Project report on

**EMPLOYEE ATTENDENCE AND PAYROLL SYSYTEM**

**Submitted in partial fulfillment of the requirements for the award of the degree**

Of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING(ARTIFICIAL**

**INTELLIGENCE AND MACHINE LEARNING)**



**Department of Computer Science and Engineering**

**(Artificial Intelligence and Machine learning)**

**BVRIT HYDERABAD College of Engineering for Women**

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

(Accredited by NBA – EEE, ECE, CSE and IT and NAAC with A grade) Bachupally, Hyderabad – 500090

2023-24

**CERTIFICATE**

This is to certify that the Technical Report titled “**EMPLOYEE ATTENDENCE AND PAYROLL SYSYTEM**” is being submitted by Team 11&12 of the II B.Tech Computer Science and Engineering, (Academic year 2023- 2024) during the II semester B.Tech in CSE(AI&ML)

|  |  |
| --- | --- |
| **Mrs.V.Asha**  **Assistant Professor,**  Department of  Computer Science  And Engineering | **DONE BY:**  Megana-22WH1A6603  Devika-22WH1A6611  Indu- 22WH1A6619  Bhavya-22WH1A6624  Kritika-22WH1A6625  Jahnvi-22WH1A6640  Prasanna-22WH1A6643  Gayathri-22WH1A6656  Sushma-23WH5A6606  Rajeshwari-23WH1A6607 |

(AI&ML)

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **Page.No** |
| **1.** | **Introduction** |  |
| **2.** | **Abstract** |  |
| **3.** | **Requirements** |  |
| **4.** | **Entity Representation Model** |  |
| **5.** | **Code Implementation** |  |
| **6.** | **Queries** |  |
| **7.** | **Testing And Results** |  |

**INTRODUCTION**

In today's competitive business environment, efficient management of employee attendance and payroll is crucial for organizational success. Companies need a robust system to track employee work hours, manage leave requests, accurately calculate payroll, and ensure compliance with tax regulations. A well-designed Employee Attendance and Payroll System not only streamlines administrative processes but also enhances productivity and employee satisfaction by providing transparent and timely payroll management.

**ABSTRACT**

This project proposes the development of an Employee Attendance and Payroll System, designed to facilitate comprehensive management of employee-related data. The system encompasses several core functionalities, including attendance tracking, work hour calculations, leave request management, payroll computations, tax deductions, and maintenance of employee information. The database-driven system aims to automate and simplify the administrative tasks associated with human resource management, ensuring accuracy and efficiency in payroll processing and attendance monitoring.

The system will feature a relational database with interconnected tables for employees, departments, attendance records, leave requests, payroll details, and tax information. Key functionalities include recording employee check-in and check-out times, calculating total hours worked, handling various types of leave requests, generating payroll based on worked hours and predefined salary rates, and applying tax deductions to compute net pay. The system will also support detailed reporting capabilities, allowing administrators to generate insights into attendance patterns, payroll expenses, and leave utilization.

By integrating these functionalities into a unified system, the proposed solution aims to reduce manual errors, enhance data accuracy, and improve overall administrative efficiency. This project will ultimately contribute to better resource management and compliance with legal and financial regulations, providing a reliable tool for organizations to manage their human resources effectively.

The system will feature a relational database with interconnected tables for employees, departments, attendance records, leave requests, payroll details, and tax information. Key functionalities include recording employee check-in and check-out times, calculating total hours worked, handling various types of leave requests, generating payroll based on worked hours and predefined salary rates, and applying tax deductions to compute net pay. The system will also support detailed reporting capabilities, allowing administrators to generate insights into attendance patterns, payroll expenses, and leave utilization.

By integrating these functionalities into a unified system, the proposed solution aims to reduce manual errors, enhance data accuracy, and improve overall administrative efficiency. This project will ultimately contribute to better resource management and compliance with legal and financial regulations, providing a reliable tool for organizations to manage their human resources effectively.

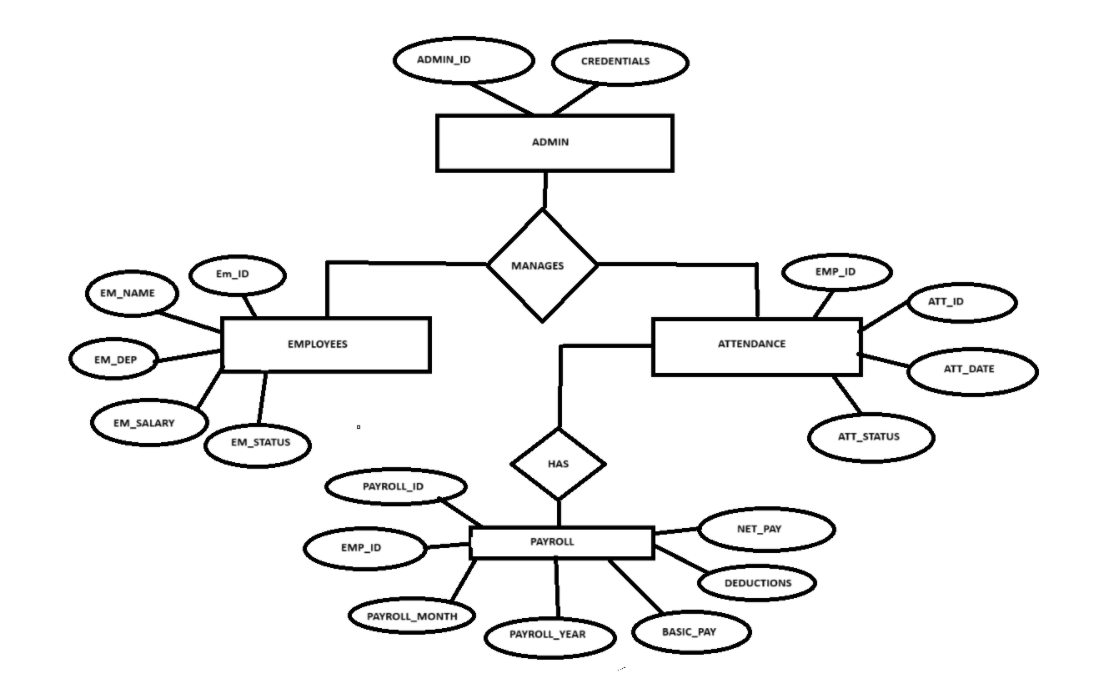
**TECHNICAL REQUIREMENTS:**

1. Database Management System (DBMS):

* Use a relational DBMS such as MySQL, PostgreSQL, or SQL Server.
* Ensure the DBMS supports ACID (Atomicity, Consistency, Isolation, Durability) properties for reliable transactions.

