Assignment no.1

**Aim:** Install and set up MySQL, Create a database and a table to store employee details. Perform basic operations like INSERT & DELETE

**Code:** CREATE TABLE employees (

Emp\_id INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(50),

Age INT,

Department VARCHAR(50),

Salary DECIMAL(10,2)

);

INSERT INTO employees (Name, Age, Department, Salary) VALUES

('shubham paithankar', 18, 'CY', 80000.00),

('rohit sharma', 42, 'HR', 4500.00),

('surya yadav', 35, 'Finance', 60000.00),

('Hardik pandya', 18, 'CSE', 70000.00),

('Shantanu sonawne', 42, 'AIML', 90000.00),

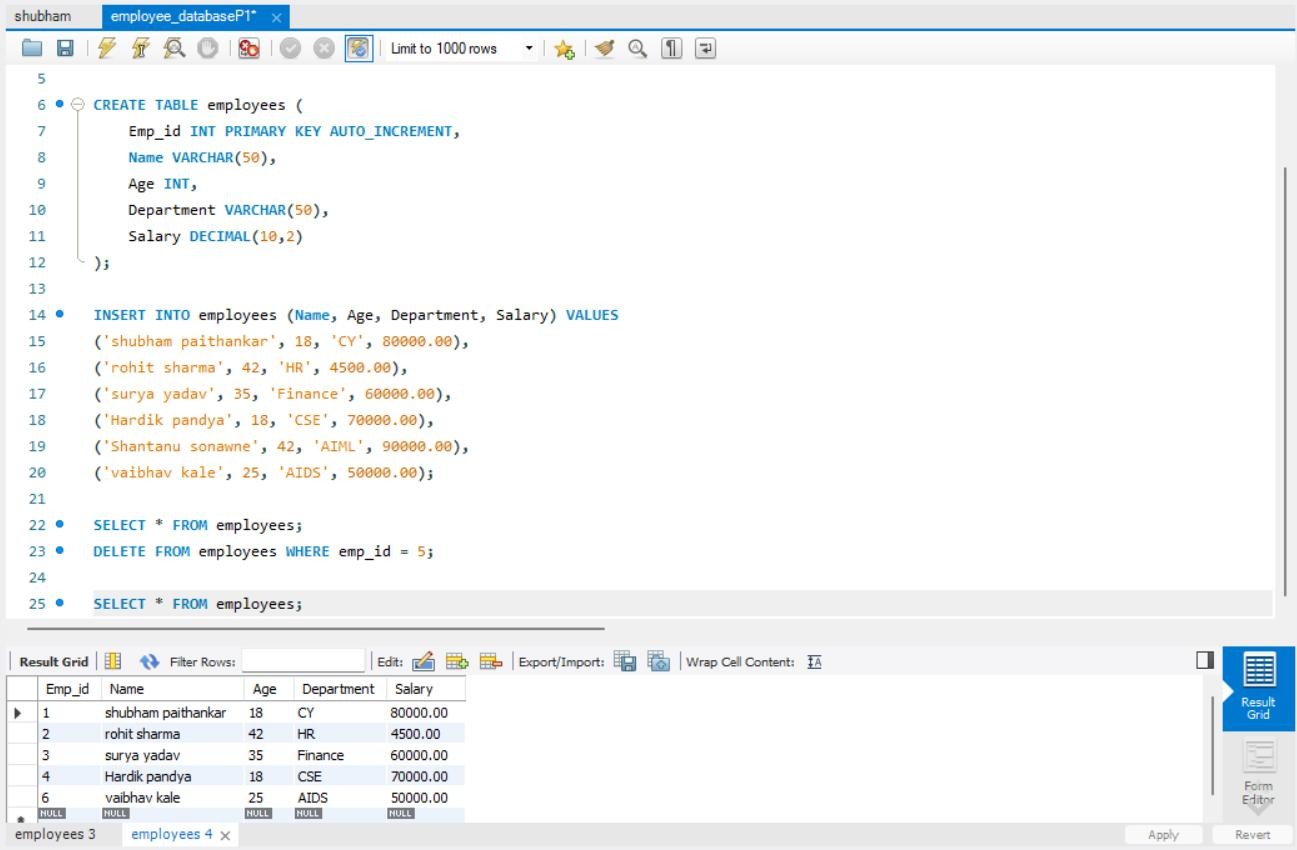
('vaibhav kale', 25, 'AIDS', 50000.00);

SELECT \* FROM employees;

DELETE FROM employees WHERE emp\_id = 5;

SELECT \* FROM employees;

**Screenshot(Output):**



# Assignment no.2

**Aim:** Create a table for storing student information. Insert sample data and perform basic operations: INSERT, UPDATE, DELETE, and SELECT

