CASE STUDY

EMPLOYEES ATTENDANCE MANAGEMENT SYSTEM

Project Overview :

Objective :

Design and implement an attendance management system to track employees' attendance, leave status, and working hours effectively.

Scope:

- Employees can log attendance.
- Managers can view attendance records.
- Admins can manage employees and attendance policies.

Requirements Analysis :

Features:

1. **Employee Module**:
   - Add new employees.
   - Update employee details.
   - Remove employees.
2. **Attendance Module**:
   - Log daily attendance (check-in/check-out time).
   - Mark absences and late arrivals.
   - Generate reports (daily, monthly, yearly).
3. **Leave Management**:
   - Track leave requests and approvals.
   - Calculate remaining leave days.
4. **Reports**:
   - Attendance summary.
   - Employee punctuality statistics.
   - Leave utilization.

Database Design :

Creating an Employee Attendance Management System using SQL can be an excellent project to showcase your database design, querying, and management skills. Here's a structured approach for your case study:

1. Project Overview

Objective:

Design and implement an attendance management system to track employees' attendance, leave status, and working hours effectively.

Scope:

- Employees can log attendance.
- Managers can view attendance records.
- Admins can manage employees and attendance policies.

2. Requirements Analysis

Features:

1. **Employee Module**:
   - Add new employees.
   - Update employee details.
   - Remove employees.
2. **Attendance Module**:
   - Log daily attendance (check-in/check-out time).
   - Mark absences and late arrivals.
   - Generate reports (daily, monthly, yearly).
3. **Leave Management**:
   - Track leave requests and approvals.
   - Calculate remaining leave days.
4. **Reports**:
   - Attendance summary.
   - Employee punctuality statistics.
   - Leave utilization.

3. Database Design

Entities and Relationships:

Tables:

1. **Employees**:

- EmployeeID (Primary Key)
- FirstName, LastName
- Department
- Position
- DateOfJoining
- ContactInfo

2. **Attendance**:
   - AttendanceID (Primary Key)
   - EmployeeID (Foreign Key)
   - Date
   - CheckInTime
   - CheckOutTime
   - Status (e.g., Present, Absent, Late)
3. **Leave**:
   - LeaveID (Primary Key)
   - EmployeeID (Foreign Key)
   - LeaveDate
   - LeaveType (e.g., Sick, Casual, Paid)
   - Status (Pending, Approved, Rejected)
4. **Departments**:
   - DepartmentID (Primary Key)
   - DepartmentName
5. **Users** (for login):
   - UserID (Primary Key)
   - Username
   - Password
   - Role (Admin, Manager, Employee)

SQL Queries :

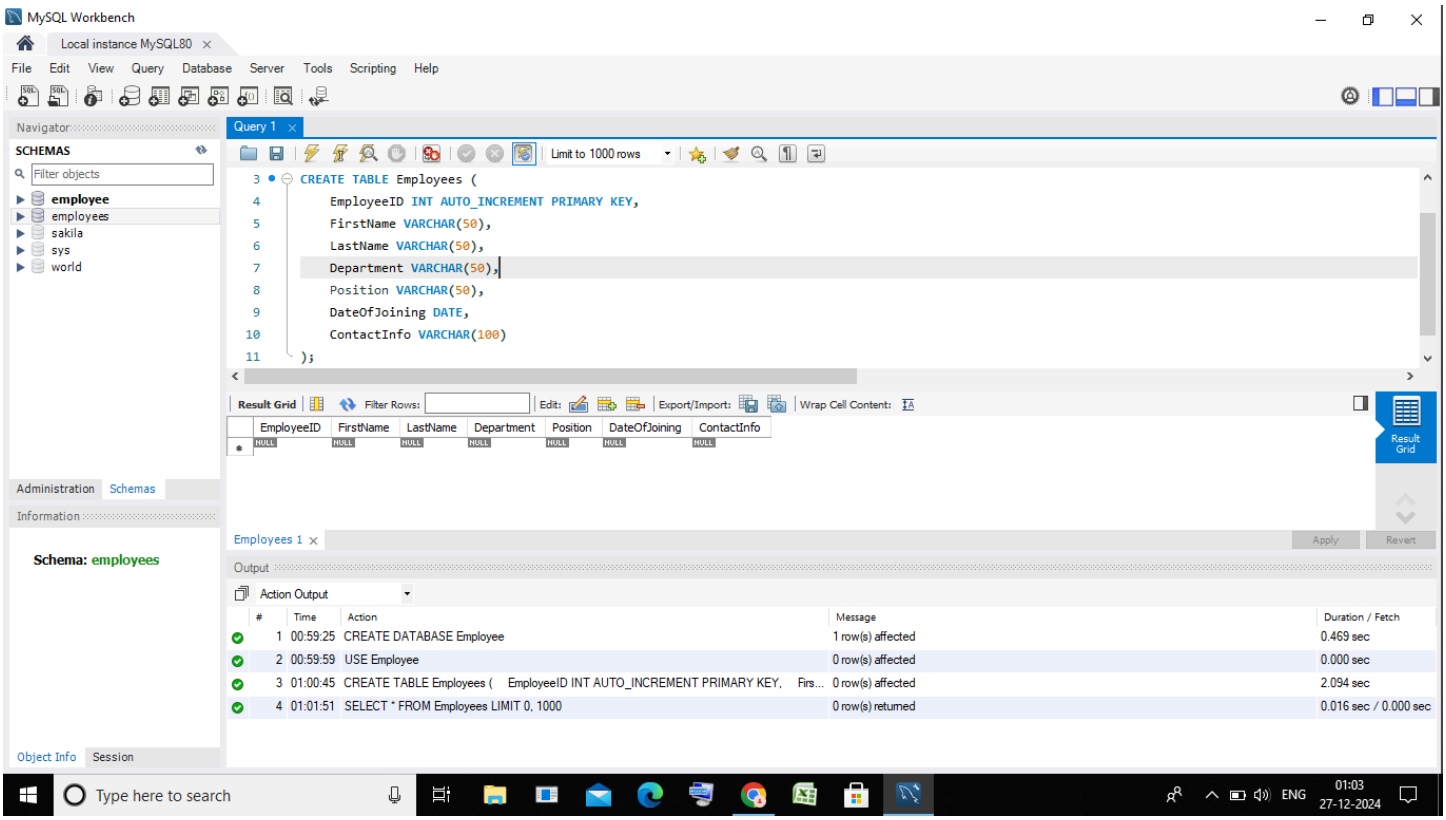
Create Tables:

Employees Table :

```
CREATE TABLE Employees (

   EmployeeID INT AUTO_INCREMENT PRIMARY KEY,

   FirstName VARCHAR(50),

   LastName VARCHAR(50),

   Department VARCHAR(50),

   Position VARCHAR(50),

   DateOfJoining DATE,
```

ContactInfo VARCHAR(100)

);



Attendance Table :

