**Server-Side Development and Databases - Project**

**Introduction**

Database management systems (DBMS) are critical components in modern server-side development, facilitating the efficient storage, retrieval, and management of data. Relational Database Management Systems (RDBMS), such as MySQL, are widely used for their robustness, scalability, and adherence to relational data modeling principles. This project aims to design and implement a database schema while addressing server-side development requirements through systematic processes. The project explores database schema creation, Entity-Relationship (E-R) modeling, and table creation with necessary constraints.

**Requirements**

1. **Database Creation:**
   * Develop a relational database in MySQL.
   * Ensure compatibility with the schema provided.
2. **Schema Design and Table Creation:**
   * Implement tables based on the schema using SQL.
   * Define appropriate constraints such as primary keys, foreign keys, unique keys, and default values.
3. **E-R Modeling:**
   * Design an Entity-Relationship (E-R) model to visualize the schema.
   * Map relationships among entities and attributes.
4. **Tools and Software:**
   * MySQL (Workbench or CLI)
   * Diagramming tools (e.g., Lucidchart, Microsoft Visio, or dbdiagram.io) for E-R modeling.
5. **Key Constraints and Features:**
   * Referential integrity.
   * Normalization to avoid redundancy.
   * Support for transactions and error handling in SQL.

**Process of Project Development:**

**Step 1: Database Schema Analysis**

* Study the given schema to understand its entities, attributes, and relationships.
* Identify primary keys, foreign keys, and constraints.

**Step 2: Database Creation in MySQL**

* Use CREATE DATABASE to set up the relational database.
* Ensure proper configuration and naming conventions.

**SQL Example:**

**CREATE DATABASE ProjectDB;**

**USE ProjectDB;**

**Step 3: Table Creation with Constraints**

* Create tables based on the schema using SQL commands.
* Add constraints to ensure data integrity.

**SQL Example:**

**CREATE TABLE Employees (**

**EmployeeID INT PRIMARY KEY,**

**Name VARCHAR(100) NOT NULL,**

**DepartmentID INT,**

**Salary DECIMAL(10,2),**

**FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)**

**);**

**Step 4: E-R Model Creation:**

* Design the E-R model to represent entities, attributes, and relationships.
* Use diagramming tools for clear visualization.

**Example:**

* Entities: Employees, Departments, Projects
* Relationships: One-to-Many (Departments to Employees), Many-to-Many (Employees to Projects)

**Step 5: Validation and Testing**

* Insert sample data to test the schema and relationships.
* Validate constraints to ensure data consistency.

**SQL Example:**

**INSERT INTO Departments (DepartmentID, DepartmentName) VALUES (1, 'IT');**

**INSERT INTO Employees (EmployeeID, Name, DepartmentID, Salary) VALUES (101, 'Alice', 1, 50000.00);**

**MAJOR PROJECT DEVELOPMENT:**

**1. Inventory system;**

code:

-- Create the database

CREATE DATABASE CompanyDBS;

USE CompanyDBS;

-- Table: location

CREATE TABLE location (

ID INT PRIMARY KEY,

PROVINCE VARCHAR(50),

CITY VARCHAR(50),

STREET VARCHAR(100)

);

-- Table: job

CREATE TABLE job (

ID INT PRIMARY KEY,

JOB\_TITLE VARCHAR(50),

SALARY DECIMAL(10, 2)

);

-- Table: category

CREATE TABLE category (

ID INT PRIMARY KEY,

NAME VARCHAR(50),

DESCRIPTION VARCHAR(100)

);

-- Table: product

CREATE TABLE product (

ID INT PRIMARY KEY,

NAME VARCHAR(50),

DESCRIPTION VARCHAR(100),

QTY\_STOCK INT,

PRICE DECIMAL(10, 2),

CATEGORY\_ID INT,

FOREIGN KEY (CATEGORY\_ID) REFERENCES category(ID)

);

-- Table: supplier

CREATE TABLE supplier (

ID INT PRIMARY KEY,

COMPANY\_NAME VARCHAR(100),

LOCATION\_ID INT,

PHONE\_NUMBER VARCHAR(15),

FOREIGN KEY (LOCATION\_ID) REFERENCES location(ID)

);

-- Table: employees

CREATE TABLE employees (

ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

EMAIL VARCHAR(100),

PHONE\_NUMBER VARCHAR(15),

JOB\_ID INT,

HIRED\_DATE DATE,

LOCATION\_ID INT,

FOREIGN KEY (JOB\_ID) REFERENCES job(ID),

FOREIGN KEY (LOCATION\_ID) REFERENCES location(ID)

);

-- Table: customer

CREATE TABLE customer (

ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

PHONE\_NUMBER VARCHAR(15),

EMPLOYEE\_ID INT,

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES employees(ID)

);

-- Table: manager

CREATE TABLE manager (

ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

LOCATION\_ID INT,

EMAIL VARCHAR(100),

PHONE\_NUMBER VARCHAR(15),

FOREIGN KEY (LOCATION\_ID) REFERENCES location(ID)

);

-- Table: type

CREATE TABLE type (

ID INT PRIMARY KEY,

TYPE VARCHAR(50)

);

-- Table: users

CREATE TABLE users (

ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

USERNAME VARCHAR(50),

PASSWORD VARCHAR(100),

TYPE\_ID INT,

LOCATION\_ID INT,

PHONE\_NUMBER VARCHAR(15),

FOREIGN KEY (TYPE\_ID) REFERENCES type(ID),

FOREIGN KEY (LOCATION\_ID) REFERENCES location(ID)

);

-- Insert sample data

-- Insert into location

INSERT INTO location (ID, PROVINCE, CITY, STREET) VALUES

(1, 'California', 'Los Angeles', 'Main Street'),

(2, 'Texas', 'Houston', 'Elm Street'),

(3, 'New York', 'New York City', 'Broadway');

-- Insert into job

INSERT INTO job (ID, JOB\_TITLE, SALARY) VALUES

(1, 'Software Engineer', 80000.00),

(2, 'Manager', 95000.00),

(3, 'Sales Representative', 60000.00);

-- Insert into category

INSERT INTO category (ID, NAME, DESCRIPTION) VALUES

(1, 'Electronics', 'Electronic gadgets and devices'),

(2, 'Furniture', 'Home and office furniture'),

(3, 'Clothing', 'Apparel and accessories');

-- Insert into employees

INSERT INTO employees (ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, JOB\_ID, HIRED\_DATE, LOCATION\_ID) VALUES

(1, 'John', 'Doe', 'john.doe@example.com', '1234567890', 1, '2023-01-15', 1),

(2, 'Jane', 'Smith', 'jane.smith@example.com', '0987654321', 2, '2022-03-20', 2),

(3, 'Alice', 'Brown', 'alice.brown@example.com', '1122334455', 3, '2021-06-10', 3);

-- Insert into customer

INSERT INTO customer (ID, FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, EMPLOYEE\_ID) VALUES

(1, 'Tom', 'Hanks', '1231231234', 1),

(2, 'Emma', 'Watson', '3213214321', 2),

(3, 'Chris', 'Hemsworth', '4566547890', 3);

-- Insert into manager

INSERT INTO manager (ID, FIRST\_NAME, LAST\_NAME, LOCATION\_ID, EMAIL, PHONE\_NUMBER) VALUES

(1, 'Michael', 'Scott', 1, 'michael.scott@example.com', '5555551234'),

(2, 'Jim', 'Halpert', 2, 'jim.halpert@example.com', '5555552345'),

(3, 'Pam', 'Beesly', 3, 'pam.beesly@example.com', '5555553456');

-- Insert into type

INSERT INTO type (ID, TYPE) VALUES

(1, 'Admin'),

(2, 'Employee'),

(3, 'Customer');

-- Insert into users

INSERT INTO users (ID, FIRST\_NAME, LAST\_NAME, USERNAME, PASSWORD, TYPE\_ID, LOCATION\_ID, PHONE\_NUMBER) VALUES

(1, 'Tom', 'Hardy', 'thardy', 'password123', 1, 1, '5551112233'),

(2, 'Emma', 'Stone', 'estone', 'password456', 2, 2, '5552223344'),

(3, 'Chris', 'Evans', 'cevans', 'password789', 3, 3, '5553334455');

-- Display data from tables

SELECT \* FROM location;

SELECT \* FROM job;

SELECT \* FROM category;

SELECT \* FROM employees;

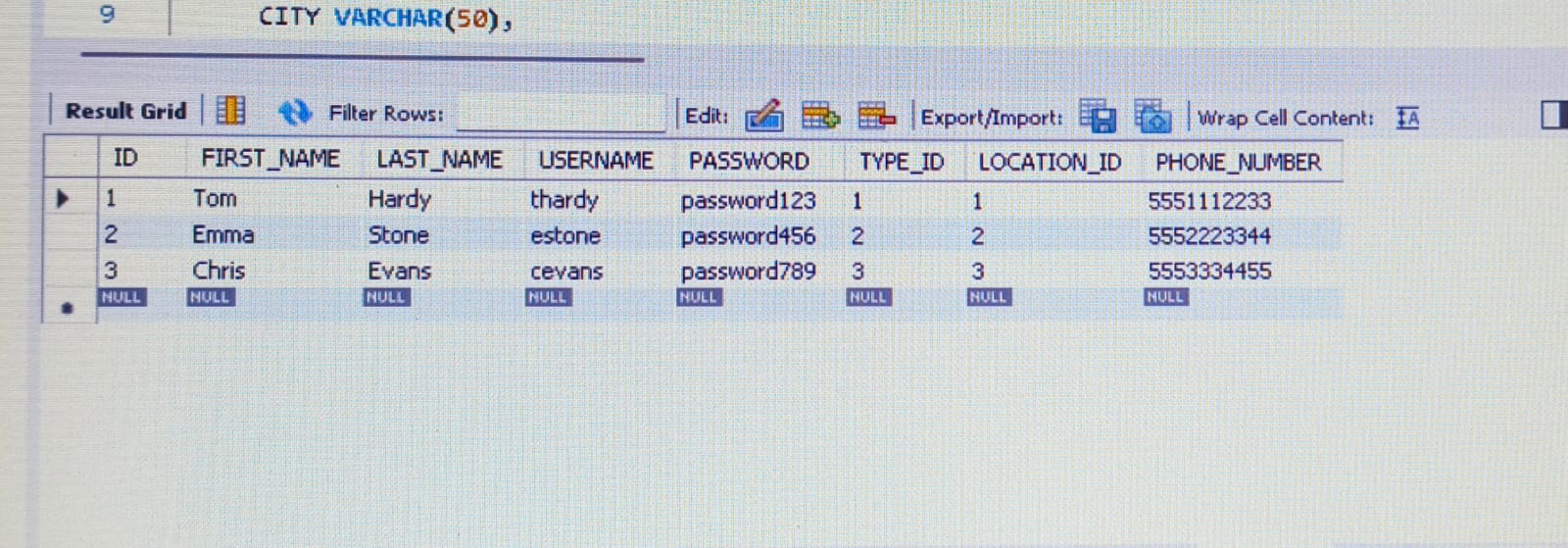
SELECT \* FROM customer;

SELECT \* FROM manager;

SELECT \* FROM type;

SELECT \* FROM users;

**OUTPUT:**

****

**2.Water refilling station system:**

-- Creating the tables for the schema

CREATE DATABASE delivery\_systems;

USE delivery\_system;