1. Perform thorough testing, including unit tests, integration tests, and end-to-end tests.

**Entity Representation Model**



CODE IMPLEMENTATION

* **Create database if not exists**

CREATE DATABASE IF NOT EXISTS employee\_management\_system;

--Use the database

USE employee\_management\_system;

* **Employees table**

CREATE TABLE IF NOT EXISTS employees (

employee\_id INT PRIMARY KEY AUTO\_INCREMENT,

employee\_name VARCHAR(100) NOT NULL,

employee\_department VARCHAR(100),

employee\_salary DECIMAL(10, 2),

employee\_status VARCHAR(20));

* **Inserting data into Employees table**

INSERT INTO employees (employee\_name, employee\_department, employee\_salary, employee\_status)

VALUES ('Sushma', 'IT', 50000.00, 'active'), ('Rajeshwari', 'HR', 450

00.00, 'active'), ('Indu', 'Finance', 55000.00, 'active'), ('Jahnvi', 'Marke

ting', 48000.00, 'active'), ('Devika', 'IT', 51000.00, 'active'), ('Prasanna', 'Finance', 56000.00, 'active');

* **Attendance table**

CREATE TABLE IF NOT EXISTS attendance (

attendance\_id INT PRIMARY KEY AUTO\_INCREMENT,

employee\_id INT,

attendance\_date DATE,

attendance\_status VARCHAR(20),

FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id));

INSERT INTO attendance (employee\_id, attendance\_date, attendance\_status)

VALUES (1, '2024-06-15', 'present'),

(2, '2024-06-15', 'present'),

(3, '2024-06-15', 'absent'),

(4, '2024-06-15', 'present'),

(5, '2024-06-15', 'present'),

(6, '2024-06-15', 'absent');

CREATE TABLE IF NOT EXISTS payroll (

payroll\_id INT PRIMARY KEY AUTO\_INCREMENT,

employee\_id INT,

payroll\_month INT,

payroll\_year INT,

basic\_pay DECIMAL(10, 2),

deductions DECIMAL(10, 2),

net\_pay DECIMAL(10, 2),

FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id));

* **Inserting data into Payroll table**

INSERT INTO payroll (employee\_id, payroll\_month, payroll\_year, basic\_pay, deductions, net\_pay)

VALUES (1, 6, 2024, 5000.00, 500.00, 4500.00),

(2, 6, 2024, 4500.00, 400.00, 4100.00),

(3, 6, 2024, 5500.00, 600.00, 4900.00),

(4, 6, 2024, 4800.00, 450.00, 4350.00),

(5, 6, 2024, 5100.00, 480.00, 4620.00),

(6, 6, 2024, 5600.00, 700.00, 4900.00);

* **Updating Employee Details**

UPDATE employees

SET employee\_salary = 52000.00

WHERE employee\_id = 1;

-- Deleting an Employee (soft delete)

UPDATE employees

SET employee\_status = 'resigned'

WHERE employee\_id = 3;

* **Fetching Employee Attendance for a Date Range**

SELECT e.employee\_name, a.attendance\_date, a.attendance\_status

FROM employees e

JOIN attendance a ON e.employee\_id = a.employee\_id

WHERE a.attendance\_date BE-- Calculating Total Payroll for a Month

SELECT e.employee\_name, p.payroll\_month, p.payroll\_year, p.net\_pay

FROM employees e

JOIN payroll p ON e.employee\_id = p.employee\_id

WHERE p.payroll\_month = 6 AND p.payroll\_year = 2024;TWEEN '2024-06-01' AND '2024-06-30';

Aggregate Functions

* **Calculating Total Payroll for a Month**

SELECT e.employee\_name, p.payroll\_month, p.payroll\_year, p.net\_pay

FROM employees e

JOIN payroll p ON e.employee\_id = p.employee\_id

WHERE p.payroll\_month = 6 AND p.payroll\_year = 2024;

* **Generating Monthly Attendance Report**

SELECT e.employee\_name, COUNT(a.attendance\_id) AS days\_present

FROM employees e

LEFT JOIN attendance a ON e.employee\_id = a.employee\_id

WHERE a.attendance\_status = 'present'

GROUP BY e.employee\_id;

* **Calculating Average Salary by Department**

SELECT employee\_department, AVG(employee\_salary) AS avg\_salary

FROM employees

GROUP BY employee\_department;

* **Regular Expressions**

-- Finding Employees with Highest Deductions

SELECT e.employee\_name, p.deductions

FROM employees e

JOIN payroll p ON e.employee\_id = p.employee\_id

ORDER BY p.deductions DESC

LIMIT 10;

-- Views

-- Example View: Employees with Active Status

CREATE VIEW active\_employees AS

SELECT \*

FROM employees

WHERE employee\_status = 'active';

-- Example View: Employees and Their Payroll Information

CREATE VIEW employee\_payroll\_info AS

SELECT e.employee\_name, p.payroll\_month, p.payroll\_year, p.net\_pay

FROM employees e

JOIN payroll p ON e.employee\_id = p.employee\_id;

-- Deleting Attendance Records Older than a Year

DELETE FROM attendance

WHERE attendance\_date < DATE\_SUB(CURDATE(), INTERVAL 1 YEAR);

**Queries**

1: Create a new table to store department details

CREATE TABLE IF NOT EXISTS departments (

department\_id INT PRIMARY KEY AUTO\_INCREMENT,

department\_name VARCHAR(100) NOT NULL,

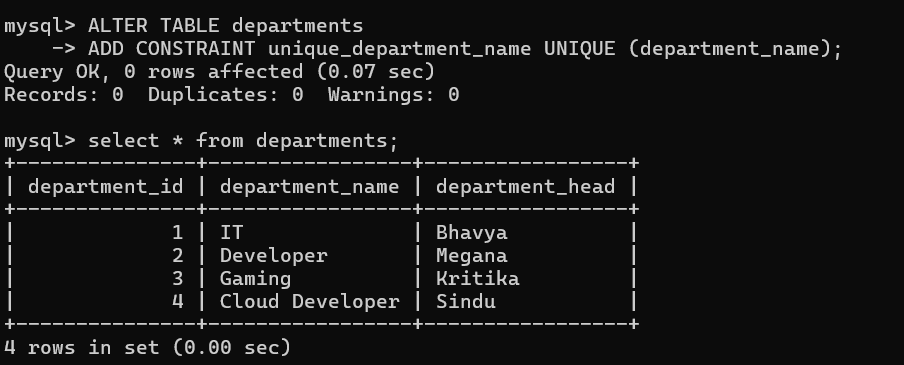
department\_head VARCHAR(100));