**Code:**

Student\_id INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(50),

Age INT,

Course VARCHAR(50),

Marks INT

);

INSERT INTO students (name, age, course, marks) VALUES

('Shubham Paithankar', 18, 'Cyber Security', 88),

('Suraj Chavhan', 21, 'Mechanical Engineering', 78),

('Prasad Kale', 23, 'Electrical Engineering', 90),

('Sanket Gaikwad', 17, 'Mechanical Engineering', 68),

('Vaibhav Thakur', 22, 'Mechanical Engineering', 78),

('Sushant Nagare', 22, 'Computer science an Engineering', 78);

SELECT \* FROM students;

UPDATE students

SET Marks = 80

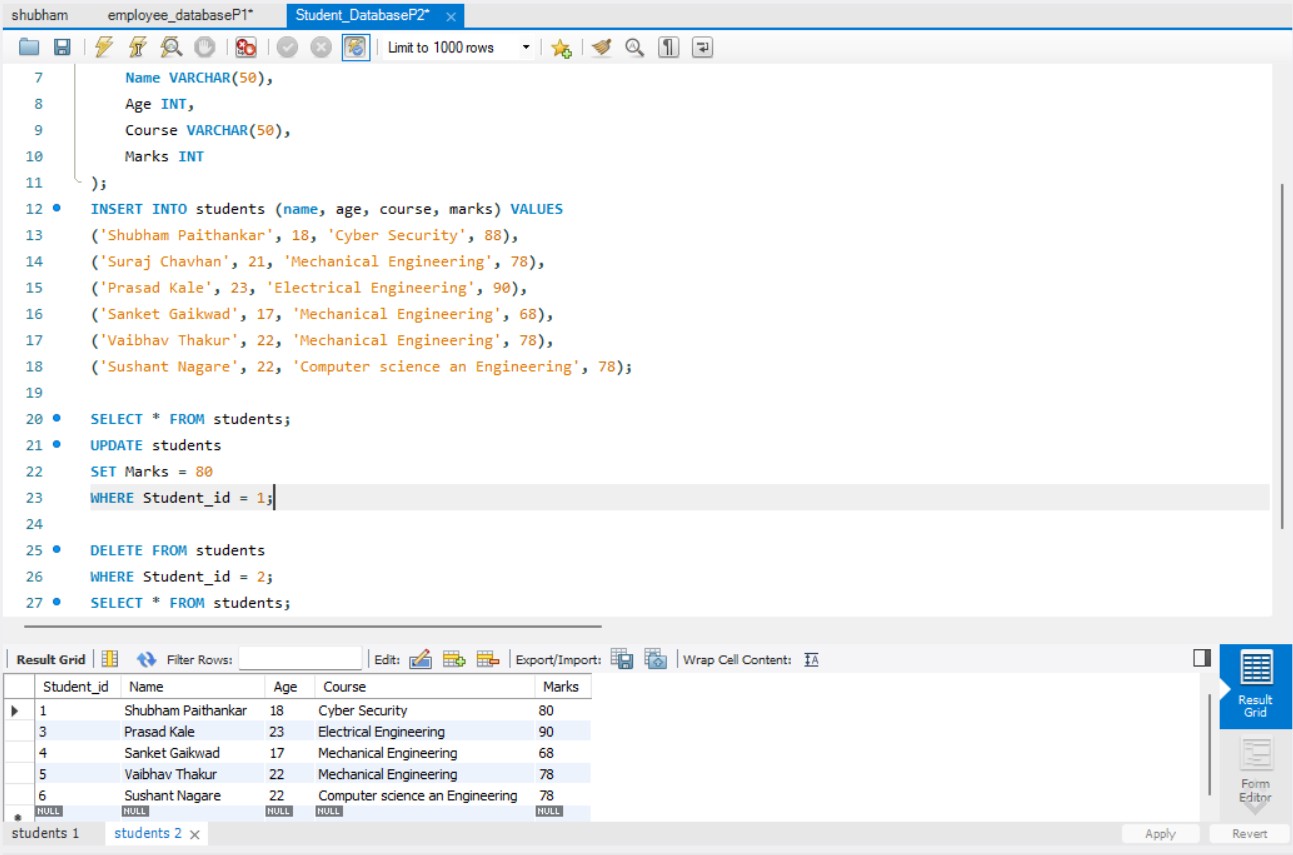
WHERE Student\_id = 1;

DELETE FROM students

WHERE Student\_id = 2;

SELECT \* FROM students;

**Screenshot(Output):**

0

# Assignment no.3

# 

**Aim:**Create a table with columns for EmployeelD. Name, Salary, Joining Date, and ActiveStatus using different data types. Insert sample data and perform queries to manipulate and retrieve data.

