ADVANCE DATABASE SYSTEM

TRIGGER

MODULE 2 LESSON 4

SQL Trigger

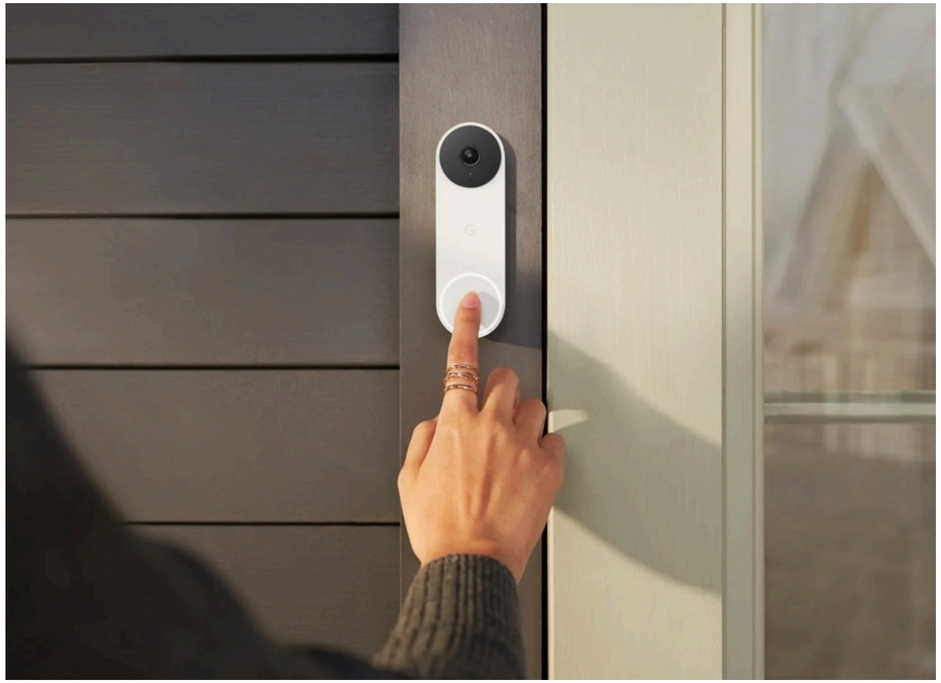
A trigger is a set of actions that are run automatically when a specified change operation (SQL INSERT, UPDATE, or DELETE statement) is performed on a specified table.

SQL Trigger

Triggers are useful for tasks such as enforcing business rules, validating input data, and keeping an audit trail.

Special stored procedures that automatically execute in response to certain events on a table.

Stored procedures vs Trigger



A stored procedure is like a doorbell button. It's a pre-compiled set of one or more SQL statements that perform a specific task such as ringing the bell

A trigger is like an opening of a door it triggers an alarm and automated actions that occur in response to database events.

