

Step 1: Create the Main Database Table

The screenshot displays the SQL Online AIDE web application interface. The browser address bar shows the URL `sqliteonline.com`. The application has a dark theme and includes a sidebar on the left with navigation options: 'Private DB +', 'Demo Memory', 'SQLite', 'DuckDB', 'PQlite', 'Demo Server', 'MariaDB', 'PostgreSQL', and 'MS SQL'. The 'MariaDB' section is expanded, showing a table named 'users_nadonza'. The main editor area contains the following SQL code:

```
1 CREATE TABLE users_nadonza (  
2   id INT PRIMARY KEY AUTO_INCREMENT,  
3   name VARCHAR(100),  
4   email VARCHAR(100) UNIQUE,  
5   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
6 );  
7
```

Below the code editor is a 'Run' button and a list of database instances: 'MariaDB', 'MariaDB.1', 'MariaDB.2', 'MariaDB.3', and 'MariaDB.4'. The 'History' panel on the right shows the executed SQL statement and the timestamp '22:07:13'. The Windows taskbar at the bottom indicates the system time as 10:07 PM on 10/3/2025.

Step 2: Create the Audit Trail Table

The screenshot displays the SQL Online AIDE web application interface. The browser address bar shows 'sqliteonline.com'. The interface is divided into several sections:

- Left Sidebar:** Contains navigation options like 'Pricing', 'Help', 'Private DB +', 'Demo Memory', and 'Demo Server'. Under 'Demo Server', 'MariaDB' is selected, showing a version of '0.19.0 beta'. Below this, a 'Table' list includes 'audit_trail_nadonza' and 'users_nadonza'.
- Top Bar:** Features a 'Run' button and tabs for different database instances: 'MariaDB', 'MariaDB.1', 'MariaDB.2', 'MariaDB.3', and 'MariaDB.4'.
- Main Editor:** Displays SQL code for creating the 'audit_trail_nadonza' table:

```
1 CREATE TABLE audit_trail_nadonza (  
2   audit_id INT NOT NULL AUTO_INCREMENT,  
3   TABLE_NAME VARCHAR(100) NOT NULL,  
4   operation_type VARCHAR(20) NOT NULL,  
5   old_data TEXT,  
6   new_data TEXT,  
7   changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
8   changed_by VARCHAR(100),  
9   PRIMARY KEY (audit_id)  
10 );  
11
```
- Right Panel:** Titled 'History', it shows a list of executed queries. The first entry, for 'MariaDB.1', shows the successful execution of the 'CREATE TABLE audit_trail_nadonza' statement at '22:07:41'. The second entry, for 'MariaDB', shows the execution of 'CREATE TABLE users_nadonza' at '22:07:13'.
- Bottom Bar:** Includes a Windows taskbar with a search bar, system icons, and a clock showing '10:07 PM 10/3/2025'.

Step 3: Create a Trigger to Log INSERT Operations

The screenshot displays the SQL Online AIDE web application interface. The main editor shows the following SQL code:

```
2 AFTER INSERT ON users_nadonza
3 FOR EACH ROW
4 BEGIN
5     INSERT INTO audit_trail_nadonza (TABLE_NAME, operation_type, old_data, new_data, changed_by)
6     VALUES (
7         'users_nadonza',
8         'INSERT',
9         NULL,
10        json_object('id', NEW.id, 'name', NEW.name, 'email', NEW.email, 'created_at', NEW.created_at),
11        'system_user'
12    );
13 END;
14
```

The interface includes a sidebar on the left with a database tree showing 'Demo Memory' and 'Demo Server' sections. The 'Demo Server' section is expanded, showing a 'MariaDB' instance with tables 'audit_trail_nadonza' and 'users_nadonza'. The 'History' panel on the right shows a list of executed queries with timestamps:

- MariaDB 2: 22:08:04
- MariaDB 1: 22:07:41
- MariaDB: 22:07:13

The bottom of the screen shows a Windows taskbar with the search bar and various application icons. The system clock indicates 10:08 PM on 10/3/2025.

