



Mini project report on

Payroll Management System

Submitted in partial fulfilment of the requirements for the award of degree of

**Bachelor of Technology
in
Computer Science & Engineering**

UE21CS351 – DBMS Project

Submitted by:

**Krishna Sphurthi Yandamuri
Himank Bansal**

**PES2UG21CS241
PES2UG21CS201**

Under the guidance of

Dr. Mannar Mannan J

Professor

Designation

PES University

AUG - DEC 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Payroll Management System

is a bonafide work carried out by

Krishna Sphurthi Yandamuri

PES2UG21CS241

Himank Bansal

PES2UG21CS201

In partial fulfilment for the completion of fifth semester DBMS Project (UE21CSS351) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2023 – DEC. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature

Dr. Mannar Mannan J

Professor

DECLARATION

We hereby declare that the DBMS Project entitled **Payroll Management System** has been carried out by us under the guidance of **Dr. Mannar Mannan J, Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2023.

Krishna Sphurthi PES2UG21CS241 <Signature>
Yandamuri

Himank Bansal PES2UG21CS201 <Signature>

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Nivedita Kasturi, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE21CS351 - DBMS Project.

I take this opportunity to thank Dr. Sandesh B J, C, Professor, Chair Person, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this DBMS Project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

The Payroll Management System is designed to automate the existing manual system using computerized equipment and cutting-edge computer software, meeting client's needs so that their valuable data and information can be stored for a longer period with easy access and manipulation. The necessary software is readily available and simple to use. This software allows users to keep track of and see computerized records without having to make duplicate entries. The project explains how to handle user data for optimal efficiency and better customer service.

The payroll management system is a web-based program that can be used by any firm to manage the records of its employees. The Payroll Application was created with the goal of keeping track of numerous employees, their allowances, and deductions that must be given to the company's employees. There will be an entry (a unique ID) for every employee of any company. The number of days will be submitted based on the date of joining and the date on which the salary is produced. Basic compensation will be determined by the employee's position and department.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	INTRODUCTION	11
	PROBLEM DEFINITION	12
	ER MODEL	13
	ER TO RELATIONAL MAPPING	14
	DDL STATEMENTS	15
	DML STATEMENTS	20
	QUERIES (SIMPLE QUERY AND UPDATE AND DELETE OPERATION, CORRELATED QUERY AND NESTED QUERY)	23
	STORED PROCEDURE, FUNCTIONS AND TRIGGERS	30
	FRONT END DEVELOPMENT	40
	REFERENCES/BIBLIOGRAPHY	41
	APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS	42

1.INTRODUCTION

A Payroll Management System is a comprehensive software solution designed to streamline and automate the process of managing an organization's payroll. It encompasses all activities related to employee compensation, including salary calculation, tax deductions, allowances, and other financial disbursements. The primary objective of a payroll management system is to ensure accurate and timely payment to employees while complying with regulatory requirements.

Key Components of a Payroll Management System:

Employee Information Management:

The system maintains a centralized database containing essential employee information such as personal details, contact information, tax-related data, and employment history.

Salary Calculation:

It automates the computation of salaries based on various parameters, including basic pay, allowances, bonuses, and deductions. This ensures accurate and consistent payroll processing.

Tax Calculation and Compliance:

The system calculates income tax, social security contributions, and other deductions in compliance with local tax laws and regulations. This helps organizations remain compliant and avoid legal issues.

Attendance and Leave Management:

Integration with attendance and leave management modules enables the system to consider attendance data and leave records when calculating salaries, ensuring that employees are compensated accurately.

Direct Deposit and Payment Processing:

Payroll management systems facilitate direct deposit to employees' bank accounts, reducing the need for manual checks. They also support various payment methods, such as electronic funds transfer.

Reporting and Analytics:

The system generates comprehensive reports and analytics related to payroll expenses, tax liabilities, and other financial aspects. This information is crucial for financial planning and decision-making.

2.PROBLEM DEFINITION

Organizations often grapple with the complexities and challenges associated with manual payroll processing, necessitating the implementation of an efficient Payroll Management System. The following issues highlight the need for such a system:

Inefficiency in Manual Processing:

Problem: The manual calculation of salaries, allowances, and deductions is time-consuming and prone to errors. This inefficiency can lead to payroll discrepancies, resulting in employee dissatisfaction and potential legal complications.

Compliance Risks:

Problem: Staying compliant with constantly evolving tax laws, labor regulations, and reporting requirements is challenging. Manual processes increase the risk of non-compliance, exposing the organization to penalties and legal ramifications.