2: Add a unique constraint to the department\_name column in the departments table

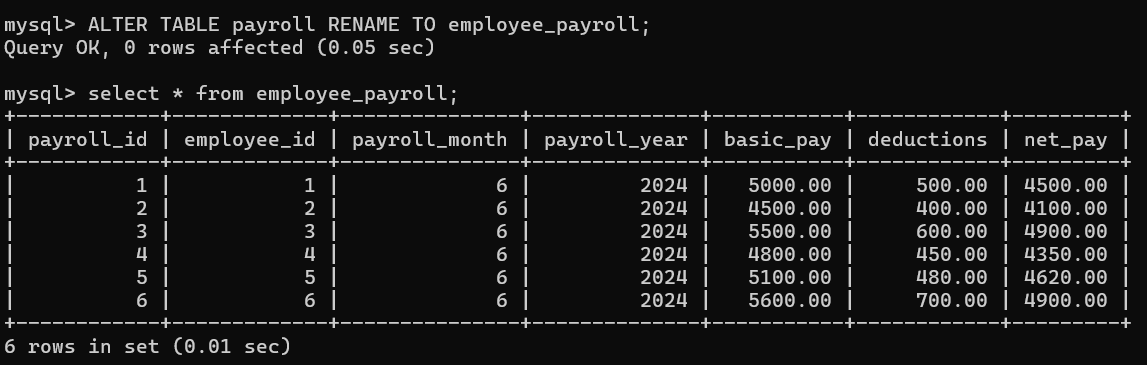
ALTER TABLE departments

ADD CONSTRAINT unique\_department\_name UNIQUE (department\_name);



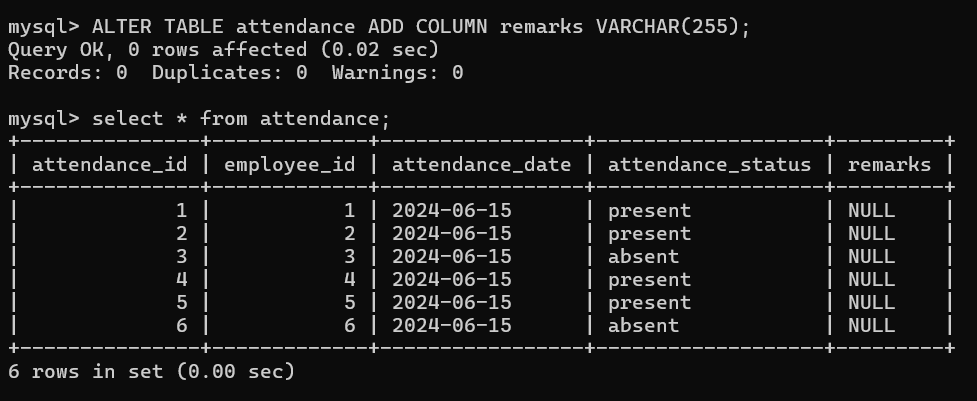
3: Rename the payroll table to employee\_payroll

ALTER TABLE payroll RENAME TO employee\_payroll;



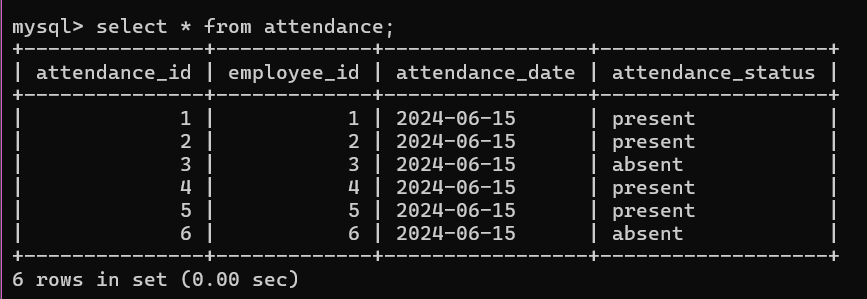
4: Add a column to the attendance table to store remarks

ALTER TABLE attendance ADD COLUMN remarks VARCHAR(255);



5: Drop the remarks column from the attendance table

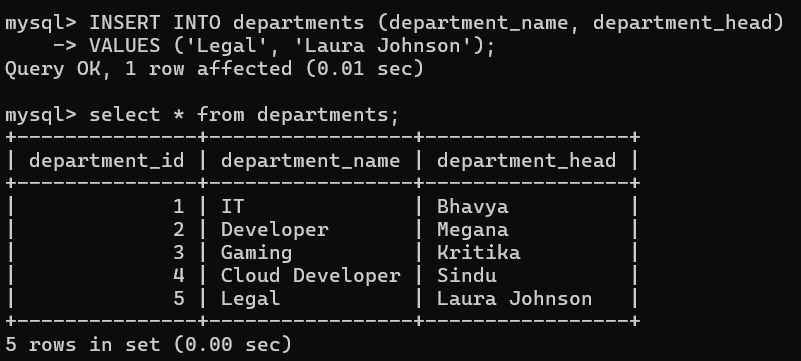
ALTER TABLE attendance DROP COLUMN remarks;



6: Insert a new department into the departments table

INSERT INTO departments (department\_name, department\_head)

VALUES ('Legal', 'Laura Johnson');

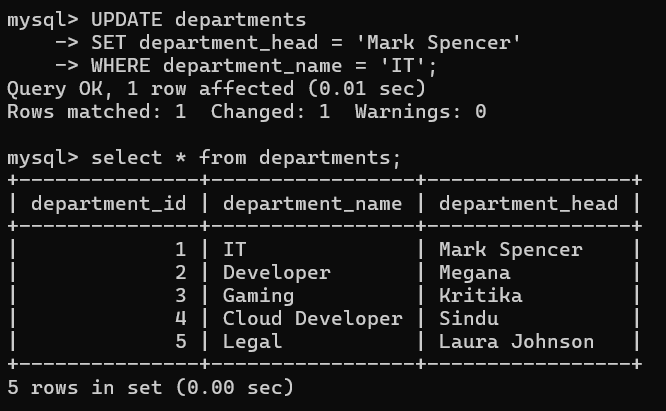


7: Update the department head for the IT department

UPDATE departments

SET department\_head = 'Mark Spencer'