Step 4: Create a Trigger to Log UPDATE Operations

The screenshot displays the SQL Online AIDE web application interface. The main editor area contains the following SQL code:

```
2 AFTER UPDATE ON users_nadonza
3 FOR EACH ROW
4 BEGIN
5     INSERT INTO audit_trail_nadonza (TABLE_NAME, operation_type, old_data, new_data, changed_by)
6     VALUES (
7         'users_nadonza',
8         'UPDATE',
9         json_object('id', OLD.id, 'name', OLD.name, 'email', OLD.email, 'created_at', OLD.created_at),
10        json_object('id', NEW.id, 'name', NEW.name, 'email', NEW.email, 'created_at', NEW.created_at),
11        'system_user'
12    );
13 END;
```

The interface includes a left sidebar with a database list (SQLite, DuckDB, PGLite, MariaDB, PostgreSQL, MS SQL) and a 'History' panel on the right showing previous queries and their execution times. The bottom status bar indicates the time as 10:09 PM on 10/3/2025.

Step 5: Create a Trigger to Log DELETE Operations

The screenshot displays the SQLiteonline web application interface. The main editor area contains the following SQL code:

```
2 AFTER DELETE ON users_nadonza
3 FOR EACH ROW
4 BEGIN
5     INSERT INTO audit_trail_nadonza (TABLE_NAME, operation_type, old_data, new_data, changed_by)
6     VALUES (
7         'users_nadonza',
8         'DELETE',
9         json_object('id', OLD.id, 'name', OLD.name, 'email', OLD.email, 'created_at', OLD.created_at),
10        NULL,
11        'system_user'
12    );
13 END;
```

The left sidebar shows a list of databases under 'Demo Memory' and 'Demo Server'. Under 'Demo Server', the 'MariaDB' instance is selected, showing tables 'audit_trail_nadonza' and 'users_nadonza'. The right sidebar, titled 'History', shows a list of executed queries for MariaDB instances, including the creation of triggers for UPDATE and INSERT operations, and the creation of the audit_trail_nadonza table.

The bottom status bar indicates the language is set to 'ENG US' and the time is 10:10 PM on 10/3/2025.

Step 6: Testing the Audit Trail

- Insert a New User

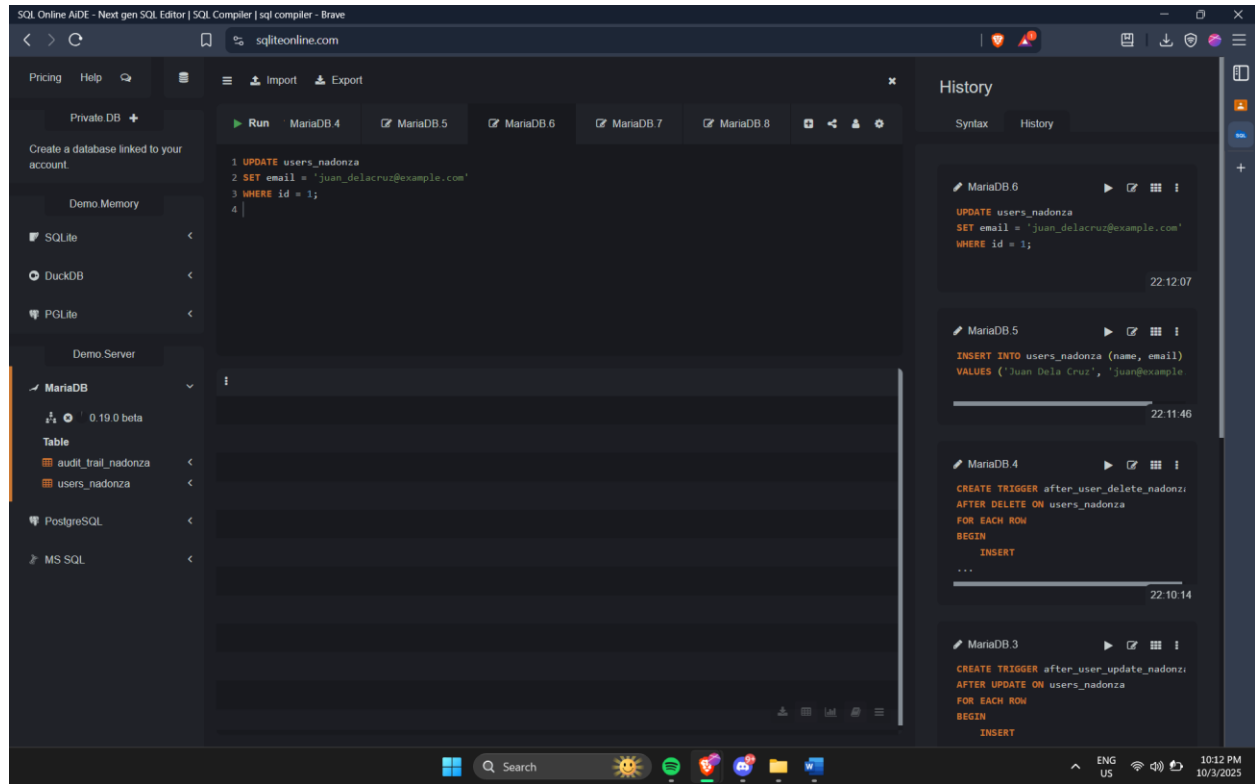
The screenshot displays the SQL Online AIDE web application interface. The main editor area contains the following SQL query:

```
1 INSERT INTO users_nadonza (name, email)
2 VALUES ('Juan Dela Cruz', 'juan@example.com');
3
```

