INTERNET APPLICATIONS DEVELOPMENT

5/23/2025

EMPLOYEE PAYROLL SYSTEM

Submitted to Mr. Irfan Hameed

Submitted by Shaheena

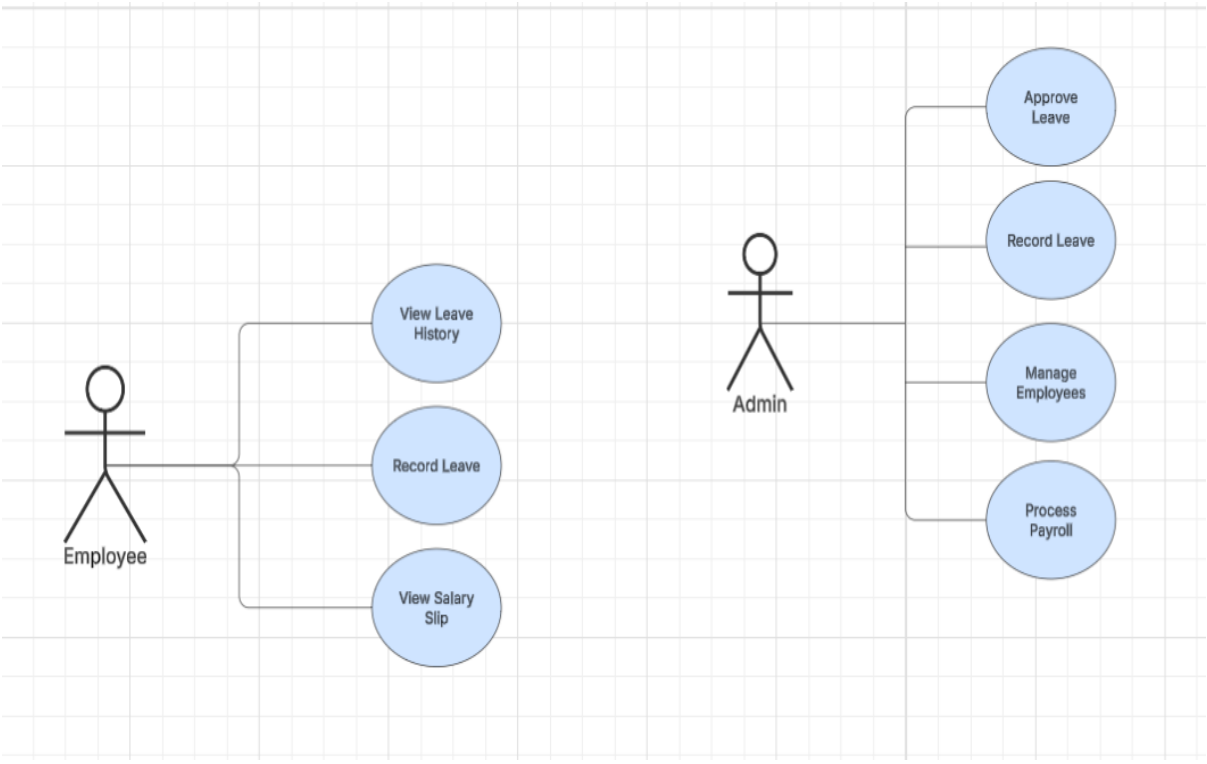
1. Project Description

The **Employee Payroll System (EPS)** is a fully integrated, database-driven system designed to automate, manage, and streamline employee-related payroll processes in an organization. This system is responsible for handling employee details, department assignments, tracking leaves, processing salaries, and generating salary slips. It ensures **data accuracy**, **efficiency**, and **security** across all payroll-related tasks, supporting the organization's human resources management.