Data Inconsistency and Lack of Centralization:

Problem: Employee data spread across various spreadsheets and documents can lead to inconsistencies and difficulties in data retrieval. A centralized system is necessary to ensure accuracy and accessibility of employee information.

Limited Reporting and Analytics:

Problem: Manual payroll systems often lack robust reporting capabilities. The absence of real-time analytics makes it challenging for organizations to derive meaningful insights into payroll expenses and trends.

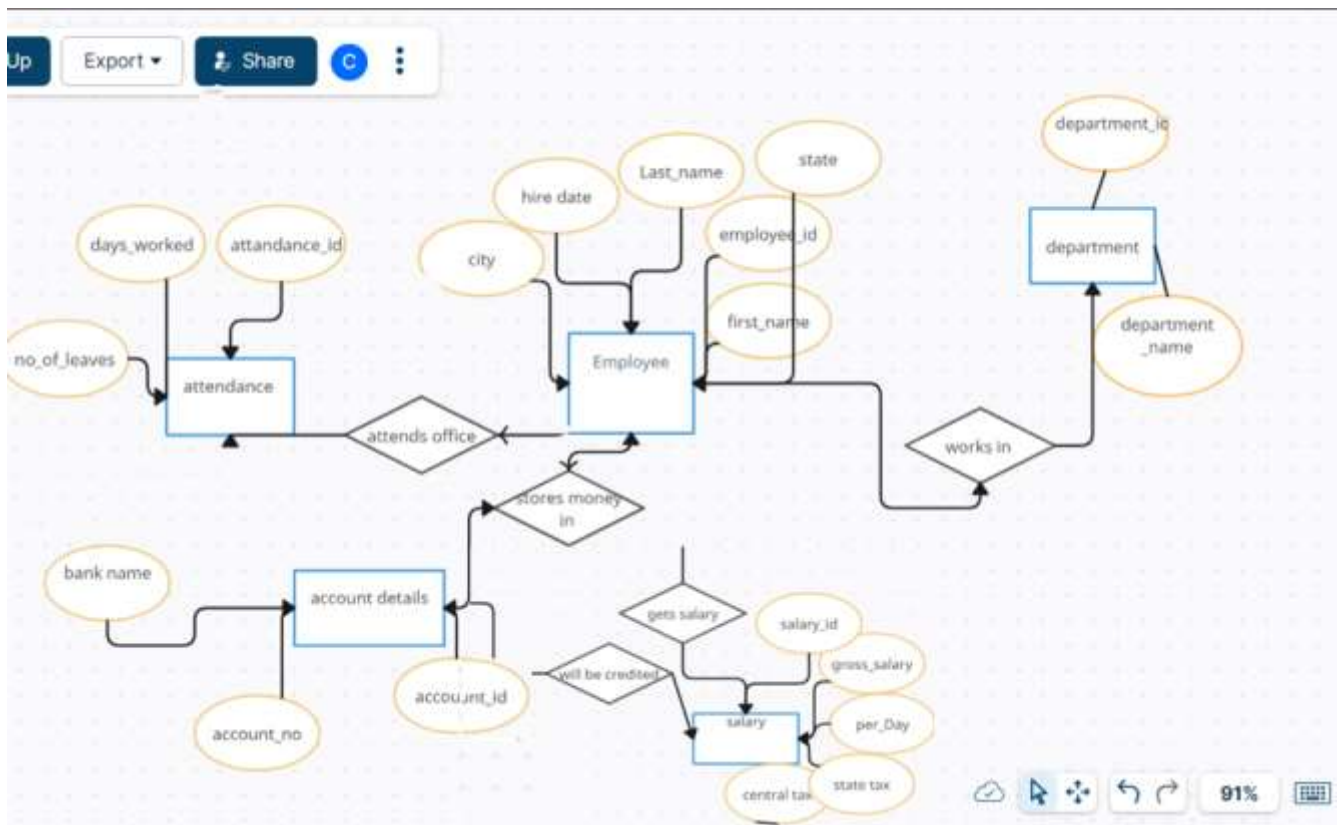
Complexity in Tax Calculations:

Problem: Computing accurate tax deductions and allowances for each employee, considering various factors such as exemptions and deductions, becomes increasingly complex. Manual calculations increase the likelihood of errors in tax-related transactions.