CREATE TABLE Attendance (

AttendanceID INT AUTO_INCREMENT PRIMARY KEY,

EmployeeID INT,

Date DATE,

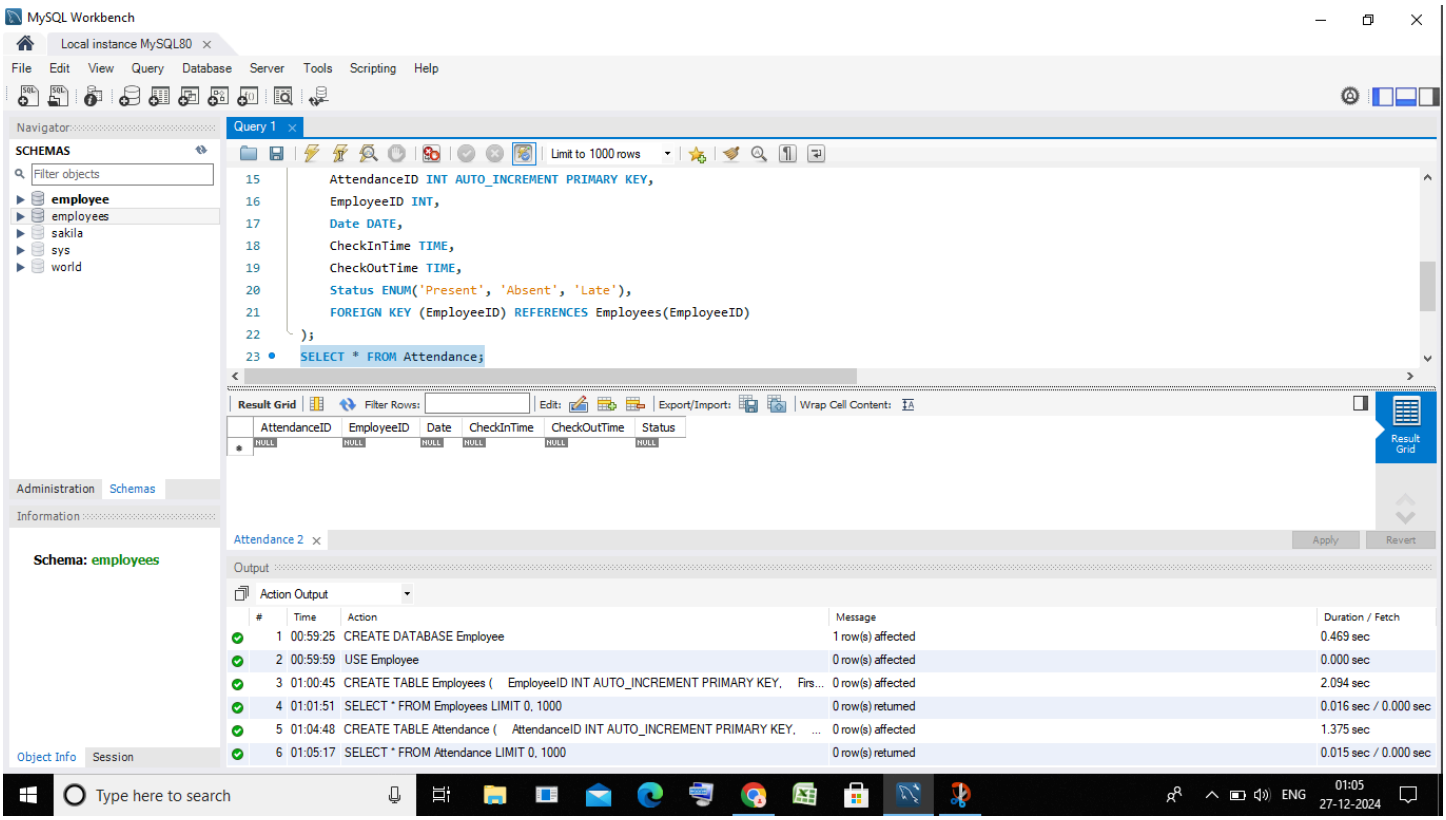
CheckInTime TIME,

CheckOutTime TIME,

Status ENUM('Present', 'Absent', 'Late'),

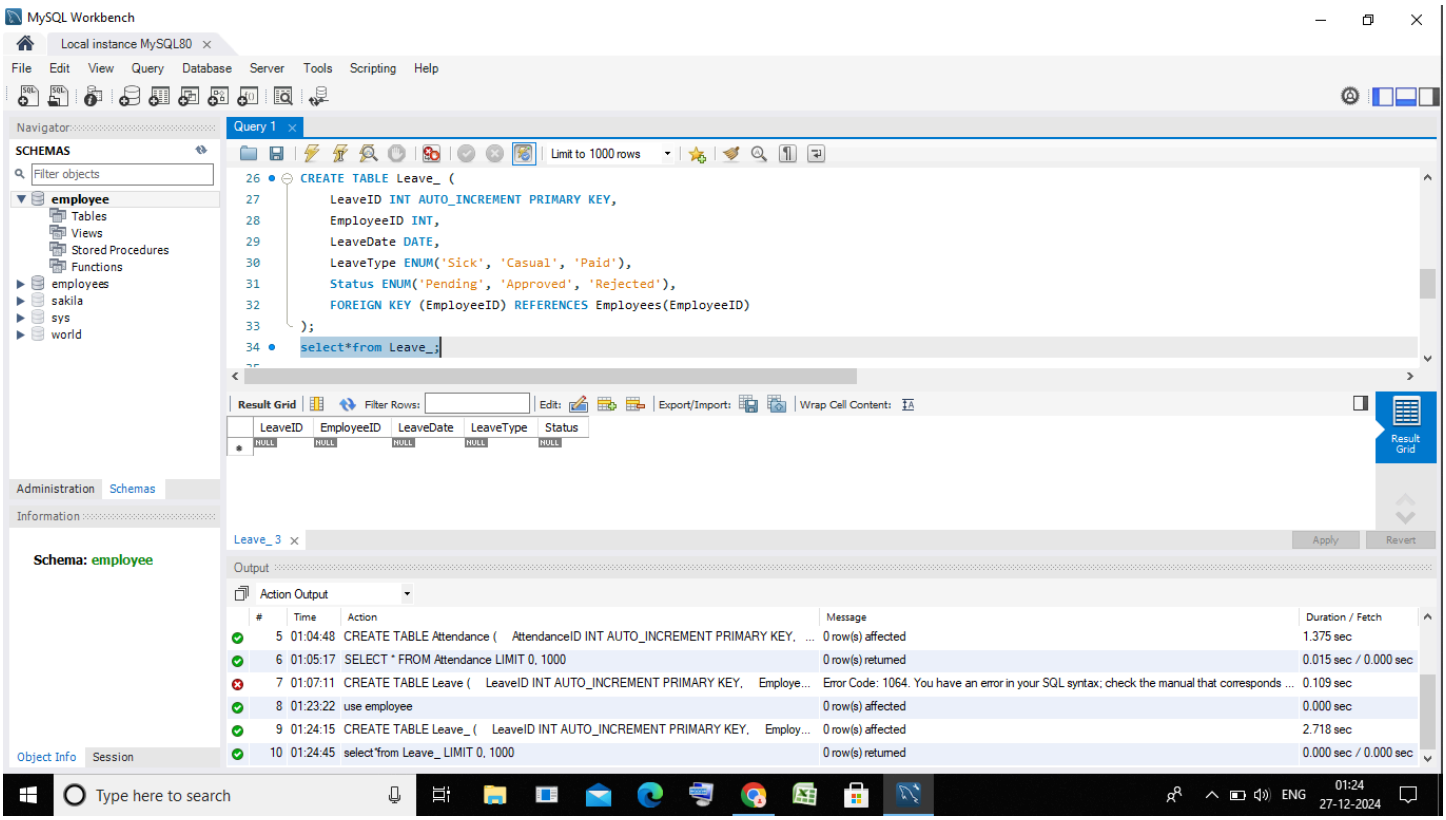
FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);