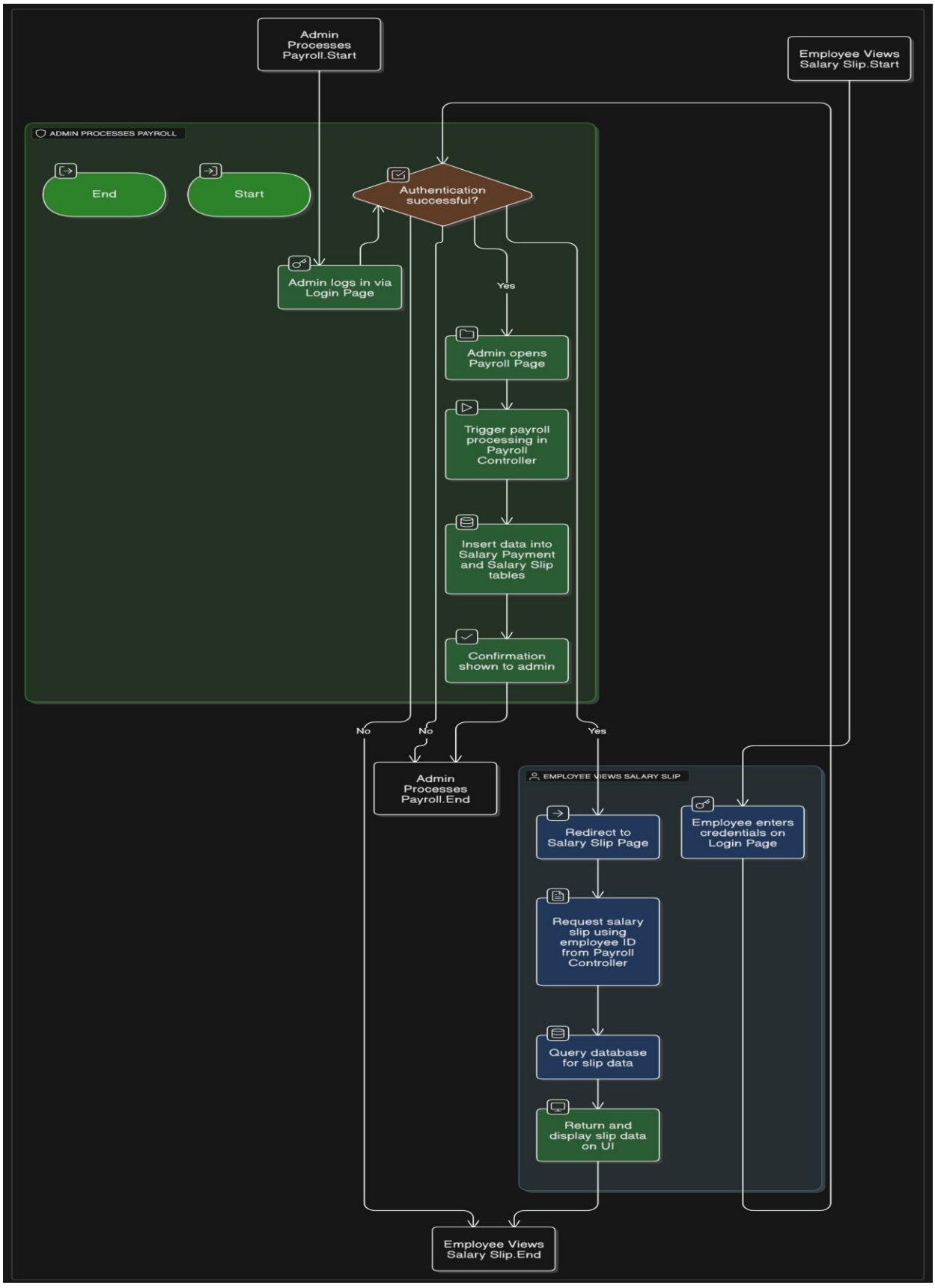
1.1. Database Design Features

- **Primary Keys**: Each table has a unique primary key to guarantee data integrity and prevent duplication.
- **Foreign Keys**: Relationships between tables are managed using **foreign keys** (e.g., EMPLOYEE_ID references the EMPLOYEE table in various entities like PAYMENT, LEAVE_RECORDS). Foreign keys enforce **referential integrity**.
- **Constraints**: **NOT NULL** constraints are applied to fields that are mandatory (e.g., `EMPLOYEE_ID`, `SALARY_PAYMENT_ID`, etc.). **CHECK** constraints can be used for salary amounts or leave days to ensure business rules are enforced (e.g., salary amount cannot be negative). **UNIQUE** constraints can be enforced on fields such as **email** or **employee ID** to maintain data integrity.
- **Default Values**: **Default values** (e.g., SYSDATE for date fields) ensure timely record creation without requiring manual data entry for every record.

