A Report on Assignment titled

"Payroll Management System"

Submitted for partial fulfillment of V semester Database Management System

by

Prajwal Subramanya S N – 1SI22AD033

Submitted to:-

Dr. Bharathi P T

Department of Computer Science & Engineering
(Program Accredited by NBA)

Siddaganga Institute of Technology, Tumakuru-3
(An autonomous institution affiliated with VTU, Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC with 'A+' grade & ISO 9001:2015 Certified)

2024-25

Table of Contents

Abstract

This project implements a menu-driven Payroll Management System using C++ for the application layer and MySQL as the relational database. The system demonstrates secure database connectivity, schema design, and core payroll operations.

Key Features of this system include:

- **Automated Payroll Calculation**: The system automatically calculates the net salary by considering base salary, allowances, bonuses, and deductions, reducing manual errors and ensuring accuracy.

- **Employee Data Management**: It stores comprehensive employee details such as personal information, salary structure, attendance, and department allocation, ensuring easy access and updates.

- **Attendance Tracking**: The system logs daily attendance, calculates days present and absent, and integrates this data into salary computation for accurate payroll generation.

- **Bonus and Deduction Handling**: The system manages employee bonuses and deductions based on predefined conditions, integrating them into the salary calculation process.

- **Payroll Generation**: Using stored procedures, the system automatically generates monthly payroll records, ensuring timely salary processing and transparency.

- **Auto-incremented Payroll IDs**: The payroll records are uniquely identified by auto-incremented IDs, ensuring no duplication and easy tracking of records.

- **Data Integrity with Foreign Keys**: The system maintains data integrity by using foreign key constraints between employee records and related tables (e.g., salaries, attendance, bonuses).

Introduction

The **Payroll Management System** is a robust software solution designed to automate and streamline the payroll process, ensuring that employees are compensated accurately and on time. By eliminating manual calculations, data entry errors, and delays, this system integrates multiple core components such as employee details, attendance, salary structure, deductions, and bonuses into a unified database system. The system leverages **Database Management Systems (DBMS)** and **SQL** to store and manage critical information and automate payroll generation.

System Design

The **Payroll Management System** uses a **relational database structure**, which consists of several interconnected tables to maintain data integrity and support the system's core functionalities. Below is an overview of the key structures and core components:

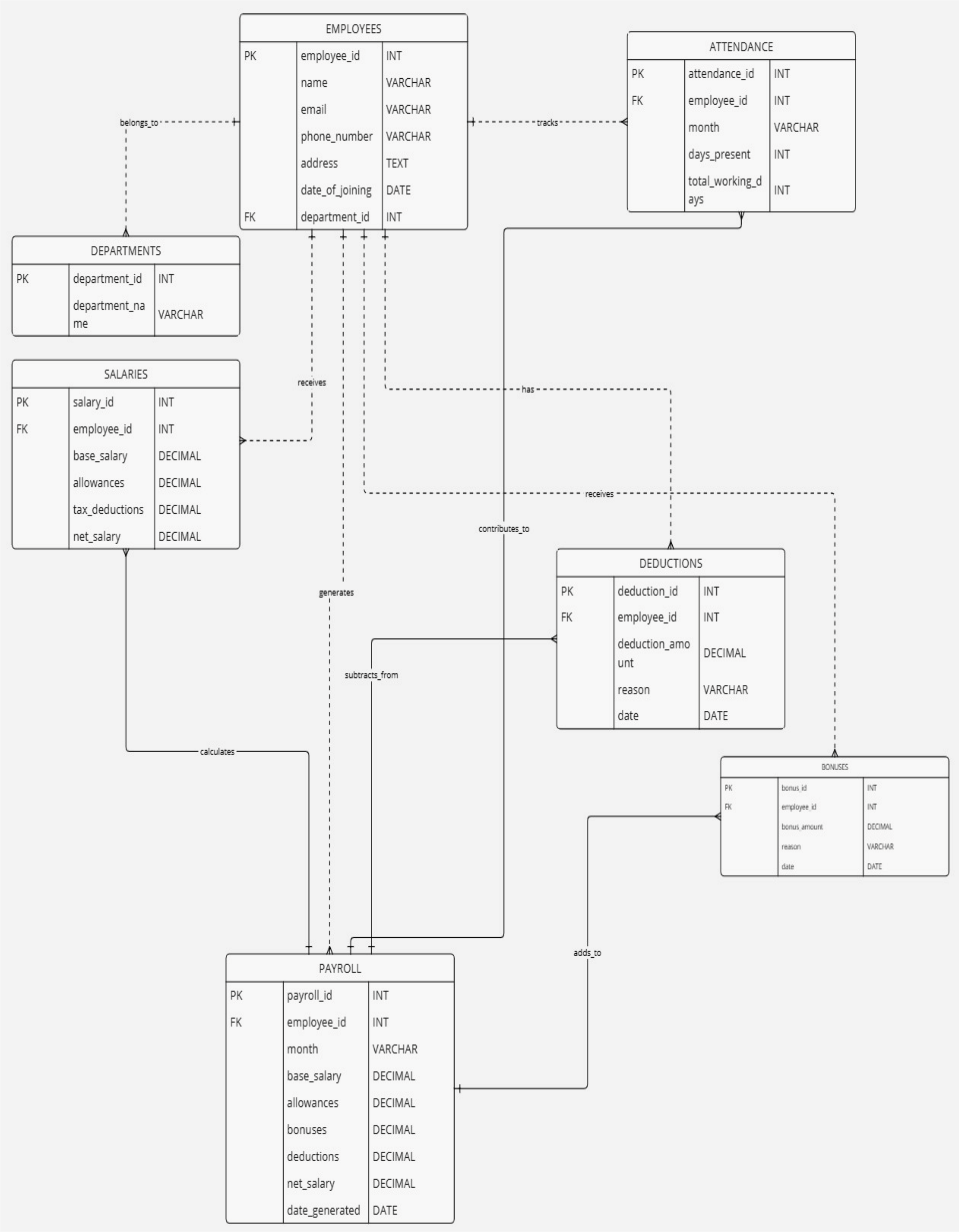
Core Components:

1. **Employees Table**: Stores personal and professional details of each employee, including employee ID, name, contact details, department, and date of joining.
2. **Salaries Table**: Stores salary-related information such as base salary, allowances, tax deductions, and calculates the net salary using the formula: `net_salary = base_salary + allowances - tax_deductions`.
3. **Attendance Table**: Tracks daily attendance data, including days present, days absent, and total working days for each employee. This data is used to calculate the salary for the given month.
4. **Deductions Table**: Manages various deductions applied to an employee's salary for reasons such as taxes, loan repayments, or other company policies.
5. **Bonuses Table**: Stores details of any bonuses awarded to employees, including the amount and reason for the bonus.
6. **Payroll Table**: Consolidates all payroll data for employees, including base salary, allowances, deductions, bonuses, and the computed net salary for each payroll cycle. The payroll records are generated monthly, linked to the employee ID, and identified by a unique auto-incremented `payroll_id`.

Data Structures Used:

- **Relational Tables**: The system employs relational tables (Employees, Salaries, Attendance, Deductions, Bonuses, Payroll) to store all relevant data.
- **Foreign Keys**: To maintain data integrity, foreign key relationships are used between tables (e.g., the `employee_id` in the **Salaries**, **Attendance**, **Deductions**, **Bonuses**, and **Payroll** tables links to the **Employees** table).
- **Auto-Increment**: The `payroll_id` in the **payroll** table is set to auto-increment to ensure unique identification of payroll records for each month.

Entity Relationship Diagram



	EMPLOYEES	
PK	employee_id	INT
	name	VARCHAR
	email	VARCHAR
	phone_number	VARCHAR
	address	TEXT
	date_of_joining	DATE
FK	department_id	INT

	ATTENDANCE	
PK	attendance_id	INT
FK	employee_id	INT
	month	VARCHAR
	days_present	INT
	total_working_days	INT

