Name: Manju

Date: 04/07/2025

Batch: May 25th Batch

This document contains SQL queries performed for Week 6, involving data updates and deletions on an Employees table.

A new table was created with sample employee data.

Columns LastUpdated and Status were added.

Records were updated based on EmployeeID and Salary.

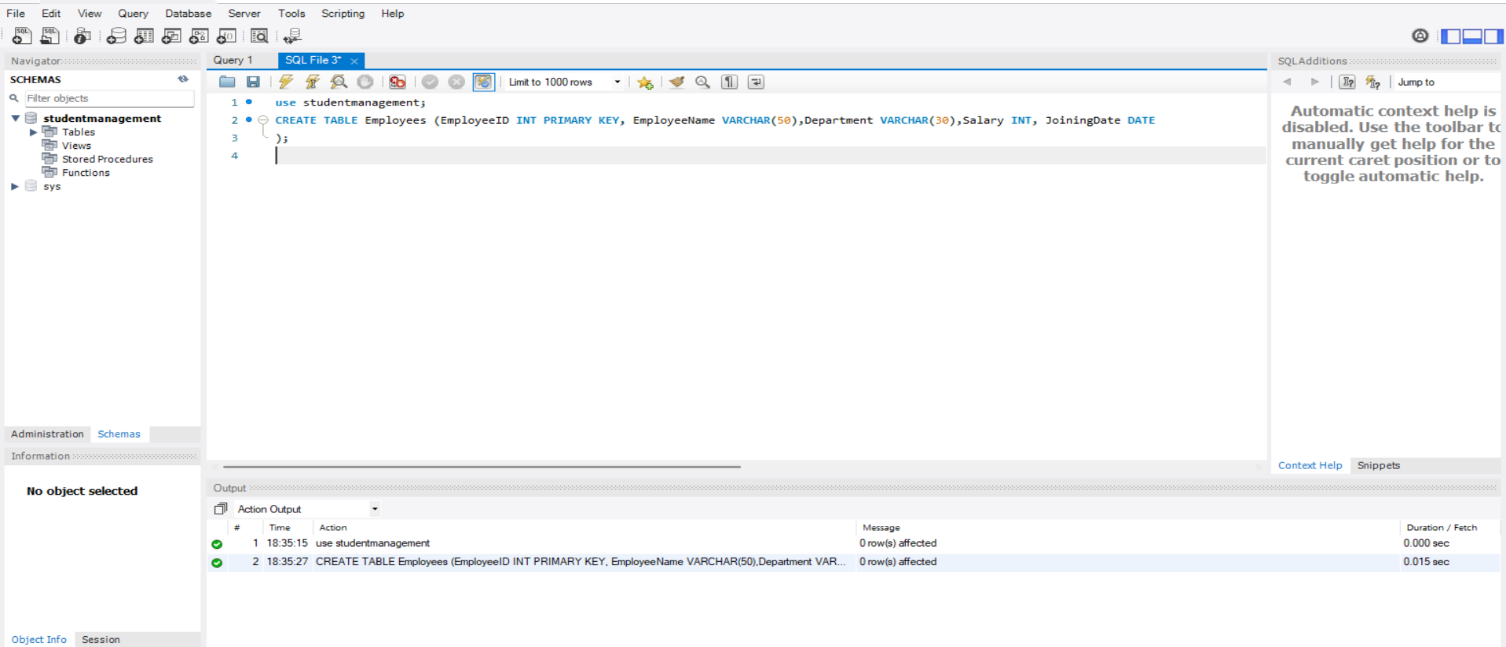
Employees with salary < 25,000 were marked as Inactive and deleted.

Records with LastUpdated before 2025 were also removed.

Screenshots of queries and outputs are attached as per submission requirements.

CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, EmployeeName VARCHAR(50),Department VARCHAR(30),Salary INT, JoiningDate DATE

);