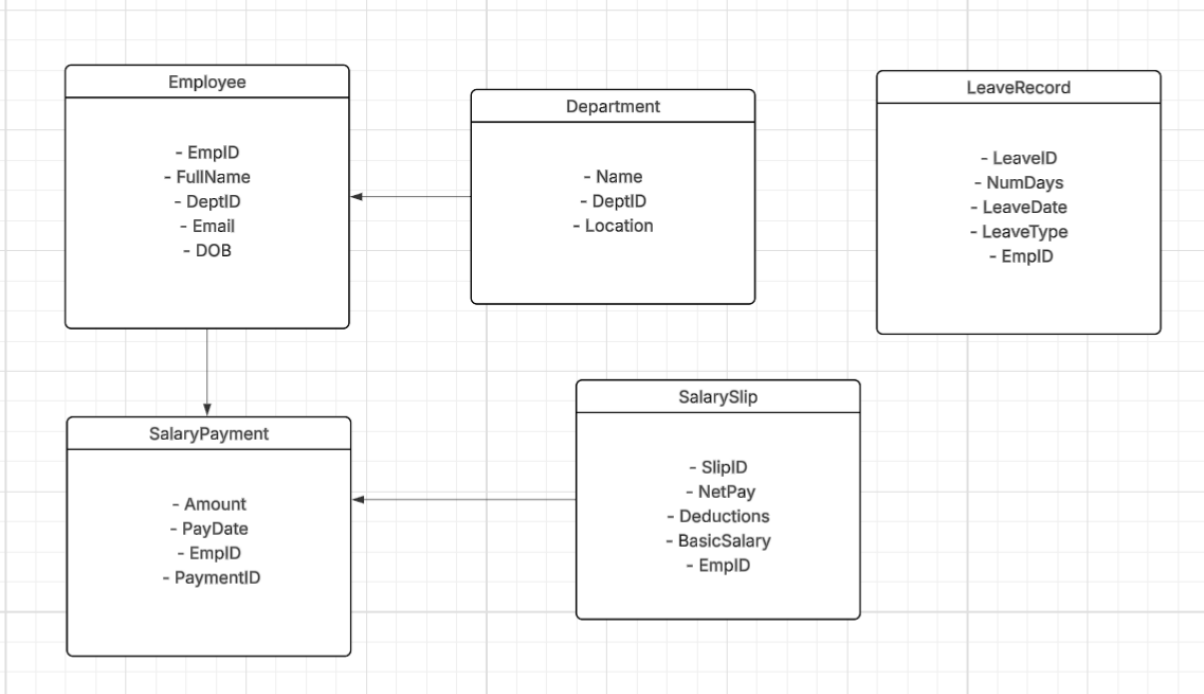
2. Use Case Diagram



3. Sequence Diagram



4. Class Diagrams



5. Enterprise Data Model

The **Enterprise Data Model (EDM)** represents a **conceptual** and **high-level view** of the organization's payroll data. It captures the major components of the payroll system:

5.1. Core Entities in the EPS:

1. **Employee**:
   - Contains core information about employees, such as ID, name, contact information, department assignments, and salary,street city and country.
2. **Department**:
   - Represents the various departments in the organization where employees work.
3. **Salary Payment**:
   - Handles the **salary transactions** for each employee, including basic salary, allowances, and total payment details.
4. **Salary Slip**:
   - Generates a document detailing the monthly salary and deductions for each employee.
5. **Leave Records**:
   - Tracks employee leaves, including leave type, days taken, and leave dates.
6. **Contact**:
   - Manages employee contact details (including multi-valued phone numbers) and addresses (which include street, city, and country).
   -

5.2. Relationships Between Entities:

- **Employee** is linked to **Department(s)** via a **many-to-many relationship** through the EMPLOYEE_DEPARTMENT table.
- **Employee** has **Leave Records** to track employee absence.
- **Employee** receives **Salary Payments** and generates **Salary Slips**.
- **Leave Records** are tied to a specific **Employee** (one-to-many relationship).
- **Salary Payments** are related to **Salary Slips** (one-to-one).
- **Employee** can have multiple **Contact Numbers**.

5.3. Entities and Attributes

1. **EMPLOYEE**:
    - EMPLOYEE_ID (PK), FIRST_NAME, LAST_NAME, EMAIL, PHONE (Multi-valued), DATE_OF_JOINING, ADDRESS (STREET, CITY, COUNTRY).

2. **EMPLOYEE_CONTACT**:
    - CONTACT_ID (PK), EMPLOYEE_ID (FK), PHONE_NUMBER (Composite key with EMPLOYEE_ID).

3. **EMPLOYEE_ADDRESS**:
    - ADDRESS_ID (PK), EMPLOYEE_ID (FK), STREET, CITY, COUNTRY.

4. **DEPARTMENT**:
    - DEPARTMENT_ID (PK), DEPARTMENT_NAME, MANAGER_ID (FK) (foreign key to the EMPLOYEE table for department manager).