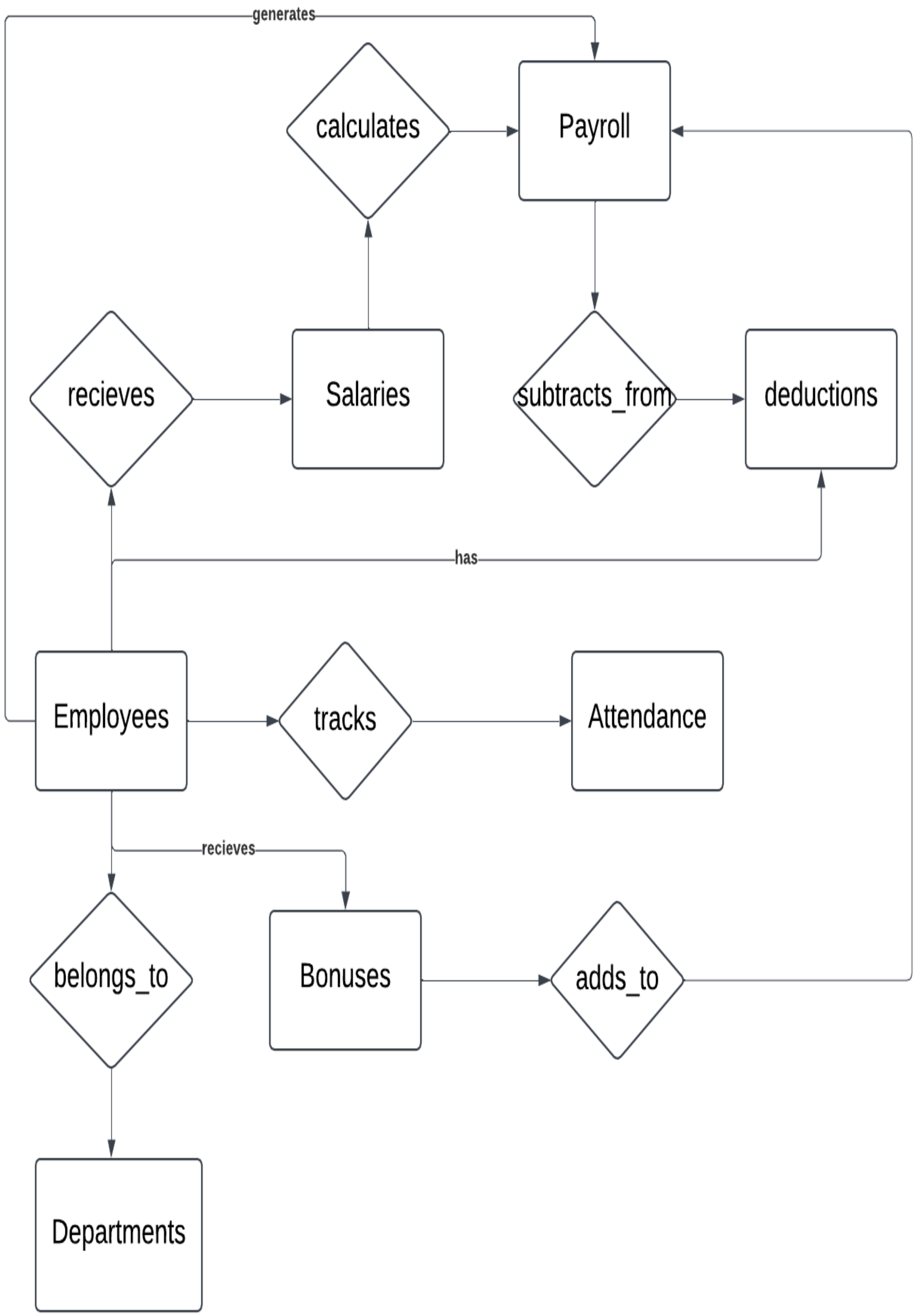
	DEPARTMENTS	
PK	department_id	INT
	department_name	VARCHAR

	SALARIES	
PK	salary_id	INT
FK	employee_id	INT
	base_salary	DECIMAL
	allowances	DECIMAL
	tax_deductions	DECIMAL
	net_salary	DECIMAL

	DEDUCTIONS	
PK	deduction_id	INT
FK	employee_id	INT
	deduction_amount	DECIMAL
	reason	VARCHAR
	date	DATE

	BONUSES	
PK	bonus_id	INT
FK	employee_id	INT
	bonus_amount	DECIMAL
	reason	VARCHAR
	date	DATE

	PAYROLL	
PK	payroll_id	INT
FK	employee_id	INT
	month	VARCHAR
	base_salary	DECIMAL
	allowances	DECIMAL
	bonuses	DECIMAL
	deductions	DECIMAL
	net_salary	DECIMAL
	date_generated	DATE


generates
calculates
Payroll
recieves
Salaries
subtracts_from
deductions
has
Employees
tracks
Attendance
recieves
belongs_to
Bonuses
adds_to
Departments

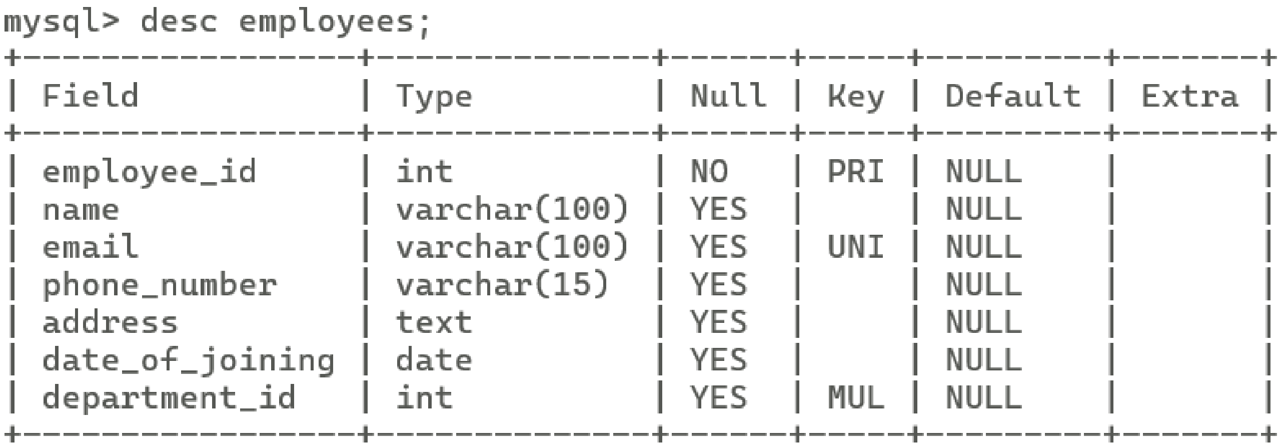
Workflow Overview:

1. **Employee Data Entry**: Employees are added to the system through the **Employees** table, where their personal and professional details (name, contact, department, etc.) are stored.
2. **Salary and Deductions Setup**: Salary components, allowances, and deductions are entered into the **Salaries** and **Deductions** tables. Bonuses are also tracked in the **Bonuses** table.
3. **Attendance Tracking**: Daily attendance records are entered into the **Attendance** table, which tracks employee presence, absences, and total working days for a specific month.
4. **Payroll Calculation**: At the end of each month, a **stored procedure** triggers the payroll generation process. The procedure calculates the **net salary** based on base salary, allowances, bonuses, and deductions. SQL triggers automatically handle calculations like net salary, reducing manual intervention.
5. **Payroll Generation**: The system then inserts the generated payroll data into the **Payroll** table, including the base salary, allowances, deductions, bonuses, and net salary, along with the payroll generation date.
6. **Payroll Reporting**: Users can retrieve monthly payroll reports for all employees, detailing their salaries, deductions, bonuses, and net pay, through **SELECT** queries.
7. **Data Integrity and Accuracy**: SQL constraints such as **foreign keys** ensure that records are valid and consistent across the system, ensuring the integrity of employee data and payroll records.

Implementation + Output

Create employee table

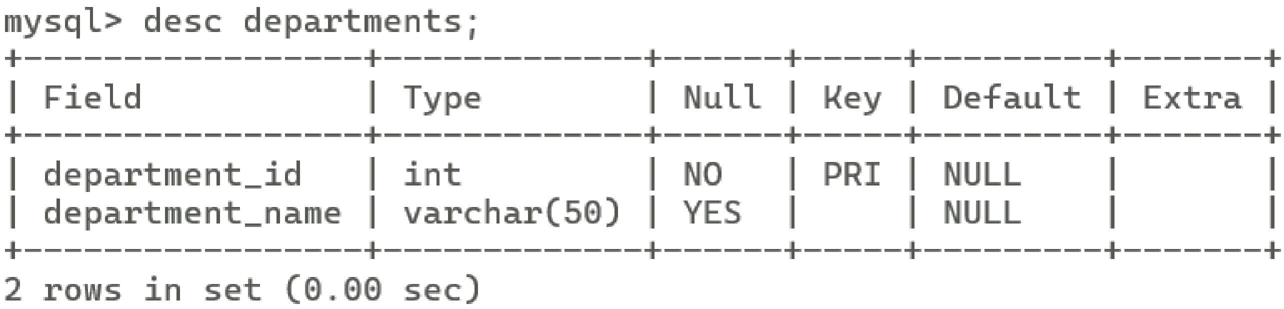
CREATE TABLE Employees (employee_id INT PRIMARY KEY, name VARCHAR(100), email VARCHAR(100) UNIQUE, phone_number VARCHAR(15), address TEXT, date_of_joining DATE, department_id INT, FOREIGN KEY (department_id) REFERENCES Departments(department_id));



```
mysql> desc employees;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| employee_id     | int          | NO   | PRI | NULL    |       |
| name            | varchar(100) | YES  |     | NULL    |       |
| email           | varchar(100) | YES  | UNI | NULL    |       |
| phone_number    | varchar(15)  | YES  |     | NULL    |       |
| address         | text         | YES  |     | NULL    |       |
| date_of_joining | date         | YES  |     | NULL    |       |
| department_id   | int          | YES  | MUL | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
```

