Task 6: Data Manipulation

Objective

Enhance skills in SQL by modifying table structures and managing records through adding columns and deleting specific records based on conditions.

Project Steps

1. Modify Table Structure

Task: Add and Populate New Columns

- **Action:** Use the `ALTER TABLE` statement to introduce new columns.
  - Example Columns:
    - `LastUpdated` (DATE) – To record when the row was last modified.
    - `Status` (VARCHAR) – To denote active or inactive records.
- **Populate New Columns:**
  - Use the `UPDATE` statement to set initial values or use default values for the new columns.
  - Example: Set `Status` to "Active" for all existing records.

---

2. Delete Records Based on Conditions

Task: Remove Records Based on Criteria

- **Action:** Use the `DELETE` statement to remove records from the table.
  - Example Conditions:
    - Delete records where `Status` = "Inactive".
    - Remove records older than a specific date in the `LastUpdated` column.
- **Confirm Deletions:**
  - Query the table to ensure that only the intended records were deleted.
  - Optionally, use a `SELECT` statement with the same condition before deletion to review records.

Explanation of Queries

- **ALTER TABLE:**
    - Used to add new columns or modify existing ones.
    - Example:
        - `ALTER TABLE Employees ADD COLUMN Status VARCHAR(10) DEFAULT 'Active';`
- **DELETE:**
    - Removes records based on specified conditions.
    - Example:
        - `DELETE FROM Employees WHERE Status = 'Inactive';`

How to Execute

1. **Setup:**
    - Ensure the table is created and populated with sample data for testing.
    - Verify existing columns and data types.
2. **Execution:**
    - Run `ALTER TABLE` to add new columns.
    - Use `UPDATE` to populate new columns with default or calculated values.
    - Execute `DELETE` statements with proper conditions.
3. **Validation:**
    - Confirm the structure change with `DESCRIBE` or `SHOW COLUMNS`.
    - Validate data modifications with appropriate `SELECT` queries.
    - Verify deletions by re-running the original condition as a `SELECT` query.

Documentation

1. **Screenshots:**
    - Capture the structure of the table before and after modifications.
    - Provide screenshots of queries and their resulting outputs.
2. **Explanations:**
    - Clearly describe the purpose of each query and the changes made.
    - Detail the conditions used for deletions and their rationale.
3. **Summary of Findings:**
    - Highlight any trends observed in the data.
    - Discuss the impact of modifications and deletions on the dataset integrity.

General Guidelines for All Tasks

- **Backup Data:** Always back up the table before performing deletions or structural changes.
- **Test with Sample Data:** Use a subset of data to test queries before applying them to the entire table.
- **Use Transactions (if supported):** Implement transactions to rollback in case of errors.
- **Maintain Audit Logs:** Consider creating an audit trail for deleted records by storing them in a separate table before removal.

Would you like more detailed instructions on performing these SQL operations or examples of best practices for data integrity? 😊

Deadline Compliance

- **Restriction**: **Submit the project within 7 days** from the start date.
- **Reason**: Meeting deadlines is crucial in the real-world software development environment. This restriction helps students practice **time management** and **task prioritization**. In professional settings, tight deadlines are often the norm, and learning to meet them without compromising quality is an essential skill.
- **Learning Outcome**: Students will learn to manage their time effectively, complete projects under pressure, and **deliver results on time**, which are all important skills in the workplace.