



PAYROLL MANAGEMENT SYSTEM

BY-ADITYA SONAWANE



PAYROLL MANAGEMENT SYSTEM

PROJECT FOR SQL MODULE

1.DESCRIPTION:

A payroll management system in SQL is designed to handle the financial aspects of paying employees within an organization. It typically involves storing employee information, tracking working hours, calculating salaries, and managing payroll-related transactions. Below is a description of a basic payroll management system in SQL:

Database Schema:

The database schema for a payroll management system typically includes the following tables:

1. **Employees:** Contains information about employees such as employee ID, name, email, department ID, salary, and hiring date.
2. **Payroll:** Stores payroll-related information such as payroll ID, employee ID (foreign key referencing the Employees table), pay date, hours worked, overtime hours, deductions, and bonuses.
3. **Departments:** Holds department information including department ID and department name.

Functionality:

1. **Employee Management:**

- Store employee details including name, email, department, salary, and hiring date.
- Update employee information as needed.
- Remove employees who are no longer with the company.

2. **Payroll Processing:**

- Calculate employee salaries based on hours worked, overtime, and any bonuses or deductions.
- Generate payrolls for each pay period, including pay dates and related financial details.
- Store payroll information for historical reference.

3. **Department Management:**

- Add, update, or remove departments as the organizational structure changes.
- Assign employees to departments.

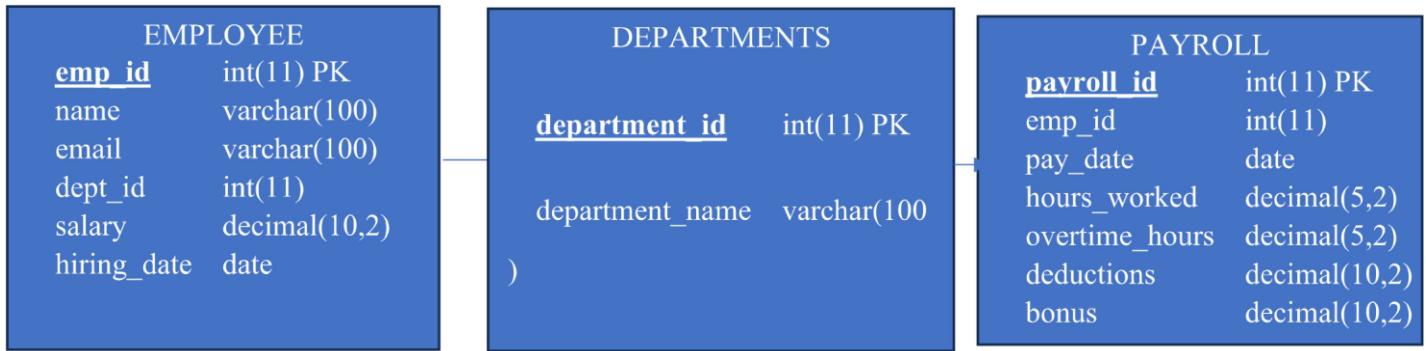
4. **Reporting:**

- Generate reports on employee salaries, payroll expenses by department, total salary expenses, etc.
- Provide insights into payroll trends and financial analysis.

5. Security and Access Control:

- Implement user authentication and authorization to ensure only authorized personnel can access sensitive payroll information.
- Enforce data security measures to protect employee and financial data.

2. ENTITY RELATION – DAIGRAM FOR PAYROLL MANGEMENT



COMMANDS:

--Create employees table

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    dept_id INT,
    salary DECIMAL(10, 2),
    hiring_date DATE
);
```

-- Insert sample data for employees