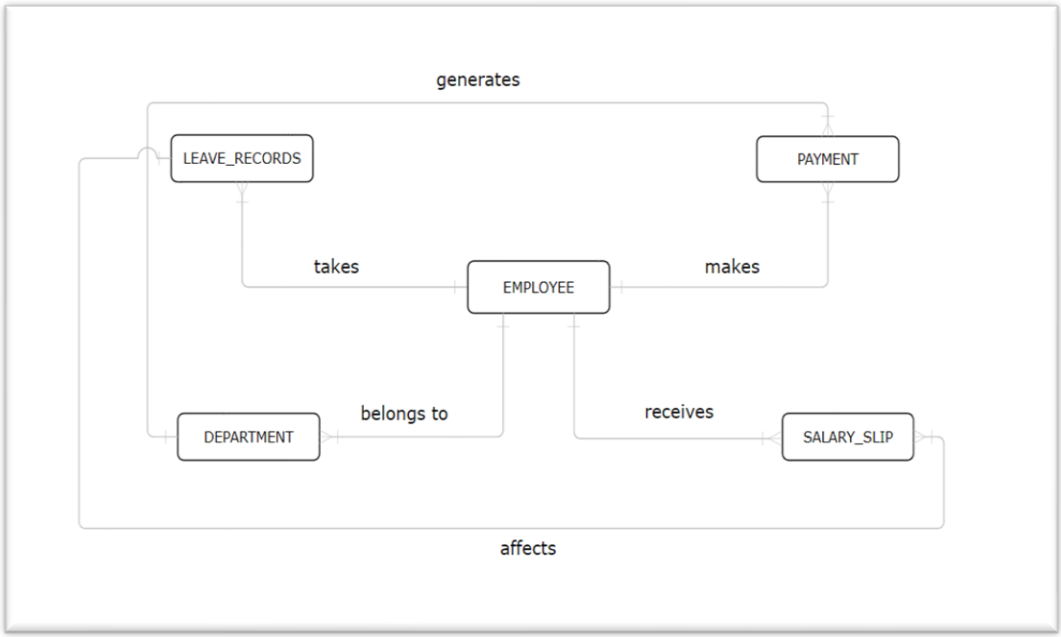
5. **EMPLOYEE_DEPARTMENT**:
    - EMPLOYEE_ID (FK), DEPARTMENT_ID (FK) (Composite key).

6. **SALARY_PAYMENT**:
    - PAYMENT_ID (PK), EMPLOYEE_ID (FK), BASIC_SALARY, TOTAL_PAY, MONTH, YEAR.

7. **SALARY_SLIP**:
    - SLIP_ID (PK), PAYMENT_ID (FK), EMPLOYEE_ID (FK), SALARY_BREAKDOWN (e.g., allowances, deductions).

8. **LEAVE_RECORDS**:
    - LEAVE_ID (PK), EMPLOYEE_ID (FK), LEAVE_TYPE, LEAVE_DAYS, LEAVE_DATE.

6. Enterprise Data Model



Enterprise Data Model

7. Entity-Relationship (ER) Model

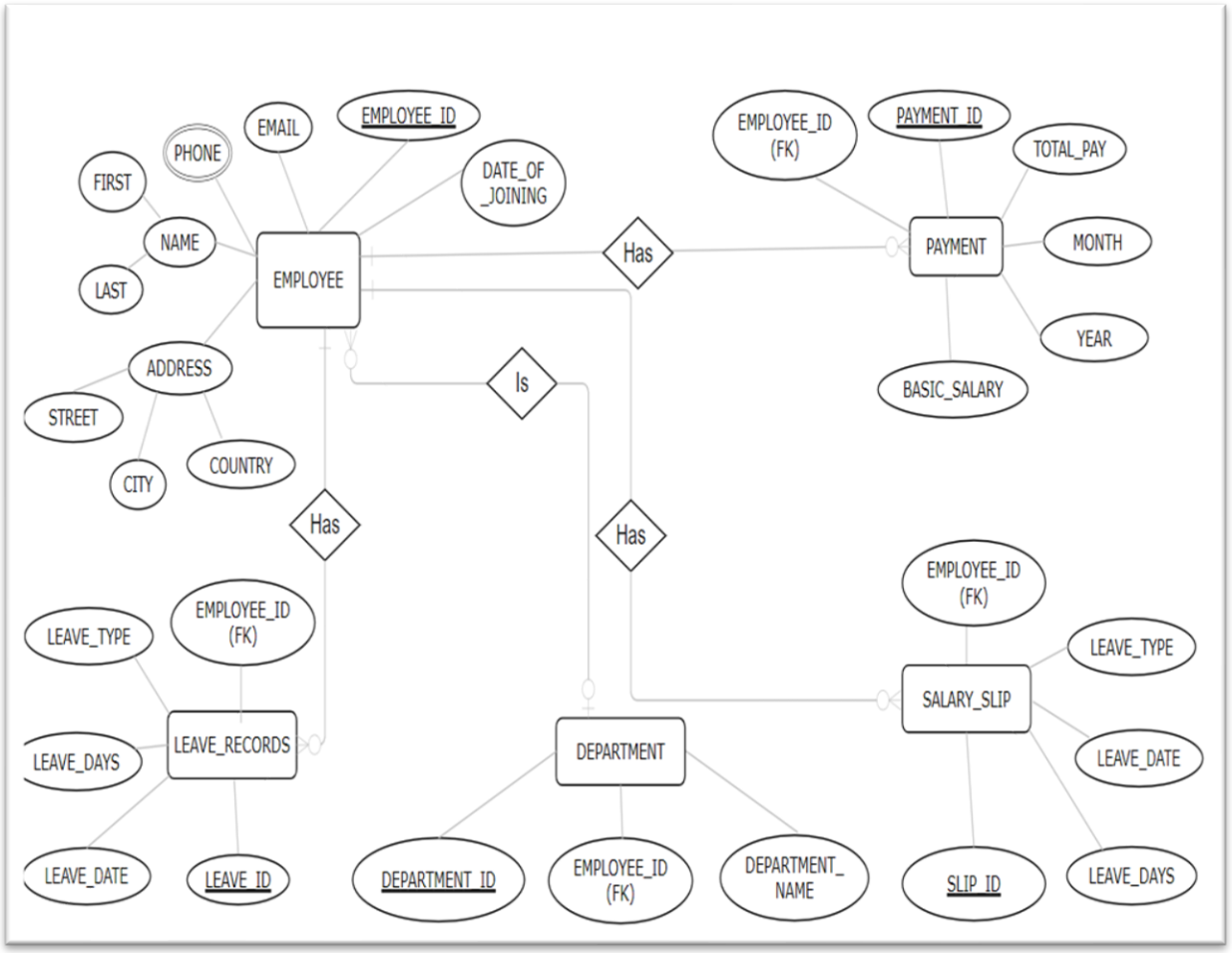
This model represents how entities interact with one another, showing their attributes and relationships:

Entities:

- **EMPLOYEE** (with attributes like EMPLOYEE_ID, FIRST_NAME, EMAIL)
- **DEPARTMENT** (with DEPARTMENT_ID, DEPARTMENT_NAME)
- **LEAVE_RECORDS** (with LEAVE_ID, LEAVE_TYPE)
- **SALARY_PAYMENT** (with PAYMENT_ID, BASIC_SALARY)
- **SALARY_SLIP** (with SLIP_ID, SALARY_BREAKDOWN)

Relationships:

- **EMPLOYEE** *belongs to* one or more **DEPARTMENTS** through EMPLOYEE_DEPARTMENT.
- **EMPLOYEE** *has* multiple **LEAVE_RECORDS**.
- **EMPLOYEE** *receives* one **SALARY_PAYMENT** each month.
- **SALARY_PAYMENT** *generates* a **SALARY_SLIP**.
- **EMPLOYEE** *has* multiple **CONTACT NUMBERS** stored in **EMPLOYEE_CONTACT**.
- **EMPLOYEE** *has* one **EMPLOYEE_ADDRESS**.



Entity Relationship Model

8. Business Rules

The system enforces various **business rules** to ensure correct and consistent data management:

1. **Department Membership**:
   - An **Employee** must belong to at least one **Department**.
   - An **Employee** can belong to multiple **Departments**.
2. **Phone Numbers**:
   - An **Employee** can have **multiple phone numbers** (multi-valued).
   - A **Contact Number** should be stored in a separate table (with a foreign key referencing EMPLOYEE).
3. **Salary Payment Generation**:
   - Each **Salary Payment** is tied to one **Employee**.
   - **Salary Payments** are created monthly with details like BASIC_SALARY and TOTAL_PAY.
4. **Leave Records**:
   - An **Employee** can have multiple **Leave Records**, each with a LEAVE_TYPE, LEAVE_DAYS, and LEAVE_DATE.

5. **Salary Slips**:
    - **Salary Slips** are generated for each **Salary Payment** and contain a detailed breakdown of the employee's pay.
    -
6. **Employee Address**:
    - **Employee Addresses** must include STREET, CITY, and COUNTRY.
7. **Constraints**:
    - **Salary** records must ensure that **Basic Salary** and **Total Pay** are positive numbers.
    - **Leave Records** cannot exceed the number of available leave days.

9. Tables for Normalization

EMPLOYEE

EMPLOYEE_ID	EMAIL	NAME	ADDRESS	DATE_OF_JOINING	PHONE

PAYMENT

PAYMENT_ID	EMPLOYEE_ID	MONTH	YEAR	TOTAL_PAY	BASIC_SALARY

LEAVE_RECORDS

LEAVE_ID	EMPLOYEE_ID	LEAVE_TYPE	LEAVE_DAYS	LEAVE_DATE

SALARY_SLIP

SLIP_ID	EMPLOYEE_ID	LEAVE_TYPE	LEAVE_DATE	LEAVE_DAYS

DEPARTMENT

DEPARTMENT_ID	EMPLOYEE_ID	DEPARTMENT_NAME

- ## Normalized Form:

1. **Phone** is multivalued.
2. **Name** is divisible into **First Name** and **Last Name**.
3. **Address** comprises **Street, City, and Country**.

