



INNOMATICS  
RESEARCH LABS

## SQL Project

# Employee Management System

## Domain Knowledge

The **Employee Management System (EMS)** is designed to streamline the management of employee data, job roles, and departmental information within an organization. This system allows for efficient tracking of employee details, job assignments, qualifications, and performance metrics. Key domain knowledge elements for this system include:

1. **Employee Information Management:** The system stores personal details of employees such as name, contact information, gender, and unique login credentials. It is crucial for ensuring secure access and easy retrieval of employee records.
2. **Job Role Assignment:** Each employee is associated with a specific job role, which is linked to the department they work in. This connection ensures that employees are correctly aligned with their job functions and responsibilities within the organization.
3. **Departmental Structure:** The organization is divided into various departments (e.g., HR, Finance, IT), each with distinct job roles. The system should manage these departments and the employees assigned to each role efficiently.
4. **Payroll and Compensation:** Employee compensation details, including salary and bonuses, are stored in the system. Payroll processing and salary allocations are automatically calculated based on the job roles and associated salary ranges.
5. **Qualifications and Skills Tracking:** The system tracks employee qualifications, certifications, and skills to ensure that employees meet the requirements for their roles and identify opportunities for professional development.
6. **Leave and Absence Management:** The system manages employee leave records, including vacation days, sick leaves, and other types of absences, with appropriate deductions applied to payroll based on the employee's leave history.

This system ensures that all employee-related information is stored securely, easily accessible for reporting, and aligned with organizational goals for performance, compensation, and growth.

# Data Description

👉 Kindly click on the [Dataset](#) to check and download

This project includes 6 interrelated tables:

| Table name    | Description   |
|---------------|---|
| JobDepartment | Stores job roles, departments, and related salary ranges                  |
| SalaryBonus   | Contains salary, bonus, and annual pay linked to specific job roles.      |
| Employee      | Maintains personal, contact, and login details of all employees.          |
| Qualification | Records qualifications and required skills for employee job positions.    |
| Leaves        | Tracks employee leave records with reasons and dates.                     |
| Payroll       | Combines employee, job, salary, and leave data to calculate net payments. |