**Code:**

CREATE TABLE employees (

EmployeeID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Salary DECIMAL(20,2),

JoiningDate DATE,

ActiveStatus BOOLEAN

);

INSERT INTO employees (Name, Salary, JoiningDate, ActiveStatus) VALUES

('Shubham Paithankar', 95000.00, '2022-06-23', TRUE),

('Rohit Kolhe', 70000.50, '2021-07-25', TRUE),

('Aarav Wable', 50000.75, '2023-01-02', True),

('Gita rajwade', 70000.25, '2019-11-05', TRUE),

('Tejas Gade', 70000.25, '2020-11-05', TRUE);

SELECT \* FROM employees;

UPDATE employees

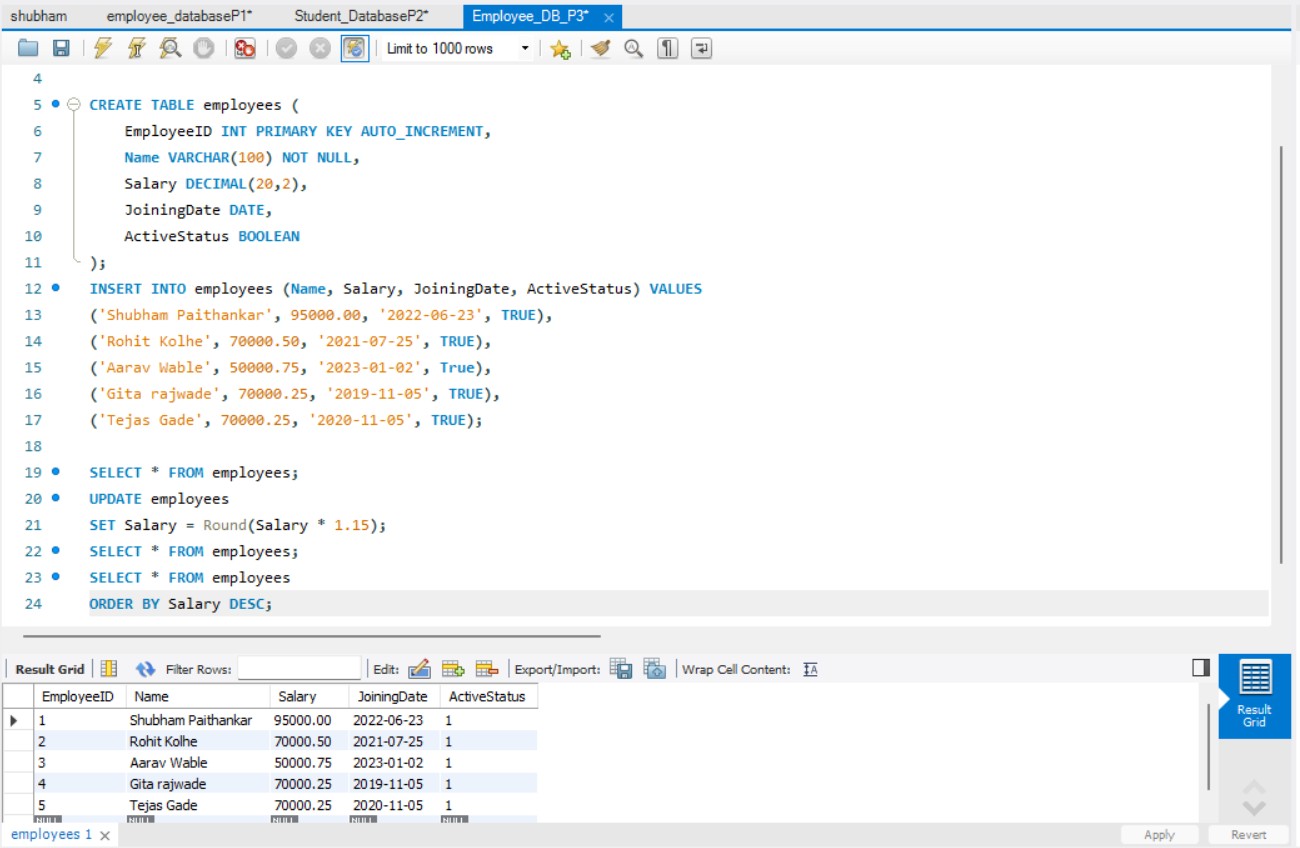
SET Salary = Round(Salary \* 1.15);

SELECT \* FROM employees;

SELECT \* FROM employees

ORDER BY Salary DESC;

**Screenshot(Output):**



# Assignment no.4

**Aim:** Create a table to store employee information with constraints like Primary Key, Foreign Key, and Unique. Insert valid and invalid data to test the constraints.