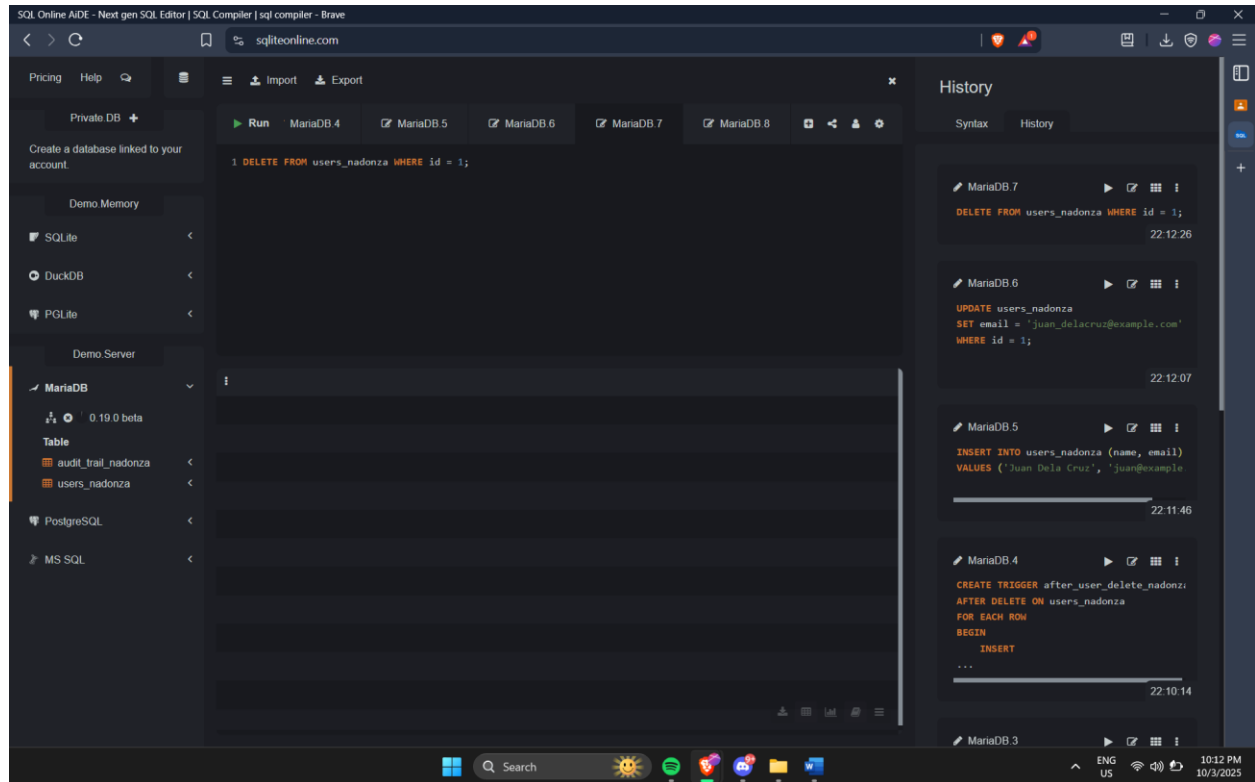
The interface includes a sidebar on the left with a 'Demo Server' section showing a list of databases: MariaDB (selected), PostgreSQL, and MS SQL. Under MariaDB, there are two tables listed: 'audit_trail_nadonza' and 'users_nadonza'. The 'History' panel on the right shows a list of executed queries, including the current one and previous ones like 'CREATE TRIGGER after_user_delete_nadonza' and 'CREATE TRIGGER after_user_update_nadonza'.