-- Creating the tables for the schema

CREATE TABLE employee (

EMPLOYEE\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

LOCATION VARCHAR(100),

AGE INT,

DATE DATE,

TIME TIME,

CONTACT\_NO VARCHAR(15)

);

CREATE TABLE customer (

CUSTOMER\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

ADDRESS VARCHAR(100),

AGE INT,

DATE DATE,

TIME TIME

);

CREATE TABLE types\_of\_container (

CONTAINER\_ID INT PRIMARY KEY,

CONTAINERS VARCHAR(50),

QUANTITY INT,

ON\_HAND INT,

UNIT\_PRICE DECIMAL(10, 2)

);

CREATE TABLE payments (

PAYMENT\_ID INT PRIMARY KEY,

CUSTOMER\_ID INT,

DATE\_ORDER DATE,

DATE\_CLAIM DATE,

QUANTITY INT,

UNIT\_PRICE DECIMAL(10, 2),

FOREIGN KEY (CUSTOMER\_ID) REFERENCES customer(CUSTOMER\_ID)

);

CREATE TABLE product (

PRODUCT\_ID INT PRIMARY KEY,

PRODUCT\_NAME VARCHAR(50),

DESCRIPTION VARCHAR(200),

QUANTITY INT,

ON\_HAND INT,

UNIT\_PRICE DECIMAL(10, 2)

);

CREATE TABLE delivery (

DELIVERY\_ID INT PRIMARY KEY,

CUSTOMER\_ID INT,

PRODUCT\_ID INT,

CONTAINER\_ID INT,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

ADDRESS VARCHAR(100),

DATE DATE,

TIME TIME,

CONTAINERS VARCHAR(50),

PRODUCT\_NAME VARCHAR(50),

FOREIGN KEY (CUSTOMER\_ID) REFERENCES customer(CUSTOMER\_ID),

FOREIGN KEY (PRODUCT\_ID) REFERENCES product(PRODUCT\_ID),

FOREIGN KEY (CONTAINER\_ID) REFERENCES types\_of\_container(CONTAINER\_ID)

);

CREATE TABLE transaction (

TRANSACTION\_ID INT PRIMARY KEY,

CUSTOMER\_ID INT,

DELIVERY\_ID INT,

PRODUCT\_ID INT,

CONTAINER\_ID INT,

EMPLOYEE\_ID INT,

FOREIGN KEY (CUSTOMER\_ID) REFERENCES customer(CUSTOMER\_ID),

FOREIGN KEY (DELIVERY\_ID) REFERENCES delivery(DELIVERY\_ID),

FOREIGN KEY (PRODUCT\_ID) REFERENCES product(PRODUCT\_ID),

FOREIGN KEY (CONTAINER\_ID) REFERENCES types\_of\_container(CONTAINER\_ID),

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES employee(EMPLOYEE\_ID)

);

-- DESC commands to display the schema for each table (MySQL-specific)

DESC employee;

DESC types\_of\_container;

DESC payments;

DESC customer;

DESC product;

DESC delivery;

DESC transaction;

-- Inserting 3 sample values into each table

INSERT INTO employee VALUES (1, 'Alice', 'Smith', 'New York', 30, '2024-01-01', '08:00:00', '1234567890');

INSERT INTO employee VALUES (2, 'Bob', 'Johnson', 'Los Angeles', 35, '2024-01-02', '09:00:00', '0987654321');

INSERT INTO employee VALUES (3, 'Charlie', 'Brown', 'Chicago', 28, '2024-01-03', '10:00:00', '5678901234');

INSERT INTO types\_of\_container VALUES (1, 'Large Box', 50, 10, 25.50);

INSERT INTO types\_of\_container VALUES (2, 'Medium Box', 100, 20, 15.00);

INSERT INTO types\_of\_container VALUES (3, 'Small Box', 200, 50, 5.00);

INSERT INTO customer VALUES (1, 'John', 'Doe', '123 Elm St', 45, '2024-01-01', '08:00:00');

INSERT INTO customer VALUES (2, 'Jane', 'Smith', '456 Oak St', 32, '2024-01-02', '09:00:00');

INSERT INTO customer VALUES (3, 'Emily', 'Johnson', '789 Pine St', 28, '2024-01-03', '10:00:00');

INSERT INTO product VALUES (1, 'Laptop', 'High-performance laptop', 10, 2, 1200.00);

INSERT INTO product VALUES (2, 'Office Chair', 'Ergonomic office chair', 50, 10, 150.00);

INSERT INTO product VALUES (3, 'Rice', 'Premium quality rice', 100, 30, 20.00);

INSERT INTO delivery VALUES (1, 1, 1, 1, 'John', 'Doe', '123 Elm St', '2024-01-01', '08:00:00', 'Large Box', 'Laptop');

INSERT INTO delivery VALUES (2, 2, 2, 2, 'Jane', 'Smith', '456 Oak St', '2024-01-02', '09:00:00', 'Medium Box', 'Office Chair');

INSERT INTO delivery VALUES (3, 3, 3, 3, 'Emily', 'Johnson', '789 Pine St', '2024-01-03', '10:00:00', 'Small Box', 'Rice');

INSERT INTO transaction VALUES (1, 1, 1, 1, 1, 1);

INSERT INTO transaction VALUES (2, 2, 2, 2, 2, 2);

INSERT INTO transaction VALUES (3, 3, 3, 3, 3, 3);

-- Displaying the data in the tables

SELECT \* FROM employee;

SELECT \* FROM types\_of\_container;

SELECT \* FROM payments;

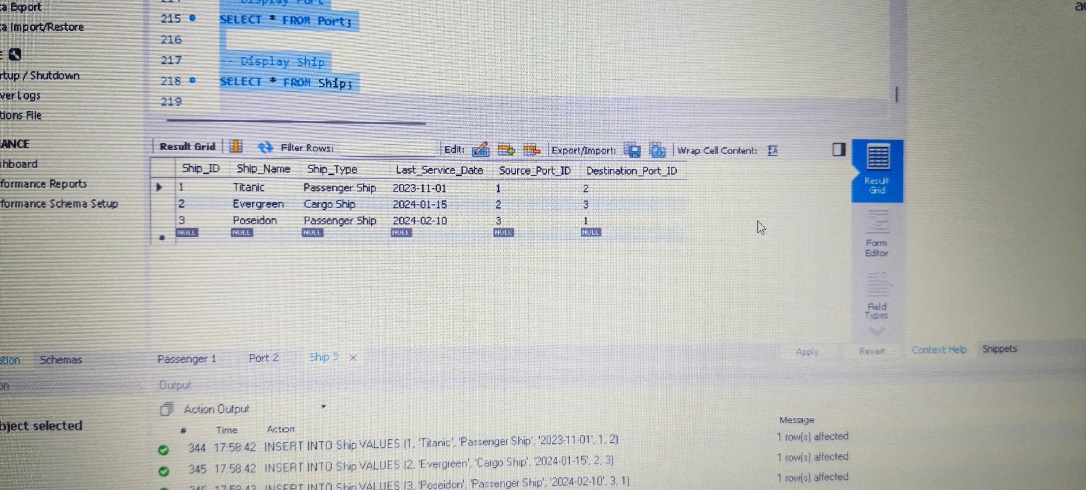
SELECT \* FROM customer;

SELECT \* FROM product;

SELECT \* FROM delivery;

SELECT \* FROM transaction;

**OUTPUT:**

****

**3.Food ordering system:**

-- Create the database

CREATE DATABASE FoodOrderingSystem;

USE FoodOrderingSystem;

-- Create the tblEmployee table

CREATE TABLE tblEmployee (

Employee\_Id INT PRIMARY KEY,

First\_Name VARCHAR(50),

Middle\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

Age INT,

Address VARCHAR(100),

Contact\_Number VARCHAR(15)

);

-- Create the tblSupplier table

CREATE TABLE tblSupplier (

Supplier\_Id INT PRIMARY KEY,

Supplier\_Name VARCHAR(50),

Supplier\_Address VARCHAR(100),

Contact\_Number VARCHAR(15),

Email\_Address VARCHAR(50)

);

-- Create the tblProducts table

CREATE TABLE tblProducts (

Products\_Id INT PRIMARY KEY,

Products\_Name VARCHAR(50),

Products\_Type VARCHAR(50),

Description VARCHAR(100),

On\_Hand INT,

Price DECIMAL(10, 2)

);

-- Create the tblCustomers table

CREATE TABLE tblCustomers (

Customer\_Id INT PRIMARY KEY,

First\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

City\_Address VARCHAR(100),

Age INT,

Phone\_Number VARCHAR(15),

Zipcode VARCHAR(10)

);

-- Create the tblPayments table

CREATE TABLE tblPayments (

Payment\_Id INT PRIMARY KEY,

Order\_Number INT,

Customer\_Id INT,

Date\_Order DATE,

Claim\_Date DATE,

Payment\_Method VARCHAR(50),

Quantity INT,

Price DECIMAL(10, 2),

FOREIGN KEY (Customer\_Id) REFERENCES tblCustomers(Customer\_Id)

);

-- Create the tblOrder table

CREATE TABLE tblOrder (

Order\_Id INT PRIMARY KEY,

Payment\_Id INT,

Customer\_Id INT,

Date\_Order DATE,

Quantity INT,

Price DECIMAL(10, 2),

FOREIGN KEY (Payment\_Id) REFERENCES tblPayments(Payment\_Id),

FOREIGN KEY (Customer\_Id) REFERENCES tblCustomers(Customer\_Id)

);

-- Create the tblTransaction table

CREATE TABLE tblTransaction (

Transaction\_Id INT PRIMARY KEY,

Employee\_Id INT,

Customer\_Id INT,

Order\_Id INT,

Products\_Id INT,

Supplier\_Id INT,

Payment\_Id INT,

FOREIGN KEY (Employee\_Id) REFERENCES tblEmployee(Employee\_Id),

FOREIGN KEY (Customer\_Id) REFERENCES tblCustomers(Customer\_Id),

FOREIGN KEY (Order\_Id) REFERENCES tblOrder(Order\_Id),

FOREIGN KEY (Products\_Id) REFERENCES tblProducts(Products\_Id),

FOREIGN KEY (Supplier\_Id) REFERENCES tblSupplier(Supplier\_Id),

FOREIGN KEY (Payment\_Id) REFERENCES tblPayments(Payment\_Id)

);

-- Insert sample data into tblEmployee

INSERT INTO tblEmployee (Employee\_Id, First\_Name, Middle\_Name, Last\_Name, Age, Address, Contact\_Number) VALUES

(1, 'John', 'A', 'Doe', 30, '123 Elm Street', '5551234567'),

(2, 'Jane', 'B', 'Smith', 25, '456 Oak Avenue', '5552345678'),

(3, 'Mark', 'C', 'Johnson', 35, '789 Pine Road', '5553456789');

-- Insert sample data into tblSupplier

INSERT INTO tblSupplier (Supplier\_Id, Supplier\_Name, Supplier\_Address, Contact\_Number, Email\_Address) VALUES

(1, 'Fresh Produce Inc.', '101 Market Lane', '5559876543', 'info@freshproduce.com'),

(2, 'Bakers Unlimited', '202 Bread Street', '5558765432', 'support@bakers.com'),

(3, 'Meat and More', '303 Grill Avenue', '5557654321', 'sales@meatandmore.com');

-- Insert sample data into tblProducts

INSERT INTO tblProducts (Products\_Id, Products\_Name, Products\_Type, Description, On\_Hand, Price) VALUES