Create Department Table

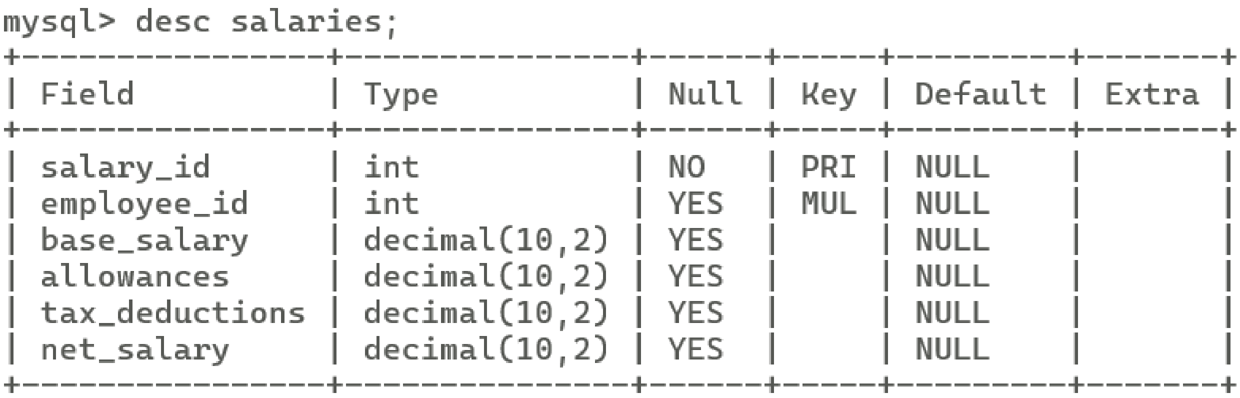
```
CREATE TABLE Departments (
        department_id INT PRIMARY KEY,
        department_name VARCHAR(50));
```



```
mysql> desc departments;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| department_id   | int         | NO   | PRI | NULL    |       |
| department_name | varchar(50) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Create Salaries Table

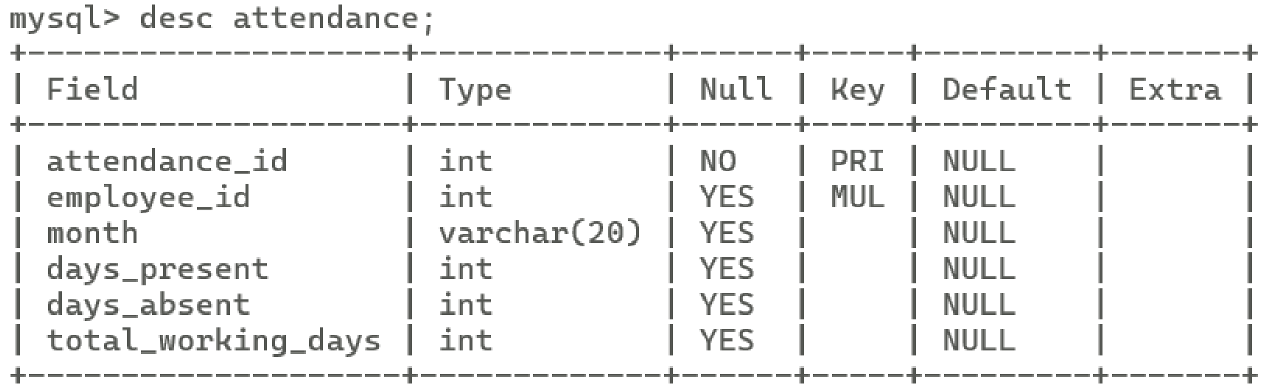
```
CREATE TABLE Salaries ( salary_id INT PRIMARY KEY, employee_id INT, base_salary DECIMAL(10, 2), allowances DECIMAL(10, 2), tax_deductions DECIMAL(10, 2), net_salary AS (base_salary + allowances - tax_deductions) STORED, FOREIGN KEY (employee_id) REFERENCES Employees(employee_id) );
```



```
mysql> desc salaries;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| salary_id      | int           | NO   | PRI | NULL    |       |
| employee_id    | int           | YES  | MUL | NULL    |       |
| base_salary    | decimal(10,2) | YES  |     | NULL    |       |
| allowances     | decimal(10,2) | YES  |     | NULL    |       |
| tax_deductions | decimal(10,2) | YES  |     | NULL    |       |
| net_salary     | decimal(10,2) | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
```

Create Attendance Table

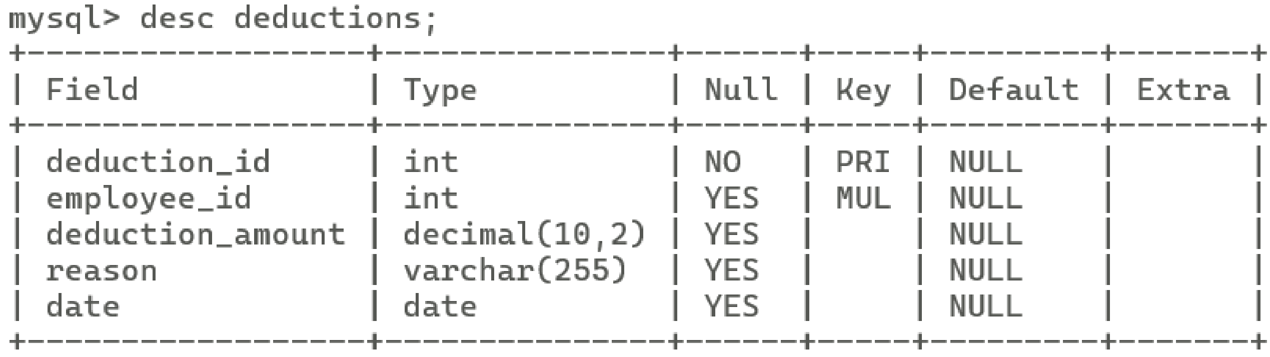
CREATE TABLE Attendance (attendance_id INT PRIMARY KEY, employee_id INT, month VARCHAR(20), days_present INT, days_absent INT, total_working_days INT, FOREIGN KEY (employee_id) REFERENCES Employees(employee_id));



```
mysql> desc attendance;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| attendance_id      | int         | NO   | PRI | NULL    |       |
| employee_id        | int         | YES  | MUL | NULL    |       |
| month              | varchar(20) | YES  |     | NULL    |       |
| days_present       | int         | YES  |     | NULL    |       |
| days_absent        | int         | YES  |     | NULL    |       |
| total_working_days | int         | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
```

Create Deductions Table

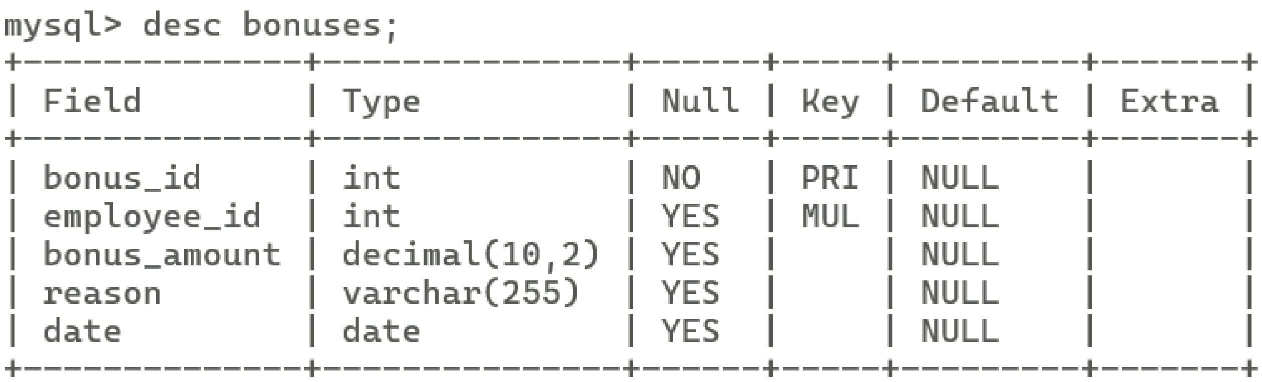
CREATE TABLE Deductions (deduction_id INT PRIMARY KEY, employee_id INT, deduction_amount DECIMAL(10, 2), reason VARCHAR(255), date DATE, FOREIGN KEY (employee_id) REFERENCES Employees(employee_id));



