includes functionalities for:

* Creating and managing employee records
* Handling salary details
* Managing departmental data
* Performing various data retrieval operations

**Audience**

This documentation is intended for developers, database administrators, and users who will interact with the system.

**System Architecture:**

**High-Level Architecture**

The Employee Management System is a database-centric application that utilizes SQL for managing and retrieving employee data.

**Components**

* **Database**: Manages employee data.
* **SQL Queries**: Perform operations on the database.

**Technology Stack**

* SQL for database management and queries

**Table Descriptions:**

**EmployeeDetails**

* **EmployeeID**: INT, Primary Key
* **FirstName**: VARCHAR(50)
* **LastName**: VARCHAR(50)
* **Department**: VARCHAR(50)
* **Salary**: DECIMAL(10,2)
* **HireDate**: DATE

**SQL Queries:**

**CRUD Operations:**

**-- Create a Database**

CREATE DATABASE Employee;

**-- Use the Database**

USE Employee;

**-- Create the EmployeeDetails table**

CREATE TABLE EmployeeDetails (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR (50),

LastName VARCHAR (50),

Department VARCHAR (50),

Salary DECIMAL (10,2),

HireDate DATE

);

**-- Insert values into EmployeeDetails table**

INSERT INTO EmployeeDetails (EmployeeID, FirstName, LastName, Department, Salary, HireDate)

VALUES

(1, 'Kunal', 'Thakur', 'HR', 50000.00, '2015-05-20'),

(2, 'Sumit', 'Puri', 'IT', 60000.00, '2017-08-15'),

(3, 'Sarthak', 'Goel', 'Marketing', 55000.00, '2020-01-10'),

(4, 'Baldeesh', 'Kaur', 'Finance', 65000.00, '2017-04-25'),

(5, 'Prachi', 'Sharma', 'HR', 52000.00, '2017-09-30'),

(6, 'Karunesh', 'Pathak', 'IT', 62000.00, '2021-11-18'),

(7, 'Ashish', 'Pratap', 'Marketing', 58000.00, '2024-02-26'),

(8, 'Darshil', 'Sidhu', 'Finance', 70000.00, '2022-07-12'),

(9, 'Majhil', 'Singh', 'HR', 53000.00, '2023-10-05'),

(10, 'Rohit', 'Yadav', 'IT', 64000.00, '2016-03-08');

**Data Retrieval Operations:**

* Retrieve all data from the table.

**SELECT \* FROM EmployeeDetails;**

**A screenshot of a computer

Description automatically generated**

* Retrieve only the FirstName and LastName of all employees.

**SELECT FirstName, LastName FROM EmployeeDetails;**

**A screenshot of a computer

Description automatically generated**

* Retrieve distinct departments from the employeeDetails table.

**SELECT DISTINCT Department FROM EmployeeDetails;**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees whose salary is greater than 55000.

**SELECT \* FROM EmployeeDetails WHERE Salary > 55000;**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees hired after 2019.

**SELECT \* FROM EmployeeDetails WHERE HireDate > '2019-12-31';**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees whose first name starts with ‘A’.

**SELECT FirstName FROM EmployeeDetails WHERE FirstName LIKE 'A%';**

**A close up of a sign

Description automatically generated**

* Retrieve employees whose last name ends with ‘aur’.

**SELECT LastName FROM EmployeeDetails WHERE LastName LIKE '%aur';**

**A close up of a name

Description automatically generated**

* Retrieve employees whose First name do not have ‘a’.

**SELECT FirstName FROM EmployeeDetails**

**WHERE FirstName NOT LIKE '%a%';**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees sorted by their salary in descending order.

**SELECT \* FROM EmployeeDetails ORDER BY Salary DESC;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the count of employees in each department.

**SELECT Department, COUNT(\*) FROM EmployeeDetails**

**GROUP BY Department;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the average salary of employees in the Finance department.

**SELECT AVG(Salary) AS Average\_Salary FROM EmployeeDetails**

**WHERE Department = 'Finance';**

**A close up of numbers

Description automatically generated**

* Retrieve the maximum salary among all employees.

**SELECT MAX(Salary) AS Maximum\_Salary FROM EmployeeDetails;**

**A number of numbers on a white background

Description automatically generated**

* Retrieve the total salary expense for the company.

**SELECT SUM(Salary) AS Total\_Salary\_Expense FROM EmployeeDetails;**

**A close up of a number

Description automatically generated**

* Retrieve the oldest and newest hire date among all employees.

**SELECT MIN(HireDate) AS Oldest\_Hire\_Date, MAX(HireDate) AS Newest\_Hire\_Date FROM EmployeeDetails;**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees with a salary between 50000 and 60000.

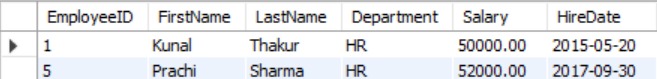
**SELECT \* FROM EmployeeDetails WHERE Salary BETWEEN 50000 AND 60000;**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees who are in the HR department and were hired before 2019.

**SELECT \* FROM EmployeeDetails WHERE Department = 'HR' AND HireDate < '2019-01-01';**

****

* Retrieve employees with a salary less than the average salary of all employees.

**SELECT \* FROM EmployeeDetails WHERE Salary < (SELECT AVG(Salary) FROM EmployeeDetails);**

**A screenshot of a computer

Description automatically generated**

* Retrieve the top 3 highest paid employees.