(1, 'Apple', 'Fruit', 'Fresh red apples', 100, 0.50),

(2, 'Bread', 'Bakery', 'Whole wheat bread', 50, 1.25),

(3, 'Chicken', 'Meat', 'Boneless chicken breast', 30, 5.00);

-- Insert sample data into tblCustomers

INSERT INTO tblCustomers (Customer\_Id, First\_Name, Last\_Name, City\_Address, Age, Phone\_Number, Zipcode) VALUES

(1, 'Alice', 'Green', '123 City Road', 28, '5551234567', '10001'),

(2, 'Bob', 'Brown', '456 Town Avenue', 35, '5552345678', '10002'),

(3, 'Charlie', 'White', '789 Country Lane', 40, '5553456789', '10003');

-- Insert sample data into tblPayments

INSERT INTO tblPayments (Payment\_Id, Order\_Number, Customer\_Id, Date\_Order, Claim\_Date, Payment\_Method, Quantity, Price) VALUES

(1, 101, 1, '2024-11-01', '2024-11-03', 'Credit Card', 5, 10.00),

(2, 102, 2, '2024-11-05', '2024-11-07', 'Cash', 3, 15.00),

(3, 103, 3, '2024-11-10', '2024-11-12', 'Debit Card', 8, 40.00);

-- Insert sample data into tblOrder

INSERT INTO tblOrder (Order\_Id, Payment\_Id, Customer\_Id, Date\_Order, Quantity, Price) VALUES

(1, 1, 1, '2024-11-01', 5, 10.00),

(2, 2, 2, '2024-11-05', 3, 15.00),

(3, 3, 3, '2024-11-10', 8, 40.00);

-- Insert sample data into tblTransaction

INSERT INTO tblTransaction (Transaction\_Id, Employee\_Id, Customer\_Id, Order\_Id, Products\_Id, Supplier\_Id, Payment\_Id) VALUES

(1, 1, 1, 1, 1, 1, 1),

(2, 2, 2, 2, 2, 2, 2),

(3, 3, 3, 3, 3, 3, 3);

-- Display data from the tables

SELECT \* FROM tblEmployee;

SELECT \* FROM tblSupplier;

SELECT \* FROM tblProducts;

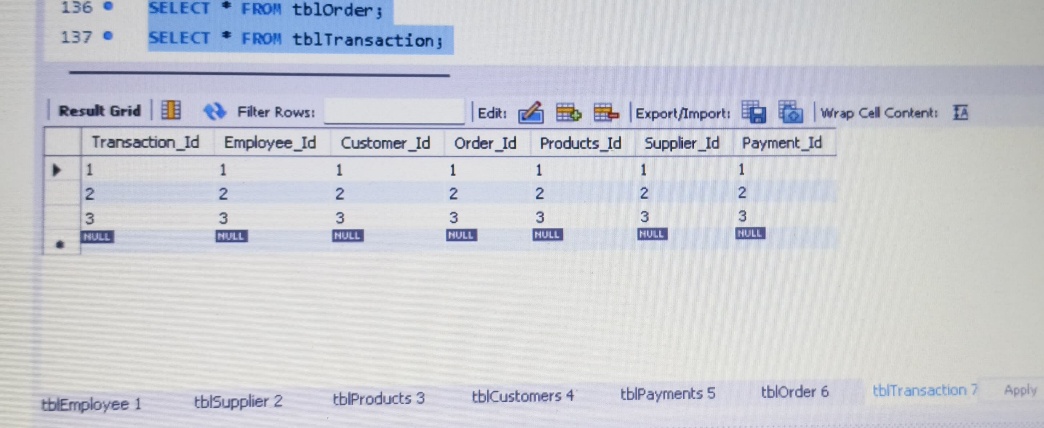
SELECT \* FROM tblCustomers;

SELECT \* FROM tblPayments;

SELECT \* FROM tblOrder;

SELECT \* FROM tblTransaction;

**OUTPUT:**

****

**4. Hotel reservation :**

-- Step 1: Create the database

CREATE DATABASE HotelManagement;

USE HotelManagement;

-- Step 2: Create tables based on the schema

-- Table: tblcashier

CREATE TABLE tblcashier (

C\_Id INT PRIMARY KEY,

Employee\_Id INT NOT NULL,

First\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

Address VARCHAR(100),

Phone\_no VARCHAR(15),

Email VARCHAR(50)

);

-- Table: tblemployee

CREATE TABLE tblemployee (

employee\_Id INT PRIMARY KEY,

employee\_Fname VARCHAR(50),

employee\_Lname VARCHAR(50),

employee\_Address VARCHAR(100),

Position VARCHAR(50),

Hire\_Date DATE,

salary DECIMAL(10, 2)

);

-- Table: tblcustomer

CREATE TABLE tblcustomer (

cust\_Id INT PRIMARY KEY,

cust\_Fname VARCHAR(50),

cust\_Lname VARCHAR(50),

cust\_Address VARCHAR(100),

cust\_Phone VARCHAR(15)

);

-- Table: tblreservation

CREATE TABLE tblreservation (

res\_Id INT PRIMARY KEY,

res\_Fname VARCHAR(50),

res\_Lname VARCHAR(50),

res\_Address VARCHAR(100),

res\_Contactno VARCHAR(15),

res\_event VARCHAR(50),

no\_of\_person INT,

res\_date DATE,

res\_time TIME

);

-- Table: tblpayments

CREATE TABLE tblpayments (

C\_Id INT,

name\_of\_customer VARCHAR(50),

name\_of\_cashier VARCHAR(50),

total\_amount DECIMAL(10, 2),

FOREIGN KEY (C\_Id) REFERENCES tblcashier(C\_Id)

);

-- Step 3: Describe the tables

DESC tblcashier;

DESC tblemployee;

DESC tblcustomer;

DESC tblreservation;

DESC tblpayments;

-- Step 4: Insert 3 records into each table

-- Insert records into tblcashier

INSERT INTO tblcashier VALUES

(1, 101, 'John', 'Doe', '123 Main St', '1234567890', 'john.doe@example.com'),

(2, 102, 'Jane', 'Smith', '456 Elm St', '0987654321', 'jane.smith@example.com'),

(3, 103, 'Alice', 'Johnson', '789 Oak St', '1122334455', 'alice.johnson@example.com');

-- Insert records into tblemployee

INSERT INTO tblemployee VALUES

(101, 'John', 'Doe', '123 Main St', 'Manager', '2023-01-01', 5000.00),

(102, 'Jane', 'Smith', '456 Elm St', 'Supervisor', '2023-02-01', 4000.00),

(103, 'Alice', 'Johnson', '789 Oak St', 'Clerk', '2023-03-01', 3000.00);

-- Insert records into tblcustomer

INSERT INTO tblcustomer VALUES

(1, 'Michael', 'Brown', '123 Pine St', '5551234567'),

(2, 'Sarah', 'Davis', '456 Maple St', '5559876543'),

(3, 'Emily', 'Clark', '789 Birch St', '5551122334');

-- Insert records into tblreservation

INSERT INTO tblreservation VALUES

(1, 'Michael', 'Brown', '123 Pine St', '5551234567', 'Wedding', 100, '2023-12-25', '18:00:00'),

(2, 'Sarah', 'Davis', '456 Maple St', '5559876543', 'Conference', 50, '2023-12-26', '09:00:00'),

(3, 'Emily', 'Clark', '789 Birch St', '5551122334', 'Birthday', 30, '2023-12-27', '13:00:00');

-- Insert records into tblpayments

INSERT INTO tblpayments VALUES

(1, 'Michael Brown', 'John Doe', 5000.00),

(2, 'Sarah Davis', 'Jane Smith', 2500.00),

(3, 'Emily Clark', 'Alice Johnson', 1500.00);

-- Step 5: Display the tables

SELECT \* FROM tblcashier;

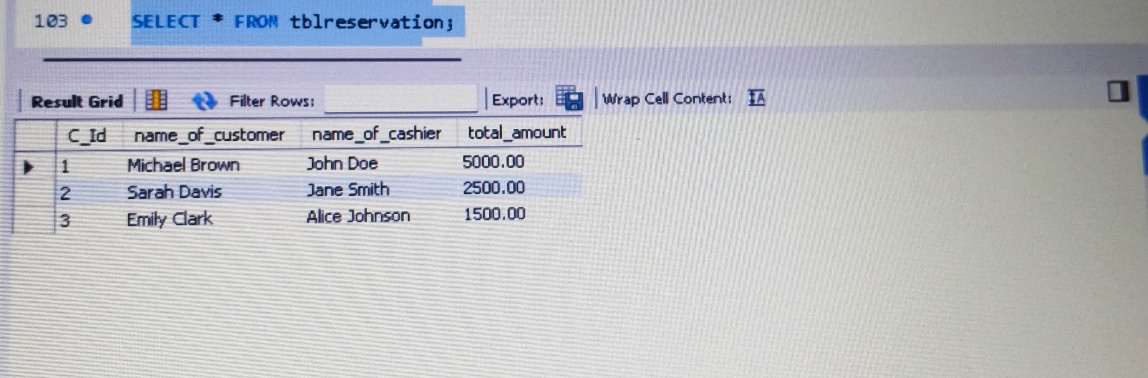
SELECT \* FROM tblemployee;

SELECT \* FROM tblcustomer;

SELECT \* FROM tblreservation;

SELECT \* FROM tblpayments;

**OUTPUT**:



**5.Supply Management system:**

- Step 1: Create the database

CREATE DATABASE SupplyManagementSystem;

USE SupplyManagementSystem;

-- Step 2: Create tables based on the schema

-- Table: Customer

CREATE TABLE Customer (

Customer\_ID INT PRIMARY KEY,

First\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

City\_Address VARCHAR(100)

);

-- Table: Products

CREATE TABLE Products (

Product\_ID INT PRIMARY KEY,

Product\_Name VARCHAR(50),

Price DECIMAL(10, 2)

);

-- Table: Employee

CREATE TABLE Employee (

Employee\_ID INT PRIMARY KEY,

First\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

Salary DECIMAL(10, 2),

Position VARCHAR(50),

Hire\_Date DATE

);

-- Table: Supplier

CREATE TABLE Supplier (

Supplier\_ID INT PRIMARY KEY,

Supplier\_Name VARCHAR(50),

Supplier\_Address VARCHAR(100)

);

-- Table: Transactions

CREATE TABLE Transactions (

Transaction\_ID INT PRIMARY KEY,

Customer\_ID INT,

Employee\_ID INT,

Product\_ID INT,

Supplier\_ID INT,

Quantity INT,

Price DECIMAL(10, 2),

Total\_Price DECIMAL(10, 2),

Date DATE,

FOREIGN KEY (Customer\_ID) REFERENCES Customer(Customer\_ID),

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID),

FOREIGN KEY (Product\_ID) REFERENCES Products(Product\_ID),

FOREIGN KEY (Supplier\_ID) REFERENCES Supplier(Supplier\_ID)

);

-- Step 3: Describe the tables

DESC Customer;

DESC Products;

DESC Employee;

DESC Supplier;

DESC Transactions;

-- Step 4: Insert 3 records into each table

-- Insert records into Customer

INSERT INTO Customer VALUES

(1, 'John', 'Doe', '123 Main St'),

(2, 'Jane', 'Smith', '456 Elm St'),

(3, 'Alice', 'Johnson', '789 Oak St');

-- Insert records into Products

INSERT INTO Products VALUES

(101, 'Laptop', 800.00),

(102, 'Smartphone', 600.00),

(103, 'Tablet', 300.00);