```
INSERT INTO employees (emp_id, name, email, dept_id, salary, hiring_date) VALUES
(1, 'John Doe', 'john.doe@example.com', 1, 60000.00, '2020-01-15'),
(2, 'Jane Smith', 'jane.smith@example.com', 2, 55000.00, '2020-02-20'),
(3, 'Michael Johnson', 'michael.johnson@example.com', 1, 62000.00, '2019-11-10'),
(4, 'Emily Brown', 'emily.brown@example.com', 3, 58000.00, '2021-03-05'),
(5, 'David Davis', 'david.davis@example.com', 2, 54000.00, '2020-09-15'),
(6, 'Sarah Wilson', 'sarah.wilson@example.com', 1, 61000.00, '2018-08-20'),
(7, 'Matthew Taylor', 'matthew.taylor@example.com', 3, 59000.00, '2017-07-25'),
(8, 'Jennifer Martinez', 'jennifer.martinez@example.com', 2, 56000.00, '2019-06-30'),
(9, 'Christopher Anderson', 'christopher.anderson@example.com', 1, 63000.00, '2018-04-12'),
(10, 'Amanda Garcia', 'amanda.garcia@example.com', 3, 57000.00, '2017-02-05'),
(11, 'James Rodriguez', 'james.rodriguez@example.com', 2, 55000.00, '2020-10-10'),
(12, 'Linda Nguyen', 'linda.nguyen@example.com', 1, 64000.00, '2019-01-20'),
(13, 'Ryan King', 'ryan.king@example.com', 3, 58000.00, '2018-06-15'),
(14, 'Kimberly Thomas', 'kimberly.thomas@example.com', 2, 57000.00, '2017-09-30'),
(15, 'Eva Martinez', 'eva.martinez@example.com', 4, 55000.00, '2021-05-10');
```

-- Create payroll table

```
CREATE TABLE payroll (
    payroll_id INT PRIMARY KEY,
    emp_id INT,
    pay_date DATE,
    hours_worked DECIMAL(5, 2),
    overtime_hours DECIMAL(5, 2),
    deductions DECIMAL(10, 2),
    bonus DECIMAL(10, 2)
);
```

-- Insert sample payroll data

```
INSERT INTO payroll (payroll_id, emp_id, pay_date, hours_worked, overtime_hours,
deductions, bonus) VALUES
(1, 1, '2024-04-01', 160.00, 10.00, 500.00, 1000.00),
(2, 2, '2024-04-01', 150.00, 5.00, 450.00, 900.00),
(3, 3, '2024-04-01', 165.00, 12.00, 550.00, 1100.00),
(4, 4, '2024-04-01', 155.00, 8.00, 520.00, 950.00),
(5, 5, '2024-04-01', 145.00, 6.00, 480.00, 800.00),
(6, 6, '2024-04-01', 170.00, 14.00, 600.00, 1200.00),
```

```
(7, 7, '2024-04-01', 155.00, 8.00, 520.00, 950.00),
(8, 8, '2024-04-01', 150.00, 5.00, 450.00, 900.00),
(9, 9, '2024-04-01', 160.00, 10.00, 500.00, 1000.00),
(10, 10, '2024-04-01', 145.00, 6.00, 480.00, 800.00),
(11, 11, '2024-04-01', 155.00, 8.00, 520.00, 950.00),
(12, 12, '2024-04-01', 170.00, 14.00, 600.00, 1200.00),
(13, 13, '2024-04-01', 155.00, 8.00, 520.00, 950.00),
(14, 14, '2024-04-01', 150.00, 5.00, 450.00, 900.00),
(15, 15, '2024-04-01', 165.00, 12.00, 550.00, 1100.00);
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(100)
);
INSERT INTO departments (department_id, department_name) VALUES
(1, 'Engineering'),
(2, 'Sales'),
(3, 'Marketing'),
(4, 'Human Resources');
```

. Join to get **Table Description**

1.Employees

Field	Type	Null	Key	Default	Extra
emp_id	int(11)	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
email	varchar(100)	YES		NULL	
dept_id	int(11)	YES		NULL	
salary	decimal(10,2)	YES		NULL	
hiring_date	date	YES		NULL	

2.Payroll

Field	Type	Null	Key	Default	Extra
payroll_id	int(11)	NO	PRI	NULL	
emp_id	int(11)	YES		NULL	
pay_date	date	YES		NULL	
hours_worked	decimal(5,2)	YES		NULL	
overtime_hours	decimal(5,2)	YES		NULL	
deductions	decimal(10,2)	YES		NULL	
bonus	decimal(10,2)	YES		NULL	

3.Department

Field	Type	Null	Key	Default	Extra
department_id	int(11)	NO	PRI	NULL	
department_name	varchar(100)	YES		NULL	

SUBQUERIES:

1. Top Earners in Each Department:

```
SELECT department_name, name, salary
FROM (
    SELECT e.name, e.salary, d.department_name,
    ROW_NUMBER() OVER (PARTITION BY e.dept_id ORDER BY e.salary DESC) AS rn
    FROM employees e
    JOIN departments d ON e.dept_id = d.department_id
) AS ranked
WHERE rn = 1;
```

RESULT:

department_name	name	salary
Engineering	Linda Nguyen	64000
Sales	Kimberly Thomas	57000
Marketing	Matthew Taylor	59000
Human Resources	Eva Martinez	55000

2. SUBQUERIES 2 :Employees With Highest Overtime Hours:

```
SELECT e.name, p.overtime_hours
FROM employees e
JOIN payroll p ON e.emp_id = p.emp_id
ORDER BY p.overtime_hours DESC
LIMIT 5;
```

RESULT:

name	overtime_hours
Linda Nguyen	14
Sarah Wilson	14
Eva Martinez	12
Michael Johnson	12
Christopher Anderson	10

SUBQUERY 3:-Average Salary of Employees in Each Department:

```
SELECT department_name, AVG(salary) AS avg_salary
FROM employees e
JOIN departments d ON e.dept_id = d.department_id
GROUP BY department_name;
```

RESULT:

department_name	avg_salary
Engineering	62000
Human Resources	55000
Marketing	58000
Sales	55400

4. Total Number of Employees in Each Department:

```
SELECT department_name, COUNT(*) AS num_employees
FROM employees e
JOIN departments d ON e.dept_id = d.department_id
GROUP BY department_name;
```

RESULT:

department_name	num_employees
Engineering	5
Human Resources	1
Marketing	4
Sales	5

- . 5 Employees With Highest Overtime Hours

```
SELECT e.name, p.overtime_hours  
FROM employees e  
JOIN payroll p ON e.emp_id = p.emp_id  
ORDER BY p.overtime_hours DESC  
LIMIT 5;
```

RESULT:

name	overtime_hours
Linda Nguyen	14
Sarah Wilson	14
Eva Martinez	12
Michael Johnson	12
Christopher Anderson	10

Join to get
JOINS

1. Join to get employee details along with their payroll information:

```
SELECT e.emp_id, e.name, p.pay_date, p.hours_worked, p.overtime_hours, p.deductions,  
p.bonus  
FROM employees e  
JOIN payroll p ON e.emp_id = p.emp_id;
```

RESULT:

emp_id	name	pay_date	hours_worked	overtime_hours	deductions	bonus
1	John Doe	01-04-2024	160	10	500	1000
2	Jane Smith	01-04-2024	150	5	450	900
3	Michael Johnson	01-04-2024	165	12	550	1100
4	Emily Brown	01-04-2024	155	8	520	950
5	David Davis	01-04-2024	145	6	480	800
6	Sarah Wilson	01-04-2024	170	14	600	1200
7	Matthew Taylor	01-04-2024	155	8	520	950
8	Jennifer Martinez	01-04-2024	150	5	450	900
9	Christopher Anderson	01-04-2024	160	10	500	1000
10	Amanda Garcia	01-04-2024	145	6	480	800
11	James Rodriguez	01-04-2024	155	8	520	950
12	Linda Nguyen	01-04-2024	170	14	600	1200
13	Ryan King	01-04-2024	155	8	520	950
14	Kimberly Thomas	01-04-2024	150	5	450	900

15	Eva Martinez	01-04-2024	165	12	550	1100
----	--------------	------------	-----	----	-----	------

2 employee details along with their department name:

```
SELECT e.emp_id, e.name, d.department_name
FROM employees e
JOIN departments d ON e.dept_id = d.department_id;
```

RESULT:

emp_id	name	department_name
1	John Doe	Engineering
2	Jane Smith	Sales
3	Michael Johnson	Engineering
4	Emily Brown	Marketing
5	David Davis	Sales
6	Sarah Wilson	Engineering
7	Matthew Taylor	Marketing
8	Jennifer Martinez	Sales
9	Christopher Anderson	Engineering
10	Amanda Garcia	Marketing
11	James Rodriguez	Sales
12	Linda Nguyen	Engineering
13	Ryan King	Marketing
14	Kimberly Thomas	Sales
15	Eva Martinez	Human Resources

3 . Join to get payroll details along with employee and department information:

```
SELECT e.name, d.department_name, p.pay_date, p.hours_worked, p.overtime_hours,  
p.deductions, p.bonus
```

```
FROM payroll p
```

```
JOIN employees e ON p.emp_id = e.emp_id
```

```
JOIN departments d ON e.dept_id = d.department_id;
```

RESULT:

name	department_name	pay_date	hours_worked	overtime_hours	deductions	bonus
John Doe	Engineering	01-04-2024	160	10	500	1000
Michael Johnson	Engineering	01-04-2024	165	12	550	1100
Sarah Wilson	Engineering	01-04-2024	170	14	600	1200
Christopher Anderson	Engineering	01-04-2024	160	10	500	1000
Linda Nguyen	Engineering	01-04-2024	170	14	600	1200
Jane Smith	Sales	01-04-2024	150	5	450	900
David Davis	Sales	01-04-2024	145	6	480	800
Jennifer Martinez	Sales	01-04-2024	150	5	450	900
James Rodriguez	Sales	01-04-2024	155	8	520	950
Kimberly Thomas	Sales	01-04-2024	150	5	450	900
Emily Brown	Marketing	01-04-2024	155	8	520	950
Matthew Taylor	Marketing	01-04-2024	155	8	520	950
Amanda Garcia	Marketing	01-04-2024	145	6	480	800
Ryan King	Marketing	01-04-2024	155	8	520	950
Eva Martinez	Human Resources	01-04-2024	165	12	550	1100

4 the total number of employees in each department along with their department names:

```
SELECT d.department_name, COUNT(e.emp_id) AS num_employees  
FROM departments d  
LEFT JOIN employees e ON d.department_id = e.dept_id  
GROUP BY d.department_name;
```

RESULT:

department_name	num_employees
Engineering	5
Human Resources	1
Marketing	4
Sales	5

5. Join to get the total salary expense for each department:

```
SELECT d.department_name, SUM(e.salary) AS total_salary_expense  
FROM departments d  
JOIN employees e ON d.department_id = e.dept_id  
GROUP BY d.department_name;
```

RESULT:

department_name	total_salary_expense
Engineering	310000
Human Resources	55000
Marketing	232000
Sales	277000

Queries

1. List all employees and their salaries

```
SELECT name , salary from employees;
```

RESULT:

NAME	SALARY
John Doe	60000
Jane Smith	55000
Michael Johnson	62000
Emily Brown	58000
David Davis	54000
Sarah Wilson	61000
Matthew Taylor	59000
Jennifer Martinez	56000
Christopher Anderson	63000
Amanda Garcia	57000
James Rodriguez	55000
Linda Nguyen	64000
Ryan King	58000
Kimberly Thomas	57000
Eva Martinez	55000

2. Calculate total salary expenditure

```
SELECT SUM(Salary) AS TotalSalaryExpenditure  
FROM Employees;
```

RESULT:

TotalSalaryExpenditure
874000

3. List employees with salary greater than the average salary

```
SELECT name , Salary  
FROM Employees  
WHERE Salary > (SELECT AVG(Salary) FROM Employees);
```

RESULT:

NAME	SALARY
John Doe	60000
Michael Johnson	62000

Sarah Wilson	61000
Matthew Taylor	59000
Christopher Anderson	63000
Linda Nguyen	64000

4. Retrieve the highest salary among all employees.

SELECT MAX(Salary) As max salary from employees;

RESULT:

MAX SALARY
64000

5. Retrieve the average salary of all employees

SELECT AVG(Salary) as average salary from employees;

RESULT:

Average salary
58266.66667