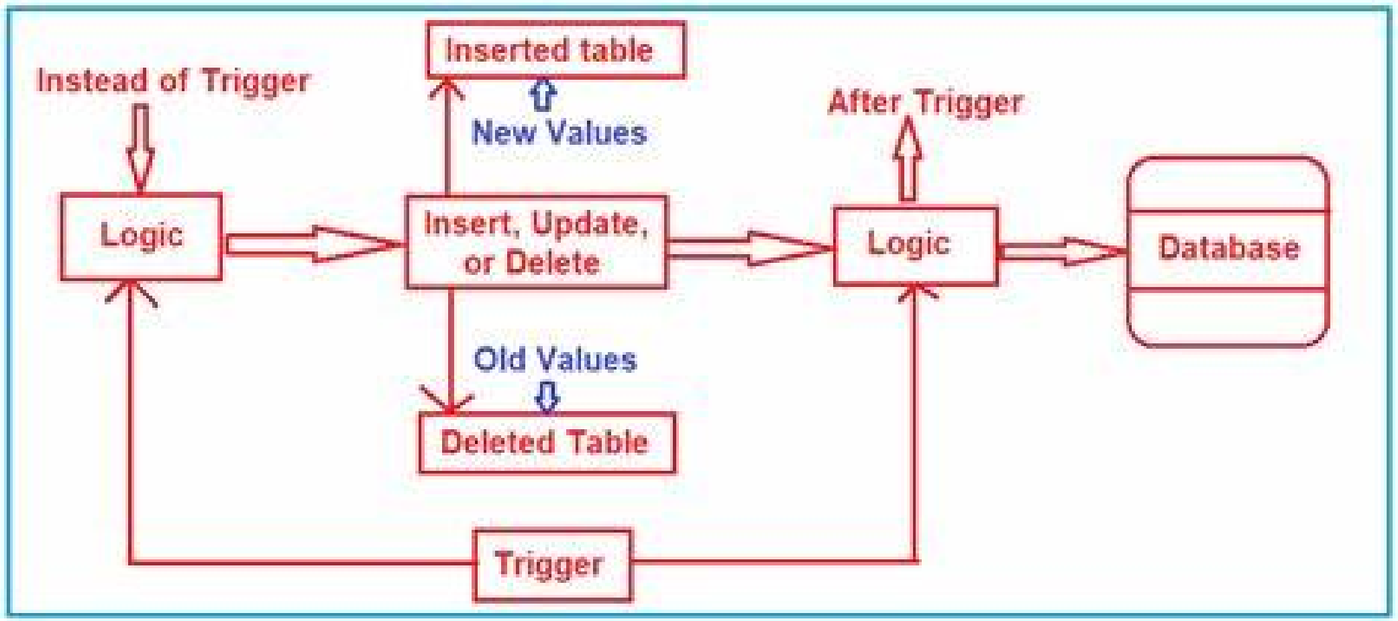
```
DELIMITER //
CREATE TRIGGER trigger_name
        trigger_time
        trigger_event ON table_name
        FOR EACH ROW
        BEGIN
        --variable declarations
        --trigger code
END //
```

SQL Trigger Syntax

trigger_name

- It is the name of the trigger that we want to create. It must be written after the CREATE TRIGGER statement.
- It is to make sure that the trigger name should be unique within the schema.

trigger_time



- It is the trigger action time, which should be either BEFORE or AFTER.
- It is the required parameter while defining a trigger.
- It indicates that the trigger will be invoked before or after each row modification occurs on the table.

trigger_event:

- It is the type of operation name that activates the trigger. It can be either INSERT, UPDATE, or DELETE operation.
- The trigger can invoke only one event at one time.
- If we want to define a trigger which is invoked by multiple events, it is required to define multiple triggers, and one for each event.

Table_name:

COUNTRY

id	name	population	area
1	France	66600000	640680
2	Germany	80700000	357000
...	...	...	...

CITY

id	name	country_id	population	rating
1	Paris	1	2243000	5
2	Berlin	2	3460000	3
...	...	...	...	...

It is the name of the table to which the trigger is associated.

It must be written after the ON keyword.

If we did not specify the table name, a trigger would not exist.

BEGIN END Block:

```
BEGIN
    INSERT INTO LocationHist
    SELECT LocationID
        ,getdate()
    FROM inserted
END
```

T-SQL block that runs against specified DML Event

- Finally, we will specify the statement for execution when the trigger is activated.
- If we want to execute multiple statements, we will use the BEGIN END block that contains a set of queries to define the logic for the trigger.

SQL Before Trigger

SYNTAX:

```
CREATE TRIGGER calculate
before INSERT
ON student
FOR EACH ROW
SET new.marks = new.marks+100;
```

SQL After Trigger

SYNTAX:

```
CREATE TRIGGER total_mark
after insert
ON student
FOR EACH ROW
insert into Final_mark values(new.marks);
```

SQL Drop Triggers

SYNTAX:

```
DROP TRIGGER trigger name;
```

EXAMPLE:

```
DROP TRIGGER calculate name;
```

SQL Show Triggers

```
SHOW TRIGGERS;
```

Project Proposal and Database Design Development

- **Propose a concept paper that will handle fundamental CRUD (Create, Read, Update, Delete) operations, incorporating primary and advanced SQL commands to fulfill the approved proposals' processes using Laravel.**
- **A minimum of ten (10) tables must be created, utilizing the XAMPP MySQL for implementation.**
- **Present an overview of the project and the Database Design.**
- **Presentation on March 12 and 14 2025 should only be taking 5-15 minutes.**