The bottom of the screen shows a Windows taskbar with the search bar, task view button, and several application icons. The system clock indicates 10:11 PM on 10/3/2025.

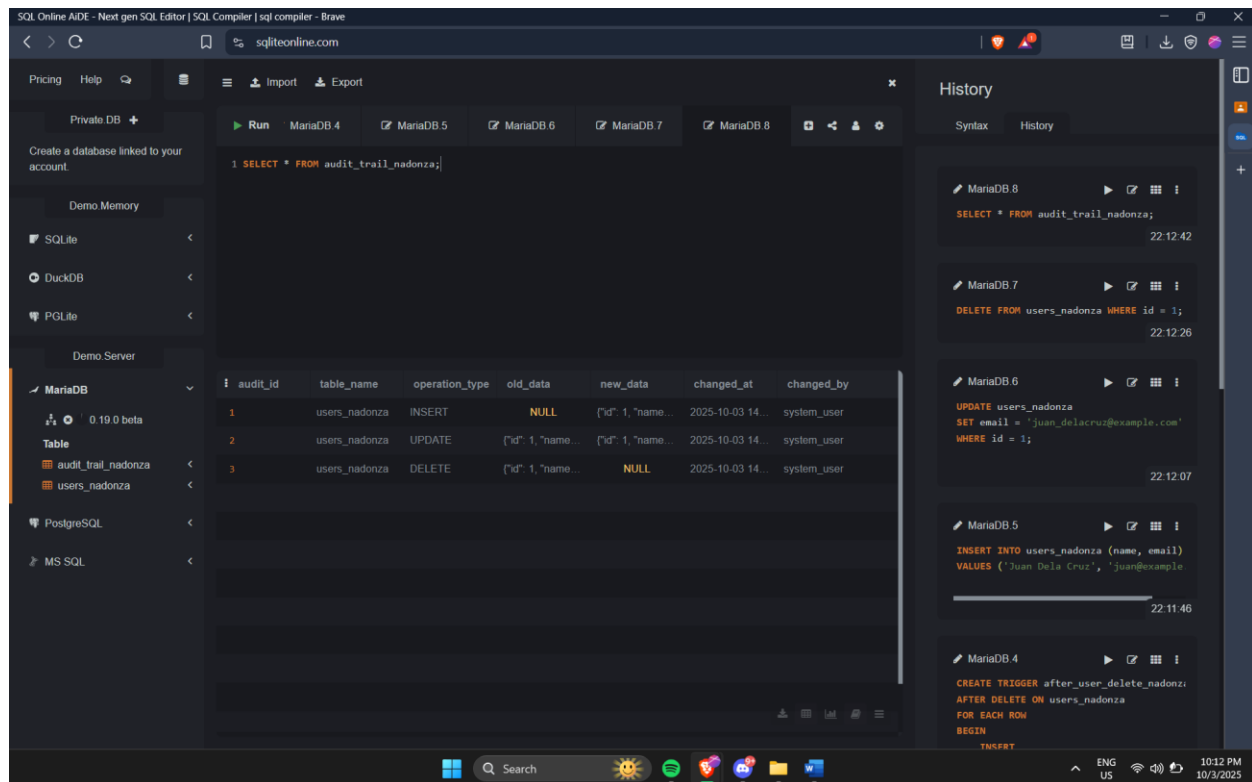
- Update a User's Email



- Delete a User



Step 7: View the Audit Trail



Screenshot the output and answer the following:

1. Do you want to track every operation (INSERT, UPDATE, DELETE)? Why?

Yes, Because it helps to keep the data to make it secured and updated.

2. Should the audit trail capture only the changed data (old and new values) or also the context of the change (e.g., who, when, why)? Why?

It should capture both the changed data and the context (who, when, why).

Because the context is crucial for security and shows who was responsible.

3. Should you log the database user or the application user? Why?

You should log the application user, why? because it provides true accountability for the individual user.

4. How long should audit data be retained?

It should be retained based on regulatory and legal requirements. Because laws mandate specific time periods, let say 5-10 years for compliance.