---

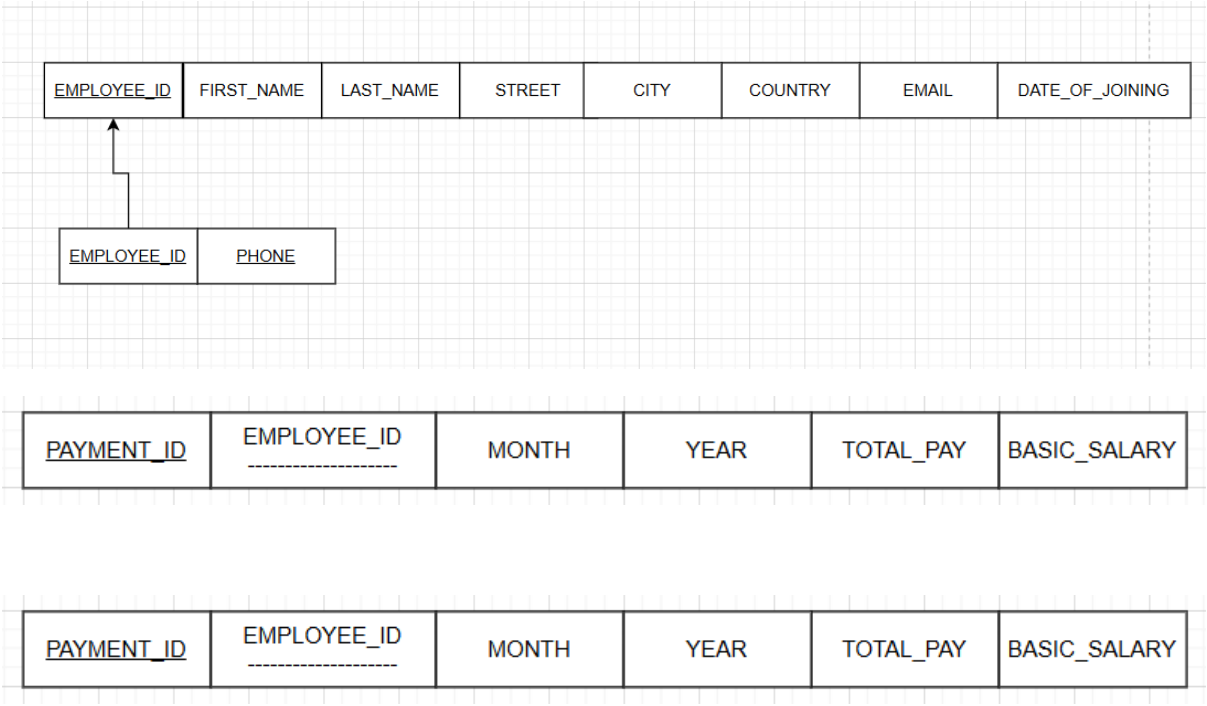
1. First Normal Form (1NF)

In 1NF:

- Atomic attributes (no multi-valued or composite attributes).
- Each row is unique.

Entities in 1NF:

As we have only one attribute which is multivalued so we created a separate table for phone number of the employee.



---

2. Second Normal Form (2NF)

- All non-key attributes depend on the entire primary key.
- The existing entities already satisfy 2NF since there are no partial dependencies.