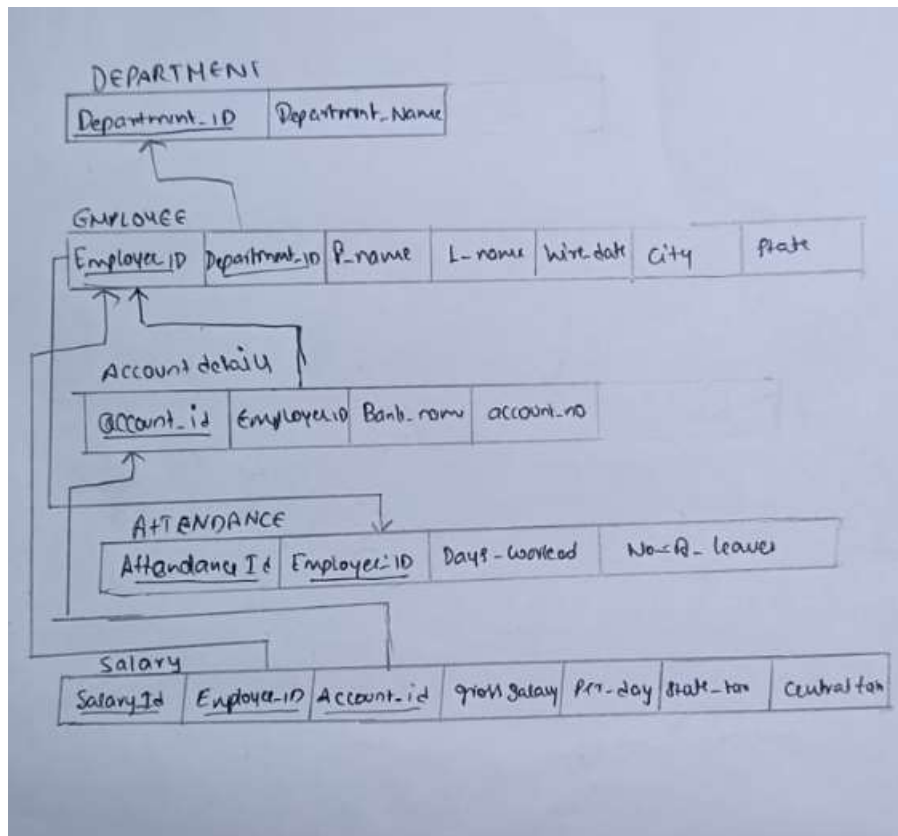
Dependency on Manual Attendance Tracking:

Problem: Relying on manual attendance tracking systems for payroll calculations can result in inaccuracies. Integrating attendance data seamlessly into the payroll process is crucial for precision in compensation.

3. ER MODEL



4. ER TO RELATIONAL MAPPING



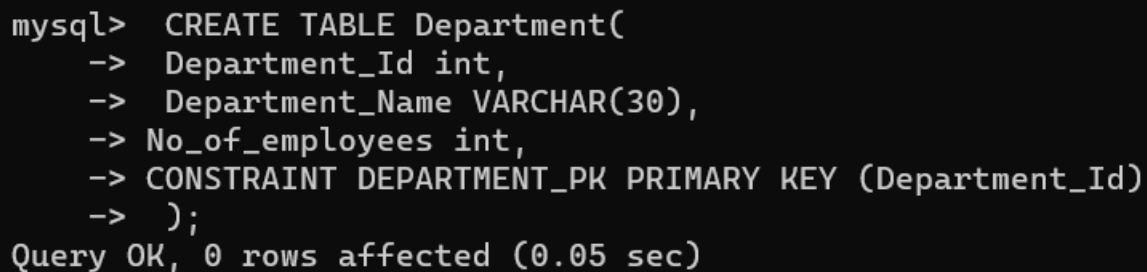
5. DDL STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE CREATION

```
CREATE DATABASE payroll_management;
```

```
USE payroll_management;
```

```
CREATE TABLE Department(  
    Department_Id int,  
    Department_Name VARCHAR(30),  
    No_of_employees int,  
    CONSTRAINT DEPARTMENT_PK PRIMARY KEY (Department_Id)  
);
```



```
mysql> CREATE TABLE Department(  
    -> Department_Id int,  
    -> Department_Name VARCHAR(30),  
    -> No_of_employees int,  
    -> CONSTRAINT DEPARTMENT_PK PRIMARY KEY (Department_Id)  
    -> );  
Query OK, 0 rows affected (0.05 sec)
```

```
CREATE TABLE Employee(  
    Employee_Id INT(6),  
    First_Name VARCHAR(25),  
    Last_Name VARCHAR(25),  
    Hire_Date DATE,  
    City VARCHAR(25),  
    State VARCHAR(25),  
    Department_id INT,  
    CONSTRAINT EMPLOYEE_PK PRIMARY KEY (Employee_Id),  
    FOREIGN KEY (Department_id)
```

REFERENCES Department(Department_id)