-- Insert records into Employee

INSERT INTO Employee VALUES

(201, 'Michael', 'Brown', 5000.00, 'Manager', '2023-01-01'),

(202, 'Sarah', 'Davis', 4000.00, 'Supervisor', '2023-02-01'),

(203, 'Emily', 'Clark', 3000.00, 'Clerk', '2023-03-01');

-- Insert records into Supplier

INSERT INTO Supplier VALUES

(301, 'TechSupply Co.', '456 Tech Park'),

(302, 'GadgetHub', '789 Silicon Ave'),

(303, 'DeviceWorld', '123 Innovation St');

-- Insert records into Transactions

INSERT INTO Transactions VALUES

(401, 1, 201, 101, 301, 2, 800.00, 1600.00, '2023-11-01'),

(402, 2, 202, 102, 302, 3, 600.00, 1800.00, '2023-11-02'),

(403, 3, 203, 103, 303, 1, 300.00, 300.00, '2023-11-03');

-- Step 5: Display the tables

SELECT \* FROM Customer;

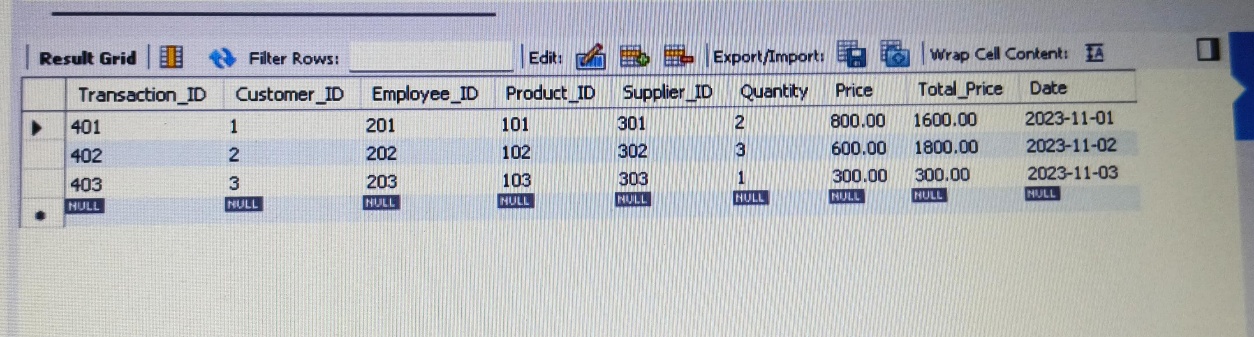
SELECT \* FROM Products;

SELECT \* FROM Employee;

SELECT \* FROM Supplier;

SELECT \* FROM Transactions;

**OUTPUT:**

****

1. **Payroll management system:**

- Step 1: Create a database

CREATE DATABASE PayrollManagementSystem;

-- Step 2: Use the database

USE PayrollManagementSystem;

-- Step 3: Create tables

-- Create the User table

CREATE TABLE User (

Employee\_ID INT PRIMARY KEY,

Last\_name VARCHAR(50),

First\_name VARCHAR(50),

Address VARCHAR(100),

Contact\_number VARCHAR(15),

Gender CHAR(1),

Email VARCHAR(100)

);

-- Create the Product table

CREATE TABLE Product (

Product\_id INT PRIMARY KEY,

Product\_name VARCHAR(50),

Product\_Price DECIMAL(10, 2)

);

-- Create the Customer table

CREATE TABLE Customer (

Customer\_id INT PRIMARY KEY,

Customer\_name VARCHAR(50),

Customer\_no VARCHAR(15),

Customer\_address VARCHAR(100)

);

-- Create the Income table

CREATE TABLE Income (

User\_id INT,

Prod\_id INT,

Price DECIMAL(10, 2),

Quantity INT,

Tax DECIMAL(10, 2),

Total\_price DECIMAL(10, 2),

Date DATE,

FOREIGN KEY (User\_id) REFERENCES User(Employee\_ID),

FOREIGN KEY (Prod\_id) REFERENCES Product(Product\_id)

);

-- Create the Transaction table

CREATE TABLE Transaction (

Customer\_id INT,

Employee\_id INT,

Product INT,

Quantity INT,

Price DECIMAL(10, 2),

Total DECIMAL(10, 2),

Date DATE,

FOREIGN KEY (Customer\_id) REFERENCES Customer(Customer\_id),

FOREIGN KEY (Employee\_id) REFERENCES User(Employee\_ID),

FOREIGN KEY (Product) REFERENCES Product(Product\_id)

);

-- Step 4: Insert values into tables

-- Insert values into User

INSERT INTO User VALUES (1, 'Smith', 'John', '123 Main St', '1234567890', 'M', 'john.smith@example.com');

INSERT INTO User VALUES (2, 'Doe', 'Jane', '456 Elm St', '0987654321', 'F', 'jane.doe@example.com');

INSERT INTO User VALUES (3, 'Brown', 'Charlie', '789 Oak St', '1122334455', 'M', 'charlie.brown@example.com');

-- Insert values into Product

INSERT INTO Product VALUES (101, 'Laptop', 1200.50);

INSERT INTO Product VALUES (102, 'Smartphone', 800.00);

INSERT INTO Product VALUES (103, 'Tablet', 450.25);

-- Insert values into Customer

INSERT INTO Customer VALUES (201, 'Alice', '5551234567', '12 Willow St');

INSERT INTO Customer VALUES (202, 'Bob', '5559876543', '34 Pine St');

INSERT INTO Customer VALUES (203, 'Eve', '5555678901', '56 Maple St');

-- Insert values into Income

INSERT INTO Income VALUES (1, 101, 1200.50, 2, 100.00, 2501.00, '2024-11-25');

INSERT INTO Income VALUES (2, 102, 800.00, 1, 80.00, 880.00, '2024-11-26');

INSERT INTO Income VALUES (3, 103, 450.25, 3, 135.00, 1485.75, '2024-11-27');

-- Insert values into Transaction

INSERT INTO Transaction VALUES (201, 1, 101, 1, 1200.50, 1200.50, '2024-11-25');

INSERT INTO Transaction VALUES (202, 2, 102, 2, 800.00, 1600.00, '2024-11-26');

INSERT INTO Transaction VALUES (203, 3, 103, 1, 450.25, 450.25, '2024-11-27');

-- Step 5: Display the tables

-- Display the contents of each table

SELECT \* FROM User;

SELECT \* FROM Product;

SELECT \* FROM Customer;

SELECT \* FROM Income;

SELECT \* FROM Transaction;

-- Step 6: Describe the tables

-- Describe each table to verify structure

DESC User;

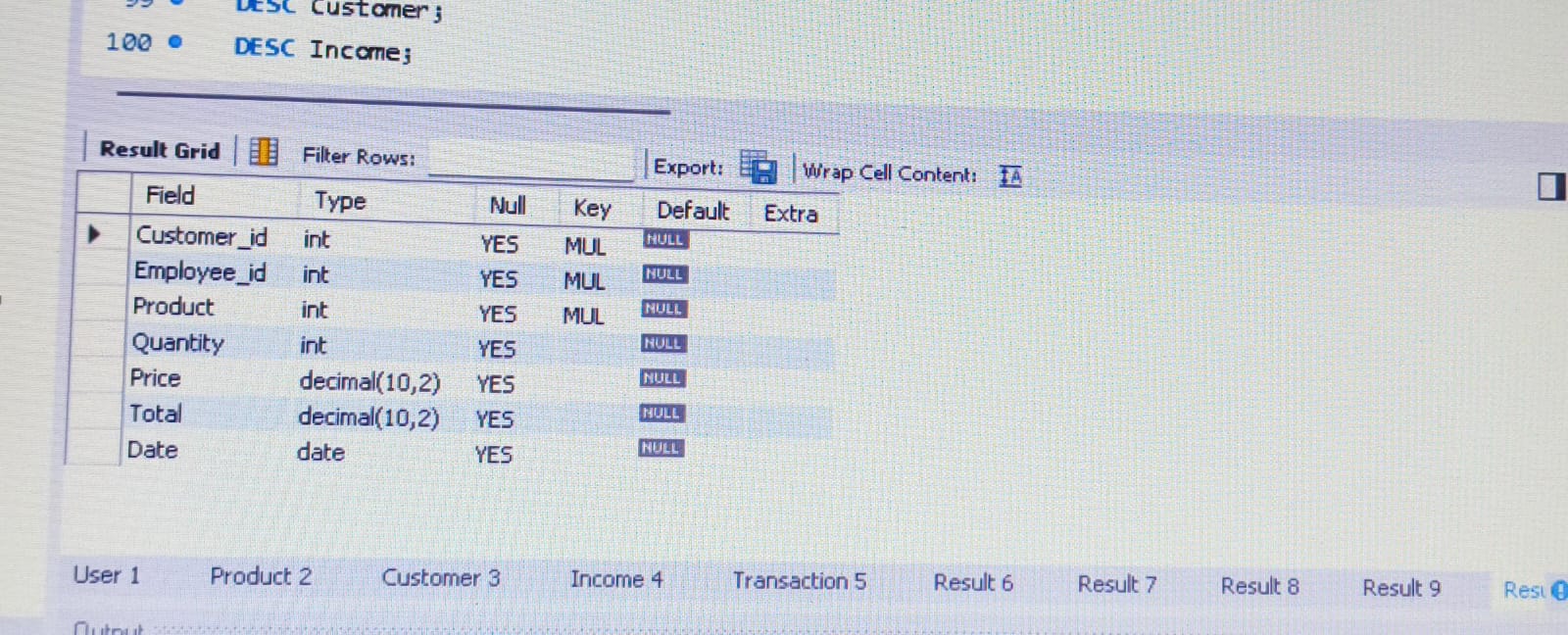
DESC Product;

DESC Customer;

DESC Income;

DESC Transaction;

**OUTPUT:**

****

1. **Clinic management:**

-- Create Database

CREATE DATABASE ClinicManagement1;

USE ClinicManagement1;

-- Patients Table

CREATE TABLE Patients (

patient\_id INT PRIMARY KEY,

fname VARCHAR(50) NOT NULL,

lname VARCHAR(50) NOT NULL,

patient\_type VARCHAR(20),

age INT,

address VARCHAR(255)

);

-- Treatments Table

CREATE TABLE Treatments (

treat\_id INT PRIMARY KEY,

treat\_type VARCHAR(50) NOT NULL

);

-- Medicines Table

CREATE TABLE Medicines (

med\_id INT PRIMARY KEY,

med\_name VARCHAR(50),

quantity INT,

available\_qty INT,

description VARCHAR(255),

expiry\_date DATE,

requested\_date DATE

);

-- Equipments Table

CREATE TABLE Equipments (

equip\_id INT PRIMARY KEY,

equip\_name VARCHAR(50),

requested\_date DATE,

date\_deleted DATE

);

-- Check\_Up Table (Create this after the referenced tables are defined)

CREATE TABLE Check\_Up (

check\_id INT PRIMARY KEY,

patient\_id INT,

complains VARCHAR(255),

findings VARCHAR(255),

treat\_id INT,

med\_id INT,

quantity INT,

date DATE,

equip\_id INT,

FOREIGN KEY (patient\_id) REFERENCES Patients(patient\_id),

FOREIGN KEY (treat\_id) REFERENCES Treatments(treat\_id),

FOREIGN KEY (med\_id) REFERENCES Medicines(med\_id),

FOREIGN KEY (equip\_id) REFERENCES Equipments(equip\_id)