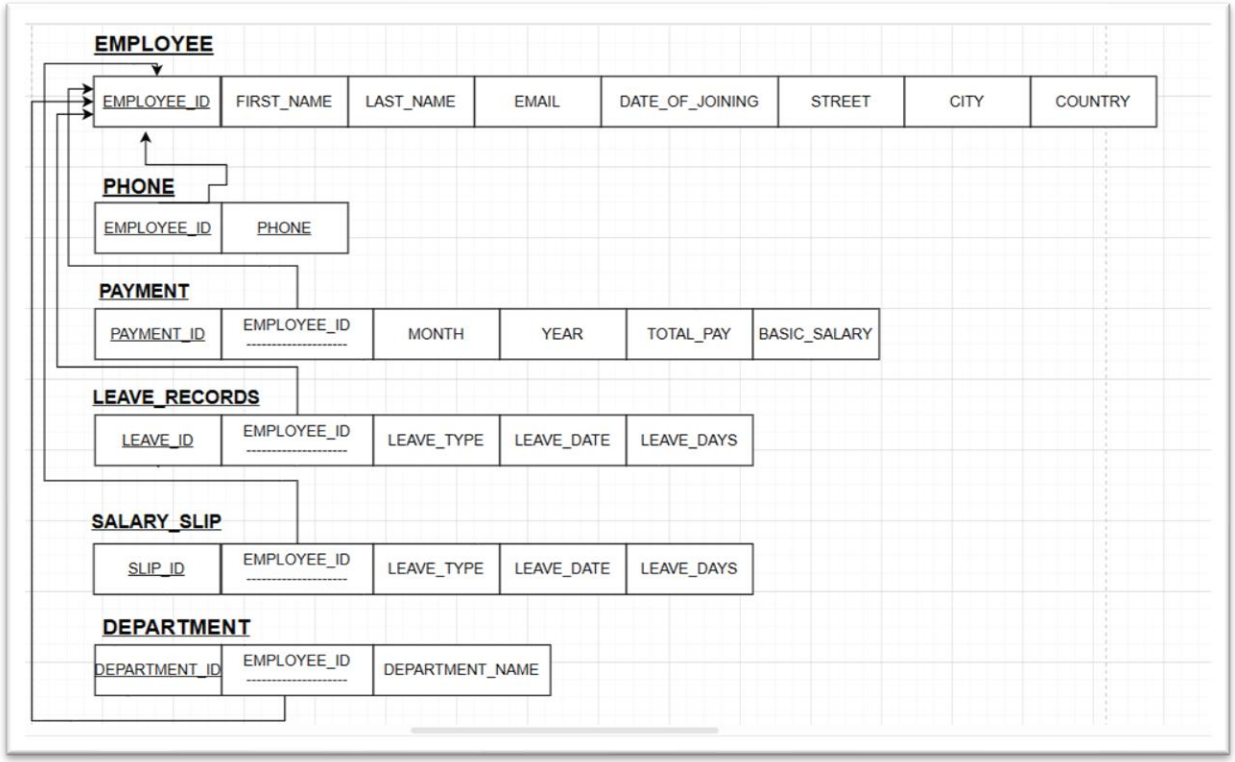
3. Third Normal Form (3NF)

- Remove transitive dependencies: non-key attributes must depend only on the primary key.

Final Normalized Entities

1. **Employee**: Main employee information.
2. **Phone**: Manages multivalued phone numbers.
3. **Payment**: Salary payment details.

4. **Leave Records**: Leave information for employees.
5. **Salary Slip**: Leave-related salary slip data.
6. **Department**: Department details.



Relational Schema

- The **Employee_Phone** table resolves the multivalued attribute issue.
- The **Address** field is decomposed into **Street, City, and Country** for atomicity.
- The **Name** attribute is split into **First Name** and **Last Name** for better clarity.