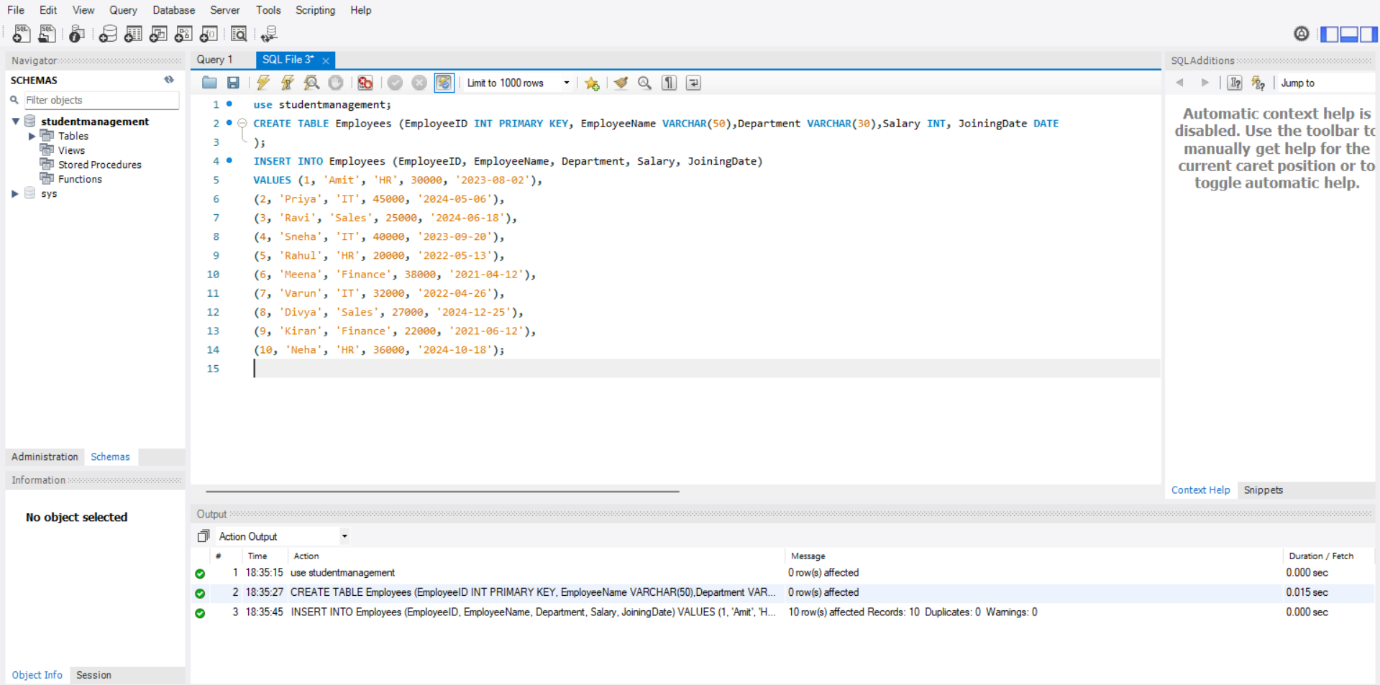
**Output:**

**Purpose:** To create a new table named Employees to store employee-related details.

**Explanation:** Defines five columns: a unique EmployeeID, name, department, salary, and joining date.

INSERT INTO Employees (EmployeeID, EmployeeName, Department, Salary, JoiningDate)

VALUES(…..);