);

```
mysql> CREATE TABLE Employee(  
-> Employee_Id INT(6),  
-> First_Name VARCHAR(25),  
-> Last_Name VARCHAR(25),  
-> Hire_Date DATE,  
-> City VARCHAR(25),  
-> State VARCHAR(25),  
-> Department_id INT,  
-> CONSTRAINT EMPLOYEE_PK PRIMARY KEY (Employee_Id),  
-> FOREIGN KEY (Department_id)  
-> REFERENCES Department(Department_id)  
-> );  
Query OK, 0 rows affected, 1 warning (0.05 sec)
```

CREATE TABLE AccountDetails(
Account_Id int,
Bank_Name VARCHAR(50),
Account_Number VARCHAR(50),
Employee_Id INT,
CONSTRAINT Account_PK PRIMARY KEY (Account_Id),
FOREIGN KEY (Employee_Id)
REFERENCES Employee(Employee_Id)
);

```
mysql> CREATE TABLE AccountDetails(  
-> Account_Id int,  
-> Bank_Name VARCHAR(50),  
-> Account_Number VARCHAR(50),  
-> Employee_Id INT,  
-> CONSTRAINT Account_PK PRIMARY KEY (Account_Id),  
-> FOREIGN KEY (Employee_Id)  
-> REFERENCES Employee(Employee_Id)  
-> );  
Query OK, 0 rows affected (0.07 sec)
```

CREATE TABLE Attendance(
Attendance_Id int,

```
Days_worked int,  
No_of_leaves int,  
Employee_id int,  
CONSTRAINT Attendance_PK PRIMARY KEY (Attendance_Id),  
FOREIGN KEY (Employee_Id)  
REFERENCES Employee(Employee_Id)  
);
```

```
mysql> CREATE TABLE Attendance(  
-> Attendance_Id int,  
-> Days_worked int,  
-> No_of_leaves int,  
-> Employee_id int,  
-> CONSTRAINT Attendance_PK PRIMARY KEY (Attendance_Id),  
-> FOREIGN KEY (Employee_Id)  
-> REFERENCES Employee(Employee_Id)  
-> );  
Query OK, 0 rows affected (0.05 sec)
```

```
CREATE TABLE Salary(  
Salary_Id int,  
Gross_Salary int,  
perday int,  
State_Tax int,  
central_Tax int,  
Account_Id int,  
Employee_id int,  
CONSTRAINT SALARY_PK PRIMARY KEY (Salary_Id),  
FOREIGN KEY (Account_Id)  
REFERENCES ACCOUNTDETAILS(Account_Id),  
FOREIGN KEY (Employee_id)  
REFERENCES Employee(employee_id) );
```

```
mysql> CREATE TABLE Salary(  
  -> Salary_Id int,  
  -> Gross_Salary int,  
  -> perday int,  
  -> State_Tax int,  
  -> central_Tax int,  
  -> Account_Id int,  
  -> Employee_id int,  
  -> CONSTRAINT SALARY_PK PRIMARY KEY (Salary_Id),  
  -> FOREIGN KEY (Account_Id)  
  -> REFERENCES ACCOUNTDETAILS(Account_Id),  
  -> FOREIGN KEY (Employee_id)  
  -> REFERENCES Employee(employee_id)  
  ->  
  -> );  
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> show tables;  
+-----+  
| Tables_in_payroll_management |  
+-----+  
| accountdetails                |  
| attendance                    |  
| department                    |  
| employee                     |  
| salary                        |  
+-----+  
5 rows in set (0.01 sec)
```


6. DML STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE WITH INSERTED VALUES

INSERT INTO Employee VALUES (101,'pranavi','Akula','2023-04-13','Bengaluru','karnataka',1);

INSERT INTO Employee VALUES (102,'Sphurthi','Yandamuri','2020-06-12','Bengaluru','karnataka',4);

INSERT INTO Employee VALUES (103,'Himank','Bansal','2021-09-11','Bengaluru','karnataka',2);

INSERT INTO Employee VALUES (104,'Abhiram','Dasika','2023-06-17','Bengaluru','karnataka',5);

INSERT INTO Employee VALUES (105,'Garima','Bajpayi','2024-08-12','Bengaluru','karnataka',2);

```
mysql> INSERT INTO Employee VALUES (102,'Sphurthi','Yandamuri','2020-06-12',  
'Bengaluru','karnataka',4);  
Query OK, 1 row affected (0.01 sec)  
  
mysql> INSERT INTO Employee VALUES (103,'Himank','Bansal','2021-09-11','Beng  
aluru','karnataka',2);  
Query OK, 1 row affected (0.01 sec)  
  
mysql> INSERT INTO Employee VALUES (104,'Abhiram','Dasika','2023-06-17','Ben  
galuru','karnataka',5);  
Query OK, 1 row affected (0.01 sec)  
  
mysql> INSERT INTO Employee VALUES (105,'Garima','Bajpayi','2024-08-12','Ben  
galuru','karnataka',2);  
Query OK, 1 row affected (0.00 sec)
```