```
mysql> desc deductions;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| deduction_id     | int           | NO   | PRI | NULL    |       |
| employee_id      | int           | YES  | MUL | NULL    |       |
| deduction_amount | decimal(10,2) | YES  |     | NULL    |       |
| reason           | varchar(255)  | YES  |     | NULL    |       |
| date             | date          | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
```

Create Bonuses Table

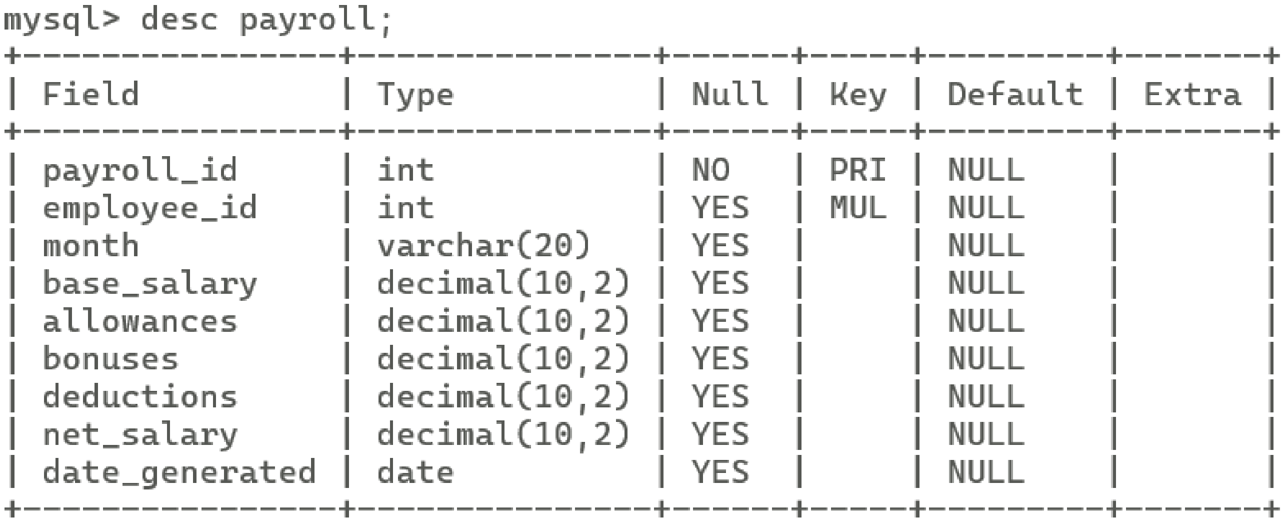
CREATE TABLE Bonuses (bonus_id INT PRIMARY KEY, employee_id INT, bonus_amount DECIMAL(10, 2), reason VARCHAR(255), date DATE, FOREIGN KEY (employee_id) REFERENCES Employees(employee_id));



```
mysql> desc bonuses;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| bonus_id     | int           | NO   | PRI | NULL    |       |
| employee_id  | int           | YES  | MUL | NULL    |       |
| bonus_amount | decimal(10,2) | YES  |     | NULL    |       |
| reason       | varchar(255)  | YES  |     | NULL    |       |
| date         | date          | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
```

Create Payroll Table

CREATE TABLE Payroll (payroll_id INT PRIMARY KEY, employee_id INT, month VARCHAR(20), base_salary DECIMAL(10, 2), allowances DECIMAL(10, 2), bonuses DECIMAL(10, 2), deductions DECIMAL(10, 2), net_salary DECIMAL(10, 2), date_generated DATE, FOREIGN KEY (employee_id) REFERENCES Employees(employee_id));

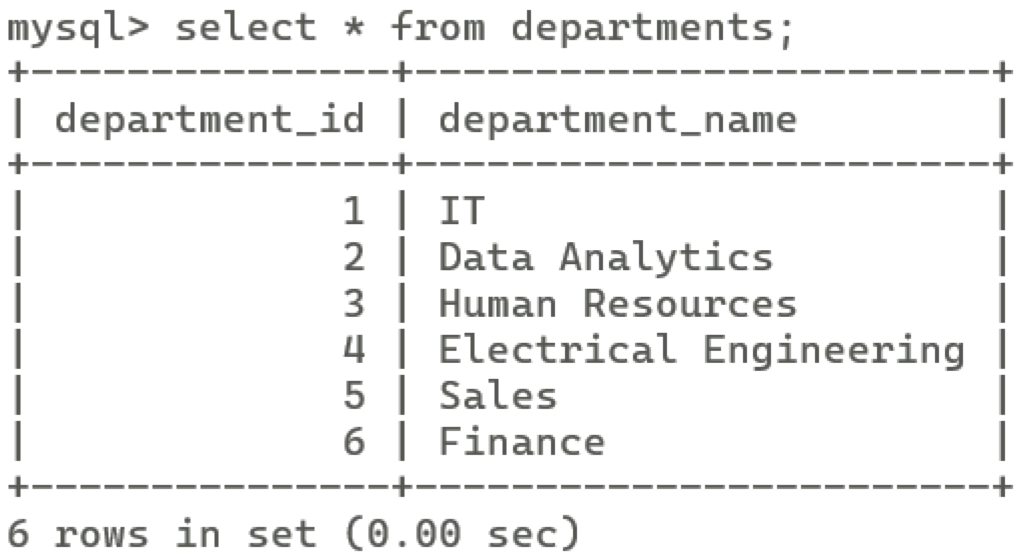


```
mysql> desc payroll;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| payroll_id     | int           | NO   | PRI | NULL    |       |
| employee_id    | int           | YES  | MUL | NULL    |       |
| month          | varchar(20)   | YES  |     | NULL    |       |
| base_salary    | decimal(10,2) | YES  |     | NULL    |       |
| allowances     | decimal(10,2) | YES  |     | NULL    |       |
| bonuses        | decimal(10,2) | YES  |     | NULL    |       |
| deductions     | decimal(10,2) | YES  |     | NULL    |       |
| net_salary     | decimal(10,2) | YES  |     | NULL    |       |
| date_generated | date          | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
```

Queries + Output

Insert into Department

INSERT INTO Departments (department_id, department_name) VALUES (1, 'IT'), (2, 'Data Analytics'), (3, 'Human Resources'), (4, 'Electrical Engineering'), (5, 'Sales'), (6, 'Finance');



```
mysql> select * from departments;
+---------------+------------------------+
| department_id | department_name        |
+---------------+------------------------+
|             1 | IT                     |
|             2 | Data Analytics         |
|             3 | Human Resources        |
|             4 | Electrical Engineering |
|             5 | Sales                  |
|             6 | Finance                |
+---------------+------------------------+
6 rows in set (0.00 sec)
```

INSERT INTO Employees

INSERT INTO EMPLOYEES(employee_id, name, email, phone_number, address, date_of_joining, department_id)

VALUES

(101, 'Amit Sharma', 'amit.sharma@company.com', '+91-9876543210', 'No. 21, MG Road, Bengaluru', '2020-06-15', 1),

(102, 'Priya Mehta', 'priya.mehta@company.com', '+91-9123456789', 'Flat 302, Andheri West, Mumbai', '2019-09-12', 2),

(103, 'Rajesh Kumar', 'rajesh.kumar@company.com', '+91-9988776655', 'Plot 18, Connaught Place, New Delhi', '2018-01-10', 3),

(104, 'Sneha Iyer', 'sneha.iyer@company.com', '+91-8877665544', 'House No. 45, Anna Nagar, Chennai', '2021-11-20', 4),

(105, 'Anil Gupta', 'anil.gupta@company.com', '+91-7766554433', 'No. 5, Koregaon Park, Pune', '2022-03-05', 5),

(106, 'Divya Nair', 'divya.nair@company.com', '+91-6655443322', 'Block B, Jubilee Hills, Hyderabad', '2020-12-10', 6);

Insert into Salaries (Auto Updated using Triggers)

INSERT INTO Salaries (salary_id, employee_id, base_salary, allowances, tax_deductions)

VALUES

(201, 101, 50000.00, 5000.00, 8000.00), -- Amit Sharma

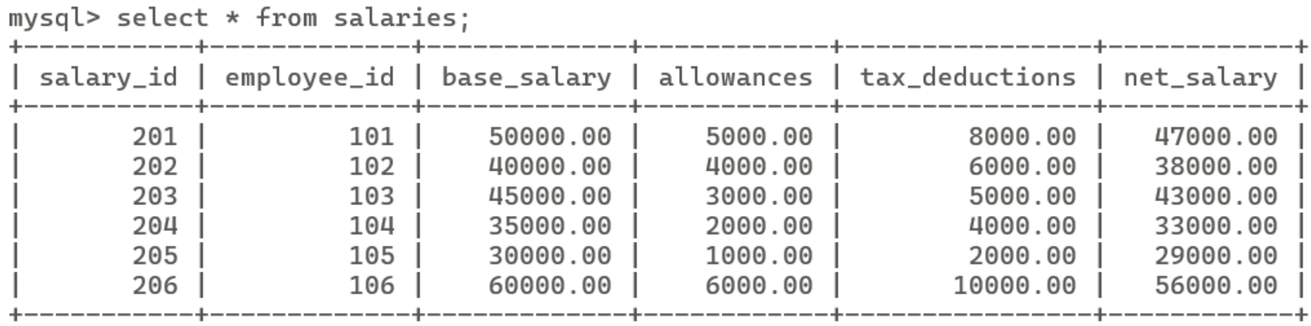
(202, 102, 40000.00, 4000.00, 6000.00), -- Priya Mehta

(203, 103, 45000.00, 3000.00, 5000.00), -- Rajesh Kumar

(204, 104, 35000.00, 2000.00, 4000.00), -- Sneha Iyer

(205, 105, 30000.00, 1000.00, 2000.00), -- Anil Gupta

(206, 106, 60000.00, 6000.00, 10000.00); -- Divya Nair