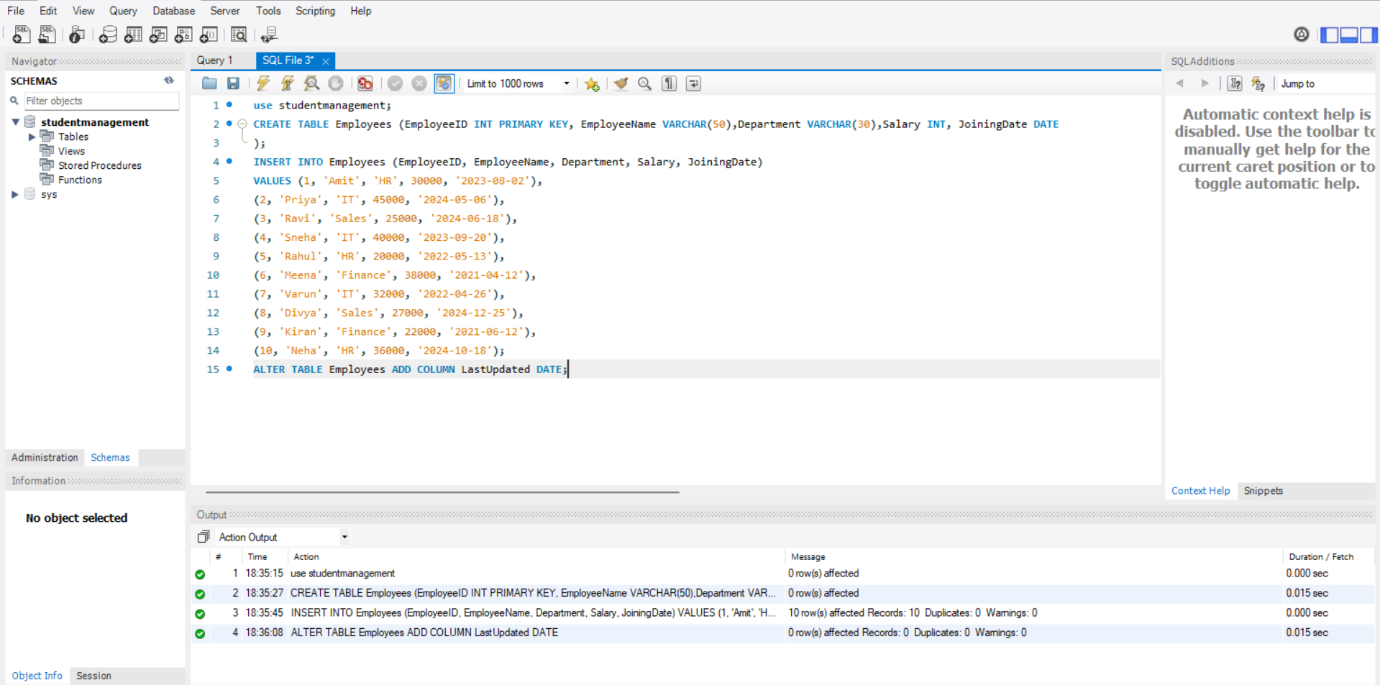
**Output:**

**Purpose:** To insert 10 employee records into the Employees table.

**Explanation:** Populates the table with sample employee data for further processing.

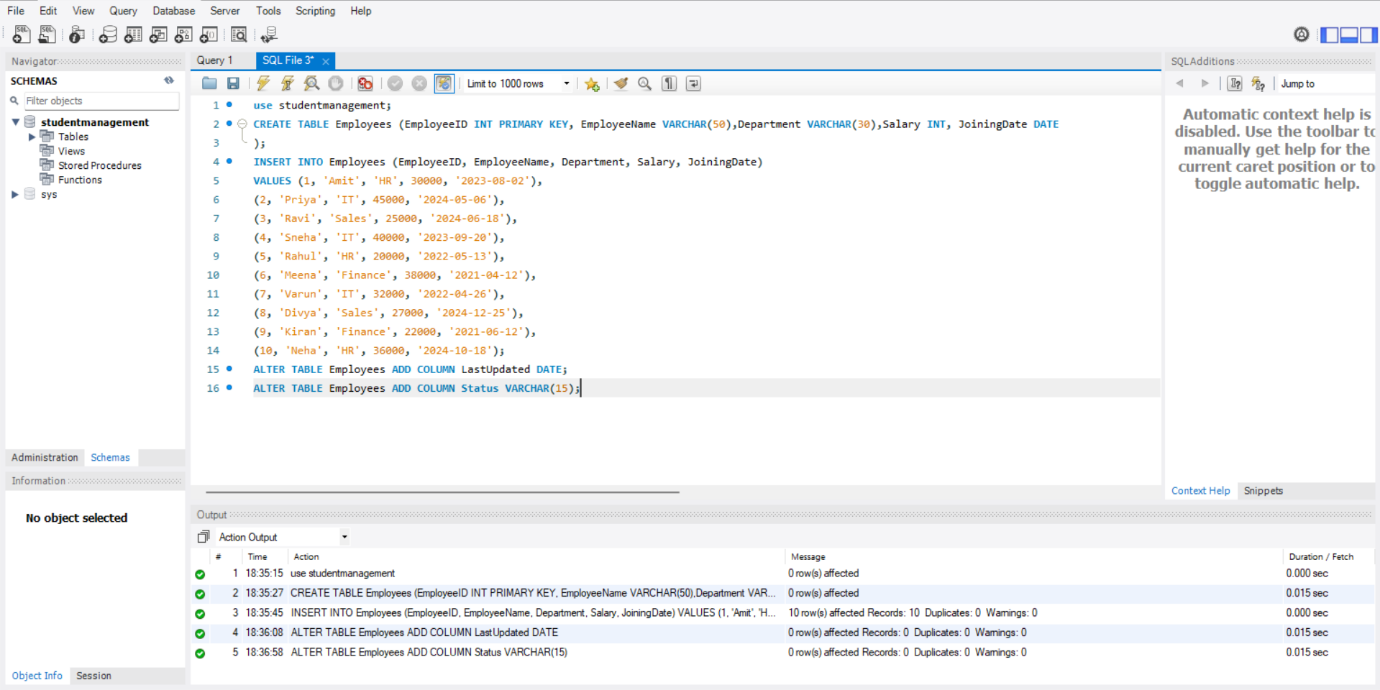
**1.Adding new columns**

ALTER TABLE Employees ADD COLUMN LastUpdated DATE;

**Output:**

**Purpose:** To add a new column LastUpdated to track when an employee record was last updated.

**Explanation:** This column is added by modifying employees table and it will be used in further queries to store date-related updates.