**Code**:

CREATE TABLE Department (

DeptID INT PRIMARY KEY,

DeptName VARCHAR(50) UNIQUE

);

CREATE TABLE Employee (

EmpID INT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

Email VARCHAR(100) UNIQUE,

Salary DECIMAL(10,2) CHECK (Salary > 0),

DeptID INT REFERENCES Department(DeptID)

);

INSERT INTO Department (DeptID, DeptName) VALUES (1, 'HR');

INSERT INTO Department (DeptID, DeptName) VALUES (2, 'IT');

INSERT INTO Employee (EmpID, Name, Email, Salary, DeptID) VALUES (101, 'Shubham','shubham.paithankar24@sanjivani.edu.in', 55000.00, 1);

INSERT INTO Employee (EmpID, Name, Email, Salary, DeptID) VALUES

(102,'Shubham','shubham.sonawne24@sanjivani.edu.in',50000.00,2);

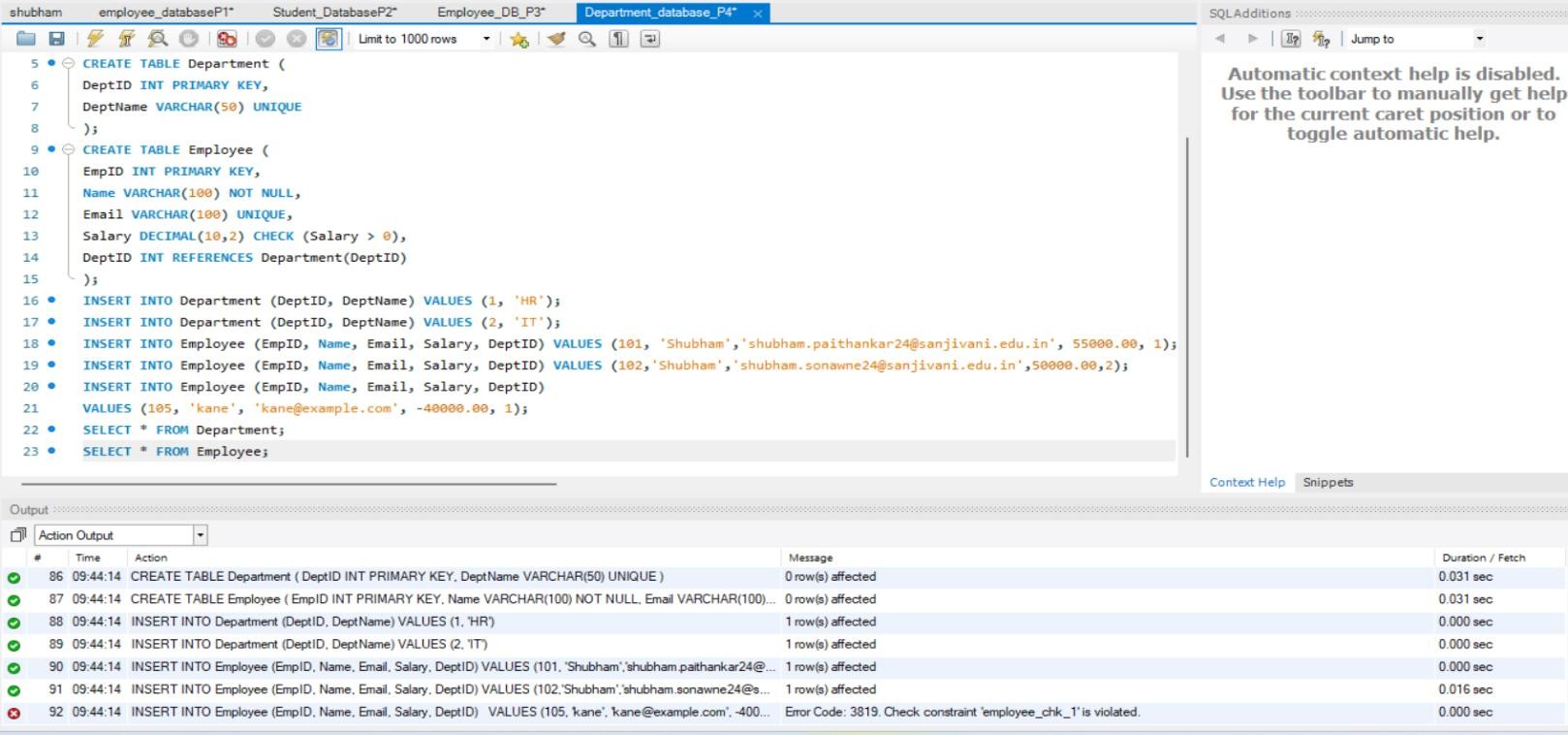
INSERT INTO Employee (EmpID, Name, Email, Salary, DeptID)

VALUES (105, 'kane', 'kane@example.com', -40000.00, 1);

SELECT \* FROM Department;

SELECT \* FROM Employee;

**Output:(Error Screenshot):**



# Assignment no.5

**Aim:** Create a table for Customer details with various integrity constraints like NOT NULL, CHECK, and DEFAULT. Insert valid and invalid data to test these constraints and ensure data integrity.

