Triggers:

Triggers in MySQL are named database objects associated with a specific table that automatically execute a predefined set of SQL statements when a particular event occurs on that table. These events include INSERT, UPDATE, or DELETE operations.

Key characteristics and components of MySQL Triggers:

Association with a Table: A trigger is always linked to a single table.

Triggering Event: The event that activates the trigger can be INSERT, UPDATE, or DELETE.

Trigger Action Time:

BEFORE: The trigger executes before the triggering event occurs. This is often used for data validation, modification, or cleansing before the data is actually inserted, updated, or deleted.

AFTER: The trigger executes after the triggering event has completed. This is commonly used for auditing, logging changes, or performing actions on other tables based on the changes in the triggered table.

Trigger Body: This is the block of SQL statements that are executed when the trigger is activated. It can contain one or more SQL statements.

FOR EACH ROW: This clause specifies that the trigger should activate for every row affected by the triggering event.

OLD and **NEW** Keywords: Within the trigger body:

NEW.column_name: Refers to the value of a column after an INSERT or UPDATE operation.

OLD.column_name: Refers to the value of a column before an UPDATE or DELETE operation.

Example:

```
DELIMITER //

CREATE TRIGGER after_employee_insert

AFTER INSERT ON employees

FOR EACH ROW

BEGIN

   INSERT INTO employee_audit (employee_id, action, timestamp)

   VALUES (NEW.employee_id, 'INSERT', NOW());

END; //

DELIMITER ;
```

- after_employee_insert is the trigger name.

- AFTER INSERT ON employees specifies that the trigger activates after an INSERT operation on the employees table.
- FOR EACH ROW ensures it runs for every inserted row.
- The BEGIN...END block contains the trigger body, which inserts an audit record into the employee_audit table, using NEW.employee_id to get the ID of the newly inserted employee.

Common uses of Triggers:

- **Data Validation:** Enforcing business rules and data integrity before data modification.
- **Auditing and Logging:** Recording changes made to tables for historical tracking or security purposes.
- **Data Synchronization:** Updating related data in other tables automatically.
- **Complex Business Logic:** Automating calculations or actions based on table events.