);

-- Insert Data into Patients Table

INSERT INTO Patients (patient\_id, fname, lname, patient\_type, age, address)

VALUES

(1, 'John', 'Doe', 'Outpatient', 30, '123 Main St'),

(2, 'Jane', 'Smith', 'Inpatient', 45, '456 Elm St'),

(3, 'Sam', 'Wilson', 'Outpatient', 50, '789 Oak St');

-- Insert Data into Treatments Table

INSERT INTO Treatments (treat\_id, treat\_type)

VALUES

(1, 'Surgery'),

(2, 'Physiotherapy'),

(3, 'Medication');

-- Insert Data into Medicines Table

INSERT INTO Medicines (med\_id, med\_name, quantity, available\_qty, description, expiry\_date, requested\_date)

VALUES

(1, 'Paracetamol', 100, 80, 'Pain reliever', '2025-01-01', '2024-10-10'),

(2, 'Ibuprofen', 200, 150, 'Anti-inflammatory', '2026-05-01', '2024-10-15'),

(3, 'Amoxicillin', 300, 250, 'Antibiotic', '2024-12-31', '2024-11-01');

-- Insert Data into Equipments Table

INSERT INTO Equipments (equip\_id, equip\_name, requested\_date, date\_deleted)

VALUES

(1, 'X-Ray Machine', '2024-10-10', NULL),

(2, 'MRI Scanner', '2024-10-20', NULL),

(3, 'Ultrasound', '2024-11-01', NULL);

-- Insert Data into Check\_Up Table

INSERT INTO Check\_Up (check\_id, patient\_id, complains, findings, treat\_id, med\_id, quantity, date, equip\_id)

VALUES

(1, 1, 'Headache', 'Migraine', 3, 1, 2, '2024-11-20', 1),

(2, 2, 'Knee Pain', 'Arthritis', 2, 2, 1, '2024-11-21', 2),

(3, 3, 'Cough', 'Flu', 3, 3, 3, '2024-11-22', 3);

-- Describe Tables

DESC Patients;

DESC Treatments;

DESC Medicines;

DESC Equipments;

DESC Check\_Up;

-- Query Data

SELECT \* FROM Patients;

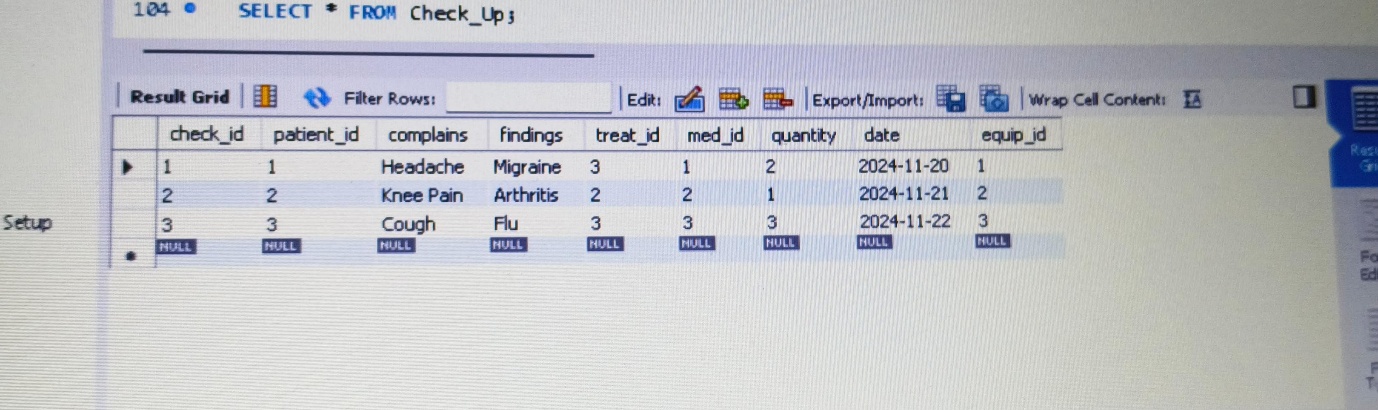
SELECT \* FROM Treatments;

SELECT \* FROM Medicines;

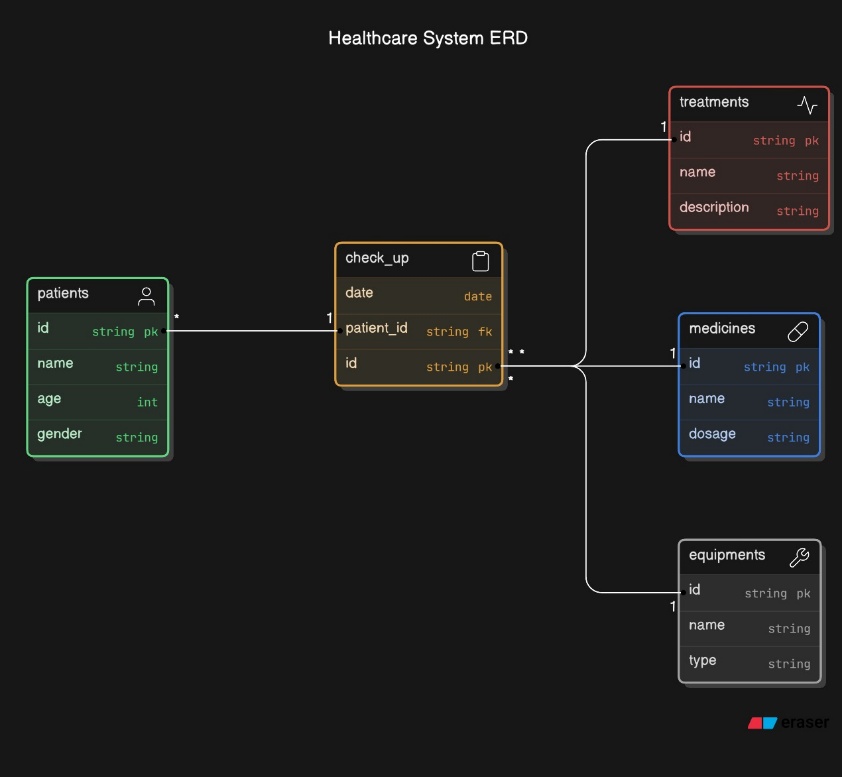
SELECT \* FROM Equipments;

SELECT \* FROM Check\_Up;

**OUTPUT:**



**8.Hospital management:**

****

- Create Database

CREATE DATABASE HospitalManagement1;

USE HospitalManagement1;

-- Create Doctor Table

CREATE TABLE Doctor (

doctor\_id INT PRIMARY KEY,

name VARCHAR(50),

age INT,

gender VARCHAR(10),

address VARCHAR(100)

);

-- Create Patient Table

CREATE TABLE Patient (

patient\_id INT PRIMARY KEY,

name VARCHAR(50),

age INT,

gender VARCHAR(10),

address VARCHAR(100),

disease VARCHAR(50),

doctor\_id INT,

FOREIGN KEY (doctor\_id) REFERENCES Doctor(doctor\_id)

);

-- Create Laboratory Table

CREATE TABLE Laboratory (

lab\_no INT PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

date DATE,

amount DECIMAL(10, 2),

FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES Doctor(doctor\_id)

);

-- Create Room Table

CREATE TABLE Room (

room\_no INT PRIMARY KEY,

room\_type VARCHAR(20),

status VARCHAR(10)

);

-- Create Inpatient Table

CREATE TABLE Inpatient (

patient\_id INT PRIMARY KEY,

room\_no INT,

date\_of\_adm DATE,

date\_of\_dis DATE,

lab\_no INT,

FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id),

FOREIGN KEY (lab\_no) REFERENCES Laboratory(lab\_no)

);

-- Create Outpatient Table

CREATE TABLE Outpatient (

patient\_id INT PRIMARY KEY,

date DATE,

lab\_no INT,

FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id),

FOREIGN KEY (lab\_no) REFERENCES Laboratory(lab\_no)

);

-- Create Bill Table

CREATE TABLE Bill (

bill\_no INT PRIMARY KEY,

patient\_id INT,

doctor\_charge DECIMAL(10, 2),

room\_charge DECIMAL(10, 2),

lab\_charge DECIMAL(10, 2),

no\_of\_days INT,

FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id)

);

-- Insert into Doctor

INSERT INTO Doctor VALUES (1, 'Dr. John', 45, 'Male', '123 Street');

INSERT INTO Doctor VALUES (2, 'Dr. Alice', 38, 'Female', '456 Avenue');

INSERT INTO Doctor VALUES (3, 'Dr. Mark', 50, 'Male', '789 Boulevard');

-- Insert into Patient

INSERT INTO Patient VALUES (1, 'Alice', 30, 'Female', '123 Main St', 'Fever', 1);

INSERT INTO Patient VALUES (2, 'Bob', 40, 'Male', '456 Elm St', 'Injury', 2);

INSERT INTO Patient VALUES (3, 'Eve', 25, 'Female', '789 Oak St', 'Flu', 1);

-- Insert into Laboratory

INSERT INTO Laboratory VALUES (1, 1, 1, '2024-11-20', 500);

INSERT INTO Laboratory VALUES (2, 2, 2, '2024-11-21', 800);

INSERT INTO Laboratory VALUES (3, 3, 3, '2024-11-22', 300);

-- Insert into Room

INSERT INTO Room VALUES (101, 'Deluxe', 'Occupied');

INSERT INTO Room VALUES (102, 'Single', 'Occupied');

INSERT INTO Room VALUES (103, 'Suite', 'Available');

-- Insert into Inpatient

INSERT INTO Inpatient VALUES (1, 101, '2024-11-20', '2024-11-25', 1);

INSERT INTO Inpatient VALUES (2, 102, '2024-11-21', '2024-11-26', 2);

INSERT INTO Inpatient VALUES (3, 103, '2024-11-22', '2024-11-27', 3);

-- Insert into Outpatient (Ensure these patients are different from Inpatient)

INSERT INTO Outpatient VALUES (2, '2024-11-21', 2);

INSERT INTO Outpatient VALUES (3, '2024-11-22', 3);

-- Insert into Bill

INSERT INTO Bill VALUES (1, 1, 2000, 5000, 1000, 5);

INSERT INTO Bill VALUES (2, 2, 1500, 4000, 800, 3);

INSERT INTO Bill VALUES (3, 3, 1800, 4500, 900, 4);

-- Display Table Structures

DESC Doctor;

DESC Patient;

DESC Laboratory;

DESC Room;

DESC Inpatient;

DESC Outpatient;

DESC Bill;

-- Display Data

SELECT \* FROM Patient;

SELECT \* FROM Doctor;

SELECT \* FROM Inpatient;