**Code:** drop database Customer\_db; create database Customer\_db; use Customer\_db; CREATE TABLE Customer (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(100) NOT NULL,

LastName VARCHAR(100) NOT NULL,

Email VARCHAR(100) UNIQUE,

Phone VARCHAR(15),

Age INT CHECK (AgE >= 18),

IsActive BOOLEAN DEFAULT TRUE

);

**--Inserting A Valid Data Into Customer table**

INSERT INTO Customer (CustomerID, FirstName, LastName, Email, Phone, Age, IsActive)

VALUES (1, 'shubham','Paithankar','shubham.paithankar24@sanjivani.edu.in', '7020268056', 18, TRUE);

INSERT INTO Customer (CustomerID, FirstName, LastName, Email, Phone, Age)

VALUES (2, 'Rohit', 'Kale', 'rohit@example.com', '9956873490', 27);

SELECT \* FROM Customer;

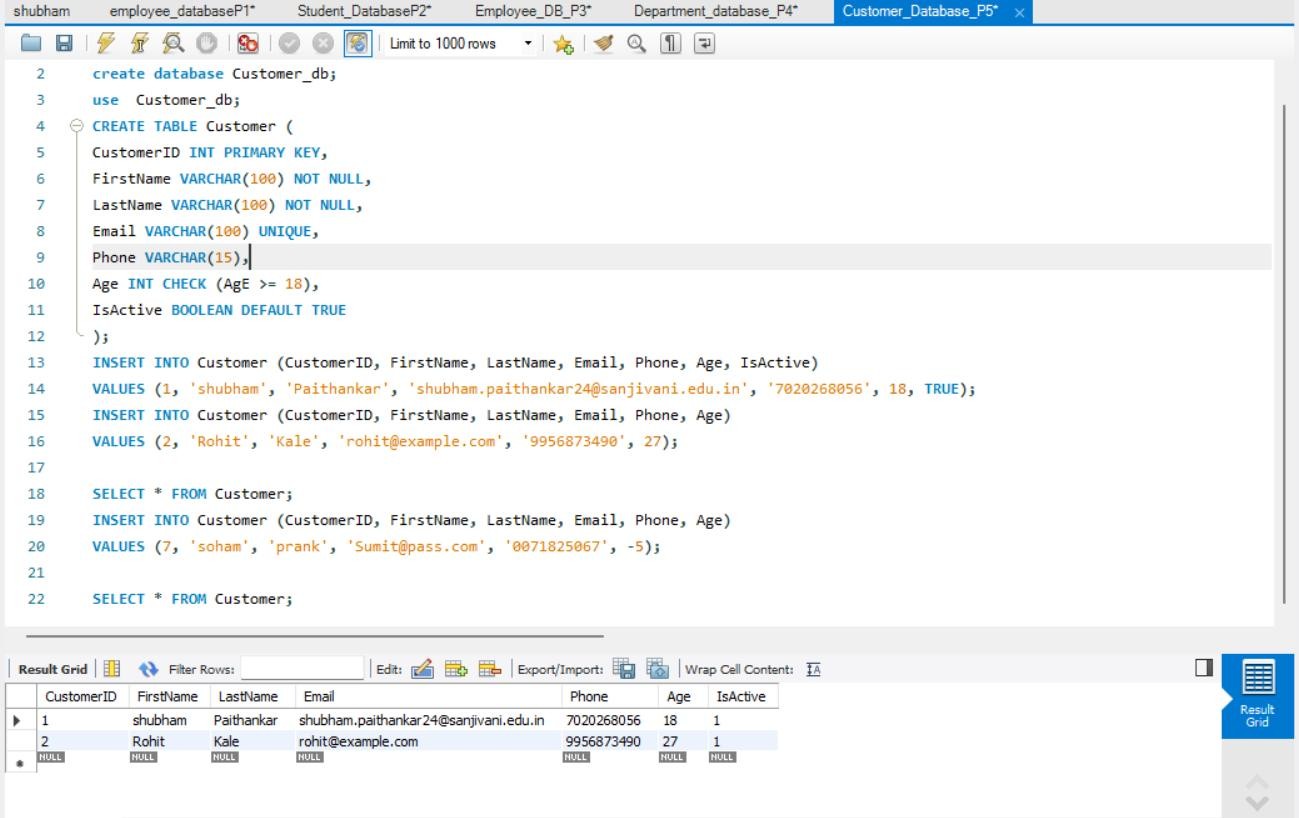
**--Inserting invalid Data to test constrains**