INSERT INTO Department VALUES (1,'Human Resources',0);

INSERT INTO Department VALUES (2,'Software Development',0);

INSERT INTO Department VALUES (3,'Data Analysis',0);

INSERT INTO Department VALUES (4,'Data Science',0);

INSERT INTO Department VALUES (5,'Business Intelligence',0);

```
mysql> INSERT INTO Department VALUES (2,'Software Development',0);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Department VALUES (3,'Data Analysis',0);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Department VALUES (4,'Data Science',0);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Department VALUES (5,'Business Intelligence',0);
Query OK, 1 row affected (0.00 sec)

mysql> select * from department;
```

```
mysql> select * from department;
+-----+-----+
| Department_Id | Department_Name |
+-----+-----+
| 1 | Human Resources |
| 2 | Software Development |
| 3 | Data Analysis |
| 4 | Data Science |
| 5 | Business Intelligence |
+-----+-----+
5 rows in set (0.00 sec)
```

INSERT INTO AccountDetails VALUES (40,'Axis','S12344',101);

INSERT INTO AccountDetails VALUES (41,'Axis','S12345',102);

INSERT INTO AccountDetails VALUES (42,'Axis','S12346',103);

INSERT INTO AccountDetails VALUES (43,'Axis','S12347',104);

INSERT INTO AccountDetails VALUES (44,'Axis','C12344',105);

```
mysql> INSERT INTO AccountDetails VALUES (40,'Axis','S12344',101);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO AccountDetails VALUES (41,'Axis','S12345',102);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO AccountDetails VALUES (42,'Axis','S12346',103);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO AccountDetails VALUES (43,'Axis','S12347',104);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO AccountDetails VALUES (44,'Axis','C12344',105);
Query OK, 1 row affected (0.00 sec)
```

INSERT INTO Attendance VALUES(20,22,4,101);

INSERT INTO Attendance VALUES(21,20,6,102);

INSERT INTO Attendance VALUES(22,24,2,103);

INSERT INTO Attendance VALUES(23,19,7,104);

INSERT INTO Attendance VALUES(24,25,1,105);

```
mysql> INSERT INTO Attendance VALUES(20,22,4,101);  
Query OK, 1 row affected (0.03 sec)
```

```
mysql> INSERT INTO Attendance VALUES(21,20,6,102);  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Attendance VALUES(22,24,2,103);  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Attendance VALUES(23,19,7,104);  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Attendance VALUES(24,25,1,105);  
Query OK, 1 row affected (0.01 sec)
```

INSERT INTO salary VALUES(1,100000,3571,1700,5000,40,101);

INSERT INTO salary VALUES(2,200000,7142,3000,7000,41,102);

INSERT INTO salary VALUES(3,130000,4642,1700,5000,42,103);

INSERT INTO salary VALUES(4,120000,4285,1700,5000,43,104);

INSERT INTO salary VALUES(5,200000,7142,3000,7000,44,105);

```
mysql> INSERT INTO salary VALUES(1,100000,3571,1700,5000,40,101);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO salary VALUES(2,200000,7142,3000,7000,41,102);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO salary VALUES(3,130000,4642,1700,5000,42,103);  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO salary VALUES(4,120000,4285,1700,5000,43,104);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO salary VALUES(5,200000,7142,3000,7000,44,105);  
Query OK, 1 row affected (0.01 sec)
```

7. QUERIES

7.1 SIMPLE QUERY WITH GROUP BY, AGGREGATE

Select first_name from employee where (hire_date>'2023-01-01') group by department_id;

The above query returns all the first names of the employees who joined the company in and after 2023

Ordered based on department_id.

```
mysql> Select first_name from employee where (hire_date>'2023-01-01') group by department_id;
+-----+
| first_name |
+-----+
| pranavi    |
| Garima     |
| Abhiram    |
+-----+
3 rows in set (0.00 sec)
```

7.2 UPDATE OPERATION

```
mysql> UPDATE Employee
-> set city='bengaluru',state='karnataka'
-> where Employee_Id=101;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

the above query updates the table employee sets city and state columns of that entry as specified

7.3 DELETE OPERATION

alter table department drop no_of_Employees;

We drop a specified column from the table using the above query