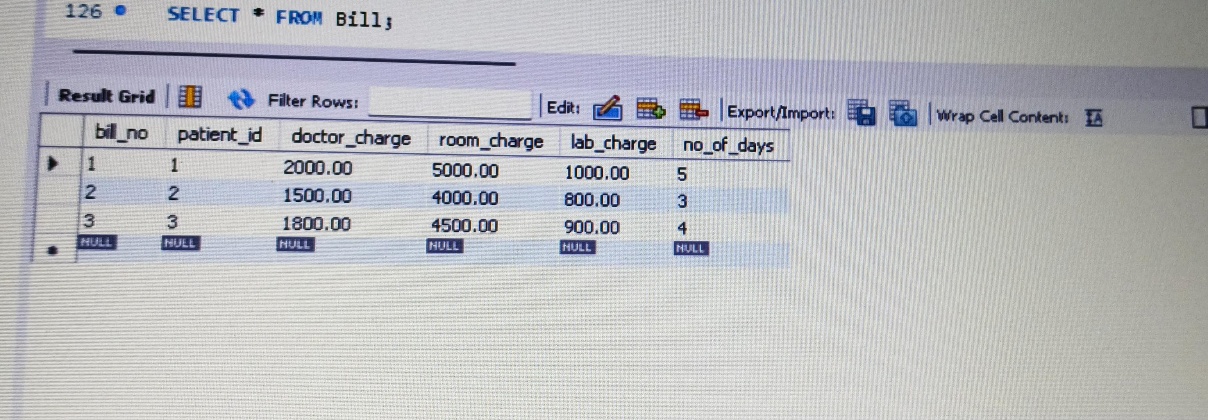
SELECT \* FROM Outpatient;

SELECT \* FROM Laboratory;

SELECT \* FROM Room;

SELECT \* FROM Bill;

**OUTPUT:**

****

**9.Libary management:**

****

-- Create Database

CREATE DATABASE LibraryManagement1;

-- Use the database

USE LibraryManagement1;

-- Create roles table (needed for permission and login)

CREATE TABLE roles (

rol\_id INT PRIMARY KEY,

rol\_name VARCHAR(15),

rol\_lvl VARCHAR(15)

);

-- Create branch table

CREATE TABLE branch (

brn\_id INT PRIMARY KEY,

brn\_name VARCHAR(20),

brn\_ctg VARCHAR(20)

);

-- Create books table

CREATE TABLE books (

book\_id INT PRIMARY KEY,

book\_name VARCHAR(30),

athr\_name VARCHAR(30),

book\_brn\_id INT,

book\_price INT,

FOREIGN KEY (book\_brn\_id) REFERENCES branch(brn\_id)

);

-- Create student table (needed for issues)

CREATE TABLE student (

stu\_id INT PRIMARY KEY,

stu\_name VARCHAR(30),

stu\_ph\_no VARCHAR(15), -- Changed from INT to VARCHAR

stu\_email VARCHAR(30),

stu\_dob DATE

);

-- Create issues table

CREATE TABLE issues (

isu\_id INT PRIMARY KEY,

isu\_date DATE,

isu\_book\_id INT,

isu\_stu\_id INT,

FOREIGN KEY (isu\_book\_id) REFERENCES books(book\_id),

FOREIGN KEY (isu\_stu\_id) REFERENCES student(stu\_id)

);

-- Create permission table

CREATE TABLE permission (

pr\_id INT PRIMARY KEY,

pr\_name VARCHAR(15),

pr\_rol\_id INT,

FOREIGN KEY (pr\_rol\_id) REFERENCES roles(rol\_id)

);

-- Create login table

CREATE TABLE login (

lgn\_id INT PRIMARY KEY,

lgn\_usr\_id INT,

lgn\_usr\_name VARCHAR(20),

lgn\_passz VARCHAR(18),

lgn\_role\_id INT,

FOREIGN KEY (lgn\_role\_id) REFERENCES roles(rol\_id)

);

-- Create user table

CREATE TABLE user (

usr\_id INT PRIMARY KEY,

usr\_name VARCHAR(30),

usr\_ph\_no VARCHAR(15), -- Changed from INT to VARCHAR

usr\_email VARCHAR(30)

);

-- Insert into branch

INSERT INTO branch VALUES (1, 'Central Library', 'Public');

INSERT INTO branch VALUES (2, 'Engineering Library', 'Private');

INSERT INTO branch VALUES (3, 'Medical Library', 'Private');

-- Insert into books

INSERT INTO books VALUES (1, 'Database Systems', 'Navathe', 1, 600);

INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 2, 700);

INSERT INTO books VALUES (3, 'AI Basics', 'Russell', 1, 800);

-- Insert into student

INSERT INTO student VALUES (1, 'John Doe', '1234567890', 'john@example.com', '2000-05-20');

INSERT INTO student VALUES (2, 'Jane Smith', '9876543210', 'jane@example.com', '2001-03-15');

INSERT INTO student VALUES (3, 'Sam Wilson', '4567891230', 'sam@example.com', '1999-11-30');

-- Insert into issues

INSERT INTO issues VALUES (1, '2024-11-01', 1, 1);

INSERT INTO issues VALUES (2, '2024-11-02', 2, 2);

INSERT INTO issues VALUES (3, '2024-11-03', 3, 3);

-- Insert into roles

INSERT INTO roles VALUES (1, 'Admin', 'High');

INSERT INTO roles VALUES (2, 'Librarian', 'Medium');

INSERT INTO roles VALUES (3, 'Assistant', 'Low');

-- Insert into permission

INSERT INTO permission VALUES (1, 'Add Book', 1);

INSERT INTO permission VALUES (2, 'Issue Book', 2);

INSERT INTO permission VALUES (3, 'View Records', 3);

-- Insert into login

INSERT INTO login VALUES (1, 1, 'admin01', 'pass123', 1);

INSERT INTO login VALUES (2, 2, 'librarian01', 'pass456', 2);

INSERT INTO login VALUES (3, 3, 'assistant01', 'pass789', 3);

-- Insert into user

INSERT INTO user VALUES (1, 'Alice', '1234509876', 'alice@example.com');

INSERT INTO user VALUES (2, 'Bob', '9876541230', 'bob@example.com');

INSERT INTO user VALUES (3, 'Charlie', '4561237890', 'charlie@example.com');

-- DESC commands

DESC branch;

DESC books;

DESC student;

DESC issues;

DESC roles;

DESC permission;

DESC login;

DESC user;

-- SELECT commands

SELECT \* FROM branch;

SELECT \* FROM books;

SELECT \* FROM student;

SELECT \* FROM issues;

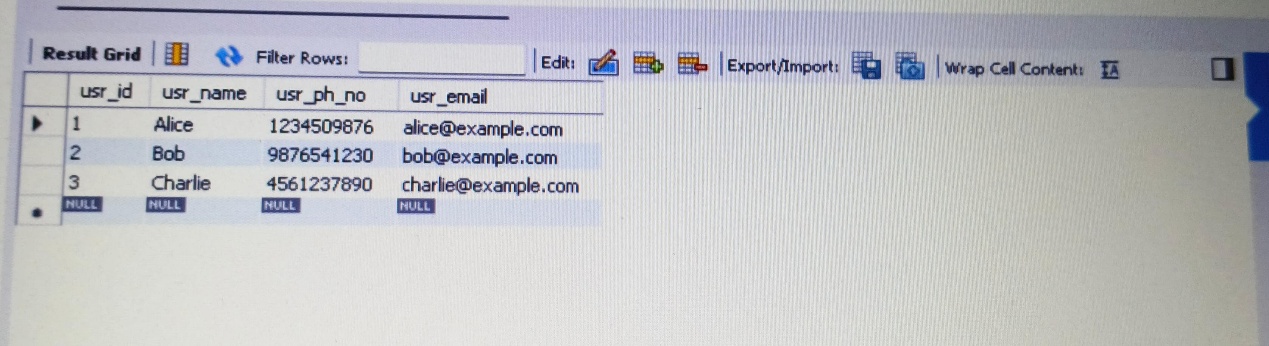
SELECT \* FROM roles;

SELECT \* FROM permission;

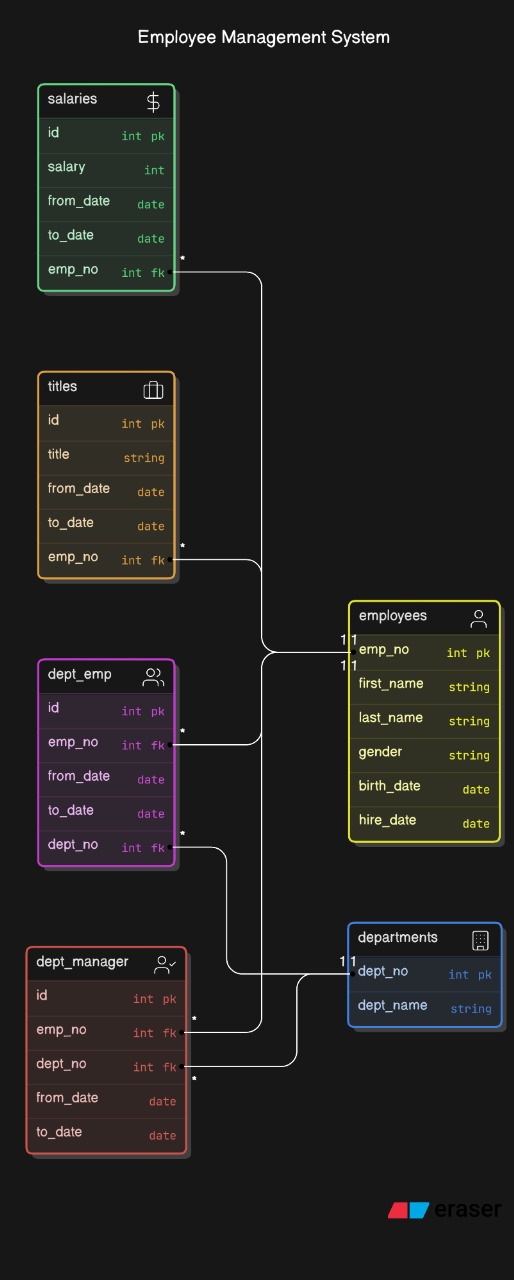
SELECT \* FROM login;

SELECT \* FROM user;

**OUTPUT:**



**10. Employee Management:**

****

- Creating the database

CREATE DATABASE EmployeeManagement;

-- Using the database

USE EmployeeManagement;

-- Creating the tables

CREATE TABLE employees (

emp\_no INT(10) PRIMARY KEY,

birth\_date DATE,

first\_name VARCHAR(14),

last\_name VARCHAR(16),

gender ENUM('M', 'F'),

hire\_date DATE

);

CREATE TABLE salaries (

emp\_no INT(10),

salary INT(11),

from\_date DATE,

to\_date DATE,

PRIMARY KEY (emp\_no, from\_date),

FOREIGN KEY (emp\_no) REFERENCES employees(emp\_no)

);

CREATE TABLE titles (

emp\_no INT(10),

title VARCHAR(50),

from\_date DATE,

to\_date DATE,

PRIMARY KEY (emp\_no, title, from\_date),

FOREIGN KEY (emp\_no) REFERENCES employees(emp\_no)

);

CREATE TABLE departments (

dept\_no CHAR(4) PRIMARY KEY,

dept\_name VARCHAR(40)

);

CREATE TABLE dept\_emp (

emp\_no INT(10),

dept\_no CHAR(4),

from\_date DATE,

to\_date DATE,

PRIMARY KEY (emp\_no, dept\_no),

FOREIGN KEY (emp\_no) REFERENCES employees(emp\_no),

FOREIGN KEY (dept\_no) REFERENCES departments(dept\_no)

);

CREATE TABLE dept\_manager (

dept\_no CHAR(4),

emp\_no INT(10),

from\_date DATE,

to\_date DATE,

PRIMARY KEY (dept\_no, emp\_no),

FOREIGN KEY (dept\_no) REFERENCES departments(dept\_no),

FOREIGN KEY (emp\_no) REFERENCES employees(emp\_no)

);

DESC employees;

DESC salaries;

