SQL TASK – 6 - Data Manipulation

Overview

This task focuses on modifying a SQL table by adding columns, updating values, and deleting outdated or inactive records in the Subscriptions table. It simulates real-world data maintenance to improve accuracy and manage active subscriptions effectively.

Table Structure

Column Name	*Data Type*	*Description*
SubID	*INT*	*Unique identifier for each subscribers*
Name	*VARCHAR*	*Name of each subscribers*
Plan	*VARCHAR*	*Type of subscription plan (e.g., Basic, Premium)*
StartDate	*DATE*	*Start date of each subscriptions*
Status	*VARCHAR*	*Current status of each subscriptions*
Last_Updated	*DATE*	*Last updated date of the subscription record*

1. Create Database and Table

SQL Query:

```
1 •  USE SQL_TASK;
2 •  CREATE TABLE Subscriptions(SubID INT AUTO_INCREMENT PRIMARY KEY,
3                               Name VARCHAR(100),
4                               Plan VARCHAR(50),
5                               StartDate DATE
6                               );
7
```

***Purpose**: Creates the Subscriptions table structure to hold subscriber details such as name, plan, start date, etc.*

***Observation**: Table structure defined successfully with necessary fields to simulate a subscription system.*

2. Insert Sample Records

SQL Query:

```
INSERT INTO Subscriptions (Name, Plan, StartDate) VALUES
('Alice', 'Premium', '2024-12-01'),
('Bob', 'Basic', '2023-11-15'),
('Charlie', 'Standard', '2025-01-10'),
('David', 'Premium', '2024-10-25'),
('Eva', 'Standard', '2025-05-20');
```

***Purpose**: Populates the Subscriptions table with 5 sample records for analysis.*

***Observation**: Data is diverse by name, plan type, and start date, enabling various realistic test cases.*

Result:

SubID	Name	Plan	StartDate
1	Alice	Premium	01-12-2024
2	Bob	Basic	15-11-2023
3	Charlie	Standard	10-01-2025
4	David	Premium	25-10-2024
5	Eva	Standard	20-05-2025

3. Task 1: Alter Table - Add Columns

SQL Query:

```
ALTER TABLE Subscriptions ADD COLUMN Status VARCHAR(10) DEFAULT 'ACTIVE',
ADD COLUMN Last_Updated DATE;
```

***Purpose**: Adds Status and Last_Updated columns to track subscription activity and record modification dates.*

***Observation**: Table structure was extended without loss of existing data; new fields allow better monitoring and life cycle tracking.*

Query Explanation:

- *Status: to track whether a subscription is Active or Inactive.*
- *Last_Updated: to store the date when the subscription record was last modified.*

Result:

SubID	Name	Plan	StartDate	Status	Last_Updated
1	Alice	Premium	01-12-2024	ACTIVE	NULL
2	Bob	Basic	15-11-2023	ACTIVE	NULL
3	Charlie	Standard	10-01-2025	ACTIVE	NULL
4	David	Premium	25-10-2024	ACTIVE	NULL
5	Eva	Standard	20-05-2025	ACTIVE	NULL

4. Task 2: Update Status of Subscription of individulas

SQL Query:

```
21 •  UPDATE Subscriptions
22    SET Status = 'Inactive'
23    WHERE SubID= 2;
24
```

***Purpose**: To simulate inactive subscriptions by changing the Status column for selected users. This helps identify and later remove or manage users who are no longer active.*

***Observation**: The Status column was updated for specific user, marking them as 'Inactive'. These records are now eligible for conditional operations like deletion.*

***Query Explanation**: This query targets specific rows based on SubID and sets their Status to 'Inactive'. It's useful for flagging records that need special handling.*

5. Task 2.1: Update Date of Subscription Update Date

SQL Query:

```
25 •  UPDATE Subscriptions
26    SET Last_Updated = '2025-01-02'
27    WHERE SubID = 1;
28 •  UPDATE Subscriptions
29    SET Last_Updated = '2024-02-05'
30    WHERE SubID = 2;
31 •  UPDATE Subscriptions
32    SET Last_Updated = '2025-01-10'
33    WHERE SubID = 3;
34 •  UPDATE Subscriptions
35    SET Last_Updated = '2024-12-25'
36    WHERE SubID = 4;
37 •  UPDATE Subscriptions
38    SET Last_Updated = '2025-05-20'
39    WHERE SubID = 5;
40
```

***Purpose**: To assign a specific last update date to each subscription record. This allows for testing operations that depend on date-based filtering, such as deleting outdated entries.*

***Observation**: A fixed/random date (e.g., '2025-01-01') was assigned to the Last_Updated column for all records. This setup enables consistent testing for identifying and removing old data.*

***Query Explanation:** This query sets the Last_Updated field to a manually chosen date for all rows. It prepares the table for future deletion or filtering operations based on this common date value.*

Result:

SubID	Name	Plan	StartDate	Status	Last_Updated
1	Alice	Premium	01-12-2024	ACTIVE	02-01-2025
2	Bob	Basic	15-11-2023	Inactive	05-02-2024
3	Charlie	Standard	10-01-2025	ACTIVE	10-01-2025
4	David	Premium	25-10-2024	ACTIVE	25-12-2024
5	Eva	Standard	20-05-2025	ACTIVE	20-05-2025

6. Task 3: Delete Records Based on Conditions

SQL Query:

```
43 •  DELETE FROM Subscriptions
44    WHERE Status = 'Inactive';
45
46 •  DELETE FROM Subscriptions
47    WHERE Last_Updated < '2025-01-01';
48
```

***Purpose**: Deletes records that are either marked as inactive or haven't been updated recently to simulate data pruning.*

***Observation**: Unwanted or outdated records were removed successfully using conditional DELETE queries, improving table cleanliness.*

Query Explanation:

- *Based on Status: Deletes all subscriptions that are no longer active (Status = 'Inactive'). It demonstrates data cleanup based on business logic.*
- *Based on Last_Updated: Removes records that haven't been updated recently. This is useful in scenarios like clearing out stale or outdated data.*

Result:

SubID	Name	Plan	StartDate	Status	Last_Updated
1	Alice	Premium	01-12-2024	ACTIVE	02-01-2025
3	Charlie	Standard	10-01-2025	ACTIVE	10-01-2025
5	Eva	Standard	20-05-2025	ACTIVE	20-05-2025