INSERT INTO Customer (CustomerID, FirstName, LastName, Email, Phone, Age)

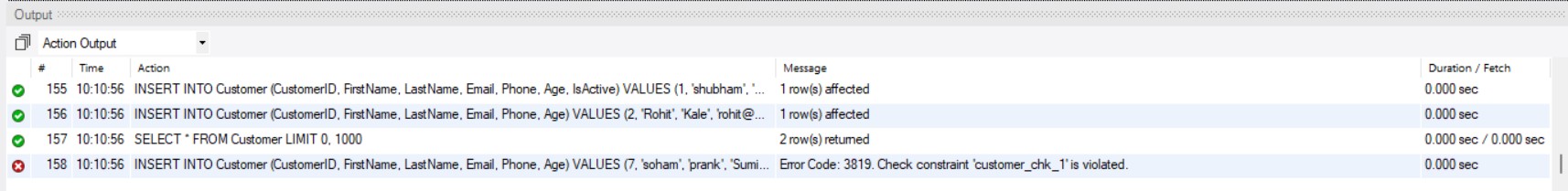
VALUES (7, 'soham', 'prank', 'Sumit@pass.com', '0071825067', -5);

SELECT \* FROM Customer;

**Screenshot(Output):**



**Error(Screenshot):**



# Assignment no.6

**Aim:** Use DDL commands to create tables and DML commands to insert, update, and delete data. Write SELECT queries to retrieve and verify data changes.

**Code:** CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Age INT,

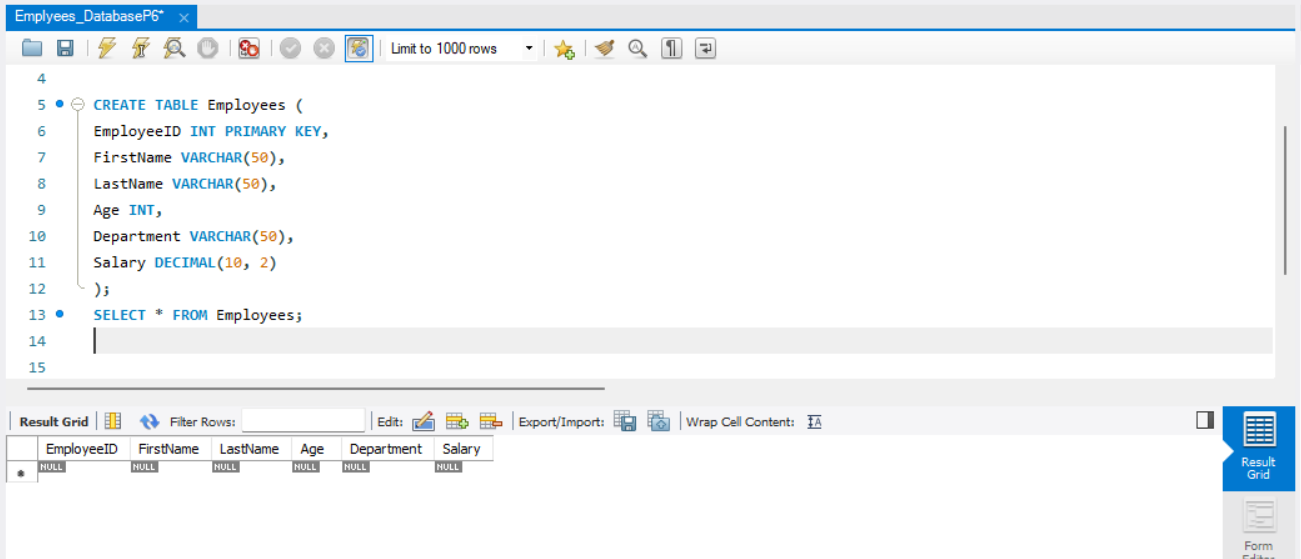
Department VARCHAR(50),

Salary DECIMAL(10, 2)

);

SELECT \* FROM Employees;

**Output:(Screenshot):**



**Data Insertion (Using DML Command):**

**--Inserting Data into Customer Table**

INSERT INTO Employees (EmployeeID, FirstName, LastName, Age, Department,

Salary)

VALUES (1, 'Shubham', 'Paithankar', 18, 'HR', 95000.00);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Age, Department,

Salary)

VALUES (2, 'Sameer', 'shaik', 28, 'IT', 65000.00);