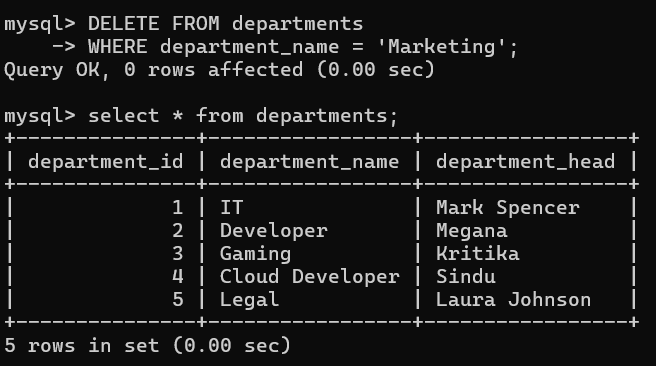
WHERE department\_name = 'IT';



8: Delete a department from the departments table

DELETE FROM departments

WHERE department\_name = 'Marketing';

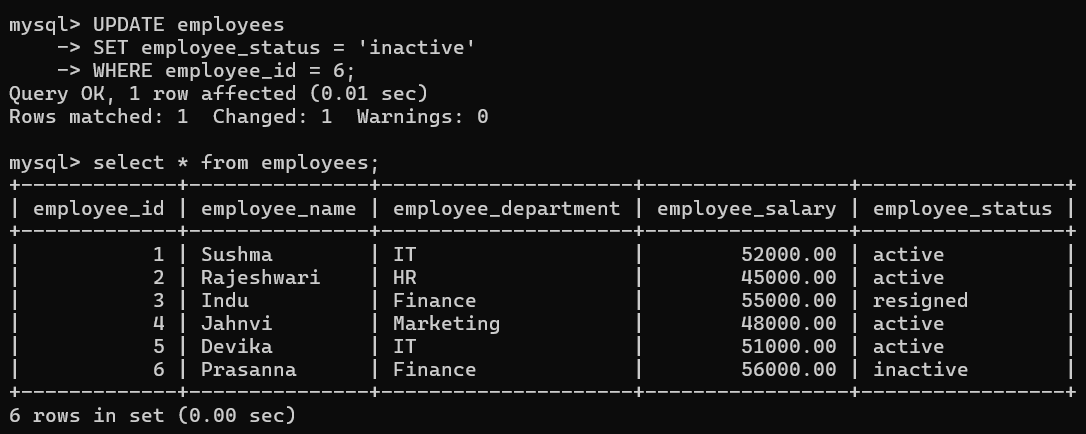


9: Soft delete an employee by setting their status to 'inactive'

UPDATE employees

SET employee\_status = 'inactive'

WHERE employee\_id = 6;



10: Bulk insert multiple employees into the employees table

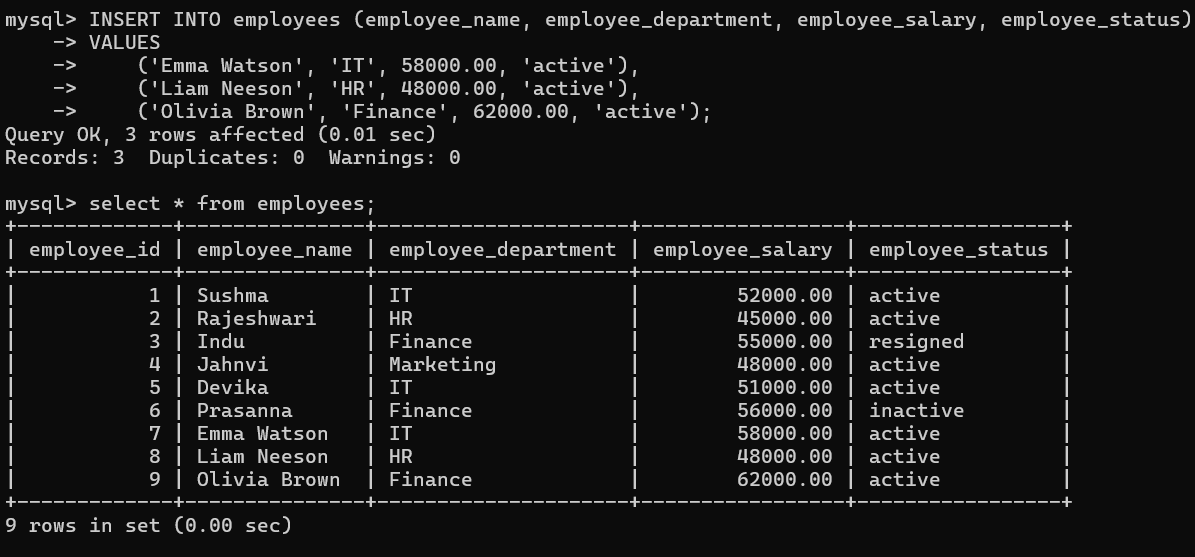
INSERT INTO employees (employee\_name, employee\_department, employee\_salary, employee\_status)

VALUES

('Emma Watson', 'IT', 58000.00, 'active'),

('Liam Neeson', 'HR', 48000.00, 'active'),

('Olivia Brown', 'Finance', 62000.00, 'active');

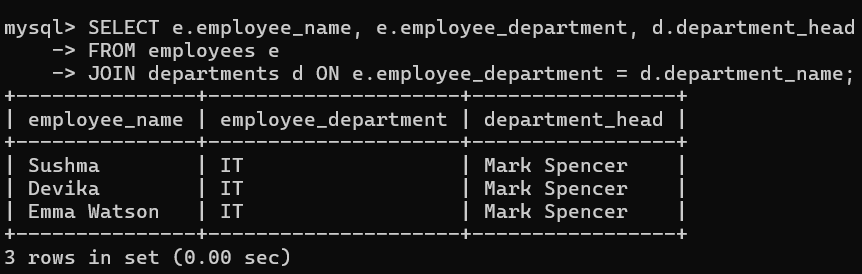


11: Fetch all employees along with their department head

SELECT e.employee\_name, e.employee\_department, d.department\_head

FROM employees e

JOIN departments d ON e.employee\_department = d.department\_name;



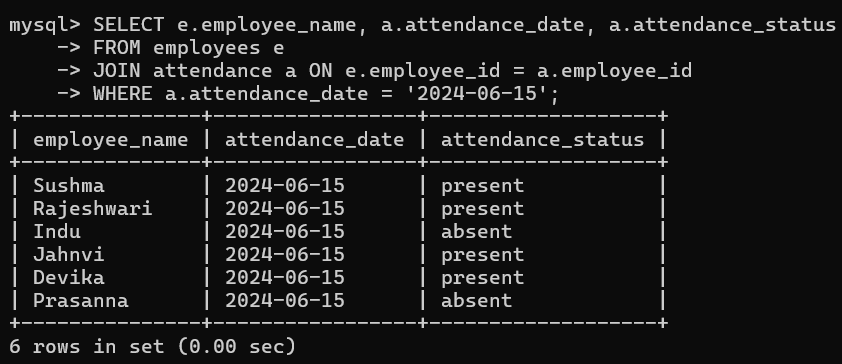
12: Fetch employees along with their attendance status for a specific date