DESC titles;

DESC departments;

DESC dept\_emp;

DESC dept\_manager;

-- Insert data into employees

INSERT INTO employees VALUES

(101, '1985-05-20', 'John', 'Doe', 'M', '2010-01-15'),

(102, '1990-07-15', 'Jane', 'Smith', 'F', '2012-03-10'),

(103, '1988-09-12', 'Mike', 'Brown', 'M', '2015-06-01');

-- Insert data into salaries

INSERT INTO salaries VALUES

(101, 60000, '2010-01-15', '2020-12-31'),

(102, 75000, '2012-03-10', '2021-12-31'),

(103, 50000, '2015-06-01', '2023-12-31');

-- Insert data into titles

INSERT INTO titles VALUES

(101, 'Software Engineer', '2010-01-15', '2020-12-31'),

(102, 'Manager', '2012-03-10', '2021-12-31'),

(103, 'Analyst', '2015-06-01', '2023-12-31');

-- Insert data into departments

INSERT INTO departments VALUES

('D001', 'Human Resources'),

('D002', 'IT'),

('D003', 'Finance');

-- Insert data into dept\_emp

INSERT INTO dept\_emp VALUES

(101, 'D002', '2010-01-15', '2020-12-31'),

(102, 'D003', '2012-03-10', '2021-12-31'),

(103, 'D001', '2015-06-01', '2023-12-31');

-- Insert data into dept\_manager

INSERT INTO dept\_manager VALUES

('D002', 102, '2015-01-01', '2021-12-31'),

('D003', 101, '2018-01-01', '2020-12-31');

SELECT \* FROM employees;

SELECT \* FROM salaries;

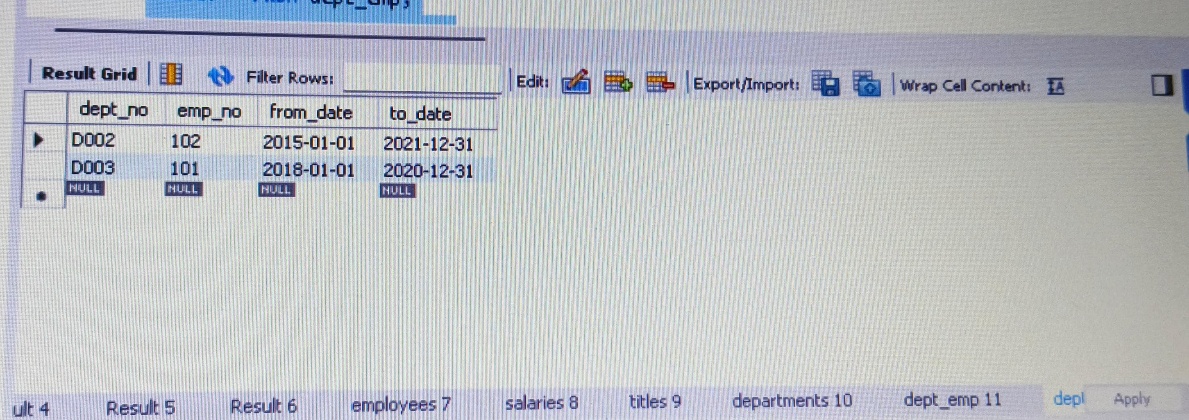
SELECT \* FROM titles;

SELECT \* FROM departments;

SELECT \* FROM dept\_emp;

SELECT \* FROM dept\_manager;

**OUTPUT:**

****

**11. Ship Management:**

-- Step 1: Create a new database

CREATE DATABASE ShipManagementSystem;

-- Step 2: Use the database

USE ShipManagementSystem;

-- Step 3: Create all required tables in the correct order

-- Department Table (Referenced by Employee)

CREATE TABLE Department (

Department\_ID INT PRIMARY KEY,

Department\_Name VARCHAR(50),

Number\_of\_Employees INT

);

-- Employee Table

CREATE TABLE Employee (

Employee\_ID INT PRIMARY KEY,

First\_Name VARCHAR(50),

Middle\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

Date\_of\_Birth DATE,

Gender CHAR(1),

Department\_ID INT,

Address VARCHAR(100),

Pin\_Code VARCHAR(10),

State VARCHAR(50),

Country VARCHAR(50),

Salary DECIMAL(10, 2),

FOREIGN KEY (Department\_ID) REFERENCES Department(Department\_ID)

);

-- Passenger Table

CREATE TABLE Passenger (

Passenger\_ID INT PRIMARY KEY,

First\_Name VARCHAR(50),

Middle\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

Date\_of\_Birth DATE,

Gender CHAR(1),

Ticket\_ID INT,

Address VARCHAR(100),

Pin\_Code VARCHAR(10),

State VARCHAR(50),

Country VARCHAR(50)

);

-- Passenger\_Phone\_Numbers Table

CREATE TABLE Passenger\_Phone\_Numbers (

Passenger\_ID INT,

Phone\_Number VARCHAR(15),

PRIMARY KEY (Passenger\_ID, Phone\_Number),

FOREIGN KEY (Passenger\_ID) REFERENCES Passenger(Passenger\_ID)

);

-- Passenger\_Emails Table

CREATE TABLE Passenger\_Emails (

Passenger\_ID INT,

Email VARCHAR(100),

PRIMARY KEY (Passenger\_ID, Email),

FOREIGN KEY (Passenger\_ID) REFERENCES Passenger(Passenger\_ID)

);

-- Port Table

CREATE TABLE Port (

Port\_ID INT PRIMARY KEY,

Port\_Name VARCHAR(50),

Port\_Capacity INT

);

-- Ship Table

CREATE TABLE Ship (

Ship\_ID INT PRIMARY KEY,

Ship\_Name VARCHAR(50),

Ship\_Type VARCHAR(20),

Last\_Service\_Date DATE,

Source\_Port\_ID INT,

Destination\_Port\_ID INT,

FOREIGN KEY (Source\_Port\_ID) REFERENCES Port(Port\_ID),

FOREIGN KEY (Destination\_Port\_ID) REFERENCES Port(Port\_ID)

);

-- Booking\_Agents Table

CREATE TABLE Booking\_Agents (

Agent\_ID INT PRIMARY KEY,

Agent\_Name VARCHAR(50)

);

-- Ticket Table

CREATE TABLE Ticket (

Ticket\_ID INT PRIMARY KEY,

Ship\_ID INT,

Port\_ID INT,

Arrival\_Time DATETIME,

Passenger\_ID INT,

Seat\_No VARCHAR(10),

Agent\_ID INT,

FOREIGN KEY (Ship\_ID) REFERENCES Ship(Ship\_ID),

FOREIGN KEY (Passenger\_ID) REFERENCES Passenger(Passenger\_ID),

FOREIGN KEY (Port\_ID) REFERENCES Port(Port\_ID),

FOREIGN KEY (Agent\_ID) REFERENCES Booking\_Agents(Agent\_ID)

);

-- Book\_Ticket Table

CREATE TABLE Book\_Ticket (

Passenger\_ID INT,

Agent\_ID INT,

Ticket\_ID INT,

Port\_ID INT,

PRIMARY KEY (Passenger\_ID, Ticket\_ID),

FOREIGN KEY (Passenger\_ID) REFERENCES Passenger(Passenger\_ID),

FOREIGN KEY (Agent\_ID) REFERENCES Booking\_Agents(Agent\_ID),

FOREIGN KEY (Port\_ID) REFERENCES Port(Port\_ID)

);

-- Route Table

CREATE TABLE Route (

Arrival\_Time DATETIME,

Departure\_Time DATETIME,

Stop\_Number INT,

Port\_ID INT,

Ship\_ID INT,

PRIMARY KEY (Stop\_Number, Port\_ID, Ship\_ID),

FOREIGN KEY (Port\_ID) REFERENCES Port(Port\_ID),

FOREIGN KEY (Ship\_ID) REFERENCES Ship(Ship\_ID)

);

-- Employee\_Phone\_Numbers Table

CREATE TABLE Employee\_Phone\_Numbers (

Employee\_ID INT,

Phone\_Number VARCHAR(15),

PRIMARY KEY (Employee\_ID, Phone\_Number),

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID)

);

-- Employee\_Emails Table

CREATE TABLE Employee\_Emails (

Employee\_ID INT,

Email VARCHAR(100),

PRIMARY KEY (Employee\_ID, Email),

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID)

);

-- Employee\_Dependent Table

CREATE TABLE Employee\_Dependent (

Dependent\_ID INT PRIMARY KEY,

Employee\_ID INT,

First\_Name VARCHAR(50),

Last\_Name VARCHAR(50),

Gender CHAR(1),

Date\_of\_Birth DATE,

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID)

);

-- Security Table

CREATE TABLE Security (

Employee\_ID INT PRIMARY KEY,

Years\_of\_Experience INT,

Security\_Company\_Name VARCHAR(50),

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID)

);

-- Drivers Table

CREATE TABLE Drivers (

Employee\_ID INT PRIMARY KEY,

Year\_of\_Experience INT,

Driving\_License\_ID VARCHAR(50),

Ship\_Type VARCHAR(20),

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID)

);

-- Other\_Staff Table

CREATE TABLE Other\_Staff (

Employee\_ID INT PRIMARY KEY,

Years\_of\_Experience INT,

FOREIGN KEY (Employee\_ID) REFERENCES Employee(Employee\_ID)

);

-- Step 4: Insert sample data into tables

-- Insert into Department

INSERT INTO Department VALUES (1, 'Operations', 50);

INSERT INTO Department VALUES (2, 'Logistics', 30);

-- Insert into Employee

INSERT INTO Employee VALUES (1, 'John', 'A.', 'Smith', '1980-05-15', 'M', 1, '123 Street', '11111', 'California', 'USA', 60000);

INSERT INTO Employee VALUES (2, 'Jane', 'B.', 'Doe', '1990-11-20', 'F', 2, '456 Avenue', '22222', 'Texas', 'USA', 50000);

-- Insert into Passenger

INSERT INTO Passenger VALUES (1, 'John', 'A.', 'Smith', '1990-05-15', 'M', 101, '123 Street', '11111', 'California', 'USA');

INSERT INTO Passenger VALUES (2, 'Jane', 'B.', 'Doe', '1985-11-20', 'F', 102, '456 Avenue', '22222', 'Texas', 'USA');

INSERT INTO Passenger VALUES (3, 'Alice', NULL, 'Johnson', '1995-07-10', 'F', 103, '789 Boulevard', '33333', 'Florida', 'USA');

-- Insert into Passenger\_Phone\_Numbers

INSERT INTO Passenger\_Phone\_Numbers VALUES (1, '1234567890');

INSERT INTO Passenger\_Phone\_Numbers VALUES (2, '9876543210');

INSERT INTO Passenger\_Phone\_Numbers VALUES (3, '4561237890');

-- Insert into Port

INSERT INTO Port VALUES (1, 'New York Port', 1000);

INSERT INTO Port VALUES (2, 'Los Angeles Port', 1500);

INSERT INTO Port VALUES (3, 'Miami Port', 1200);

-- Insert into Ship

INSERT INTO Ship VALUES (1, 'Titanic', 'Passenger Ship', '2023-11-01', 1, 2);

INSERT INTO Ship VALUES (2, 'Evergreen', 'Cargo Ship', '2024-01-15', 2, 3);

INSERT INTO Ship VALUES (3, 'Poseidon', 'Passenger Ship', '2024-02-10', 3, 1);