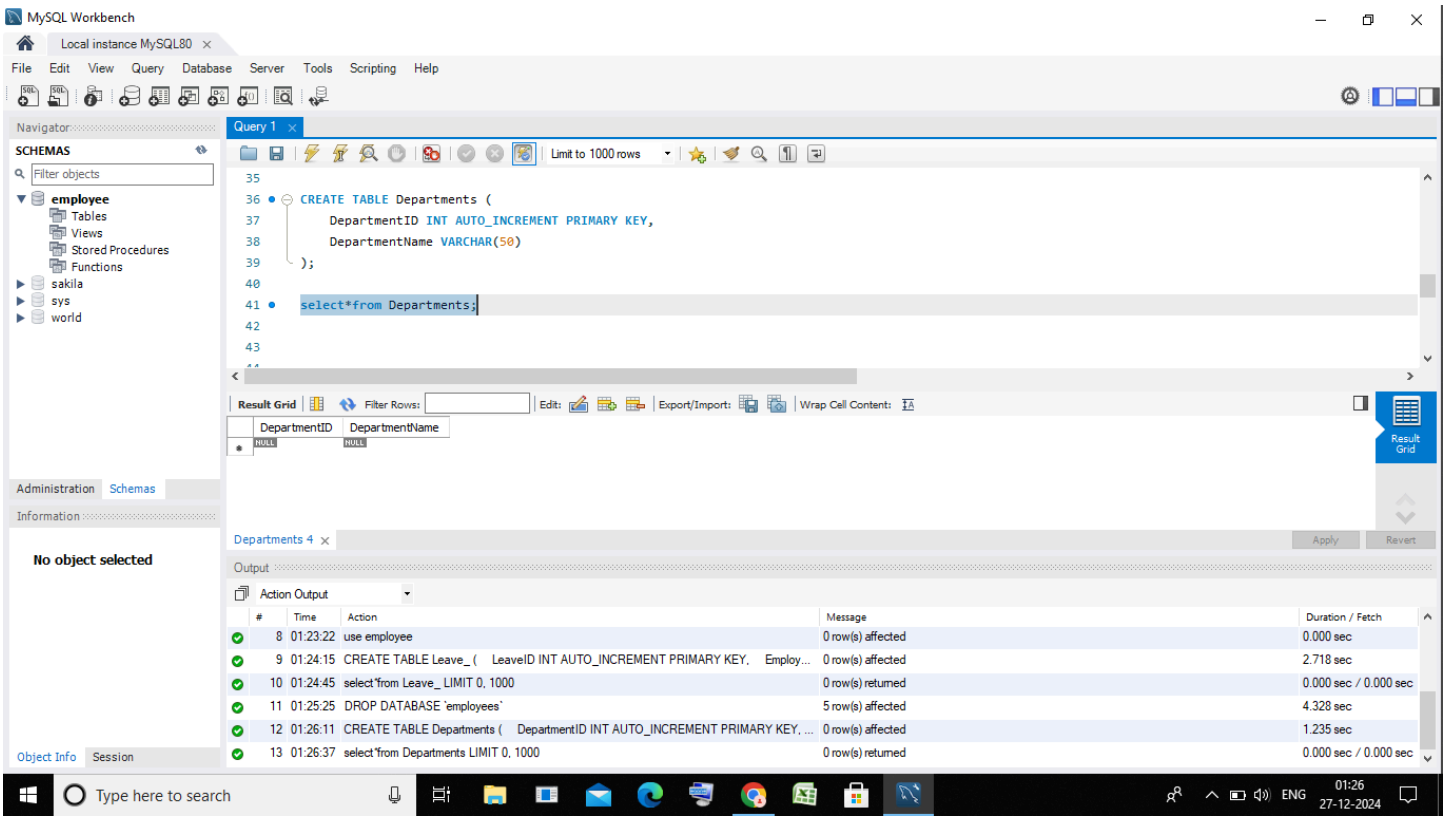
Leave Table :

```
CREATE TABLE Leave _(
    LeaveID INT AUTO_INCREMENT PRIMARY KEY,
    EmployeeID INT,
    LeaveDate DATE,
    LeaveType ENUM('Sick', 'Casual', 'Paid'),
    Status ENUM('Pending', 'Approved', 'Rejected'),
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
```



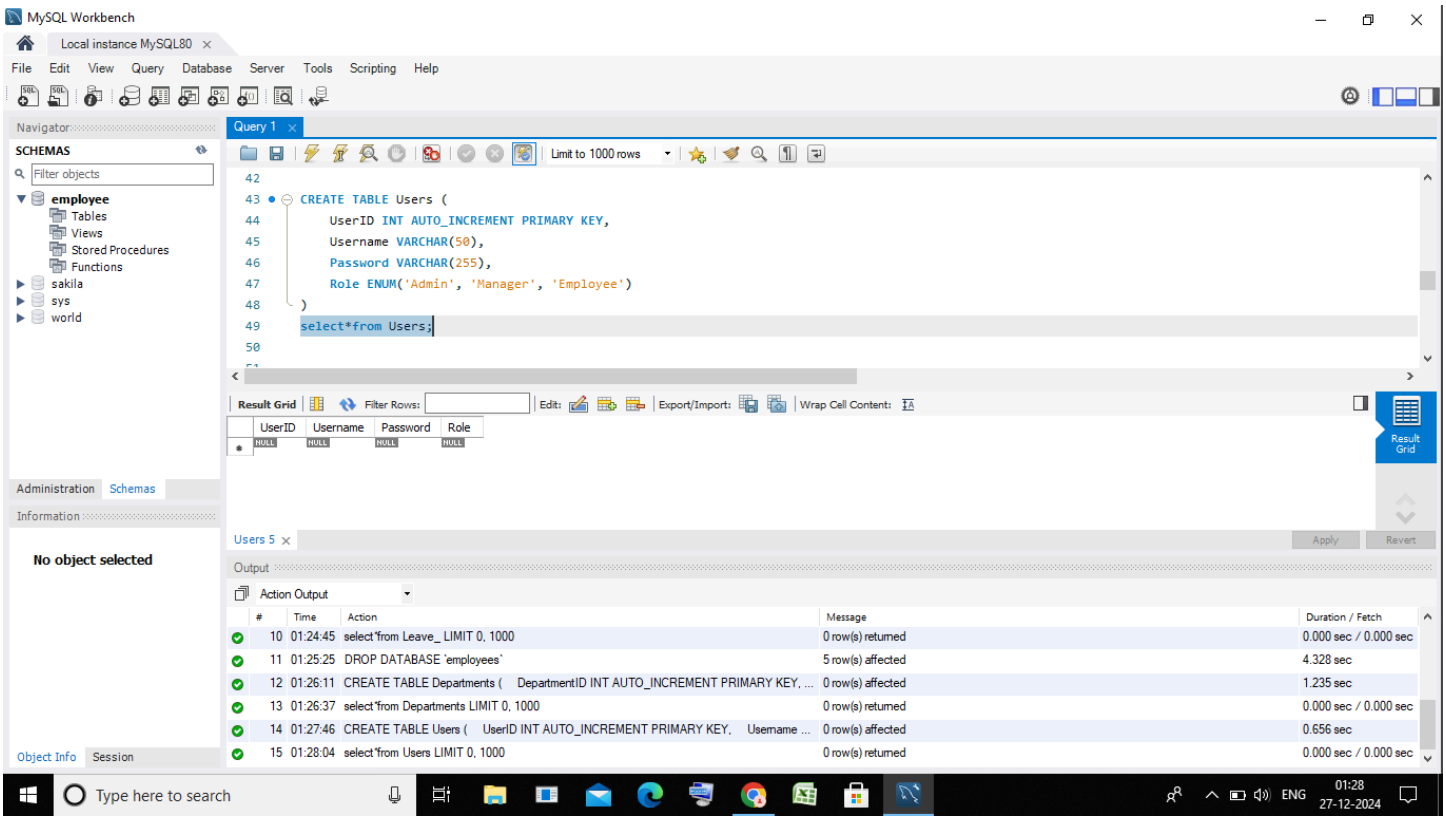
Department Table :

```
CREATE TABLE Departments (
    DepartmentID INT AUTO_INCREMENT PRIMARY KEY,
    DepartmentName VARCHAR(50)
);
```



User Table :

```
CREATE TABLE Users (
    UserID INT AUTO_INCREMENT PRIMARY KEY,
    Username VARCHAR(50),
    Password VARCHAR(255),
    Role ENUM('Admin', 'Manager', 'Employee')
);
```



Sample Queries:

Employees Details :

INSERT INTO Employees(FirstName, LastName, Department, Position, DateOfJoining, ContactInfo) VALUES