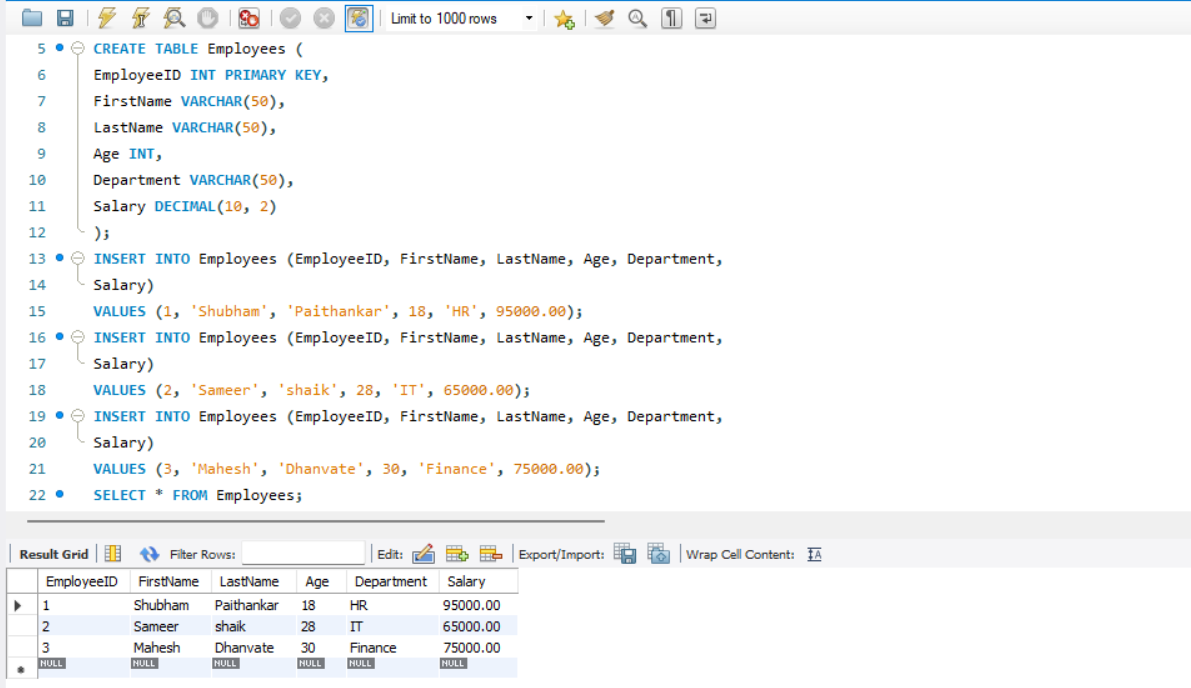
INSERT INTO Employees (EmployeeID, FirstName, LastName, Age, Department,

Salary)

VALUES (3, 'Mahesh', 'Dhanvate', 30, 'Finance', 75000.00);

SELECT \* FROM Employees;

**Output:(Screenshot):**

****

**--Updating Data into Customer Table**

* Update a single column
* Update multiple columns for a specific row
* Update entire tuple
* Update with a condition
* Update with a subquery
* Update using a CASE statement

**Code:(Updation):**

**-- 1. Update a single column (e.g., update salary for EmployeeID 1)**

UPDATE Employees

SET Salary = 70000.00

WHERE EmployeeID = 1;

SELECT \* FROM Employees;

**-- 2. Update multiple columns for a specific row (e.g., update name and salary for EmployeeID 2)**

UPDATE Employees

SET FirstName = 'James', LastName = 'Will', Salary = 75000.00

WHERE EmployeeID = 2;

SELECT \* FROM Employees;

**-- 3. Update entire tuple (all columns for EmployeeID 3)**

UPDATE Employees

SET FirstName = 'Michael', LastName = 'Bro', Age = 40, Department =

'Management', Salary = 80000.00

WHERE EmployeeID = 3;

SELECT \* FROM Employees;

**-- 4. Update with a condition (e.g., increase salary by 10% for EmployeeID=1)**

UPDATE Employees

SET Salary = Salary \* 1.10

WHERE EmployeeID = 1;

SELECT \* FROM Employees;