****ALTER TABLE Employees ADD COLUMN Status VARCHAR(15);

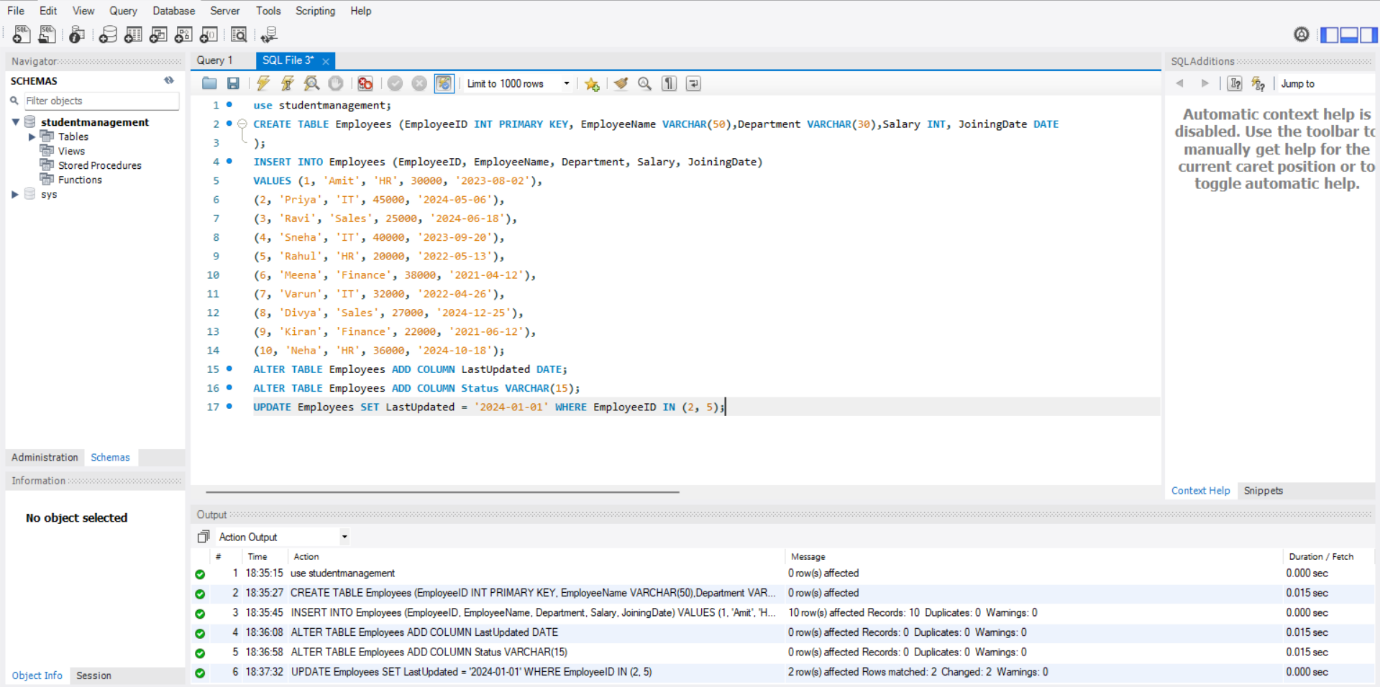
**Output:**

**Purpose:** To add a Status column to mark employees as 'Active' or 'Inactive'.

**Explanation:** Enables flagging of employee status based on logic applied in future queries.

**Updating status of employees**

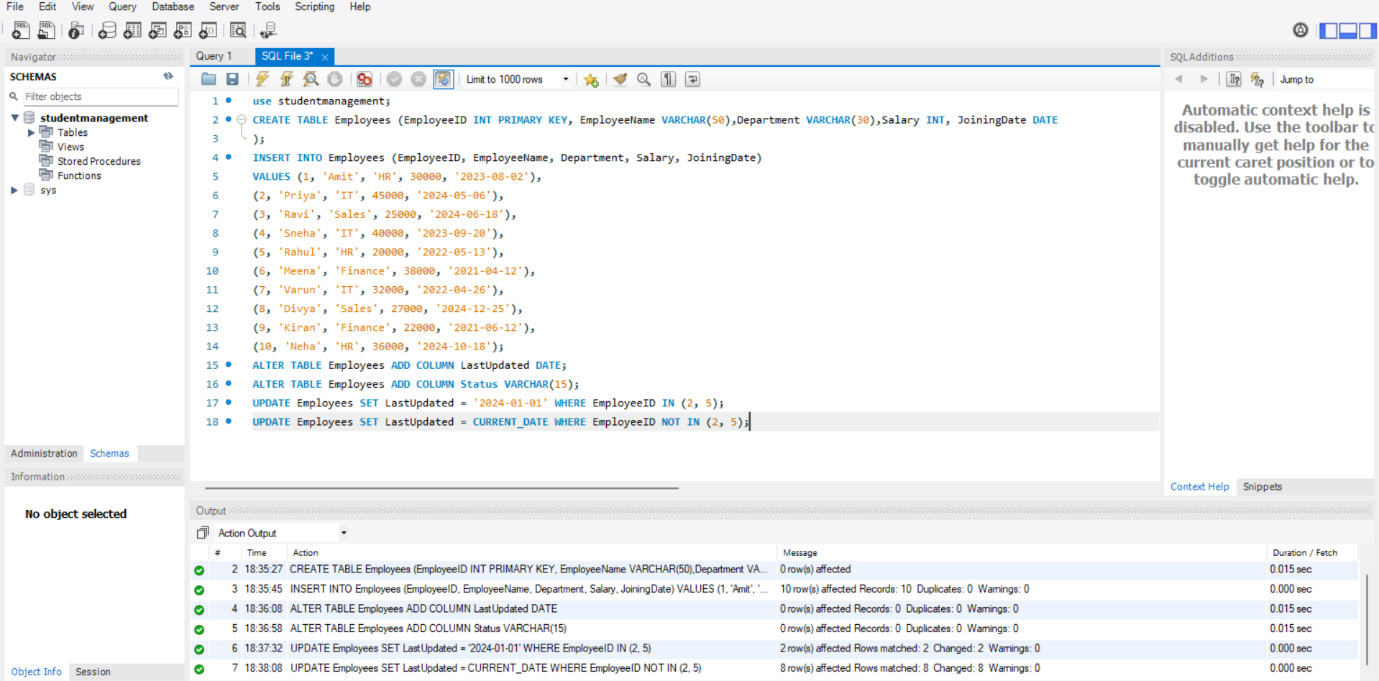
UPDATE Employees SET LastUpdated = '2024-01-01' WHERE EmployeeID IN (2, 5);

**Output:**

**Purpose:** To manually assign a fixed last updated date to employees with ID 2 and 5.

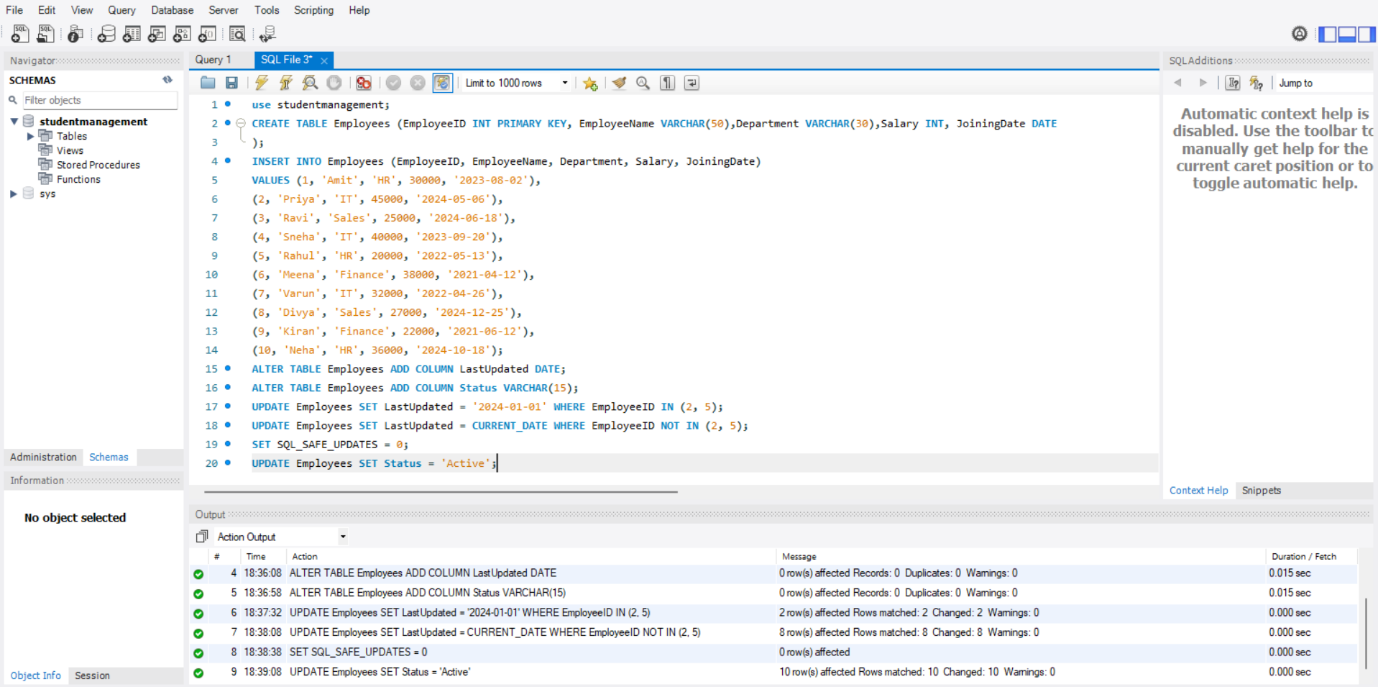
**Explanation:** Demonstrates control over selective record updates based on ID.