-- Step 5: Display the contents of tables

-- Display Passenger

SELECT \* FROM Passenger;

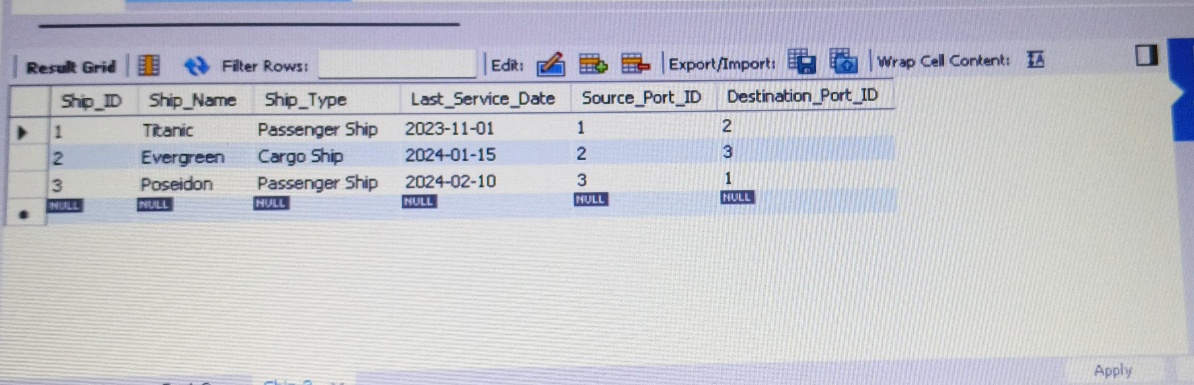
-- Display Port

SELECT \* FROM Port;

-- Display Ship

SELECT \* FROM Ship;

**OUTPUT:**

****

**12. Sales Relation:**

-- Step 1: Create a new database

CREATE DATABASE SalesRelationManagement;

-- Step 2: Use the database

USE SalesRelationManagement;

-- Step 3: Create all required tables

-- Customers Table

CREATE TABLE Customers (

Customer\_ID INT PRIMARY KEY,

First\_Name VARCHAR(255),

Last\_Name VARCHAR(255),

Email\_Address VARCHAR(255) UNIQUE,

Number\_Of\_Companies INT,

Date\_Of\_Birth DATE

);

-- Products Table

CREATE TABLE Products (

Product\_ID INT PRIMARY KEY,

Product\_Name VARCHAR(255),

Unit\_Price FLOAT,

Product\_Description VARCHAR(255),

Supp\_ID INT,

FOREIGN KEY (Supp\_ID) REFERENCES Product\_Suppliers(Supp\_ID)

);

-- Product\_Suppliers Table

CREATE TABLE Product\_Suppliers (

Supp\_ID INT PRIMARY KEY,

Supp\_Name VARCHAR(255),

Supp\_Telephone VARCHAR(15),

Supp\_Email VARCHAR(255) UNIQUE

);

-- Stores Table

CREATE TABLE Stores (

Store\_ID INT PRIMARY KEY,

Store\_Name VARCHAR(255),

Address VARCHAR(255),

SubSec INT,

Store\_Email\_Address VARCHAR(255) UNIQUE,

Store\_Telephone VARCHAR(15)

);

-- Employees Table

CREATE TABLE Employees (

Emp\_ID INT PRIMARY KEY,

First\_Name VARCHAR(255),

Last\_Name VARCHAR(255),

Date\_Of\_Birth DATE,

Hire\_Date DATE,

Emp\_Telephone VARCHAR(15),

Store\_ID INT,

FOREIGN KEY (Store\_ID) REFERENCES Stores(Store\_ID)

);

-- Sales Table

CREATE TABLE Sales (

Purchase\_ID INT PRIMARY KEY,

Date\_Of\_Purchase DATE,

Product\_ID INT,

Quantity INT,

Customer\_ID INT,

Store\_ID INT,

FOREIGN KEY (Product\_ID) REFERENCES Products(Product\_ID),

FOREIGN KEY (Customer\_ID) REFERENCES Customers(Customer\_ID),

FOREIGN KEY (Store\_ID) REFERENCES Stores(Store\_ID)

);

-- Step 4: Insert sample data into tables

-- Insert into Customers

INSERT INTO Customers (Customer\_ID, First\_Name, Last\_Name, Email\_Address, Number\_Of\_Companies, Date\_Of\_Birth)

VALUES

(1, 'John', 'Smith', 'john.smith@example.com', 2, '1990-05-15'),

(2, 'Jane', 'Doe', 'jane.doe@example.com', 1, '1985-11-20'),

(3, 'Alice', 'Johnson', 'alice.johnson@example.com', 3, '1995-07-10');

-- Insert into Product\_Suppliers

INSERT INTO Product\_Suppliers (Supp\_ID, Supp\_Name, Supp\_Telephone, Supp\_Email)

VALUES

(1, 'ABC Supplies', '1234567890', 'abc@supplies.com'),

(2, 'XYZ Wholesalers', '9876543210', 'xyz@wholesalers.com'),

(3, 'Global Traders', '4561237890', 'global@traders.com');

-- Insert into Products

INSERT INTO Products (Product\_ID, Product\_Name, Unit\_Price, Product\_Description, Supp\_ID)

VALUES

(1, 'Laptop', 1200.50, 'High performance laptop', 1),

(2, 'Smartphone', 800.00, 'Latest smartphone', 2),

(3, 'Headphones', 150.75, 'Noise-cancelling headphones', 3);

-- Insert into Stores

INSERT INTO Stores (Store\_ID, Store\_Name, Address, SubSec, Store\_Email\_Address, Store\_Telephone)

VALUES

(1, 'Tech Store', '123 Tech Street', 101, 'techstore@example.com', '123456789'),

(2, 'Gadget World', '456 Gadget Avenue', 102, 'gadgetworld@example.com', '987654321'),

(3, 'ElectroMart', '789 Electro Boulevard', 103, 'electromart@example.com', '456789123');

-- Insert into Employees

INSERT INTO Employees (Emp\_ID, First\_Name, Last\_Name, Date\_Of\_Birth, Hire\_Date, Emp\_Telephone, Store\_ID)

VALUES

(1, 'Michael', 'Brown', '1980-01-15', '2010-06-01', '1234567890', 1),

(2, 'Sarah', 'Wilson', '1992-04-20', '2015-08-12', '9876543210', 2),

(3, 'David', 'Lee', '1988-12-05', '2018-01-22', '4561237890', 3);

-- Insert into Sales

INSERT INTO Sales (Purchase\_ID, Date\_Of\_Purchase, Product\_ID, Quantity, Customer\_ID, Store\_ID)

VALUES

(1, '2024-11-01', 1, 2, 1, 1),

(2, '2024-11-05', 2, 1, 2, 2),

(3, '2024-11-10', 3, 3, 3, 3);

-- Step 5: Display the contents of tables

-- Display Customers

SELECT \* FROM Customers;

-- Display Product\_Suppliers

SELECT \* FROM Product\_Suppliers;

-- Display Products

SELECT \* FROM Products;

-- Display Stores

SELECT \* FROM Stores;

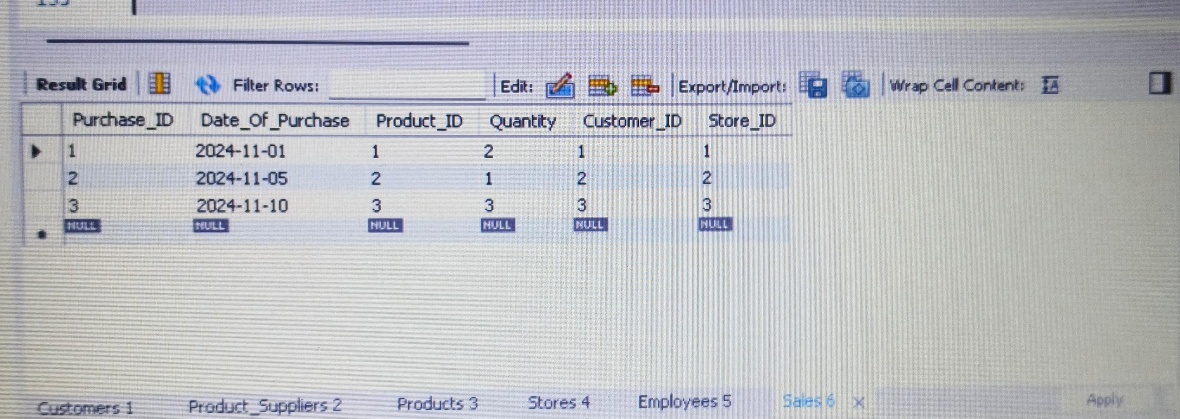
-- Display Employees

SELECT \* FROM Employees;

-- Display Sales

SELECT \* FROM Sales;

**OUTPUT:**

****

**Learning Outcomes :**

1. **Database Creation:**

* Gained proficiency in creating and managing relational databases using MySQL.
* Understood schema translation into a functional database.

1. **Table Creation with Constraints:**

* Learned to define primary keys, foreign keys, and other constraints to maintain data integrity.
* Ensured normalization and optimized table design.

3. **Entity-Relationship Modeling:**

* Developed skills in visualizing and modeling database relationships.
* Used E-R diagrams to ensure clarity in database design.

4. **Practical Skills:**

* Enhanced SQL skills for database manipulation and management.
* Understood the importance of error handling and transaction management.

**Project Summary:**

**Objective:**

The database is designed to efficiently manage and organize data for various entities and their relationships, ensuring seamless data storage, retrieval, and manipulation. It supports scalability, data integrity, and accurate representation of real-world scenarios, making it easy to maintain and query the system for informed decision-making.

**General Database Schema Overview**

A database schema represents the logical structure of a database, organizing its tables, relationships, and constraints. It typically includes:

1. **Entities (Tables):** Represent objects or concepts (e.g., employees, students, books). Each table contains rows (records) and columns (attributes).
2. **Attributes (Columns):** Define the properties or characteristics of each entity, such as names, IDs, or dates.
3. **Primary Key (PK):** A unique identifier for each record in a table.
4. **Foreign Key (FK):** Establishes relationships between tables, linking records in one table to another.
5. **Relationships:** Show associations between entities (e.g., one-to-many, many-to-many).
6. **Constraints:** Rules to maintain data integrity, such as NOT NULL, UNIQUE, and CHECK constraints.

The schema provides a blueprint for creating, managing, and querying the database effectively. It ensures consistency, scalability, and reliability of data across applications.

**Team Members and Roles:**

* 1. **Sadia:**
* Responsible for creating the schema and establishing table relationships, including primary and foreign key constraints.
* Conducted testing to verify the database functionality, including queries to display and validate data.
  1. **Bhuvaneswari:**
* Handled the insertion of sample data and ensured data integrity across tables.
  1. **Jasmine:**
* Designed the E-R model to visually represent the schema, including entity relationships and constraints.
  1. **Aryan:**
* Created DDL and DML queries for table creation, data insertion, and retrieval**.**

**Outcome:**

The database schema successfully fulfills the project requirements for managing employee records and their organizational associations. With the implemented design, querying and maintaining thesystem is efficient and scalable. The team collaborated effectively to ensure high-quality results.

**Conclusion**

This project reinforced core concepts of RDBMS and server-side database development. By creating a MySQL database, designing an E-R model, and implementing schema constraints, we achieved a structured approach to managing data efficiently. These skills form the foundation for advanced applications in server-side programming and enterprise solutions.