SELECT e.employee\_name, a.attendance\_date, a.attendance\_status

FROM employees e

JOIN attendance a ON e.employee\_id = a.employee\_id

WHERE a.attendance\_date = '2024-06-15';



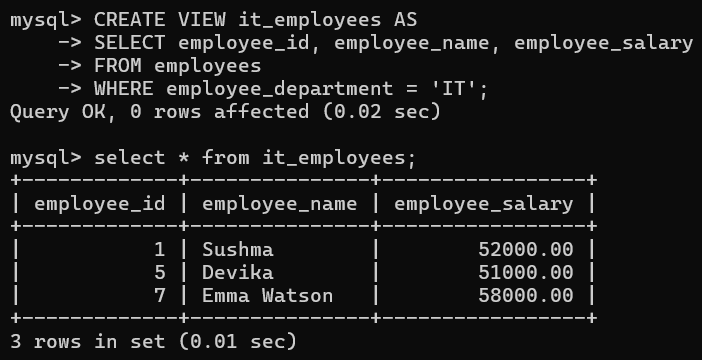
13: Create a view to show employees who are in the IT department

CREATE VIEW it\_employees AS

SELECT employee\_id, employee\_name, employee\_salary

FROM employees

WHERE employee\_department = 'IT';



14: Create a view to display the payroll details of employees with a net pay above 4500

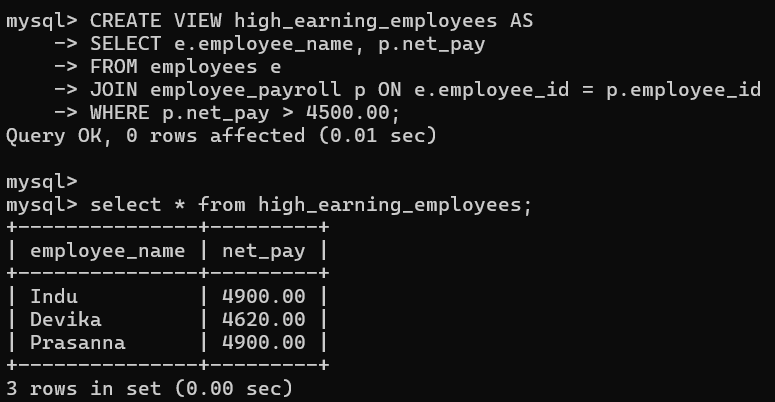
CREATE VIEW high\_earning\_employees AS

SELECT e.employee\_name, p.net\_pay

FROM employees e

JOIN employee\_payroll p ON e.employee\_id = p.employee\_id

WHERE p.net\_pay > 4500.00;

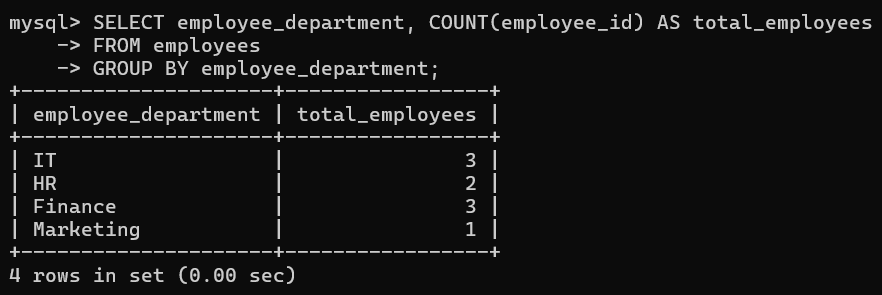


15: Calculate the total number of employees in each department

SELECT employee\_department, COUNT(employee\_id) AS total\_employees

FROM employees

GROUP BY employee\_department;

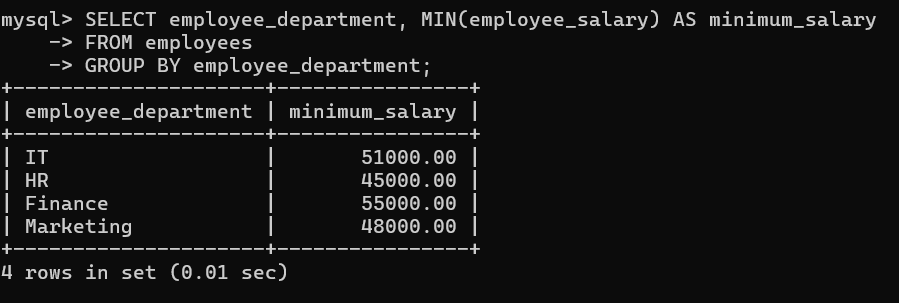


16: Calculate the minimum salary in each department

SELECT employee\_department, MIN(employee\_salary) AS minimum\_salary

FROM employees

GROUP BY employee\_department;

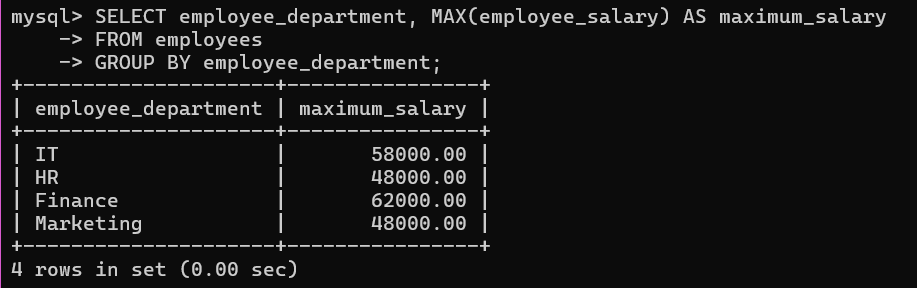


17: Calculate the maximum salary in each department

SELECT employee\_department, MAX(employee\_salary) AS maximum\_salary

FROM employees

GROUP BY employee\_department;

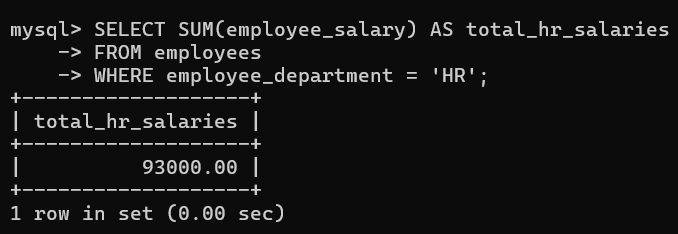


18: Calculate the sum of salaries for the HR department

SELECT SUM(employee\_salary) AS total\_hr\_salaries

FROM employees

WHERE employee\_department = 'HR';