## Table Structures

```
-- Table 1: Job Department
CREATE TABLE JobDepartment (
  Job_ID INT PRIMARY KEY,
  jobdept VARCHAR(50),
  name VARCHAR(100),
  description TEXT,
  salaryrange VARCHAR(50)
);

-- Table 2: Salary/Bonus
CREATE TABLE SalaryBonus (
  salary_ID INT PRIMARY KEY,
  Job_ID INT,
  amount DECIMAL(10,2),
  annual DECIMAL(10,2),
  bonus DECIMAL(10,2),
  CONSTRAINT fk_salary_job FOREIGN KEY (job_ID) REFERENCES JobDepartment(Job_ID)
  ON DELETE CASCADE ON UPDATE CASCADE
);

-- Table 3: Employee
CREATE TABLE Employee (
  emp_ID INT PRIMARY KEY,
  firstname VARCHAR(50),
  lastname VARCHAR(50),
  gender VARCHAR(10),
  age INT,
  contact_add VARCHAR(100),
  emp_email VARCHAR(100) UNIQUE,
  emp_pass VARCHAR(50),
  Job_ID INT,
  CONSTRAINT fk_employee_job FOREIGN KEY (Job_ID)
```

```

        REFERENCES JobDepartment(Job_ID)
        ON DELETE SET NULL
        ON UPDATE CASCADE
    );

-- Table 4: Qualification
CREATE TABLE Qualification (
    QualID INT PRIMARY KEY,
    Emp_ID INT,
    Position VARCHAR(50),
    Requirements VARCHAR(255),
    Date_In DATE,
    CONSTRAINT fk_qualification_emp FOREIGN KEY (Emp_ID)
        REFERENCES Employee(emp_ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- Table 5: Leaves
CREATE TABLE Leaves (
    leave_ID INT PRIMARY KEY,
    emp_ID INT,
    date DATE,
    reason TEXT,
    CONSTRAINT fk_leave_emp FOREIGN KEY (emp_ID) REFERENCES Employee(emp_ID)
        ON DELETE CASCADE ON UPDATE CASCADE
);

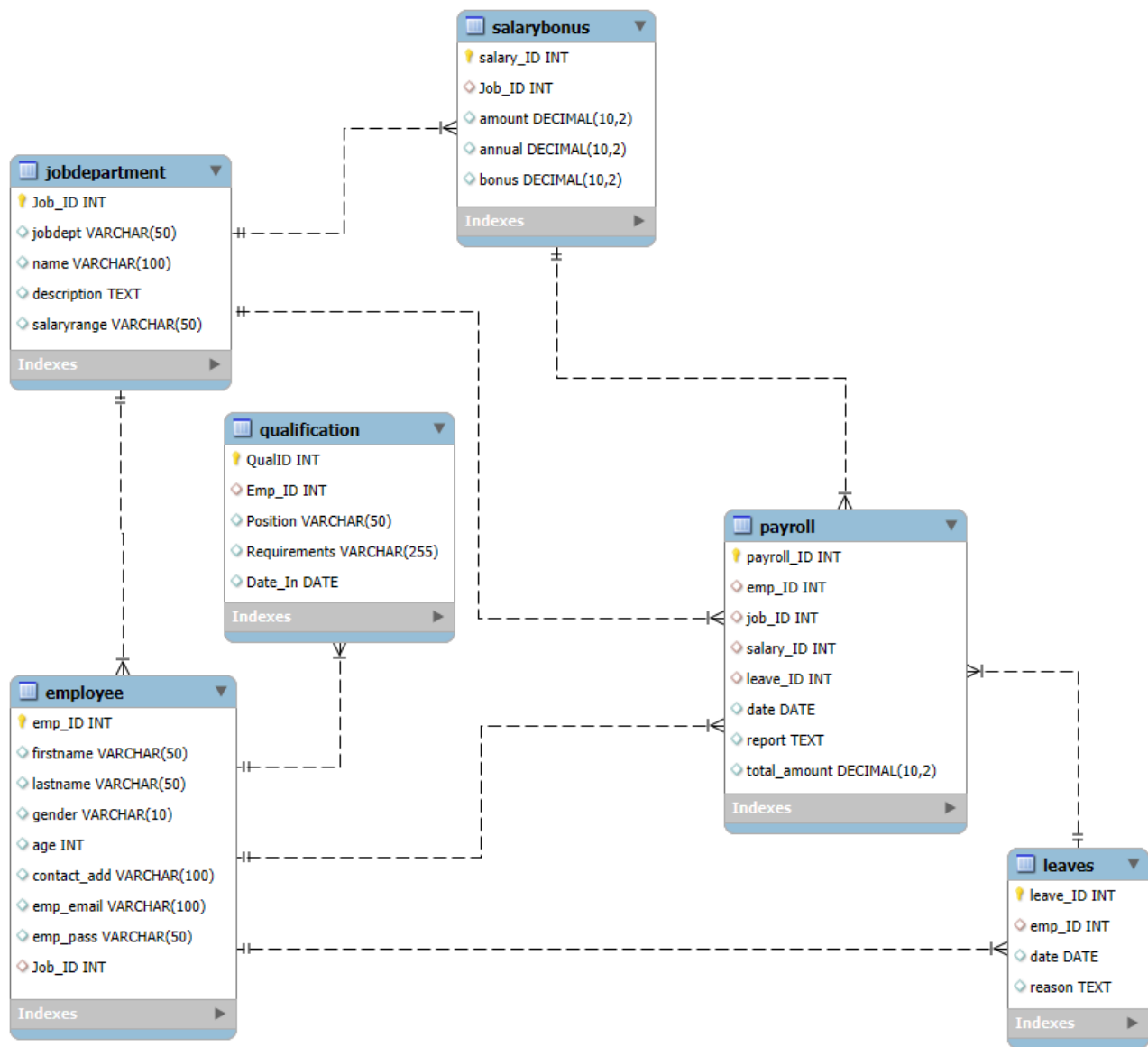
-- Table 6: Payroll
CREATE TABLE Payroll (
    payroll_ID INT PRIMARY KEY,
    emp_ID INT,
    job_ID INT,
    salary_ID INT,
    leave_ID INT,
    date DATE,
    report TEXT,
    total_amount DECIMAL(10,2),
    CONSTRAINT fk_payroll_emp FOREIGN KEY (emp_ID) REFERENCES Employee(emp_ID)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_payroll_job FOREIGN KEY (job_ID) REFERENCES JobDepartment(job_ID)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_payroll_salary FOREIGN KEY (salary_ID) REFERENCES
SalaryBonus(salary_ID)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_payroll_leave FOREIGN KEY (leave_ID) REFERENCES Leaves(leave_ID)
        ON DELETE SET NULL ON UPDATE CASCADE
);

