100 21600 90

Triggers in SQL is a special type of storage procedure that is automatically executed or fired when certain event occurs on a specified table or view in a database.
Trigger are used to enforce business rules, validate data or keep data in sync accross tables or database.

Trigger can be set to execute when :-

Insert Operation happens
UPDATE " "
DELETE " "

There are 6 types of triggers in MYSQL.

1) Before trigger :-
A BEFORE TRIGGER is executed before the specified action (Insert, Update, Delete) is performed on table.

This allows you to modify the data before it is commited to the table, such as validating or altering values.

```
CREATE TRIGGER before_insert_trigger
BEFORE INSERT ON employees.
 FOR EACH ROW
 BEGIN
      SET NEW.salary = ROUND(NEW.salary ,2);
 END;
```

2) AFTER Trigger:- A AFTER trigger is executed after the specified action (INSERT, UPDATE, DELETE) has been performed on the table.

- You can use this trigger operation like logging or updating other related tables after the primary action is complete.

```
CREATE TRIGGER after_update_trigger
AFTER UPDATE ON employees
FOR EACH ROW
BEGIN
        INSERT INTO audit_log (action, old_salary, new_salary)
         values ('Update', OLD.salary, NEW.salary);
END;
```

3) BEFORE INSERT TRIGGER:-

- A Trigger specifically for when an INSERT operation is about to occur on a table.
- You can use this type of trigger to validate or adjust the data before it is inserted.

```
CREATE TRIGGER before_insert_trigger
BEFORE INSERT ON employees
 FOR EACH ROW
 BEGIN
      IF NEW.Salary < 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT 'Salary can't
                                                  be Negative';
   END IF;
END;
```

4) AFTER INSERT TRIGGER :- A bigger specifically for when an INSERT operation has been completed on a table.

- This type can be used for logging or performing additional tasks based on the inserted Data.

```
CREATE TRIGGER after_insert_emp
AFTER INSERT ON employees
FOR EACH ROW
BEGIN
    INSERT INTO log_table (action, employee_id)
    VALUES ('INSERT', New.id);
END;
```

5) BEFORE UPDATE Trigges :- A bigger specifically for when an UPDATE operation ~~ha~~ is about to be performed on a table.

- You can be use this type to validate or modify data befor the update happen.

```
CREATE TRIGGER before_update_emp
BEFORE UPDATE ON employees.
FOR EACH ROW
BEGIN
    If NEW.salary < 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Salary Can't be Negative';
    ENDIF;
END;
```

6 AFTER UPDATE TRIGGER :- A trigger that is executed after an UPDATE operation has been completed.

- This type can be used for actions like auditing or performing additional operations after the update.

```
CREATE TRIGGER after_update_emp
AFTER UPDATE ON employees.
FOR EACH ROW
BEGIN
    INSERT INTO employee_audit (emp_id, old_salary, new_salary)
    VALUES (NEW.id, OLD.salary, NEW.salary);
END;
```

7 BEFORE DELETE TRIGGER :- A trigger that is executed before an DELETE operation is performed on a table.

- This type can be useful for checking conditions or ensuring related data is properly managed before deletion.

8 AFTER DELETE TRIGGER :- A trigger that is executed after a DELETE operation is completed.

- You can use this to clean up related data or log the deletion.