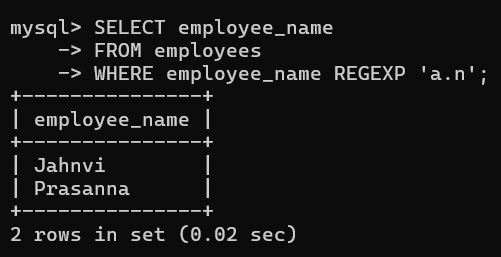


19: Find employees whose names contain 'a' followed by any character and then 'n'

SELECT employee\_name

FROM employees

WHERE employee\_name REGEXP 'a.n';

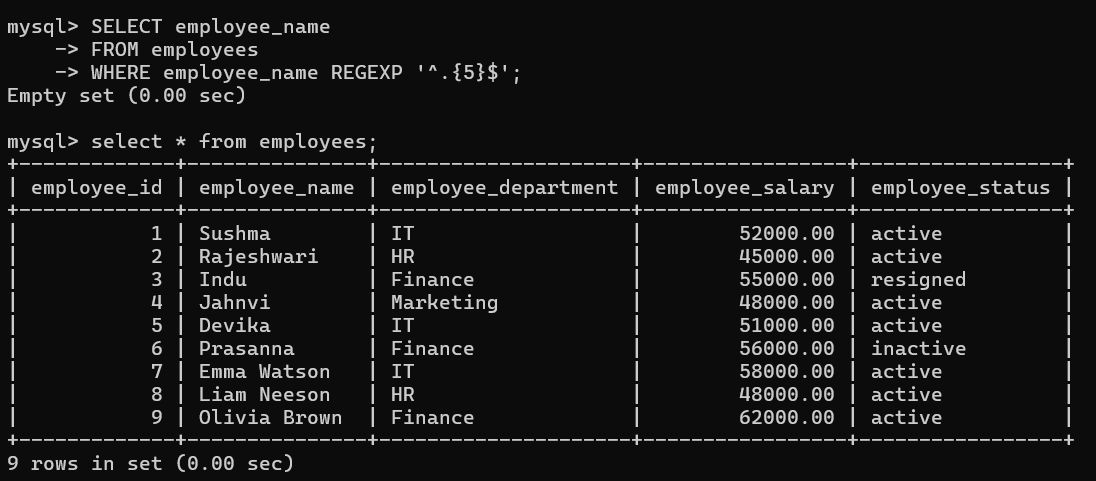


20: Find employees whose names contain exactly five characters