```
mysql> alter table department
-> drop No_of_employees;
Query OK, 0 rows affected (0.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from department;
+-----+-----+
| Department_Id | Department_Name |
+-----+-----+
| 1 | Human Resources |
| 2 | Software Development |
| 3 | Data Analysis |
| 4 | Data Science |
| 5 | Business Intelligence |
+-----+-----+
5 rows in set (0.00 sec)
```

7.4 CORRELATED QUERY

select * from abavg where gross_salary>(select avg(gross_salary) from abavg);

```
mysql> select * from abavg where gross_salary>(select avg(gross_salary) from abavg);
```

first_name	Gross_Salary
Sphurthi	200000
Garima	200000

```
2 rows in set (0.01 sec)
```

Create view abavg

As Select e.first_name , s.Gross_Salary from employee e inner join salary s on e.employee_id=s.employee_id;

7.5 NESTED QUERY

create or replace view used_paid_leaves

as

select e.first_name,e.last_name from employee e

inner join attendance a on a.employee_id=e.employee_id where a.no_of_leaves>3;

```
mysql> create or replace view used_paid_leaves
-> as
-> select e.first_name,e.last_name from employee e
-> inner join attendance a on a.employee_id=e.employee_id
-> where a.no_of_leaves>3;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from used_paid_leaves;
```

first_name	last_name
pranavi	Akula
Sphurthi	Yandamuri
Abhiram	Dasika

```
3 rows in set (0.01 sec)

mysql> |
```

The above query creates a view of first name and last name of employees who took a leave of more than 3 days

8. STORED PROCEDURES, FUNCTIONS AND TRIGGERS

8.1 STORED PROCEDURES OR FUNCTIONS

```
CREATE PROCEDURE CalculateSalary(IN empId INT, IN daysWorked INT)
-> BEGIN
->   DECLARE grossSalary INT;
->   DECLARE perDaySalary INT;
->   DECLARE stateTax FLOAT;
->   DECLARE centralTax FLOAT;
->
->   -- Fetch employee's salary-related information from the Salary table
->   SELECT s.Gross_Salary, s.PerDay, s.State_Tax, s.Central_Tax
->   INTO grossSalary, perDaySalary, stateTax, centralTax
->   FROM Salary s
->   WHERE s.Employee_id = empId;
->
->   -- Calculate gross salary
->   SET grossSalary = (daysWorked * perDaySalary) - stateTax - centralTax;
->
->   -- Update the Salary table with the calculated salary
->   UPDATE Salary
->   SET Gross_Salary = grossSalary
->   WHERE Employee_Id = empId;
END //
```

```
mysql> select * from salary;
+-----+-----+-----+-----+-----+-----+-----+
| Salary_Id | Gross_Salary | perday | State_Tax | central_Tax | Account_Id | Employee_id |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | -550208967 | 3571 | 1700 | 5000 | 40 | 101 |
| 2 | 200000 | 7142 | 3000 | 7000 | 41 | 102 |
| 3 | 130000 | 4642 | 1700 | 5000 | 42 | 103 |
| 4 | 120000 | 4285 | 1700 | 5000 | 43 | 104 |
| 5 | 200000 | 7142 | 3000 | 7000 | 44 | 105 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> call CalculateSalary(101, 30);
Query OK, 1 row affected (0.00 sec)

mysql> select * from salary;
+-----+-----+-----+-----+-----+-----+-----+
| Salary_Id | Gross_Salary | perday | State_Tax | central_Tax | Account_Id | Employee_id |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 100430 | 3571 | 1700 | 5000 | 40 | 101 |
| 2 | 200000 | 7142 | 3000 | 7000 | 41 | 102 |
| 3 | 130000 | 4642 | 1700 | 5000 | 42 | 103 |
| 4 | 120000 | 4285 | 1700 | 5000 | 43 | 104 |
| 5 | 200000 | 7142 | 3000 | 7000 | 44 | 105 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