**-- 5. Update using a CASE statement (e.g., increase salary based on Employee Id)**

UPDATE Employees

SET Salary = CASE

WHEN EmployeeID = 1 THEN Salary \* 1.05

WHEN EmployeeID = 2 THEN Salary \* 1.08

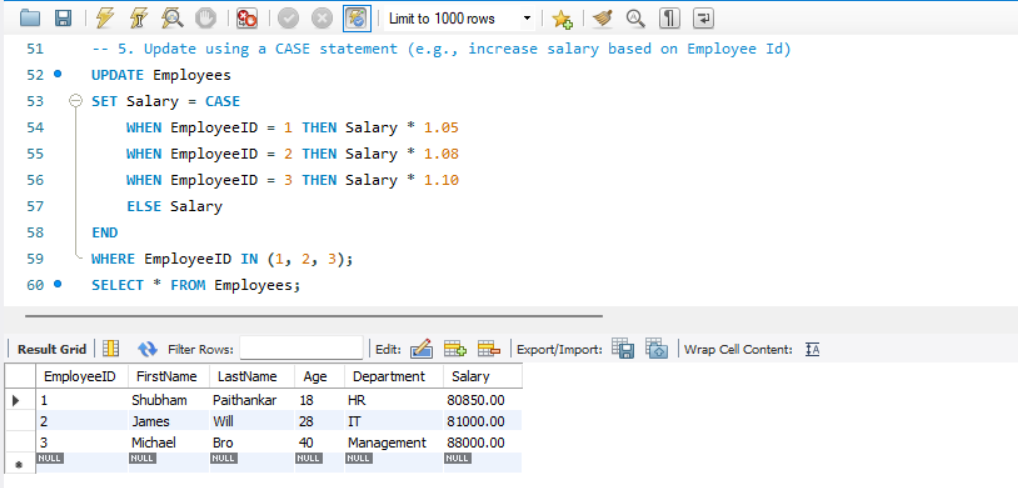
WHEN EmployeeID = 3 THEN Salary \* 1.10

ELSE Salary END

WHERE EmployeeID IN (1, 2, 3);

SELECT \* FROM Employees;

**Output:(Screenshot):**



**--Deleting Data into Customer Table (DML Command)**

DELETE FROM Employees

WHERE EmployeeID = 3;

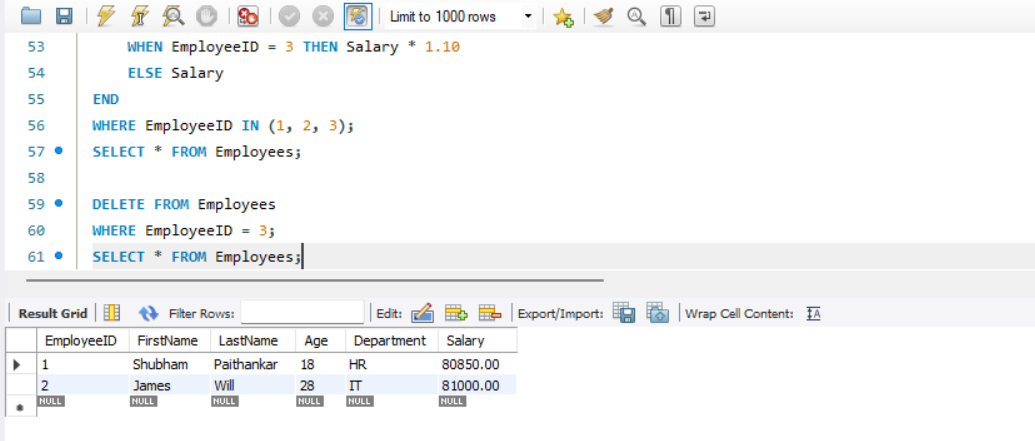
SELECT \* FROM Employees;

* **Select and Verify Data (SELECT Query)**

**-- To retrieve all data from the table**

SELECT \* FROM Employees;

**Output:(Screenshot):**

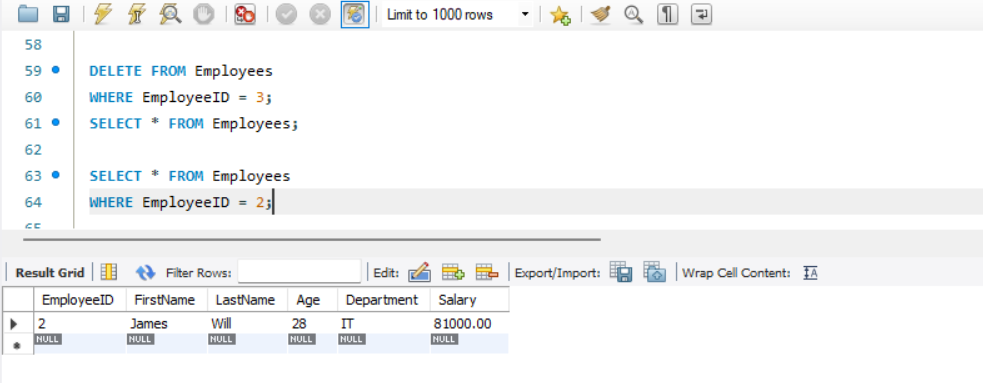


**-- To verify the update (checking updated values for EmployeeID 2)**

SELECT \* FROM Employees

WHERE EmployeeID = 2;

**Output:(Screenshot):**



**-- To verify the deletion (checking if EmployeeID 1 exists)**

SELECT \* FROM Employees

WHERE EmployeeID = 3;

**Output:(Screenshot):**