SELECT employee\_name

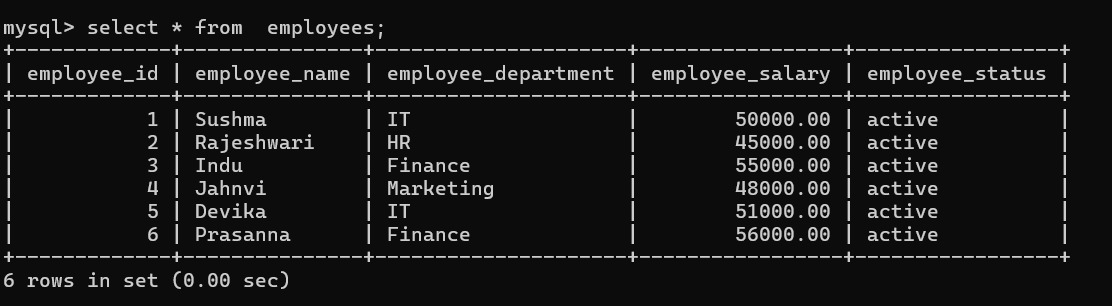
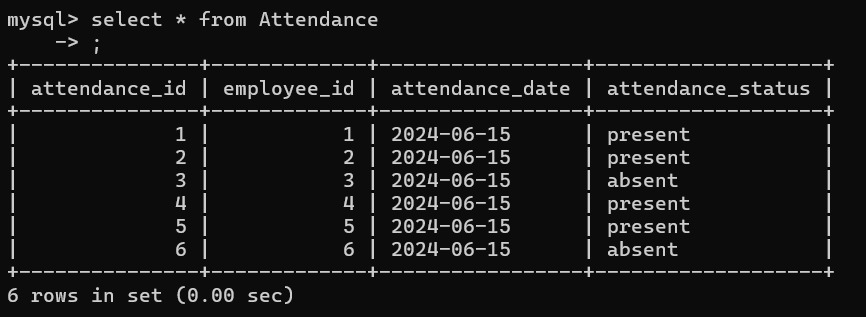
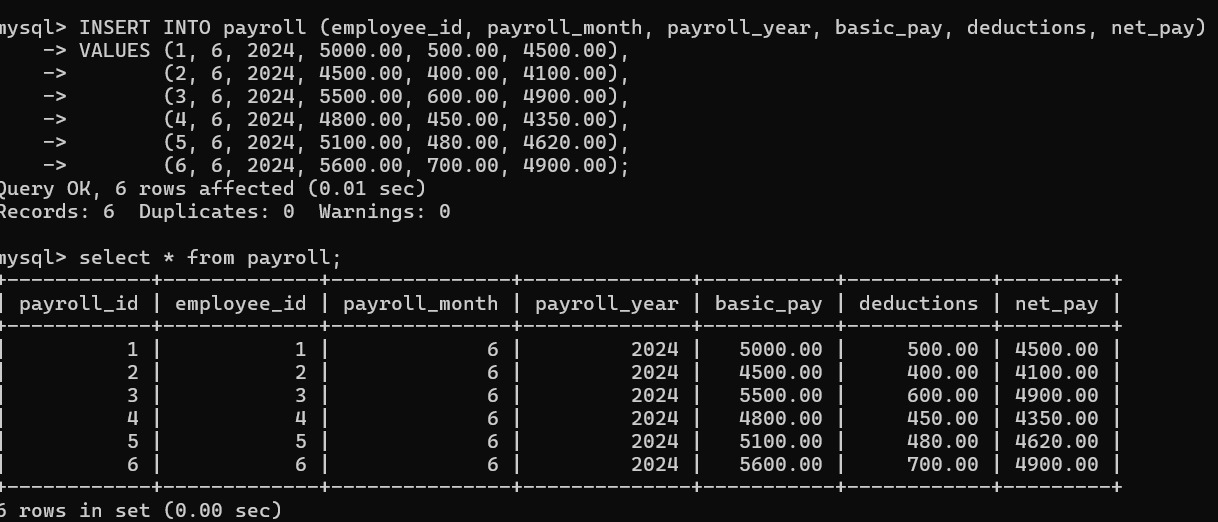
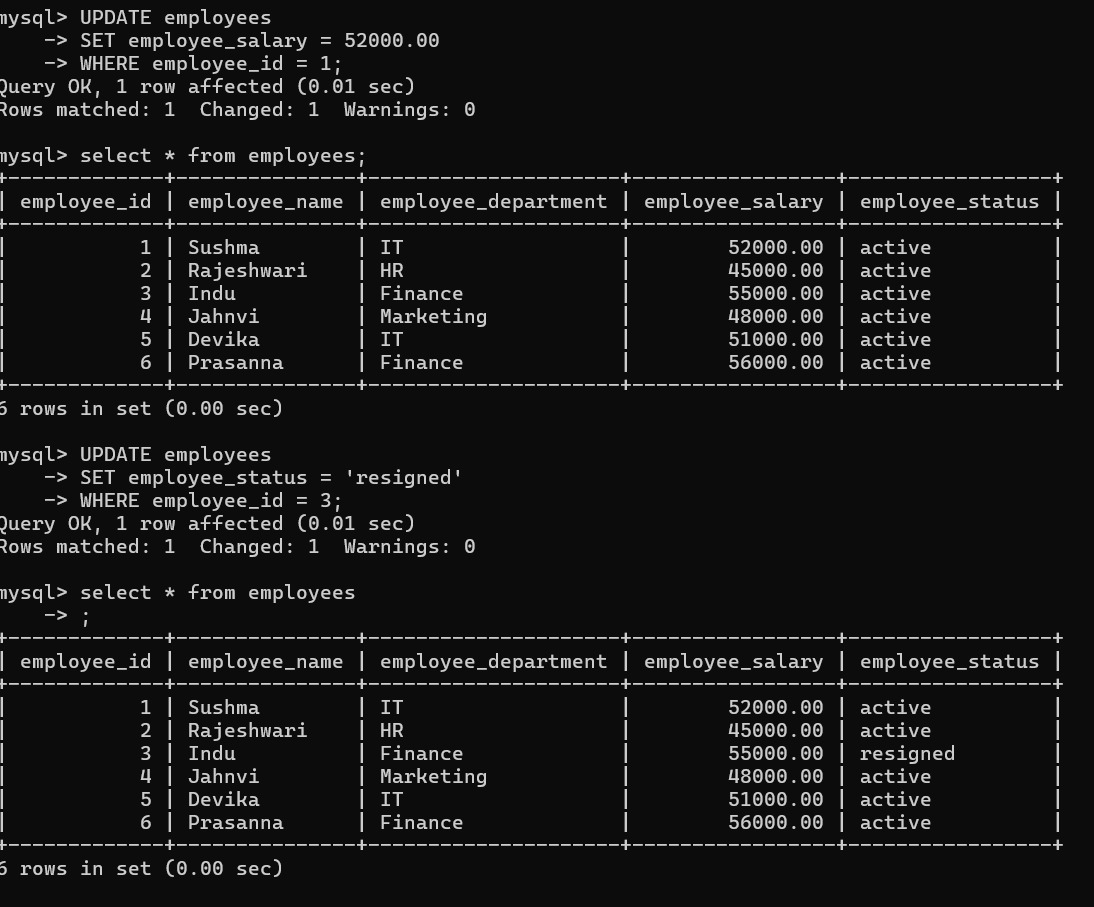
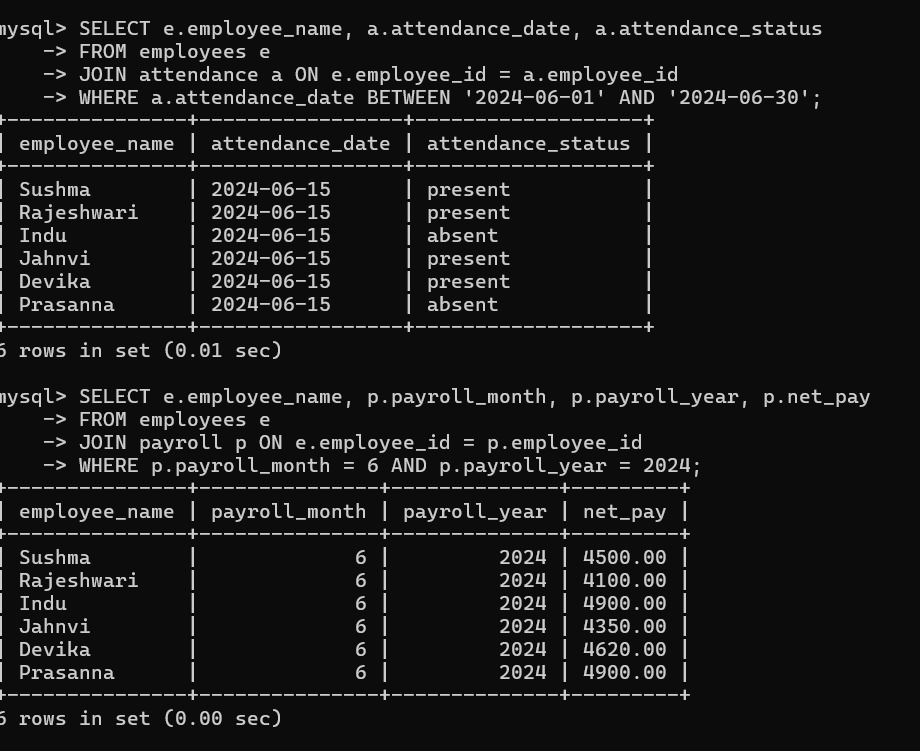