8.2 TRIGGERS

DELIMITER \$\$

CREATE TRIGGER empno

AFTER INSERT on employee

for each row

begin

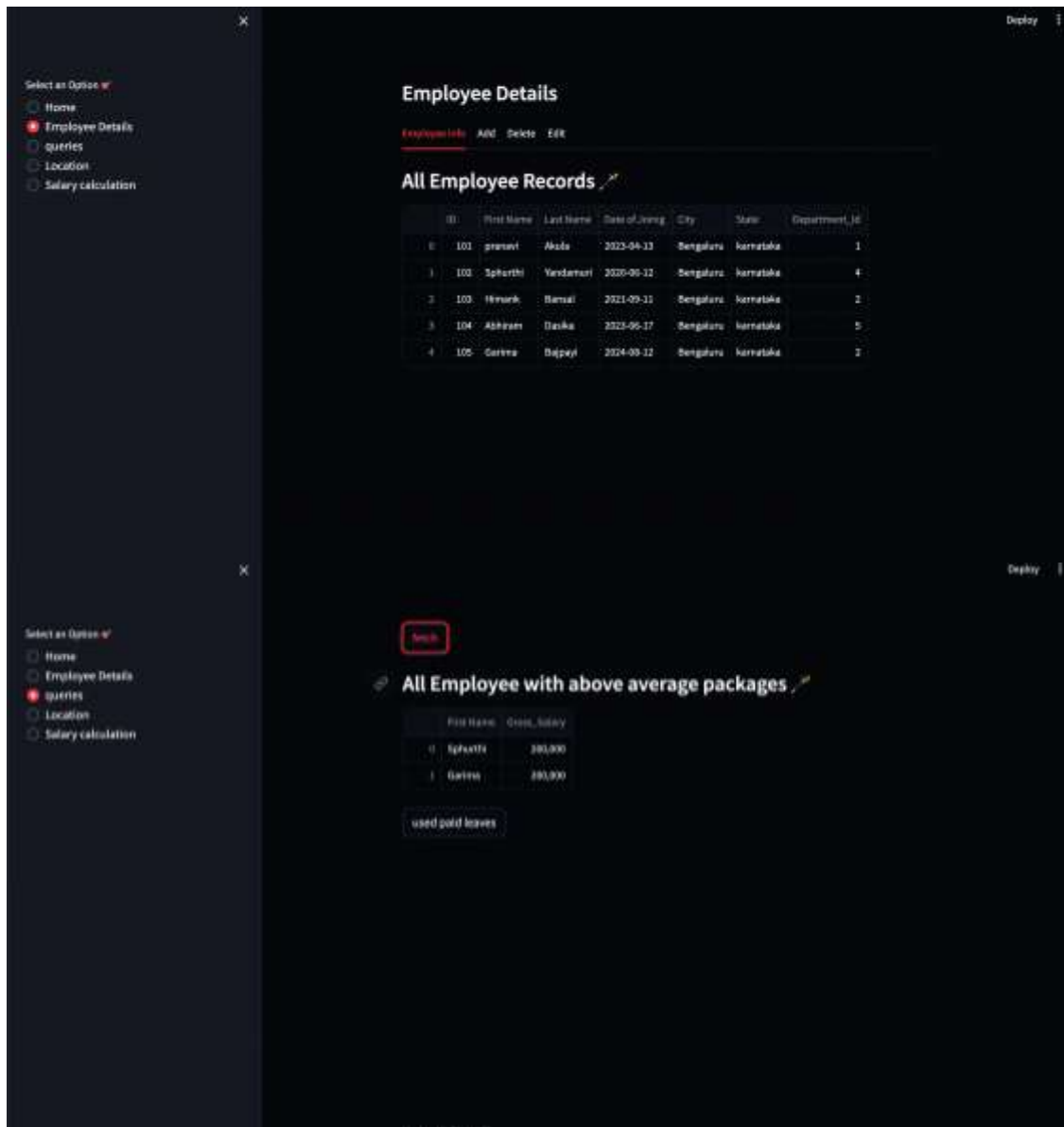
update total_employees e

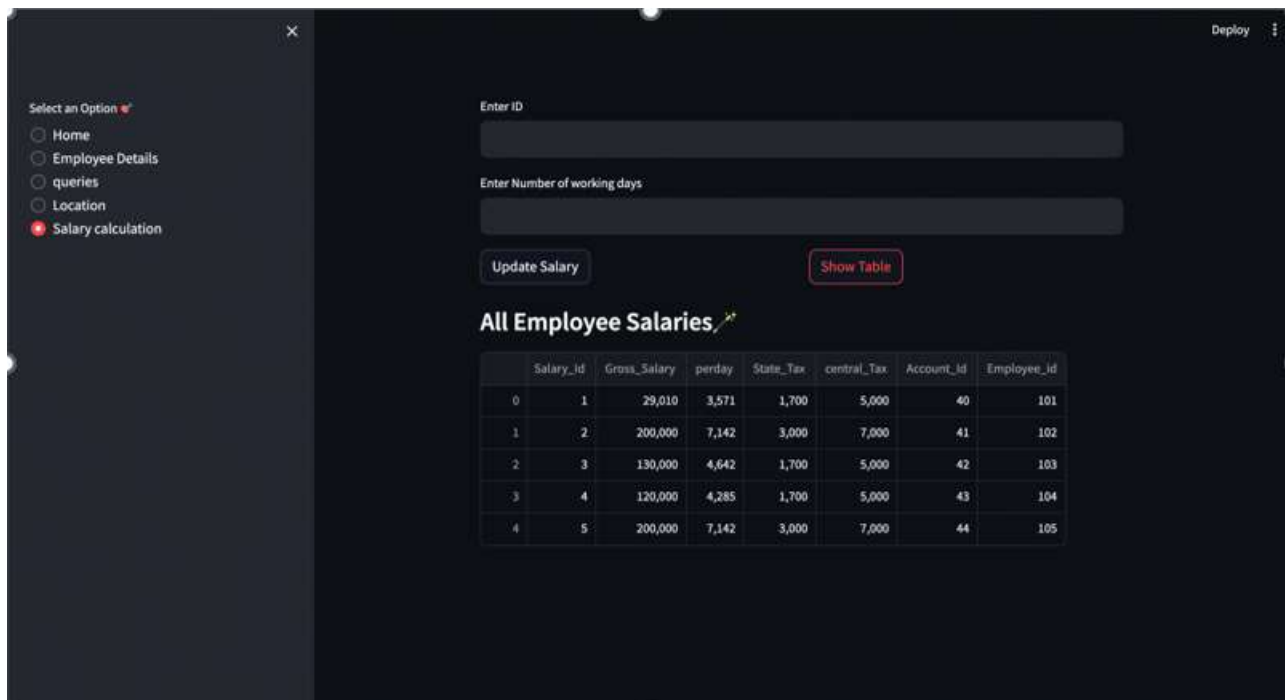
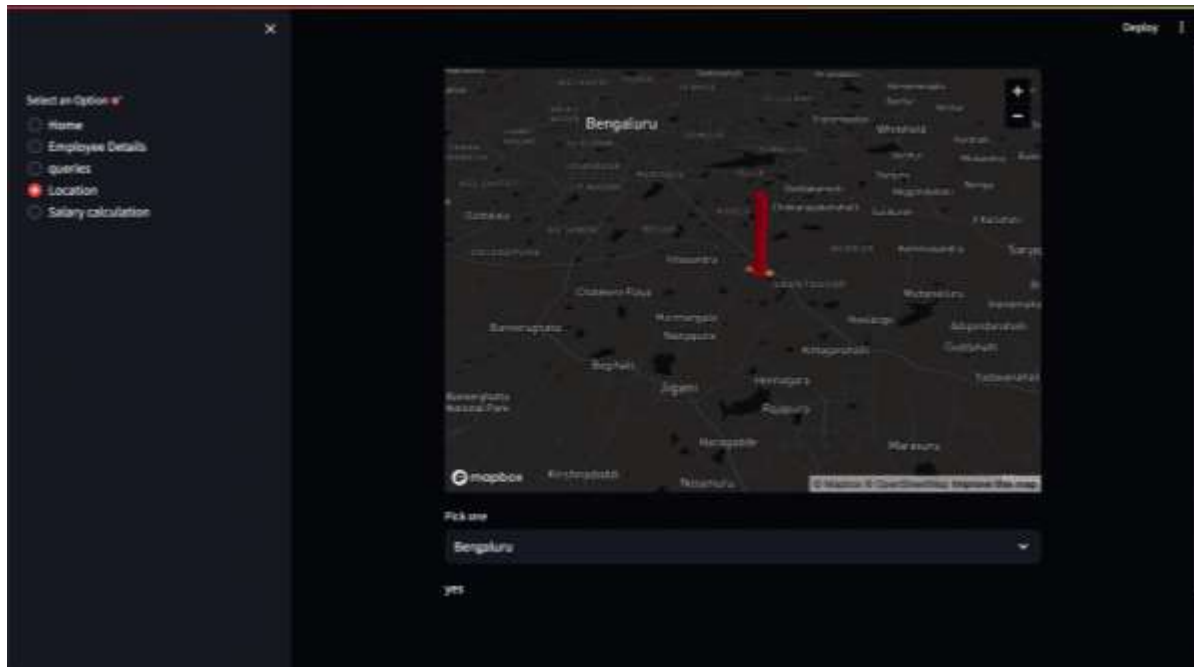
set e.no_of_employees=e.no_of_employees+1;

end \$\$

```
mysql> DELIMITER $$
mysql> CREATE TRIGGER empno
  -> AFTER INSERT on employee
  -> for each row
  -> begin
  -> update total_employees e
  -> set e.no_of_employees=e.no_of_employees+1;
  -> end $$
Query OK, 0 rows affected (0.01 sec)
```


9. FRONT END DEVELOPEMNT





REFERENCES

- [1] <https://www.w3schools.com/sql>
- [2] <https://www.simplilearn.com/tutorials/sql-tutorial/stored-procedure-in-sql>