```
mysql> select * from salaries;
+-----------+-------------+-------------+------------+----------------+------------+
| salary_id | employee_id | base_salary | allowances | tax_deductions | net_salary |
+-----------+-------------+-------------+------------+----------------+------------+
|       201 |         101 |    50000.00 |    5000.00 |        8000.00 |   47000.00 |
|       202 |         102 |    40000.00 |    4000.00 |        6000.00 |   38000.00 |
|       203 |         103 |    45000.00 |    3000.00 |        5000.00 |   43000.00 |
|       204 |         104 |    35000.00 |    2000.00 |        4000.00 |   33000.00 |
|       205 |         105 |    30000.00 |    1000.00 |        2000.00 |   29000.00 |
|       206 |         106 |    60000.00 |    6000.00 |       10000.00 |   56000.00 |
+-----------+-------------+-------------+------------+----------------+------------+
```

Insert into Attendance(Auto Calculate Remaining Days using Trigger)

INSERT INTO Attendance (attendance_id, employee_id, month, days_present, total_working_days)

VALUES

(301, 101, 'January 2025', 22, 25), -- Amit Sharma

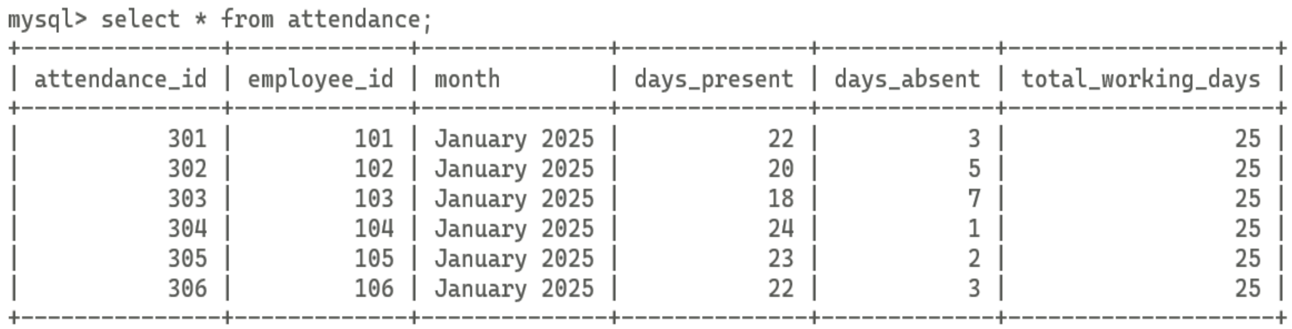
(302, 102, 'January 2025', 20, 25), -- Priya Mehta

(303, 103, 'January 2025', 18, 25), -- Rajesh Kumar

(304, 104, 'January 2025', 24, 25), -- Sneha Iyer

(305, 105, 'January 2025', 23, 25), -- Anil Gupta

(306, 106, 'January 2025', 22, 25); -- Divya Nair



```
mysql> select * from attendance;
+---------------+-------------+--------------+--------------+-------------+--------------------+
| attendance_id | employee_id | month        | days_present | days_absent | total_working_days |
+---------------+-------------+--------------+--------------+-------------+--------------------+
|           301 |         101 | January 2025 |           22 |           3 |                 25 |
|           302 |         102 | January 2025 |           20 |           5 |                 25 |
|           303 |         103 | January 2025 |           18 |           7 |                 25 |
|           304 |         104 | January 2025 |           24 |           1 |                 25 |
|           305 |         105 | January 2025 |           23 |           2 |                 25 |
|           306 |         106 | January 2025 |           22 |           3 |                 25 |
+---------------+-------------+--------------+--------------+-------------+--------------------+
```

Insert into Deductions

INSERT INTO Deductions (deduction_id, employee_id, deduction_amount, reason, date)

VALUES

(401, 101, 2000.00, 'Late submission of reports', '2025-01-15'),

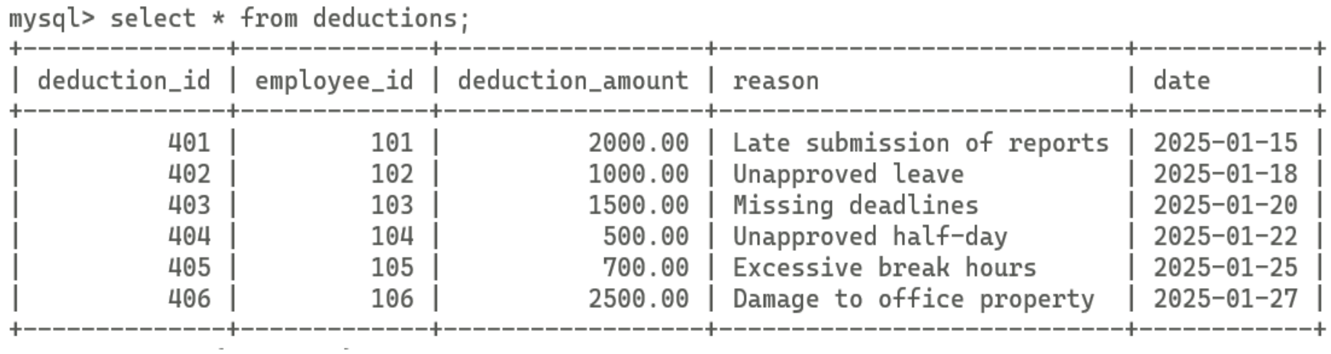
(402, 102, 1000.00, 'Unapproved leave', '2025-01-18'),

(403, 103, 1500.00, 'Missing deadlines', '2025-01-20'),

(404, 104, 500.00, 'Unapproved half-day', '2025-01-22'),

(405, 105, 700.00, 'Excessive break hours', '2025-01-25'),

(406, 106, 2500.00, 'Damage to office property', '2025-01-27');



```
mysql> select * from deductions;
+--------------+-------------+------------------+-----------------------------+------------+
| deduction_id | employee_id | deduction_amount | reason                      | date       |
+--------------+-------------+------------------+-----------------------------+------------+
|          401 |         101 |          2000.00 | Late submission of reports  | 2025-01-15 |
|          402 |         102 |          1000.00 | Unapproved leave            | 2025-01-18 |
|          403 |         103 |          1500.00 | Missing deadlines           | 2025-01-20 |
|          404 |         104 |           500.00 | Unapproved half-day         | 2025-01-22 |
|          405 |         105 |           700.00 | Excessive break hours       | 2025-01-25 |
|          406 |         106 |          2500.00 | Damage to office property   | 2025-01-27 |
+--------------+-------------+------------------+-----------------------------+------------+
```

Insert into Bonuses

INSERT INTO Bonuses (bonus_id, employee_id, bonus_amount, reason, date)

VALUES

(501, 101, 5000.00, 'Project completion bonus', '2025-01-31'),

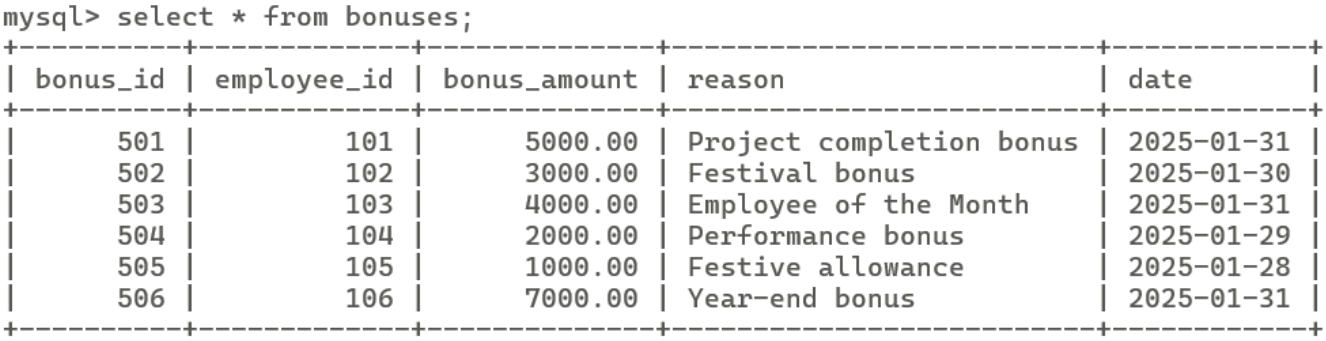
(502, 102, 3000.00, 'Festival bonus', '2025-01-30'),

(503, 103, 4000.00, 'Employee of the Month', '2025-01-31'),

(504, 104, 2000.00, 'Performance bonus', '2025-01-29'),

(505, 105, 1000.00, 'Festive allowance', '2025-01-28'),

(506, 106, 7000.00, 'Year-end bonus', '2025-01-31');



