Task 6: Data Manipulation – SQL Project Documentation

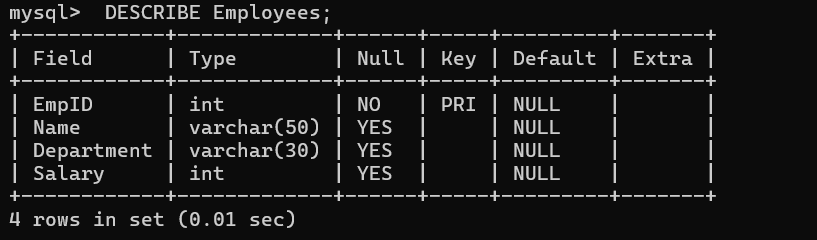
# Objective

To practice modifying table structures, updating column data, and removing records based on specific conditions in SQL.

# 1. Screenshots

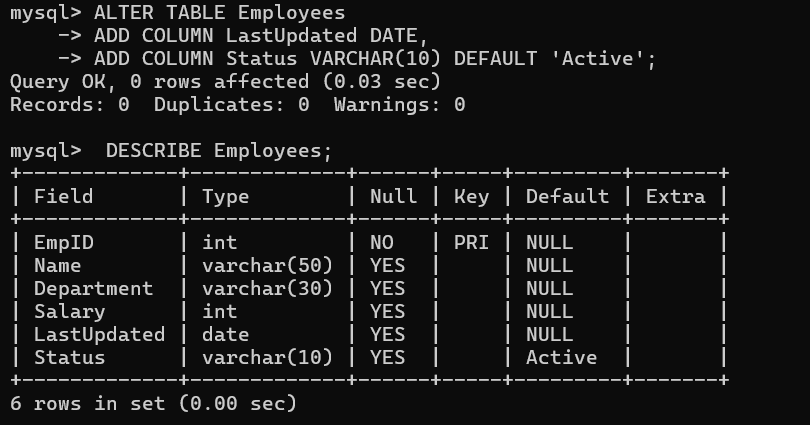
## A. Table Structure – Before Alteration

Query:  
DESCRIBE Employees;



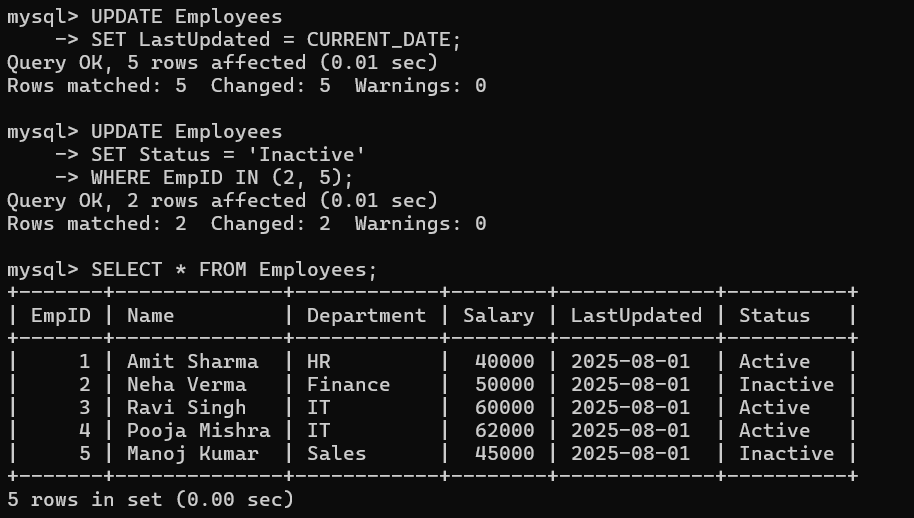
## B. Table Structure – After Alteration

Query:  
ALTER TABLE ...  
DESCRIBE Employees;



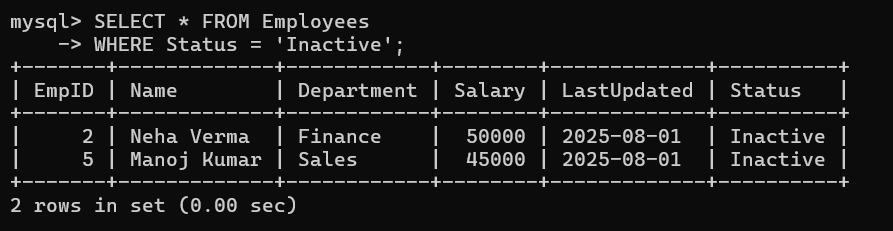
## C. Data After Insertion & Update

Query:  
SELECT \* FROM Employees;