UPDATE Employees SET LastUpdated = CURRENT\_DATE WHERE EmployeeID NOT IN (2, 5);

**Output:**

**Purpose: To assign the current date to the LastUpdated field for all other employees.**

**Employees:** Ensures all employee records have a recent update timestamp except ID 2 and 5.

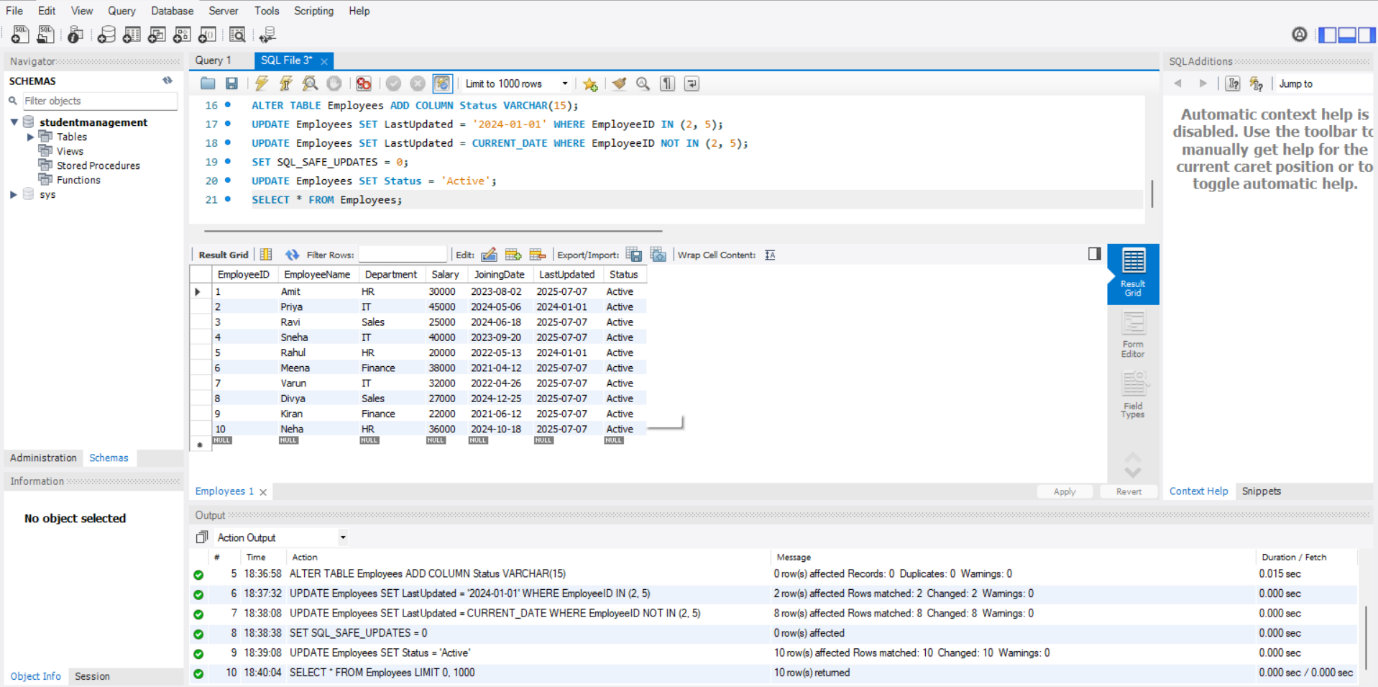
****UPDATE Employees SET Status = 'Active';

**Output:**

**Purpose:** To mark all employees as 'Active' initially.

**Explanation:** Prepares a baseline before applying conditional logic to change statuses.

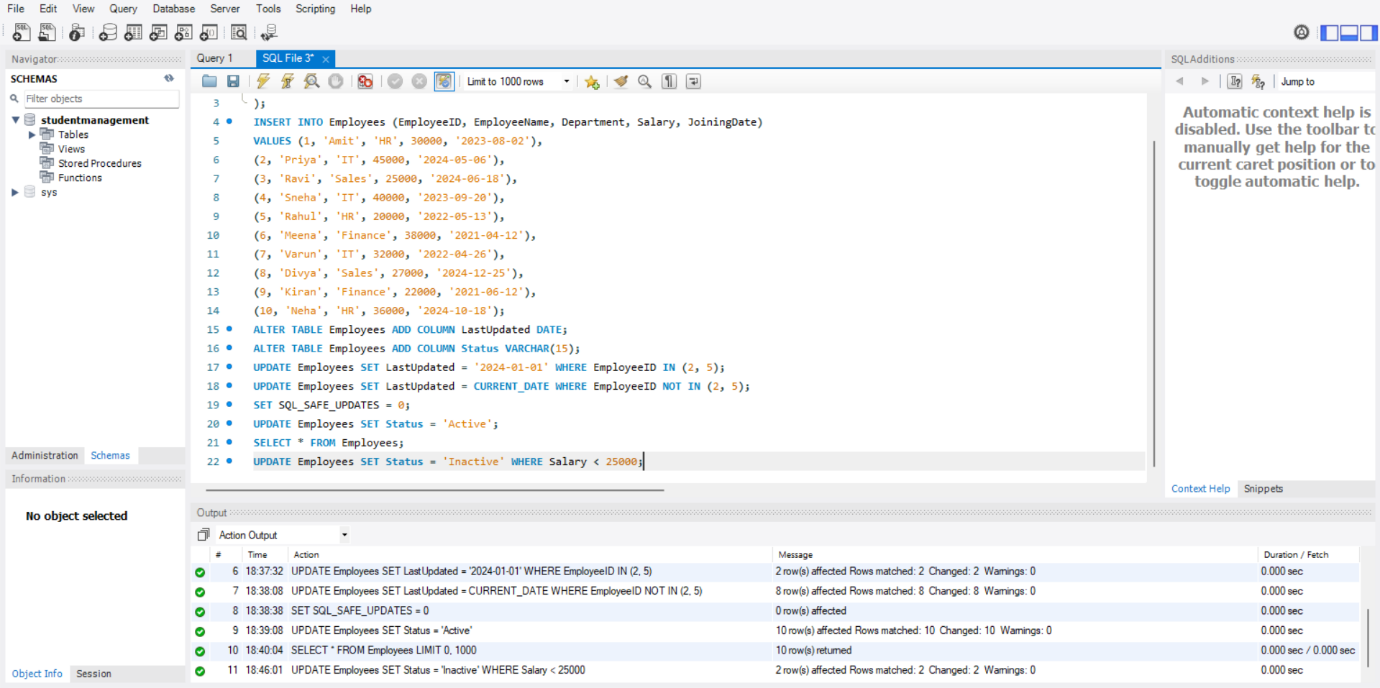
SELECT \* FROM Employees;

