

8. Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers). Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library\_Audit table.

📝 Create an **audit log table** `Library_Audit` where the old values will be stored whenever a record in `Library` is **updated** or **deleted**.

🏢 We used **row-level triggers** (`FOR EACH ROW`).

🏢 We handled both **UPDATE** and **DELETE**.

🏢 Old values are preserved in `Library_Audit`.

### Step 1: Create Tables

```
CREATE TABLE Library ( BookID INT PRIMARY KEY, Title VARCHAR(100), Author VARCHAR(100),  
  
Price DECIMAL(8,2));
```

```
CREATE TABLE Library_Audit ( AuditID INT AUTO_INCREMENT PRIMARY KEY,  
  
BookID INT, Title VARCHAR(100), Author VARCHAR(100), Price DECIMAL(8,2),  
  
ActionType VARCHAR(20), ActionDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP );
```

### Step 2: Trigger for UPDATE

```
DELIMITER $$
```

```
CREATE TRIGGER before_library_update
```

```
BEFORE UPDATE ON Library
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO Library_Audit(BookID, Title, Author, Price, ActionType)
```

```
VALUES (OLD.BookID, OLD.Title, OLD.Author, OLD.Price, 'UPDATE');

END$$

DELIMITER ;
```

### **Step 3: Trigger for DELETE**

```
DELIMITER $$

CREATE TRIGGER before_library_delete

BEFORE DELETE ON Library

FOR EACH ROW

BEGIN

    INSERT INTO Library_Audit(BookID, Title, Author, Price, ActionType)

    VALUES (OLD.BookID, OLD.Title, OLD.Author, OLD.Price, 'DELETE');

END$$

DELIMITER ;
```

### **Step 4: Test the Triggers**

```
-- Insert some records

INSERT INTO Library VALUES (1, 'DBMS Concepts', 'Navathe', 500.00);

INSERT INTO Library VALUES (2, 'Operating Systems', 'Galvin', 650.00);

-- Update a record          UPDATE Library SET Price = 550.00 WHERE BookID = 1;

-- Delete a record          DELETE FROM Library WHERE BookID = 2;
```

### **Step 5: View Audit Table**

```
SELECT * FROM Library_Audit;
```