



Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Fall, Year:2024), B.Sc. in CSE (Day)

Lab Report NO: 01
Course Title: Database
Course Code: CSE 209 Section: 231_D1

Assignment Name: Trigger and Function

Student Details

Name		ID
1.	Promod Chandra Das	231002005

Submission Date : 17.12.2024
Course Teacher's Name : Dr. Faiz Al Faisal

Lab Report Status

Marks:
Comments:.....

Signature:.....
Date:.....

1. TITLE OF THE ASSIGNMENT NAME :

Trigger and Function

2. OBJECTIVES :

1. Understand Database Automation

- Learn how to create **triggers** that respond to specific database events (e.g., INSERT, UPDATE, DELETE).
- Understand how **functions** can encapsulate reusable logic that triggers can invoke.

2. Enhance Data Integrity

- Automatically enforce rules and constraints (e.g., keeping logs, preventing invalid operations).
- Ensure that critical business processes are executed without manual intervention.

3. Practice Advanced SQL Concepts

- Gain proficiency in procedural SQL by writing functions.
- Understand how triggers and functions work together to automate backend processes.

4. Implement Real-World Use Cases

- Simulate scenarios where triggers and functions are used for:
 - Auditing changes in data (e.g., stock updates, salary changes).
 - Ensuring consistency across tables (e.g., cascading updates, validations).
 - Automating notifications or alerts.

5. Demonstrate Practical Skills

- Showcase the ability to design a database with advanced features.
- Highlight skills useful in real-world database administration and application development.

3. Step 1: Create Tables

products Table

```
CREATE TABLE products (  
  product_id SERIAL PRIMARY KEY,  
  product_name VARCHAR(100) NOT NULL,  
  stock INT NOT NULL CHECK (stock >= 0),  
  price DECIMAL(10, 2) NOT NULL  
);
```

stock_audit Table

```
CREATE TABLE stock_audit (  
    audit_id SERIAL PRIMARY KEY,  
    product_id INT NOT NULL,  
    old_stock INT NOT NULL,  
    new_stock INT NOT NULL,  
    change_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

Step 2: Create the Function

The function will log stock changes into the stock_audit table.

```
CREATE OR REPLACE FUNCTION log_stock_change()  
RETURNS TRIGGER AS $$  
BEGIN  
    -- Check if the stock has changed  
    IF OLD.stock IS DISTINCT FROM NEW.stock THEN  
        INSERT INTO stock_audit (product_id, old_stock, new_stock)  
        VALUES (OLD.product_id, OLD.stock, NEW.stock);  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

Step 3: Create the Trigger

The trigger will invoke the function after any update to the products table.

```
CREATE TRIGGER stock_update_trigger  
AFTER UPDATE OF stock ON products  
FOR EACH ROW  
EXECUTE FUNCTION log_stock_change();
```

Step 4: Test the Trigger and Function

Insert Data into products

```
INSERT INTO products (product_name, stock, price) VALUES  
( 'Laptop', 50, 1000.00),  
( 'Smartphone', 100, 500.00);
```

Update the Stock

```
UPDATE products SET stock = 45 WHERE product_name = 'Laptop';
```

Check the stock_audit Table

```
SELECT * FROM stock_audit;
```

You should see a record showing the change in stock for the product.

Stock_audit Table After the Update

audit_id	product_id	old_stock	new_stock	change_date
1	1	50	45	2024-12-15 10:30:00

Key Concepts in the Example

1. Function:

- Encapsulates logic to log changes.
- Can be reused for other triggers or extended with additional functionality.

2. Trigger:

- Ensures the logging happens automatically.
- Tied to specific events (e.g., updates to the products table).

➤ DISCUSSION :

Triggers and functions in databases enable automation, enforce business rules, and maintain data integrity. Triggers act on events like INSERT or UPDATE, while functions encapsulate reusable logic. Together, they streamline workflows, such as logging changes or validating data. Proper design ensures efficiency, but overuse can lead to performance issues and complexity.