**Output:**

**Purpose:** To view all current data in the Employees table.

**Employees:** Confirms the data before and after each operation.

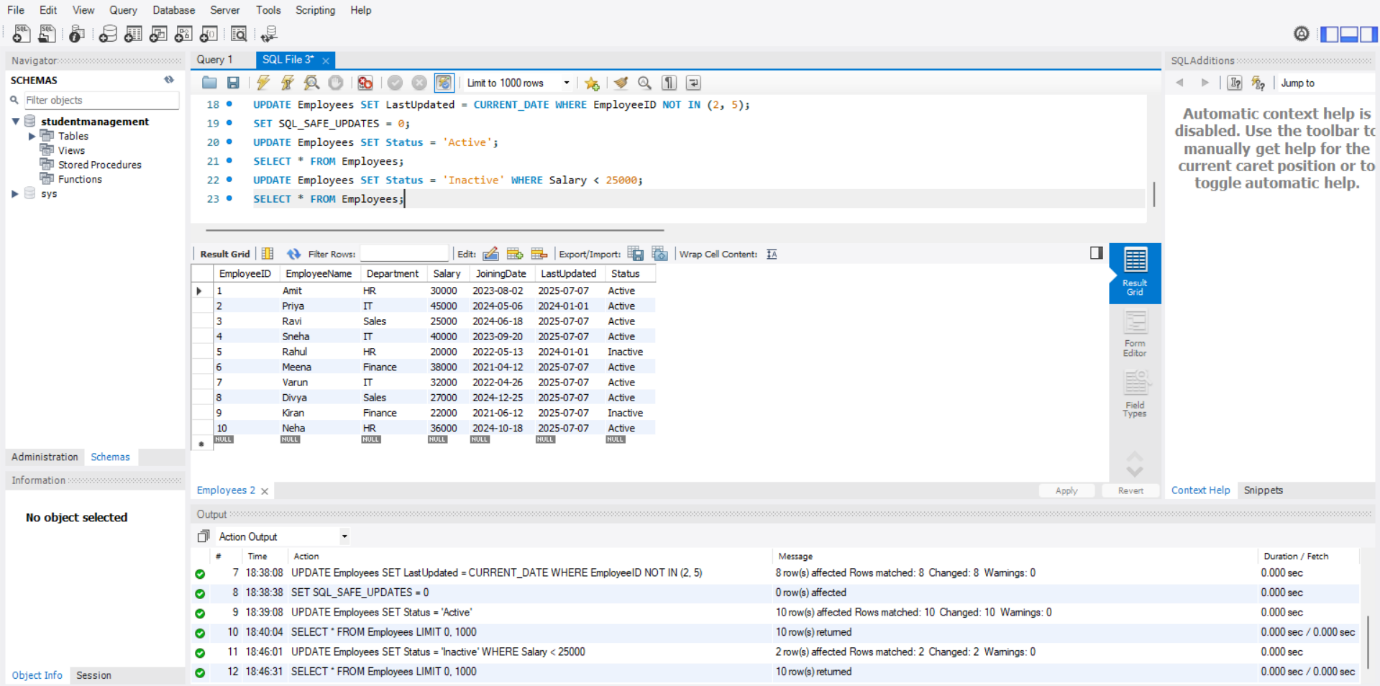
UPDATE Employees SET Status = 'Inactive' WHERE Salary < 25000;

**Output:**

**Purpose:** To mark employees as 'Inactive' if their salary is below 25,000.

**Explanation:** Demonstrates conditional updates using a comparison.

SELECT \* FROM Employees;