```
mysql> select * from bonuses;
+----------+-------------+--------------+--------------------------+------------+
| bonus_id | employee_id | bonus_amount | reason                   | date       |
+----------+-------------+--------------+--------------------------+------------+
|      501 |         101 |      5000.00 | Project completion bonus | 2025-01-31 |
|      502 |         102 |      3000.00 | Festival bonus           | 2025-01-30 |
|      503 |         103 |      4000.00 | Employee of the Month    | 2025-01-31 |
|      504 |         104 |      2000.00 | Performance bonus        | 2025-01-29 |
|      505 |         105 |      1000.00 | Festive allowance        | 2025-01-28 |
|      506 |         106 |      7000.00 | Year-end bonus           | 2025-01-31 |
+----------+-------------+--------------+--------------------------+------------+
```

Insert into Payroll(Auto Calculate Net Salary using Triggers)

INSERT INTO Payroll (payroll_id, employee_id, month, base_salary, allowances, bonuses, deductions, date_generated)

VALUES

(601, 101, 'January 2025', 50000.00, 5000.00, 5000.00, 2000.00, '2025-01-31'),

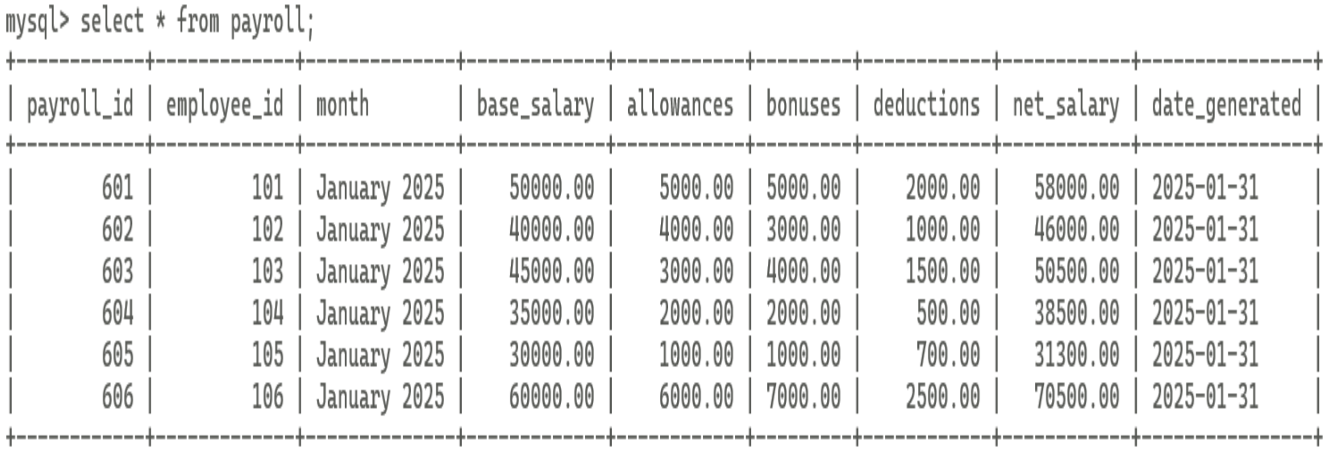
(602, 102, 'January 2025', 40000.00, 4000.00, 3000.00, 1000.00, '2025-01-31'),

(603, 103, 'January 2025', 45000.00, 3000.00, 4000.00, 1500.00, '2025-01-31'),

(604, 104, 'January 2025', 35000.00, 2000.00, 2000.00, 500.00, '2025-01-31'),

(605, 105, 'January 2025', 30000.00, 1000.00, 1000.00, 700.00, '2025-01-31'),

(606, 106, 'January 2025', 60000.00, 6000.00, 7000.00, 2500.00, '2025-01-31');

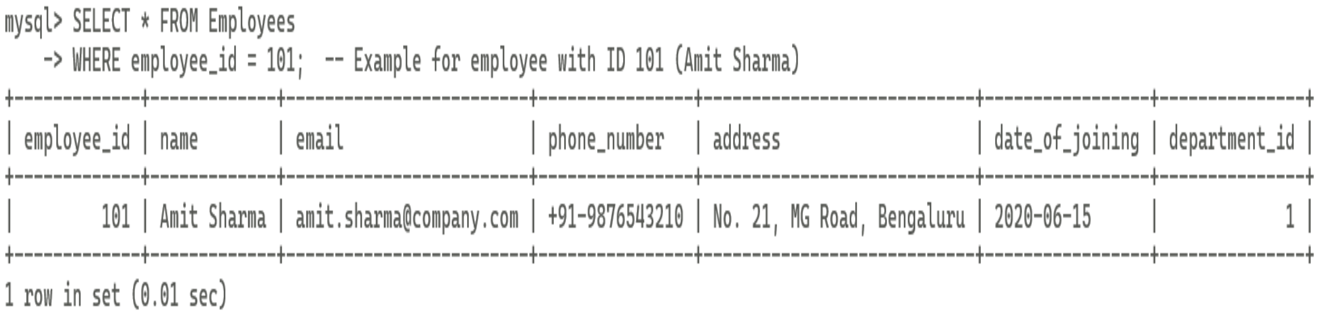


```
mysql> select * from payroll;
+------------+-------------+--------------+-------------+------------+---------+------------+------------+----------------+
| payroll_id | employee_id | month        | base_salary | allowances | bonuses | deductions | net_salary | date_generated |
+------------+-------------+--------------+-------------+------------+---------+------------+------------+----------------+
|        601 |         101 | January 2025 |    50000.00 |    5000.00 | 5000.00 |    2000.00 |   58000.00 | 2025-01-31     |
|        602 |         102 | January 2025 |    40000.00 |    4000.00 | 3000.00 |    1000.00 |   46000.00 | 2025-01-31     |
|        603 |         103 | January 2025 |    45000.00 |    3000.00 | 4000.00 |    1500.00 |   50500.00 | 2025-01-31     |
|        604 |         104 | January 2025 |    35000.00 |    2000.00 | 2000.00 |     500.00 |   38500.00 | 2025-01-31     |
|        605 |         105 | January 2025 |    30000.00 |    1000.00 | 1000.00 |     700.00 |   31300.00 | 2025-01-31     |
|        606 |         106 | January 2025 |    60000.00 |    6000.00 | 7000.00 |    2500.00 |   70500.00 | 2025-01-31     |
+------------+-------------+--------------+-------------+------------+---------+------------+------------+----------------+
```

Filter Queries

Employee Details for Employee ID 101 (Amit Sharma)

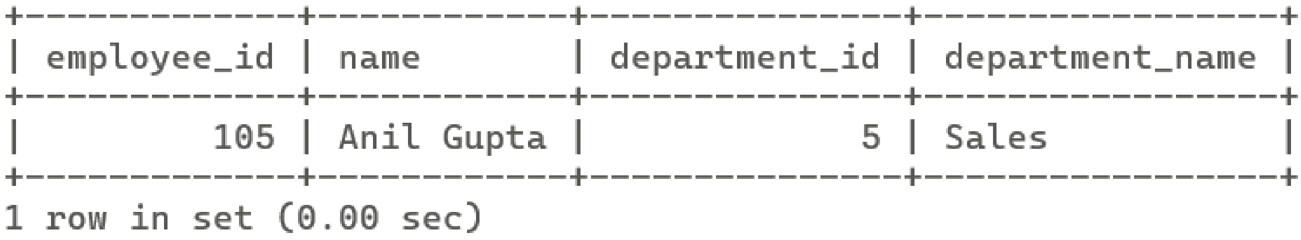
SELECT * FROM Employees WHERE employee_id = 101;



```
mysql> SELECT * FROM Employees
    -> WHERE employee_id = 101;  -- Example for employee with ID 101 (Amit Sharma)
+-------------+------------+-------------------------+---------------+----------------------------+-----------------+---------------+
| employee_id | name       | email                   | phone_number  | address                    | date_of_joining | department_id |
+-------------+------------+-------------------------+---------------+----------------------------+-----------------+---------------+
|         101 | Amit Sharma | amit.sharma@company.com | +91-9876543210 | No. 21, MG Road, Bengaluru | 2020-06-15      |             1 |
+-------------+------------+-------------------------+---------------+----------------------------+-----------------+---------------+
1 row in set (0.01 sec)
```

Employees in Sales Department

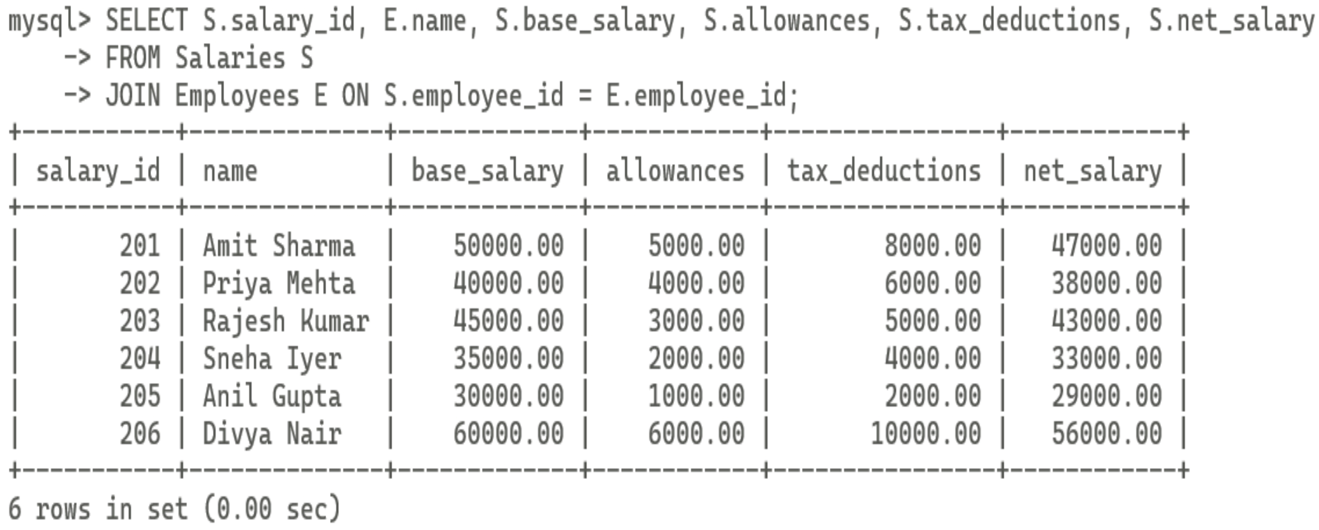
SELECT E.employee_id, E.name, E.department_id, D.department_name FROM Employees E JOIN Departments D ON E.department_id = D.department_id WHERE D.department_name = ‘Sales’;