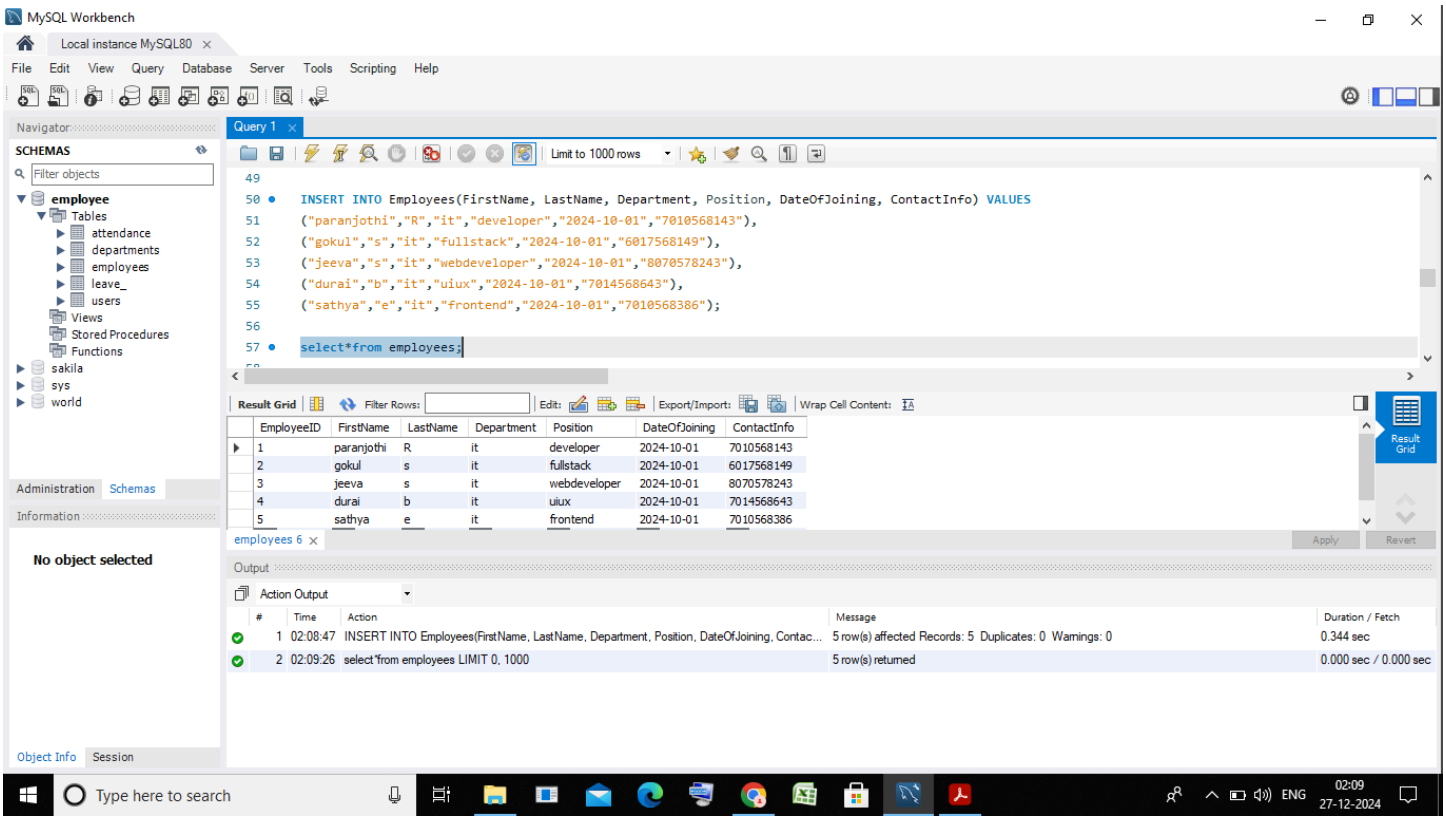
("paranjothi","R","it","developer","2024-10-01","7010568143"),

("gokul","s","it","fullstack","2024-10-01","6017568149"),

("jeeva","s","it","webdeveloper","2024-10-01","8070578243"),

("durai","b","it","uiux","2024-10-01","7014568643"),

("sathya","e","it","frontend","2024-10-01","7010568386");



Insert Attendance :

INSERT INTO Attendance (EmployeeID, Date, CheckInTime, Status)

VALUES (1, '2024-12-01', '09:00:00', 'Present'),

(2, '2024-12-01', '09:00:00', 'Present'),

(3, '2024-12-01', '09:00:00', 'Present'),

(4, '2024-12-01', '09:00:00', 'Present'),

(5, '2024-12-01', '09:00:00', 'Present'),

(1, '2024-12-02', '09:00:00', 'Present'),

(2, '2024-12-02', '09:00:00', 'Present'),

(3, '2024-12-02', '09:00:00', 'Present'),

(4, '2024-12-02', '09:00:00', 'Present'),

(5, '2024-12-02', '09:00:00', 'Present'),

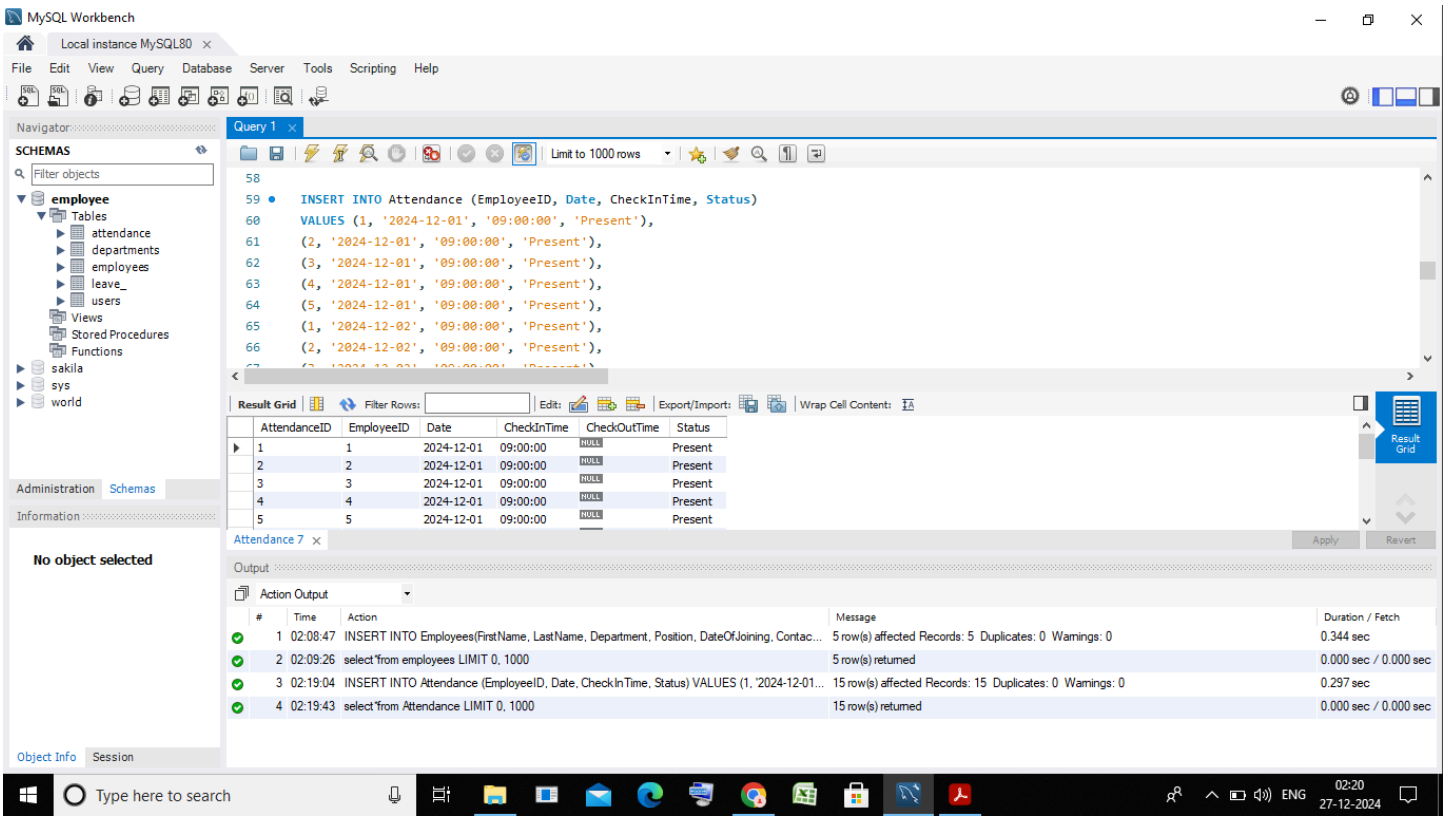
(1, '2024-12-03', '09:00:00', 'Present'),

(2, '2024-12-03', '09:00:00', 'Absent'),

(3, '2024-12-03', '09:00:00', 'Present'),

(4, '2024-12-03', '09:00:00', 'Present'),

(5, '2024-12-03', '09:00:00', 'Absent');