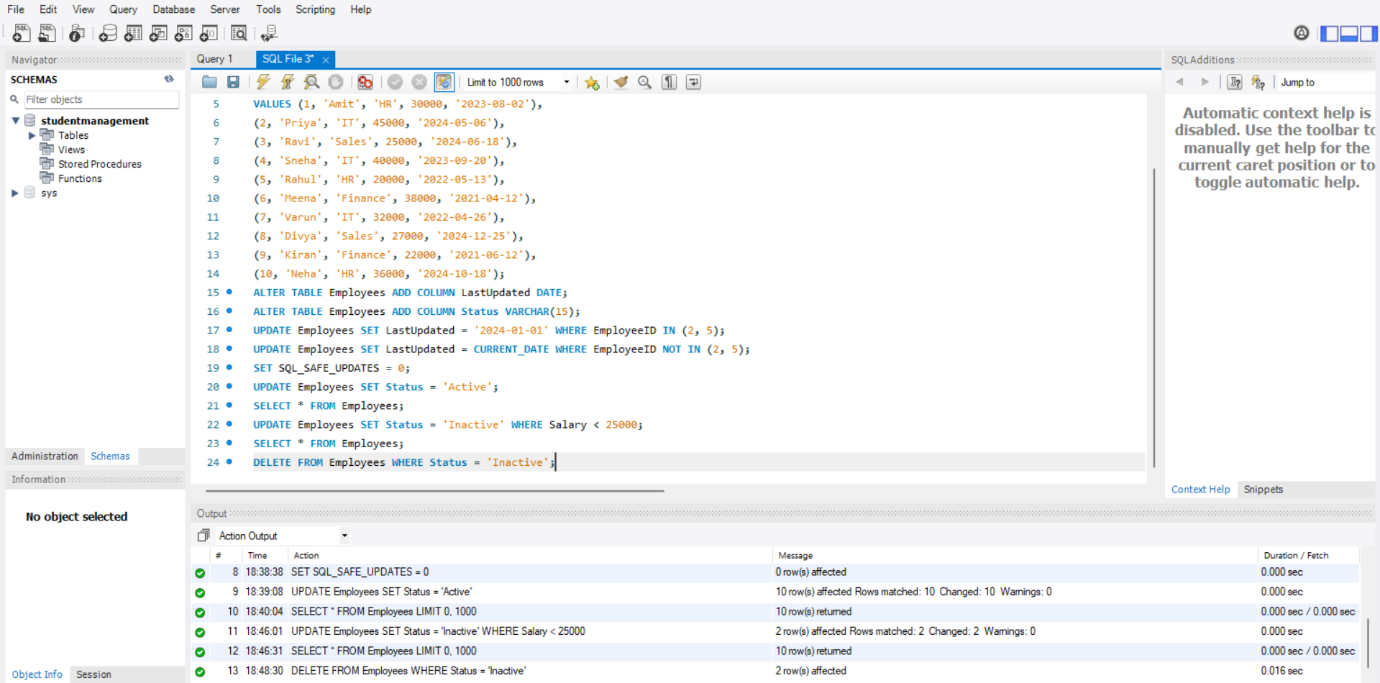
**Output:**

**Purpose:** To verify which employees have been marked as 'Inactive'.

**Explanation:** Helps track changes made in the previous query.

**Deleting records based on certain conditions**

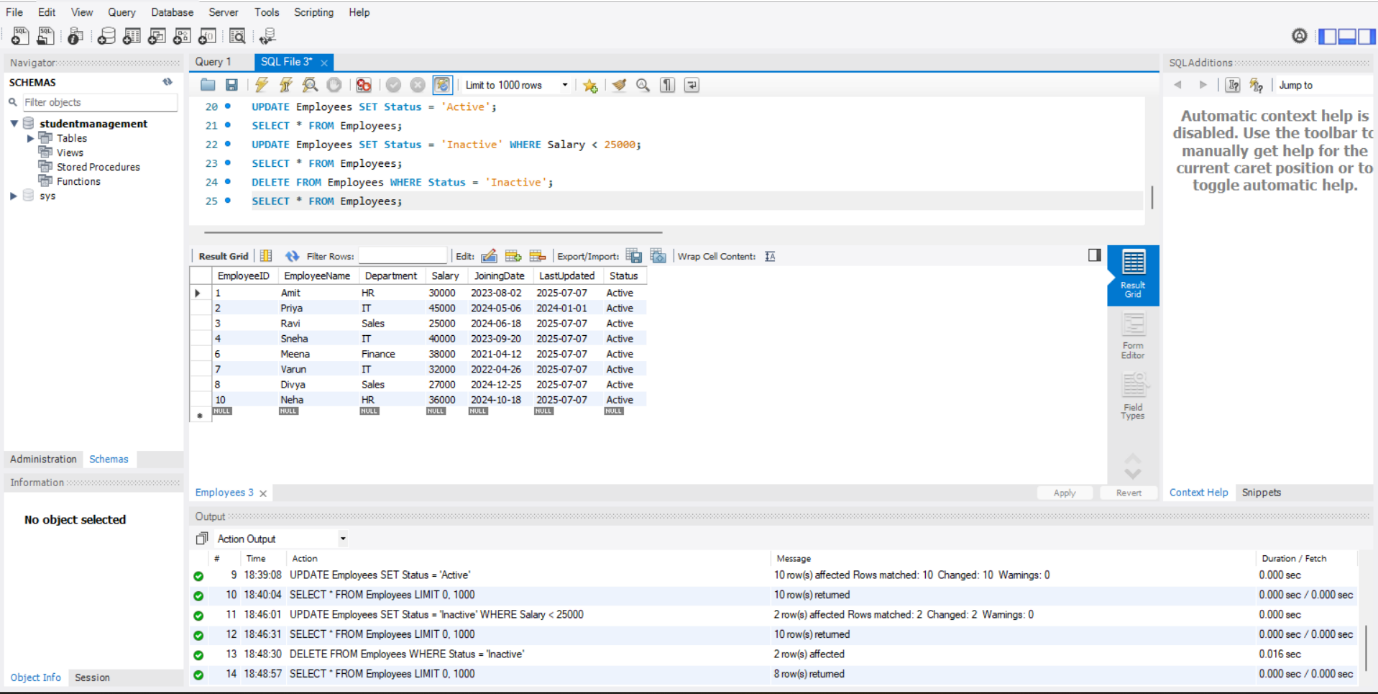
DELETE FROM Employees WHERE Status = 'Inactive';

**Output:**

**Purpose:** To remove employees whose status is 'Inactive'.

**Explanation:** Deletes records from the table.

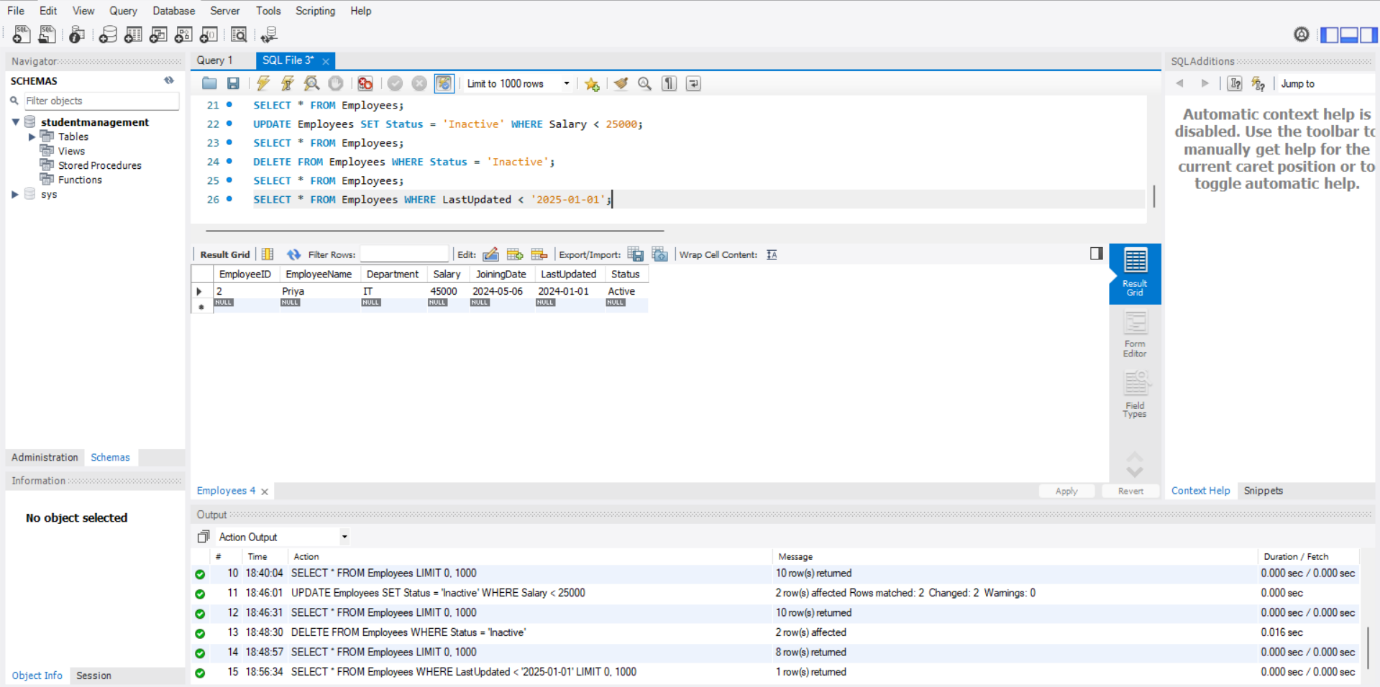
SELECT \* FROM Employees;