```

## Foreign Key Constraints:

- **Cascading Delete/Update:** For most relationships, the ON DELETE CASCADE and ON UPDATE CASCADE actions are used, meaning that if a record in the parent table (e.g., Employee, JobDepartment) is deleted or updated, the corresponding records in the child tables (e.g., SalaryBonus, Payroll) are automatically deleted or updated as well.
- **Set Null on Delete:** For the relationship between the Payroll and Leaves tables, ON DELETE SET NULL is used. If a leave record is deleted, the related payroll record will not be deleted, but the reference to the leave record will be set to null.

## ER DIAGRAM



## **Problem Statement:**

The objective of this project is to design and implement an **Employee Management System** that efficiently stores and manages employee-related data within an organization. The system needs to track various aspects of employee information, including personal details, job roles, salary structures, qualifications, leave records, and payroll data. The system should ensure the integrity and consistency of data by using relational tables with appropriate foreign keys and cascading actions.

The system should allow for easy management and querying of employee data, providing insights such as payroll calculation, leave tracking, and department-specific job roles. The goal is to streamline HR operations, ensuring that all relevant employee data is accessible and accurately updated across different modules.

## **Analysis Questions**

### **1. EMPLOYEE INSIGHTS**

- How many unique employees are currently in the system?
- Which departments have the highest number of employees?
- What is the average salary per department?
- Who are the top 5 highest-paid employees?
- What is the total salary expenditure across the company?

### **2. JOB ROLE AND DEPARTMENT ANALYSIS**

- How many different job roles exist in each department?
- What is the average salary range per department?
- Which job roles offer the highest salary?
- Which departments have the highest total salary allocation?

### **3. QUALIFICATION AND SKILLS ANALYSIS**

- How many employees have at least one qualification listed?
- Which positions require the most qualifications?
- Which employees have the highest number of qualifications?

### **4. LEAVE AND ABSENCE PATTERNS**

- Which year had the most employees taking leaves?
- What is the average number of leave days taken by its employees per department?
- Which employees have taken the most leaves?
- What is the total number of leave days taken company-wide?
- How do leave days correlate with payroll amounts?

## 5. PAYROLL AND COMPENSATION ANALYSIS

- What is the total monthly payroll processed?
- What is the average bonus given per department?
- Which department receives the highest total bonuses?
- What is the average value of `total_amount` after considering leave deductions?

## Challenges

- Defining correct table relationships and ensuring accurate use of foreign keys.
- Maintaining data consistency with cascading updates and deletes.
- Writing complex joins for reports involving employee roles, leaves, and payroll.
- Ensuring all date fields follow the YYYY-MM-DD format for reliable time-based analysis.
- Preventing duplicate entries using unique constraints, especially on email fields.

## Project Presentation Template

As part of this project, you are required to create and present the analysis findings. Use the following PowerPoint template to structure your presentation:

👉 Click here to [find the PPT Template for the Project Presentation](#)

## Submission

After completion of the project, Zip the **.sql query file** and **PPT**, upload the zip file with your name and batch number In LMS.