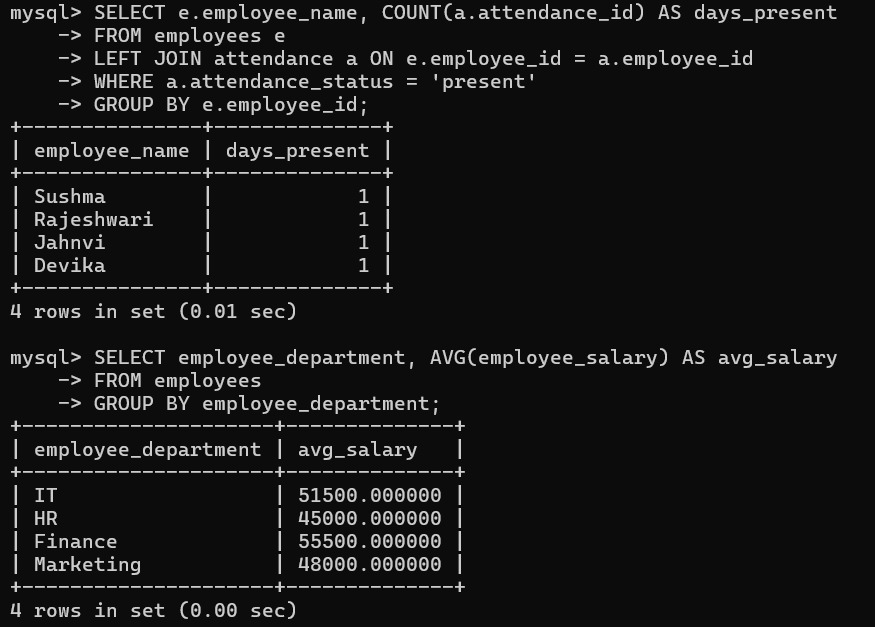
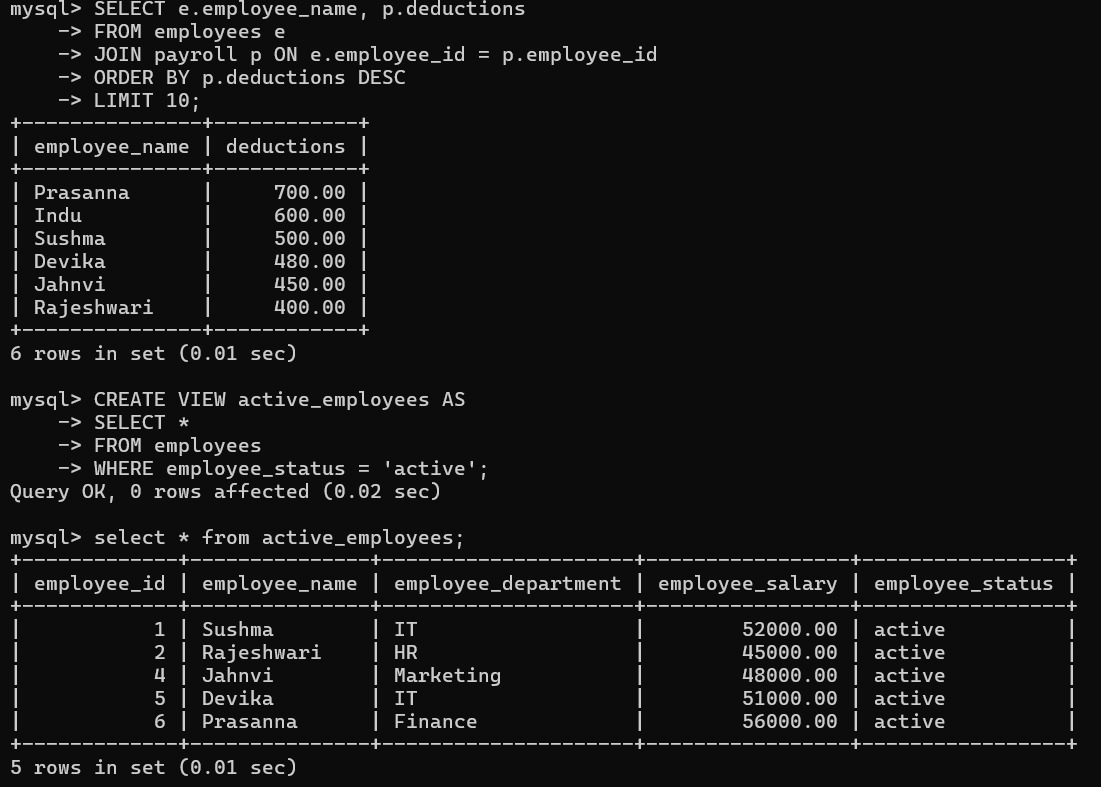
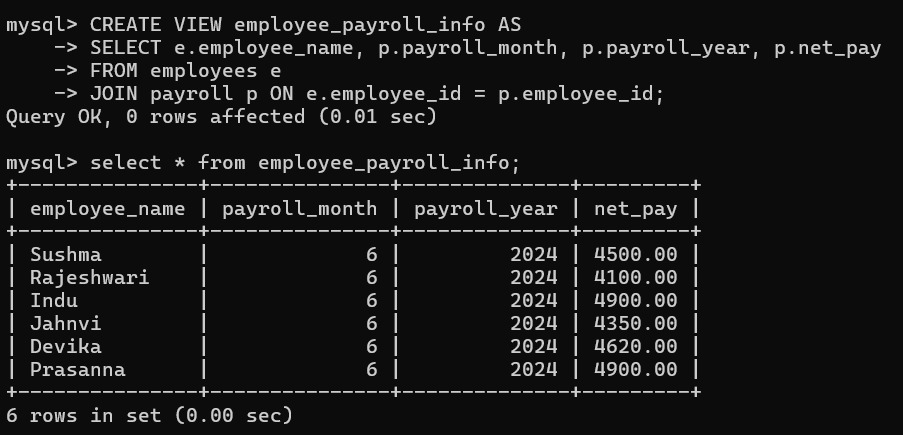
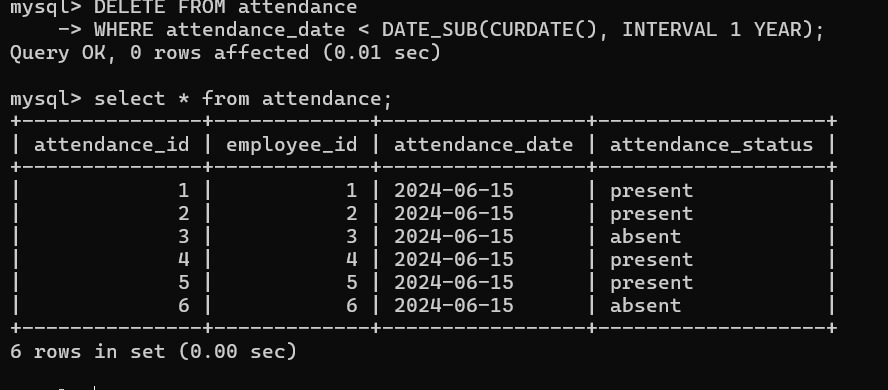
FROM employees

WHERE employee\_name REGEXP '^.{5}$';



**TESTING & RESULTS:**

**Here the Outputs of Code implementation**

****