```
+-------------+------------+---------------+-----------------+
| employee_id | name       | department_id | department_name |
+-------------+------------+---------------+-----------------+
|         105 | Anil Gupta |             5 | Sales           |
+-------------+------------+---------------+-----------------+
1 row in set (0.00 sec)
```

Employee Salary Details

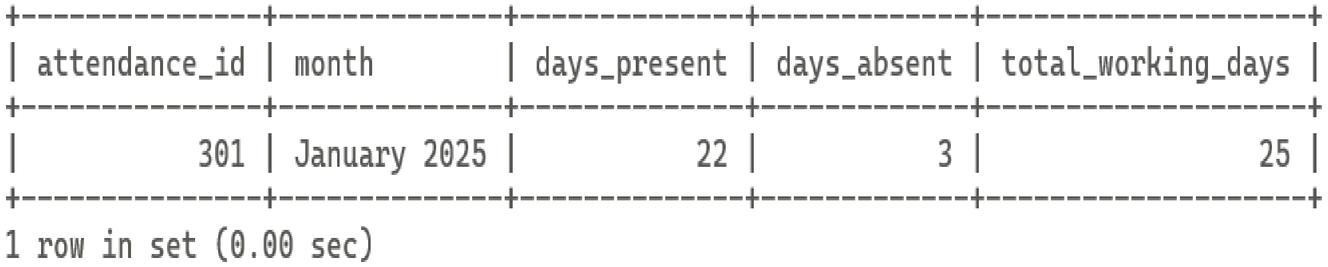
SELECT S.salary_id, E.name, S.base_salary, S.allowances, S.tax_deductions, S.net_salary FROM Salaries S JOIN Employees E ON S.employee_id = E.employee_id;



```
mysql> SELECT S.salary_id, E.name, S.base_salary, S.allowances, S.tax_deductions, S.net_salary
    -> FROM Salaries S
    -> JOIN Employees E ON S.employee_id = E.employee_id;
+-----------+--------------+-------------+------------+----------------+------------+
| salary_id | name         | base_salary | allowances | tax_deductions | net_salary |
+-----------+--------------+-------------+------------+----------------+------------+
|       201 | Amit Sharma  |    50000.00 |    5000.00 |        8000.00 |   47000.00 |
|       202 | Priya Mehta  |    40000.00 |    4000.00 |        6000.00 |   38000.00 |
|       203 | Rajesh Kumar |    45000.00 |    3000.00 |        5000.00 |   43000.00 |
|       204 | Sneha Iyer   |    35000.00 |    2000.00 |        4000.00 |   33000.00 |
|       205 | Anil Gupta   |    30000.00 |    1000.00 |        2000.00 |   29000.00 |
|       206 | Divya Nair   |    60000.00 |    6000.00 |       10000.00 |   56000.00 |
+-----------+--------------+-------------+------------+----------------+------------+
6 rows in set (0.00 sec)
```

Attendance Details for Employee ID 101 (Amit Sharma) for January 2025

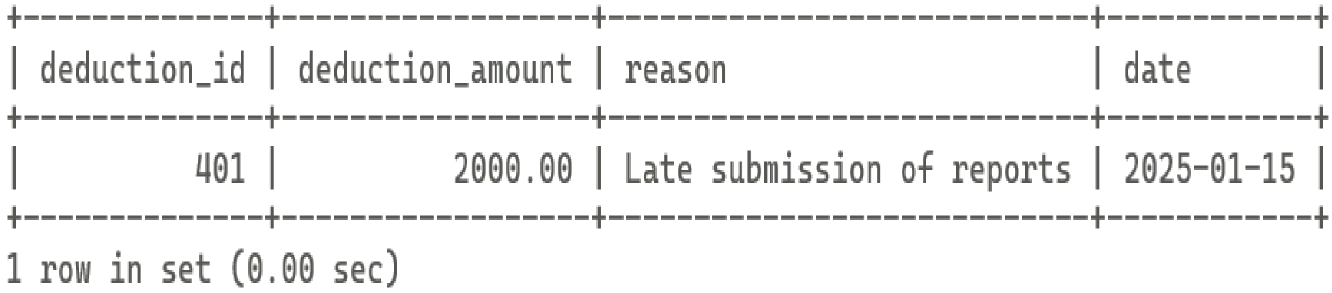
SELECT A.attendance_id, A.month, A.days_present, A.days_absent, A.total_working_days FROM Attendance A WHERE A.employee_id = 101 AND A.month = 'January 2025';



```
+---------------+--------------+--------------+-------------+--------------------+
| attendance_id | month        | days_present | days_absent | total_working_days |
+---------------+--------------+--------------+-------------+--------------------+
|           301 | January 2025 |           22 |           3 |                 25 |
+---------------+--------------+--------------+-------------+--------------------+
1 row in set (0.00 sec)
```

Deductions for Employee ID 101 (Amit Sharma)

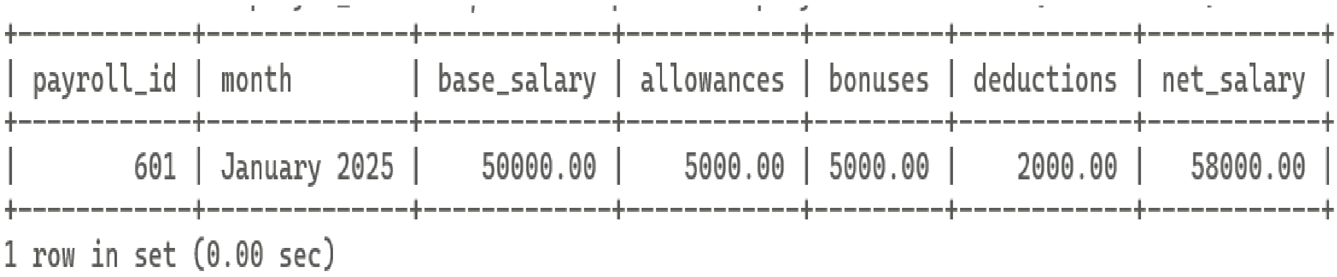
SELECT B.bonus_id, B.bonus_amount, B.reason, B.date FROM Bonuses B WHERE B.employee_id = 101;



```
+--------------+------------------+------------------------------+------------+
| deduction_id | deduction_amount | reason                       | date       |
+--------------+------------------+------------------------------+------------+
|          401 |          2000.00 | Late submission of reports   | 2025-01-15 |
+--------------+------------------+------------------------------+------------+
1 row in set (0.00 sec)
```

Payroll Details for Employee ID 101 (Amit Sharma) for January 2025

SELECT P.payroll_id, P.month, P.base_salary, P.allowances, P.bonuses, P.deductions, P.net_salary FROM Payroll P WHERE P.employee_id = 101;



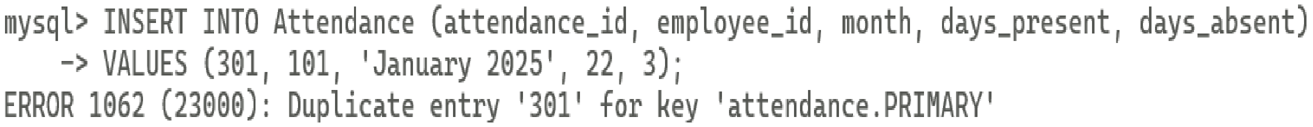
```
+------------+--------------+-------------+------------+---------+------------+------------+
| payroll_id | month        | base_salary | allowances | bonuses | deductions | net_salary |
+------------+--------------+-------------+------------+---------+------------+------------+
|        601 | January 2025 |    50000.00 |    5000.00 | 5000.00 |    2000.00 |   58000.00 |
+------------+--------------+-------------+------------+---------+------------+------------+
1 row in set (0.00 sec)
```

Trigger Queries

Trigger to Auto-Calculate Total Working Days in Attendance

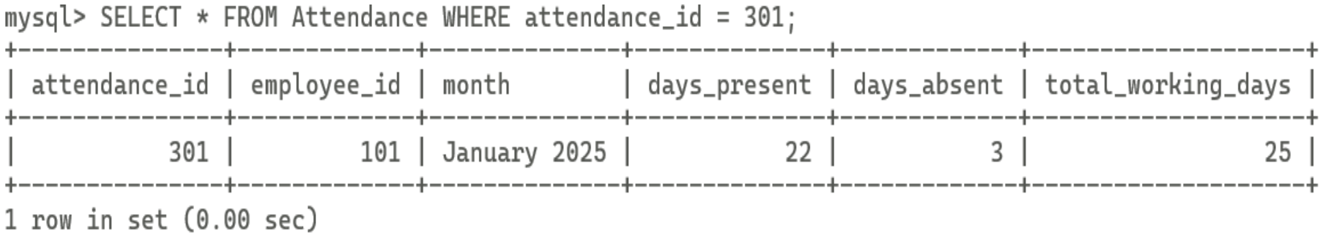
```
DELIMITER $$