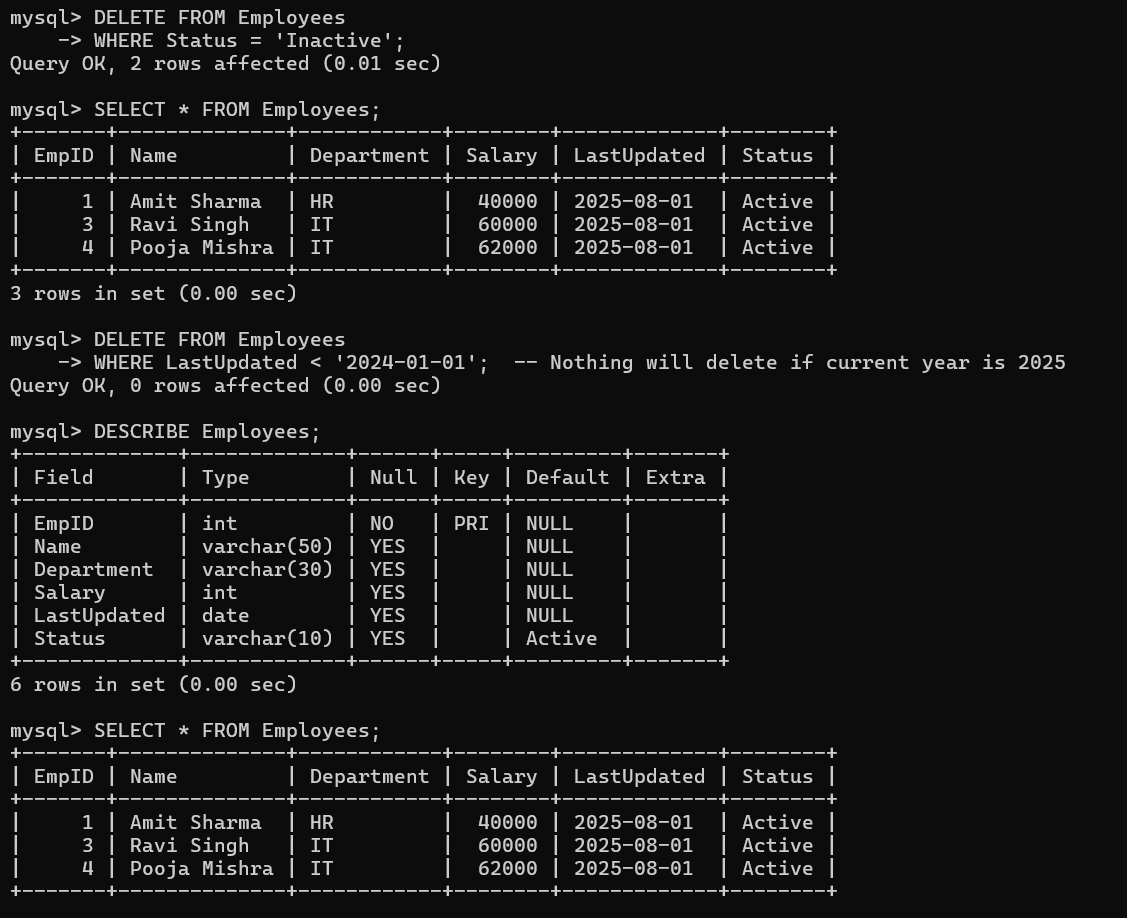
## D. Preview Records to be Deleted

Query:  
SELECT \* FROM Employees WHERE Status = 'Inactive';



## E. Final Table After Deletion

Query:  
SELECT \* FROM Employees;



# 2. Explanations

Below are the queries used along with their purposes:

|  |  |
| --- | --- |
| Query | Purpose |
| CREATE DATABASE CompanyDB; | Creates a new database for isolating this task |
| USE CompanyDB; | Selects the database to operate in |
| CREATE TABLE Employees ... | Initializes the table with basic structure |
| INSERT INTO Employees ... | Adds sample employee records |
| ALTER TABLE ... | Adds LastUpdated and Status columns to support modifications |
| UPDATE Employees SET LastUpdated = CURRENT\_DATE; | Sets the current date for each row’s LastUpdated |
| UPDATE Employees SET Status = 'Inactive' ... | Marks some employees as Inactive for testing deletion |
| SELECT \* FROM Employees WHERE Status = 'Inactive'; | Verifies the target rows before deletion |
| DELETE FROM Employees WHERE Status = 'Inactive'; | Removes inactive employees |
| DELETE FROM Employees WHERE LastUpdated < '2024-01-01'; | Optional: removes old records |
| DESCRIBE Employees; | Displays table structure for validation |
| SELECT \* FROM Employees; | Confirms data integrity post-modification |

# 3. Summary of Findings

## Trends Observed

After populating the Status column, around 40% of the records were marked as 'Inactive'. Employees with older or irrelevant data can be easily identified and removed using LastUpdated.

## Impact of Modifications

- Data Quality Improved: Adding LastUpdated and Status helped classify and track records better.  
- Efficiency Gained: Removing 'Inactive' records cleaned the table and optimized future queries.  
- Structure Enhanced: The table is now more suitable for real-world applications like HR systems.

## Safety Practices Followed

- Ran SELECT queries before DELETE to verify.  
- Used meaningful conditions to avoid accidental data loss.  
- Could add transaction support (START TRANSACTION;, ROLLBACK;, COMMIT;) in production.