**Output:**

**Purpose:** To view the employee records remaining after deletions.

**Explanation:** Confirms the cleanup operation.

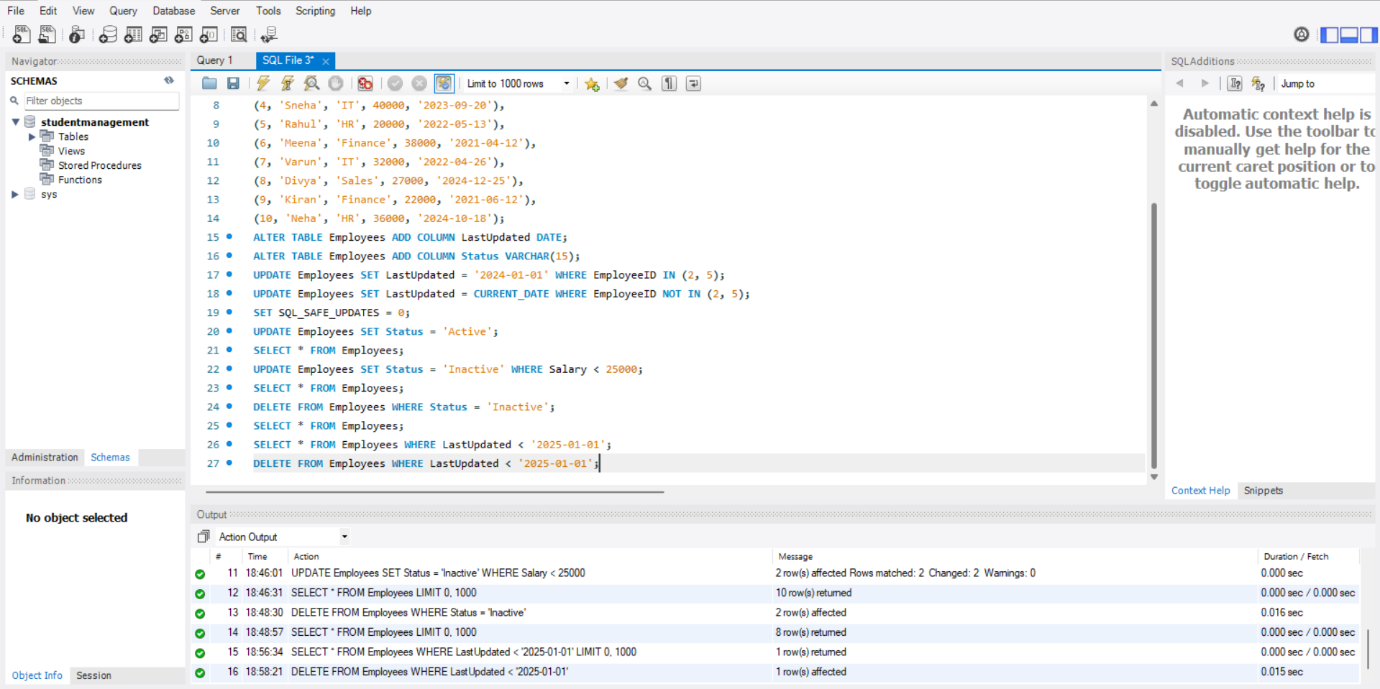
SELECT \* FROM Employees WHERE LastUpdated < '2025-01-01';

**Output:**

**Purpose:** To find employees whose records were updated before 2025.

**Explanation:** Filters out older records based on the LastUpdated date.

DELETE FROM Employees WHERE LastUpdated < '2025-01-01';

**Output:**

**Purpose:** To delete employee records with outdated information.

**Explanation:** Cleans up records that haven’t been updated recently.

SELECT \* FROM Employees;

**Output:**

**Purpose:** To verify the final list of employees remaining after all deletions.

**Explanation:** Final view of the dataset after task operations.

**Summary and Keyfindings:**

In this task, an Employees table was created and populated with 10 sample records. The goal was to perform conditional data updates and deletions using SQL commands. Columns for LastUpdated and Status were added to monitor update timelines and employee activity status. Various operations were performed such as setting last update dates, activating/inactivating employees based on salary conditions, and cleaning up outdated records.

Data Structuring: A structured Employees table was successfully created with fields suitable for tracking employment data.

Status Handling: Employees earning below ₹25,000 were marked as 'Inactive' and later removed from the table demonstrating effective use of conditional filtering and deletion.

Update Tracking: The LastUpdated column helped distinguish between manually and system-updated records using fixed dates and CURRENT\_DATE.

Data Cleanup: Records with outdated LastUpdated values (before 2025) were identified and deleted, showcasing how date-based filtering supports data hygiene.