Insert Users :

INSERT INTO Users(Username, Password, Role) VALUES(

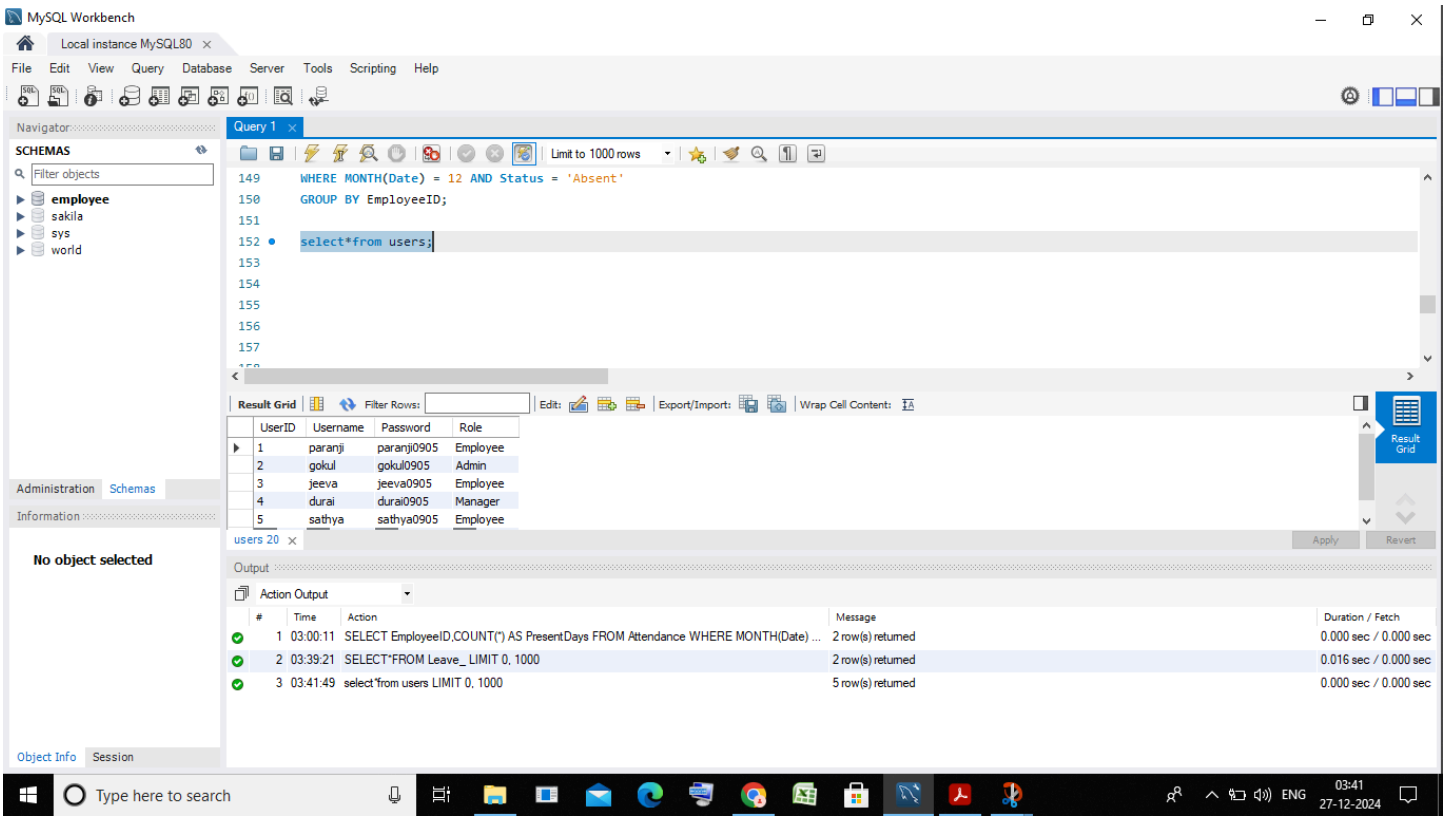
"paranji","paranji0905","employee"),

("gokul","gokul0905","employee"),

("jeeva","jeeva0905","employee"),

("durai","durai0905","employee"),

("sathya","sathya0905","employee");

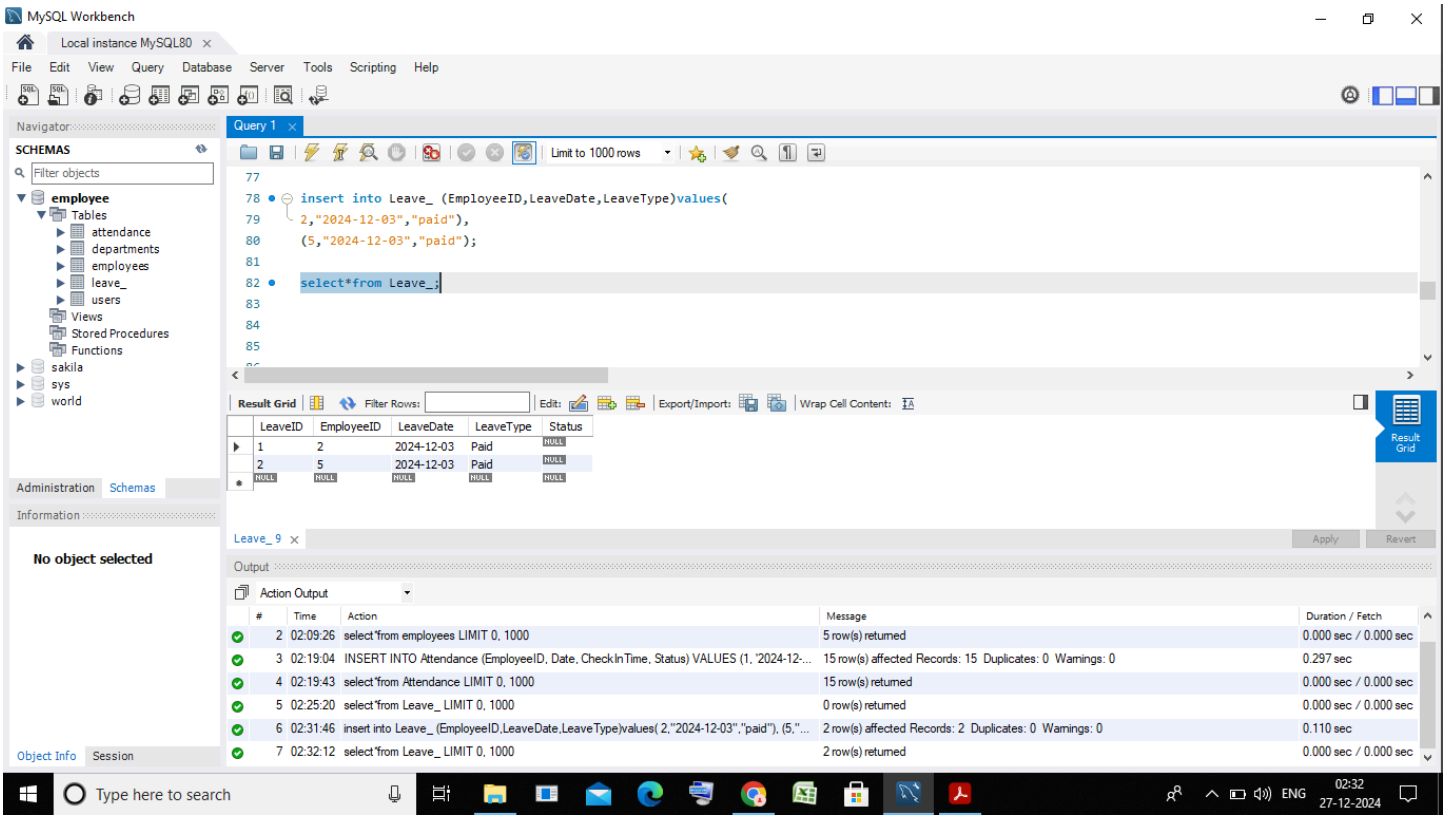


Insert Leave_ :