10. SQL Queries:

```
1> CREATE TABLE Users (
2>     UserID INT IDENTITY(1,1) PRIMARY KEY,
3>     Username NVARCHAR(50) NOT NULL UNIQUE,
4>     Password NVARCHAR(100) NOT NULL
5> );
6> GO
1> INSERT INTO Users (Username, Password)
2> VALUES ('payrollAdmin', 'SecurePass2025!');
```

```
3> GO

(1 rows affected)
1> CREATE TABLE Departments (
2>     DeptID INT IDENTITY(1,1) PRIMARY KEY,
3>     DeptName NVARCHAR(100) NOT NULL
4> );
5> GO
1> INSERT INTO Departments (DeptName)
2> VALUES ('HR'), ('IT'), ('Finance'), ('Sales');
3> GO

(4 rows affected)
1> CREATE TABLE Employees (
2>     EmpID INT IDENTITY(1,1) PRIMARY KEY,
3>     FullName NVARCHAR(100) NOT NULL,
4>     Email NVARCHAR(100) NOT NULL UNIQUE,
5>     Address NVARCHAR(255),
6>     Phone NVARCHAR(20),
7>     DOJ DATE NOT NULL,
8>     DeptID INT FOREIGN KEY REFERENCES Departments(DeptID)
9> );
10> GO
1> CREATE TABLE Leaves (
2>     LeaveID INT IDENTITY(1,1) PRIMARY KEY,
3>     EmpID INT FOREIGN KEY REFERENCES Employees(EmpID),
4>     LeaveType NVARCHAR(50),
5>     LeaveDate DATE NOT NULL,
6>     NumDays INT CHECK (NumDays > 0)
7> );
8> GO
```

```
1> CREATE TABLE Payments (
2>     PaymentID INT IDENTITY(1,1) PRIMARY KEY,
3>     EmpID INT FOREIGN KEY REFERENCES Employees(EmpID),
4>     Month NVARCHAR(20) NOT NULL,
5>     Year INT NOT NULL CHECK (Year >= 1900),
6>     BasicSalary DECIMAL(18,2) NOT NULL CHECK (BasicSalary >= 0),
7>     TotalPay DECIMAL(18,2) NOT NULL CHECK (TotalPay >= 0)
8> );
9> GO
1> INSERT INTO Users (Username, Password)
2> VALUES ('payrollAdmin', 'SecurePass2025!');
3> GO
Msg 2627, Level 14, State 1, Server Computer\SQLEXPRESS01, Line 1
Violation of UNIQUE KEY constraint 'UQ__Users__536C85E4CAFF33B1'. Cannot insert duplicate key in object 'dbo.Users'. The duplicate key value is (payrollAdmin).
The statement has been terminated.
1> INSERT INTO Departments (DeptName)
2> VALUES
3> ('HR'),
4> ('IT'),
5> ('Finance'),
6> ('Sales'),
7> ('Operations');
8> GO

(5 rows affected)
1> INSERT INTO Employees (FullName, Email, Address, Phone, DOJ, DeptID)
2> VALUES
3> ('Ali Raza', 'ali.raza@example.com', 'Street 12, Lahore', '03121234567', '2023-05-15', 1),
4> ('Sara Khan', 'sara.khan@example.com', 'Street 45, Karachi', '03219876543', '2022-10-10', 2),
5> ('Zain Ahmad', 'zain.ahmad@example.com', 'Block A, Islamabad', '03001112222', '2021-08-01', 3);
```

```
6> GO

(3 rows affected)
1> INSERT INTO Leaves (EmpID, LeaveType, LeaveDate, NumDays)
2> VALUES
3> (1, 'Sick Leave', '2024-01-10', 2),
4> (2, 'Casual Leave', '2024-03-05', 1),
5> (3, 'Annual Leave', '2024-04-15', 5);
6> GO

(3 rows affected)
1> INSERT INTO Payments (EmpID, Month, Year, BasicSalary, TotalPay)
2> VALUES
3> (1, 'January', 2025, 50000.00, 55000.00),
4> (1, 'February', 2025, 50000.00, 55000.00),
5> (2, 'January', 2025, 60000.00, 65000.00),
6> (3, 'January', 2025, 70000.00, 75000.00);
7> GO

(4 rows affected)
1> SELECT * FROM Payments;
2> GO
PaymentID   EmpID       Month               Year        BasicSalary         TotalPay
----------- ----------- ------------------- ----------- ------------------- --------------------
          1           1 January                    2025            50000.00            55000.00
          2           1 February                   2025            50000.00            55000.00
          3           2 January                    2025            60000.00            65000.00
          4           3 January                    2025            70000.00            75000.00
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
1	HR	1
2	Finance	2
3	IT	3

PaymentID	EmpID	Month	Year	BasicSalary	TotalPay
1	1	January	2025	50000.00	55000.00
2	1	February	2025	50000.00	55000.00
3	2	January	2025	60000.00	65000.00
4	3	January	2025	70000.00	75000.00
5	1	january	2025	112761.00	21783.00

```
SELECT * FROM Departments;
```

Set 1. Total rows: 9

DeptID	DeptName
1	HR
2	IT
3	Finance
4	Sales
5	HR
6	IT
7	Finance
8	Sales
9	Operations

Enter your SQL query or T-SQL batch. It will be executed on database: ShaheenaDatabase

```
SELECT * FROM Employees;
```

EXECUTE T-SQL

Set 1. Total rows: 3

EmpID	FullName	Email	Address	Phone	DOJ	DeptID
1	Ali Raza	ali.raza@example.com	Street 12, Lahore	03121234567	5/15/2023 12:00:00 AM	1
2	Sara Khan	sara.khan@example.com	Street 45, Karachi	03219876543	10/10/2022 12:00:00 AM	2
3	Zain Ahmad	zain.ahmad@example.com	Block A, Islamabad	03001112222	8/1/2021 12:00:00 AM	3

Enter your SQL query or T-SQL batch. It will be executed on database: ShaheenaDatabase

```
SELECT * FROM Leaves;
```

Set 1. Total rows: 3

LeaveID	EmpID	LeaveType	LeaveDate	NumDays
1	1	Sick Leave	1/10/2024 12:00:00 AM	2
2	2	Casual Leave	3/5/2024 12:00:00 AM	1
3	3	Annual Leave	4/15/2024 12:00:00 AM	5

Enter your SQL query or T-SQL batch. It will be executed on database: ShaheenaDatabase

```
SELECT * FROM Payments;
```

EXECUTE

Set 1. Total rows: 4

PaymentID	EmpID	Month	Year	BasicSalary	TotalPay
1	1	January	2025	50000.00	55000.00
2	1	February	2025	50000.00	55000.00
3	2	January	2025	60000.00	65000.00
4	3	January	2025	70000.00	75000.00