**SELECT \* FROM EmployeeDetails ORDER BY Salary DESC LIMIT 3;**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees whose hire date is not in 2017.

**SELECT \* FROM EmployeeDetails WHERE YEAR(HireDate) <> 2017;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the nth highest salary (you can choose the value of n).

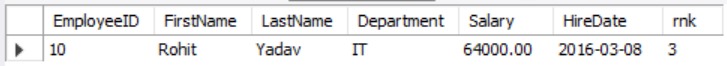
**SELECT \* FROM (**

**SELECT \*, DENSE\_RANK() OVER(ORDER BY Salary DESC) rnk**

**FROM EmployeeDetails**

**) AS SUB**

**WHERE rnk = 3; -- Change 3 to any desired rank**

****

* Retrieve employees who were hired in the same year as ‘Sumit Puri’.

**SELECT \* FROM EmployeeDetails WHERE YEAR(HireDate) = (**

**SELECT YEAR(HireDate) FROM EmployeeDetails WHERE FirstName = 'Sumit' AND LastName = 'Puri'**

**);**

**A screenshot of a computer

Description automatically generated**

* Retrieve employees who have been hired on weekends (Saturday or Sunday).

**SELECT \* FROM EmployeeDetails WHERE DAYOFWEEK(HireDate) IN (1, 7);**

****

* Retrieve employees who have been hired in the last 6 years.

**SELECT \* FROM EmployeeDetails**

**WHERE HireDate >= CURDATE() - INTERVAL 6 YEAR;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the department with the highest average salary.

**SELECT Department, AVG(Salary) AS Average\_Salary FROM EmployeeDetails**

**GROUP BY Department**

**ORDER BY Average\_Salary DESC**

**LIMIT 1;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the top 2 highest paid employees in each department.

**WITH HIGHEST\_SALARY AS (**

**SELECT \*, DENSE\_RANK() OVER(PARTITION BY Department ORDER BY Salary DESC) rnk**

**FROM EmployeeDetails**

**)**

**SELECT \* FROM HIGHEST\_SALARY WHERE rnk <= 2;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the cumulative salary expense for each department sorted by department and hire date.

**WITH CTE AS (**

**SELECT \*, SUM(Salary) OVER(PARTITION BY Department ORDER BY HireDate) AS Cumulative\_Salary**

**FROM EmployeeDetails**

**)**

**SELECT \* FROM CTE ORDER BY Department, HireDate;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the employee ID, salary, and the next highest salary for each employee.

**SELECT EmployeeID, Salary, LEAD(Salary) OVER(ORDER BY Salary DESC) AS NEXT\_HIGHEST\_SALARY**

**FROM EmployeeDetails**

**ORDER BY Salary DESC;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the employee ID, salary, and the difference between the current salary and the next highest salary for each employee.

**SELECT EmployeeID, Salary,**

**Salary - LEAD(Salary) OVER(ORDER BY Salary DESC) AS Salary\_Difference**

**FROM EmployeeDetails;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the employee(s) with the highest salary in each department.

**WITH CTE AS (**

**SELECT \*, DENSE\_RANK() OVER(PARTITION BY Department ORDER BY Salary DESC) rnk**

**FROM EmployeeDetails**

**)**

**SELECT \* FROM CTE WHERE rnk = 1;**

**A screenshot of a computer

Description automatically generated**

* Retrieve the department(s) where the total salary expense is greater than the average total salary expense across all departments.

**WITH DEP\_TOTAL\_SALARY AS (**

**SELECT Department, SUM(Salary) AS Total\_Salary**

**FROM EmployeeDetails**

**GROUP BY Department**

**),**

**AVERAGE\_TOTAL\_SALARY AS (**

**SELECT AVG(Total\_Salary) AS avg\_salary\_total**

**FROM DEP\_TOTAL\_SALARY**

**)**

**SELECT Department, Total\_Salary**

**FROM DEP\_TOTAL\_SALARY d, AVERAGE\_TOTAL\_SALARY a**

**WHERE d.Total\_Salary > a.avg\_salary\_total;**

* Retrieve the employee(s) who have the same first name and last name as any other employee.

**SELECT e1.FirstName, e1.LastName, e1.EmployeeID**

**FROM EmployeeDetails e1**

**JOIN EmployeeDetails e2**

**ON e1.FirstName = e2.FirstName**

**AND e1.LastName = e2.LastName**

**AND e1.EmployeeID != e2.EmployeeID;**

**A close up of a computer screen

Description automatically generated**

* Retrieve the employee(s) who have been with the company for more than 7 Years.

**SELECT \* FROM EmployeeDetails**

**WHERE HireDate <= DATE\_SUB(CURDATE(), INTERVAL 7 YEAR);**

**A screenshot of a computer

Description automatically generated**

* Retrieve the department with the highest salary range (Difference b/w highest and lowest).

**SELECT Department, MAX(Salary) - MIN(Salary) AS DIFFERENCE**

**FROM EmployeeDetails**

**GROUP BY Department**

**ORDER BY DIFFERENCE DESC;**

**A screenshot of a computer

Description automatically generated**

**Maintenance and Troubleshooting**

**Regular Maintenance**

* Regularly back up the database.
* Monitor database performance and optimize queries as needed.

**Common Issues**

* **Connection Issues**: Ensure the database server is running and accessible.
* **Query Errors**: Verify the syntax and logic of SQL queries.