insert into Leave_ (EmployeeID,LeaveDate,LeaveType)values(

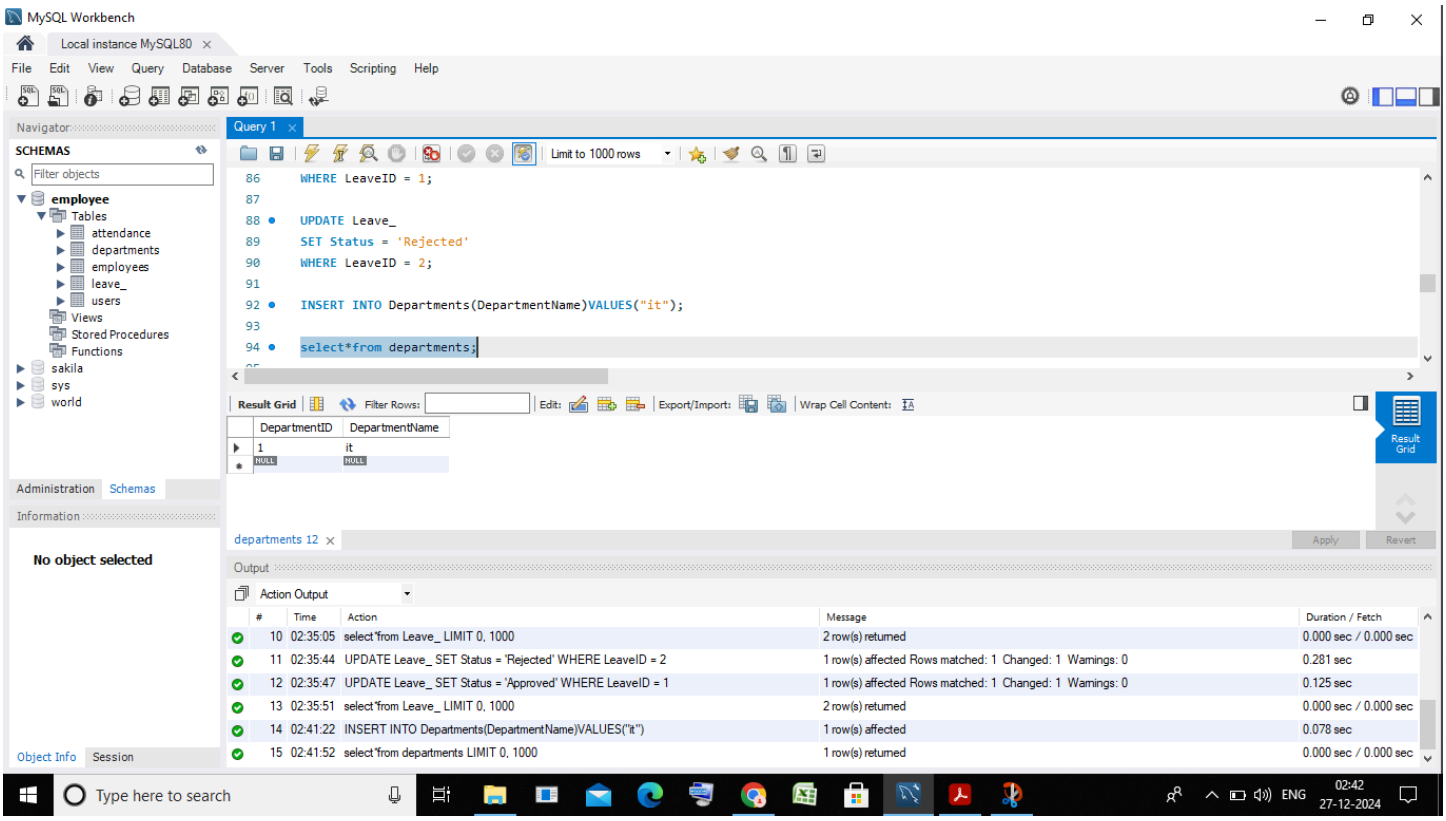
2,"2024-12-03","paid"),

(5,"2024-12-03","paid");



Insert Department :

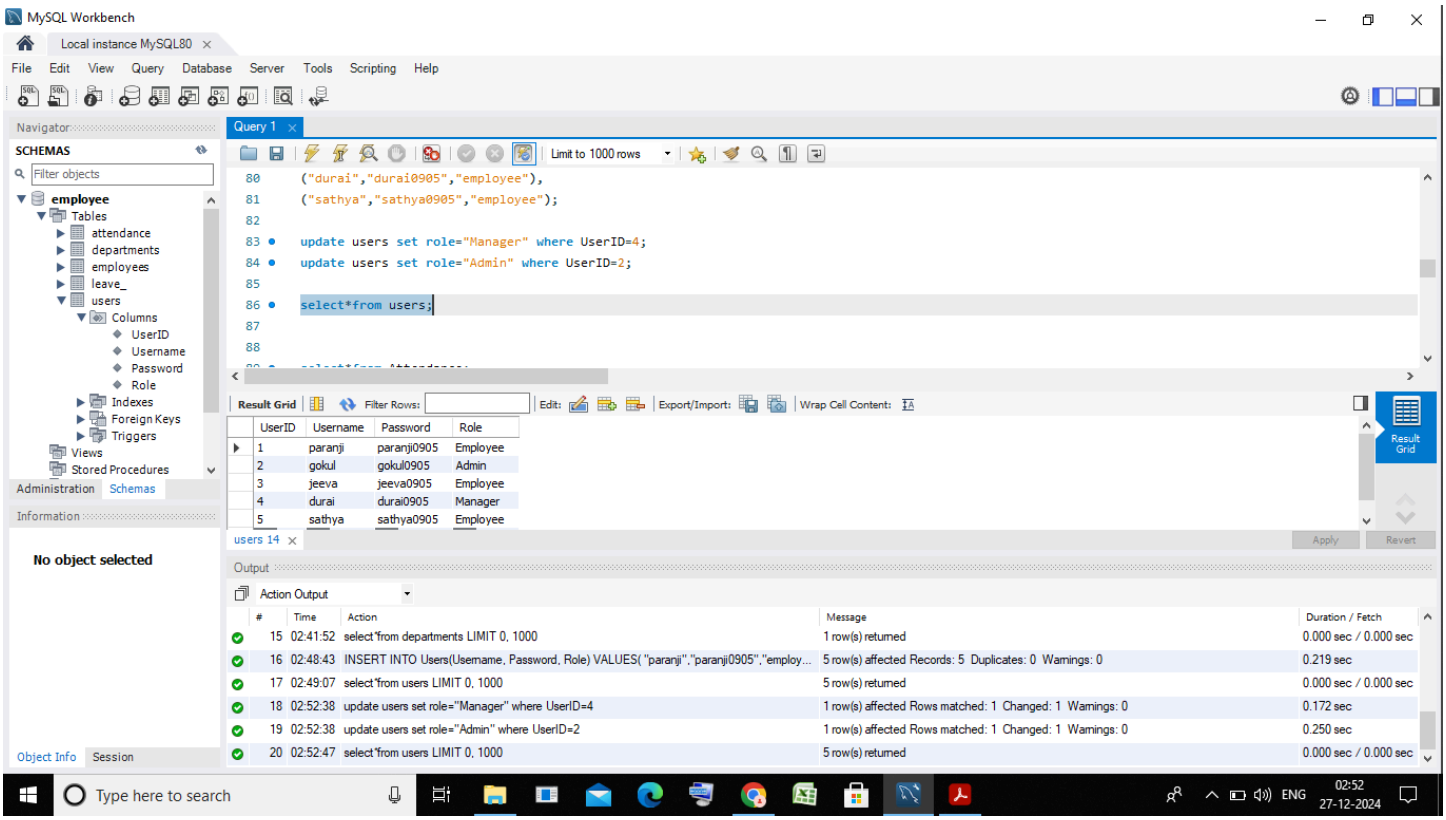
INSERT INTO Departments(DepartmentName)VALUES("it");



Update :

update users set role="Manager" where UserID=4;

update users set role="Admin" where UserID=2;



Update Leave :