CREATE TRIGGER calculate_total_working_days

BEFORE INSERT ON Attendance

FOR EACH ROW

BEGIN

    SET NEW.total_working_days = NEW.days_present + NEW.days_absent;

END$$

DELIMITER ;
```

Firing the Trigger



```
mysql> INSERT INTO Attendance (attendance_id, employee_id, month, days_present, days_absent)
    -> VALUES (301, 101, 'January 2025', 22, 3);
ERROR 1062 (23000): Duplicate entry '301' for key 'attendance.PRIMARY'
```

Output

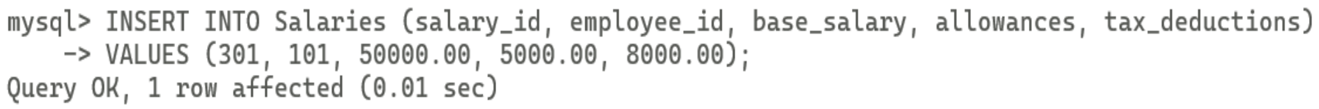


```
mysql> SELECT * FROM Attendance WHERE attendance_id = 301;
+---------------+-------------+--------------+--------------+-------------+--------------------+
| attendance_id | employee_id | month        | days_present | days_absent | total_working_days |
+---------------+-------------+--------------+--------------+-------------+--------------------+
|           301 |         101 | January 2025 |           22 |           3 |                 25 |
+---------------+-------------+--------------+--------------+-------------+--------------------+
1 row in set (0.00 sec)
```

Trigger to Update net_salary in Salaries

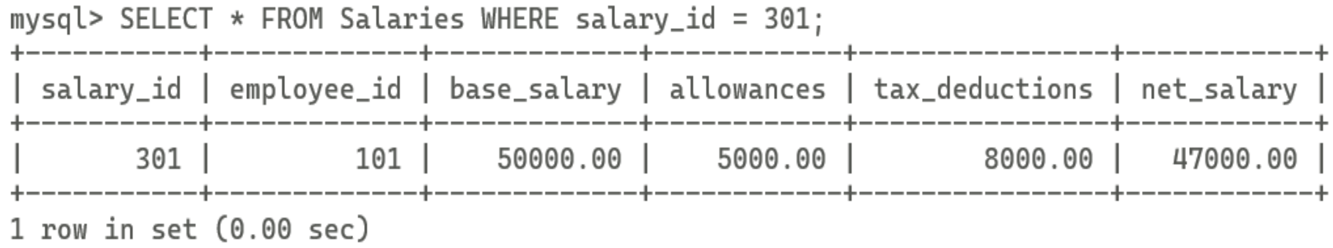
```
DELIMITER $$

CREATE TRIGGER calculate_net_salary

BEFORE INSERT ON Salaries

FOR EACH ROW

BEGIN

    SET NEW.net_salary = NEW.base_salary + NEW.allowances - NEW.tax_deductions;

END$$

DELIMITER ;
```

Firing the Trigger



```
mysql> INSERT INTO Salaries (salary_id, employee_id, base_salary, allowances, tax_deductions)
    -> VALUES (301, 101, 50000.00, 5000.00, 8000.00);
Query OK, 1 row affected (0.01 sec)
```

Output



```
mysql> SELECT * FROM Salaries WHERE salary_id = 301;
+-----------+-------------+-------------+------------+----------------+------------+
| salary_id | employee_id | base_salary | allowances | tax_deductions | net_salary |
+-----------+-------------+-------------+------------+----------------+------------+
|       301 |         101 |    50000.00 |    5000.00 |        8000.00 |   47000.00 |
+-----------+-------------+-------------+------------+----------------+------------+
1 row in set (0.00 sec)
```

Procedure Queries

Procedure to Generate Payroll for a Specific Month

DELIMITER $$

CREATE PROCEDURE GeneratePayrollForMonth(IN payroll_month VARCHAR(20), IN generation_date DATE) BEGIN DECLARE done INT DEFAULT 0; DECLARE emp_id INT; DECLARE emp_base_salary DECIMAL(10, 2); DECLARE emp_allowances DECIMAL(10, 2); DECLARE emp_bonuses DECIMAL(10, 2); DECLARE emp_deductions DECIMAL(10, 2); DECLARE emp_net_salary DECIMAL(10, 2); DECLARE cur CURSOR FOR SELECT employee_id, base_salary, allowances, bonuses, deductions FROM Salaries;

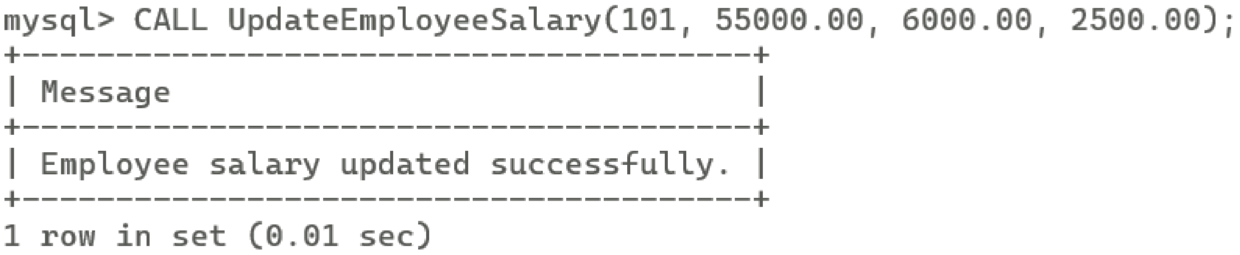
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1; OPEN cur; read_loop: LOOP FETCH cur INTO emp_id, emp_base_salary, emp_allowances, emp_bonuses, emp_deductions;

IF done THEN LEAVE read_loop;

END IF;

SET emp_net_salary = emp_base_salary + emp_allowances + emp_bonuses - emp_deductions; INSERT INTO Payroll (employee_id, month, base_salary, allowances, bonuses, deductions, net_salary, date_generated)

VALUES (emp_id, payroll_month, emp_base_salary, emp_allowances, emp_bonuses, emp_deductions, emp_net_salary, generation_date); END LOOP; CLOSE cur; END$$ DELIMITER ;



```
mysql> CALL UpdateEmployeeSalary(101, 55000.00, 6000.00, 2500.00);
+-------------------------------------+
| Message                             |
+-------------------------------------+
| Employee salary updated successfully. |
+-------------------------------------+
1 row in set (0.01 sec)
```

Explanation of the Code

The Payroll Management System developed using SQL and triggers is designed to efficiently manage various aspects of employee data, including salaries, attendance, deductions, bonuses, and payroll processing. It begins with an Employees table that stores crucial employee details such as their unique employee ID, name, email, phone number, address, date of joining, and department. This table serves as the core reference point for linking all employee-related data in the system. The Departments table organizes the employees into different departments, facilitating easy categorization and reporting. Each employee's salary details are stored in the Salaries table, which includes the base salary, allowances, tax deductions, and a computed net_salary, calculated automatically by a trigger when new salary records are inserted. This trigger ensures that the net_salary is always up to date, minimizing the chances of errors.

The Attendance table is used to log the number of days each employee was present and absent within a given month. It also includes a trigger to calculate the total_working_days automatically based on the provided days present and absent, ensuring that the data is consistent and accurate. The Deductions and Bonuses tables are responsible for storing any salary deductions and bonuses related to each employee, such as penalties or rewards for performance, respectively. These tables are integrated into the payroll calculations, ensuring that deductions and bonuses are accurately reflected in the final salary statement. The Payroll table serves as the final summary of all salary data, aggregating information from various tables like salaries, attendance, deductions, and bonuses. This table not only stores the monthly payroll but also provides the final net_salary after all necessary calculations, with triggers ensuring these values are calculated automatically.

The system is further enhanced by the use of SQL stored procedures, which help automate tasks such as payroll generation for each employee at the end of the month. Procedures like GeneratePayrollForMonth automate the process of aggregating data from multiple tables and generating the final payroll report. By using triggers to calculate the net salary, total working days, and other relevant fields, the system reduces manual intervention and ensures that calculations are accurate and consistent. Additionally, the use of SQL joins enables the seamless retrieval of employee information, such as linking employee records to their respective departments, salary, and attendance data. This system not only ensures efficiency but also eliminates the chances of errors in payroll processing, making it a reliable tool for managing employee payroll in any organization. With its automated features and structured database design, the Payroll Management System significantly enhances the efficiency and accuracy of HR departments, providing a comprehensive solution for managing employee records and payroll processing.

Conclusion

The Payroll Management System designed using DBMS and SQL provides an efficient and automated solution for managing employee payroll, attendance, salaries, deductions, bonuses, and related data. By utilizing SQL tables, triggers, and stored procedures, the system streamlines the entire payroll process, ensuring accurate calculations of salaries, allowances, deductions, and attendance. Triggers like calculate_total_working_days and calculate_net_salary are used to automatically compute values when new records are inserted, reducing the likelihood of errors and manual input. The system allows for smooth management and retrieval of employee data, ensuring timely and accurate payroll generation. This approach not only saves time but also increases operational efficiency, making it a reliable tool for HR departments in organizations.