UPDATE Leave_

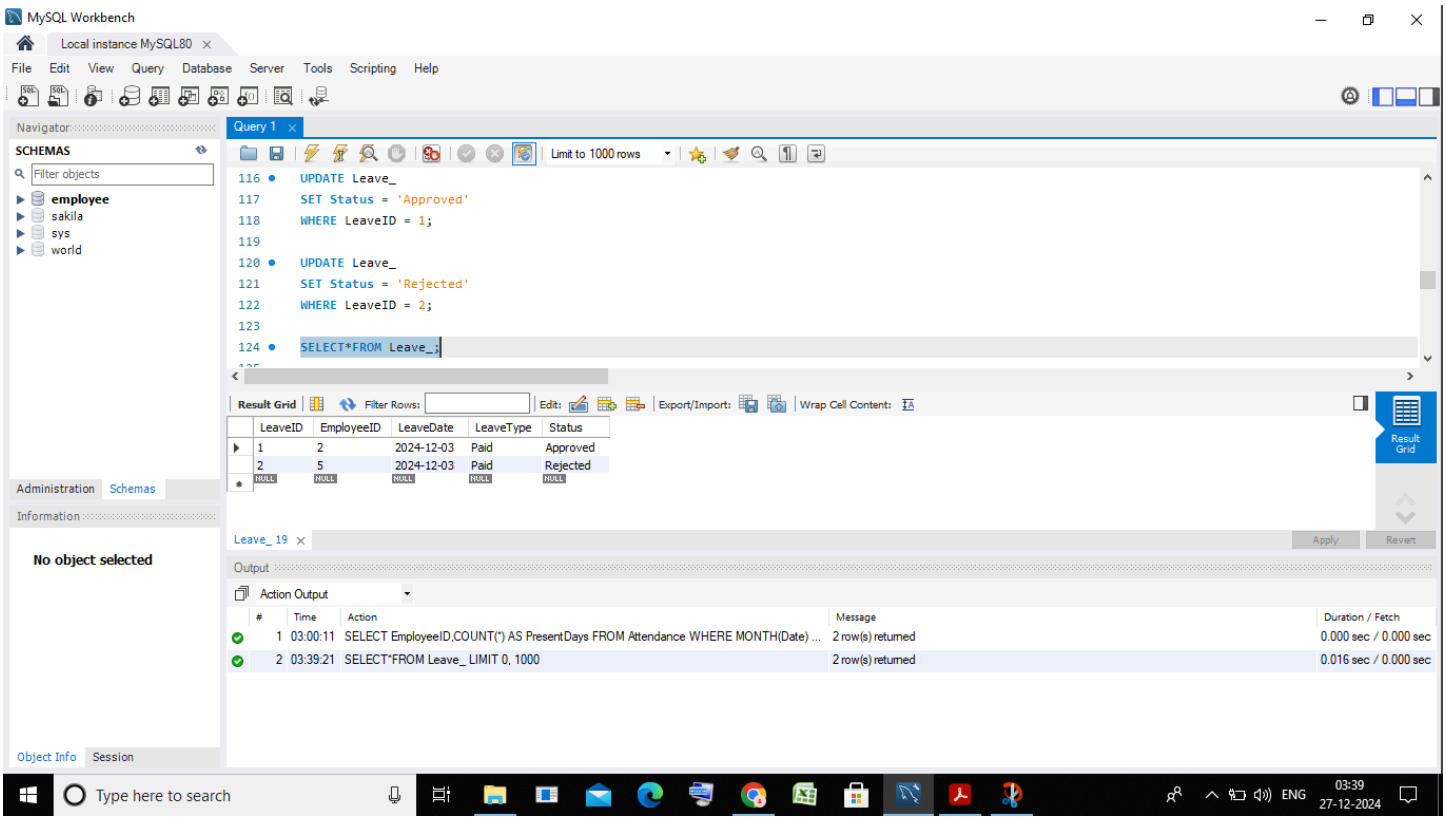
SET Status = 'Approved'

WHERE LeaveID = 1;

UPDATE Leave_

SET Status = 'Rejected'

WHERE LeaveID = 2;



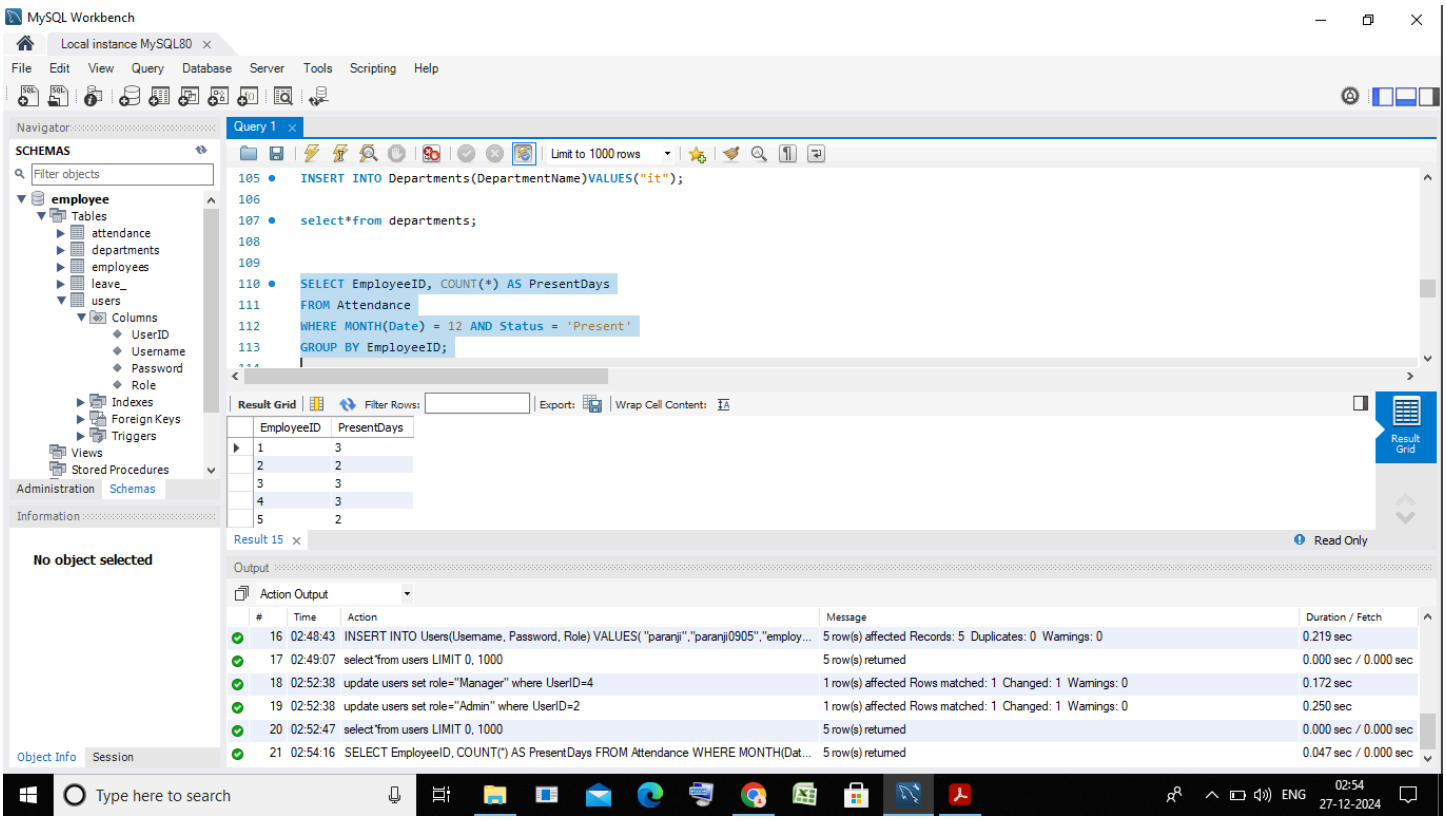
Present days Count :

SELECT EmployeeID, COUNT(*) AS PresentDays

FROM Attendance

WHERE MONTH(Date) = 12 AND Status = 'Present'

GROUP BY EmployeeID;



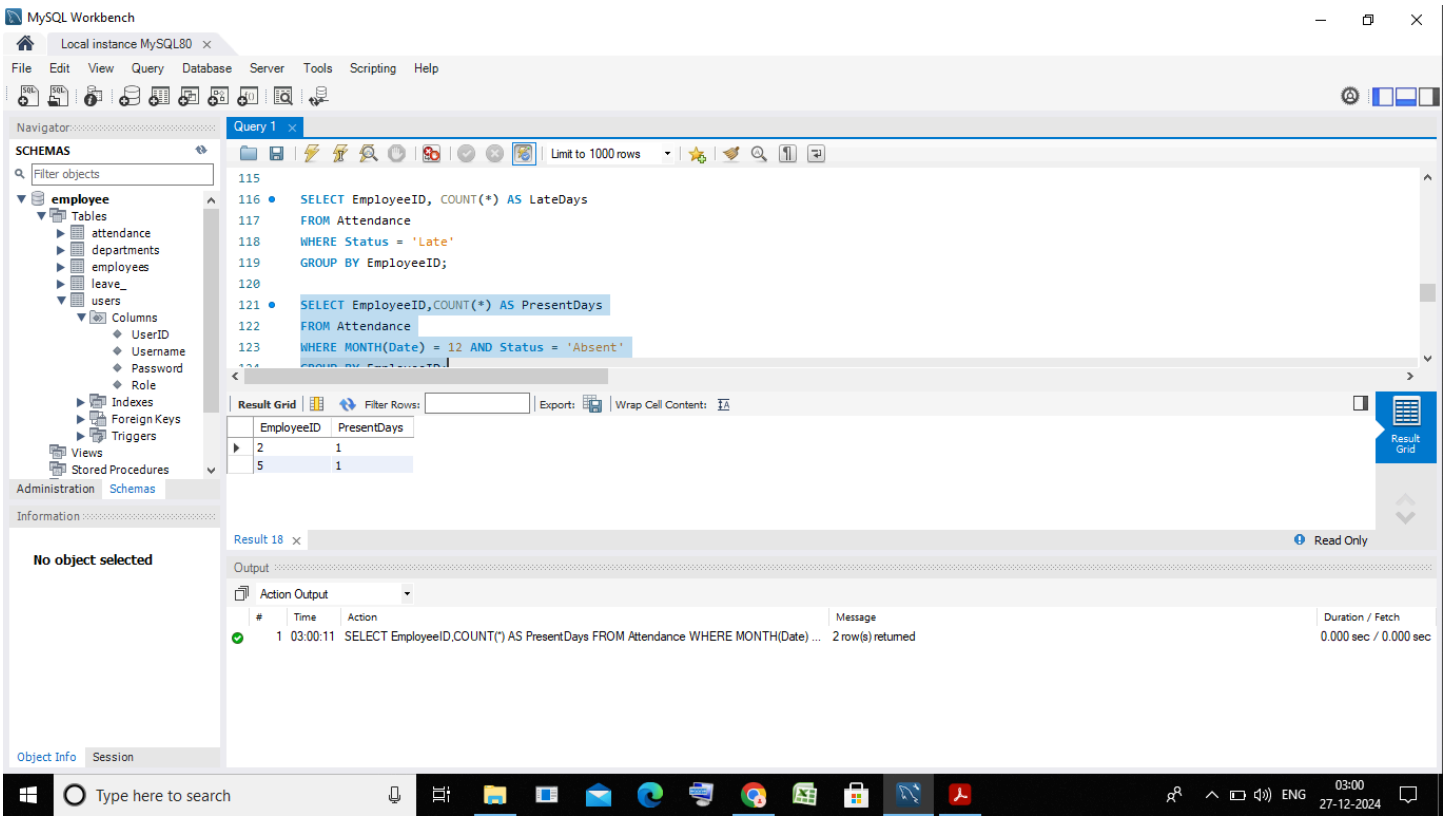
Absent days Count :

SELECT EmployeeID,COUNT(*) AS PresentDays

FROM Attendance

WHERE MONTH(Date) = 12 AND Status = 'Absent'

GROUP BY EmployeeID;



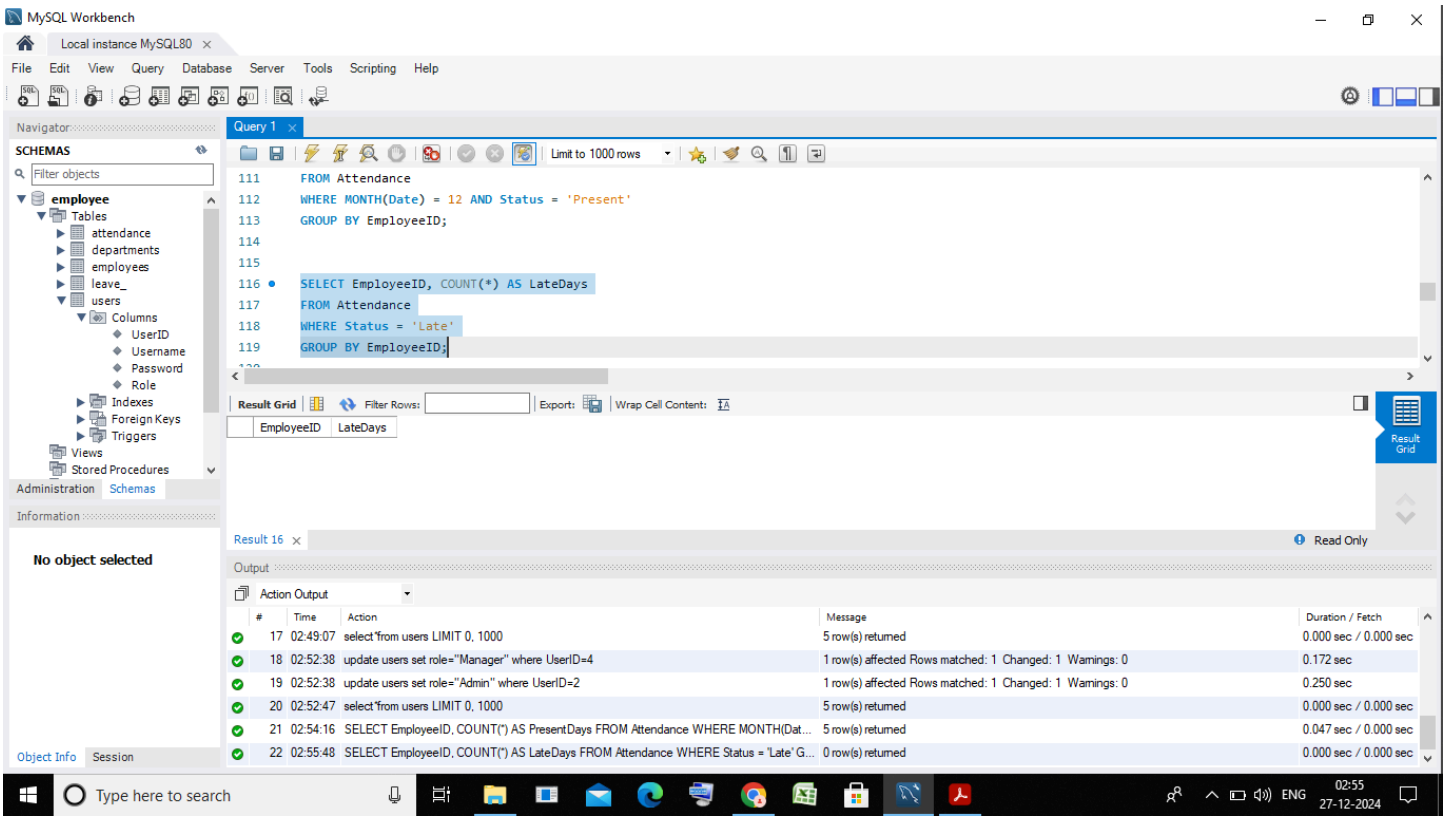
Late Count :

SELECT EmployeeID, COUNT(*) AS LateDays

FROM Attendance

WHERE Status = 'Late'

GROUP BY EmployeeID;



Additional Features

- Use **triggers** to auto-update employee leave balances.
- Add **stored procedures** for recurring reports.
- Implement role-based access control with **views**.