Assignment No.-5 PLSQL (Triggers)

1. What is PL/SQL trigger?

Triggers are stored programs which are automatically executed or fired when some events occur. Triggers are, written to be executed in response to any of the following events -

(a) A database manipulation (DML) statement
(DELETE, INSERT, UPDATE)

(b) A database definition (DDL) statement
(CREATE, ALTER, DROP)

(c) A database operation -
(SERVERERROR, LOGON, LOGOFF, STARTUP or SHUTDOWN).

Triggers could be defined on the table, view, schema or database with which the event is associated.

2. Explain the benefits of triggers.

Advantages of triggers are -

a) Triggers generates some derived column values automatically.

b) Enforces referential integrity

c) Event logging & storing information on table access.

d) Auditing

e) Synchronous replication of tables

f) Imposing security authorizations.

g) Preventing invalid transactions

3. What are the different types of triggers?
There are 6 different types of triggers in MySQL.

1. Before Update Trigger - As the name implies, it is a trigger that enacts before an update is invoked. If we write an update statement, then the actions of the trigger will be performed before the update is implemented.

~~eg → create table customer~~

2. After Update trigger - As the name implies, this trigger is invoked after an updation occurs (i.e. it gets implemented after an update statement is executed).

3. Before Insert trigger - As the name implies the trigger is invoked before an insert statement is executed.

4. After Insert trigger - The trigger is invoked after an insert is implemented.

5. Before Delete trigger – The trigger is invoked before a delete occurs or before delete statement is implemented.

6. After Delete trigger – The trigger is invoked after a delete occurs or after a delete operation is implemented.

4. Explain the syntax of a trigger with example.
We can created a trigger in MySQL using the CREATE TRIGGER statement. Basic syntax to create a trigger is –

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

the create trigger syntax contains the following parameters –

→ trigger_name – It is the name of the trigger that we want to create. It 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.

→ trigger_event – It is the type of operation that activates the trigger. It can be either INSERT, UPDATE or DELETE.

→ table_name – It is the name of the table to which the

trigger is associated. It must be written after the ON keyword. If we do not specify the table name, a trigger would not exist.

BEGIN END block - Finally we 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 defines a set of queries (logic) for the trigger.

MySQL Trigger example - 1st we'll create a table called employee.

```
CREATE TABLE employee (
    name varchar(45) NOT NULL,
    occupation varchar(35) NOT NULL,
    working_date date,
    working_hours varchar(10)
);

INSERT INTO employee VALUES
('Peter', 'Actor', '2020-10-04', 14),
('Marco', 'Doctor', '2020-10-04', 12);
```

Now execute SELECT statement to verify inserted record

```
SELECT * FROM employee;
```

name	occupation	working_date	working_hours
Peter	Actor	2020-10-04	14
Marco	Doctor	2020-10-04	12

Now create a Before Insert trigger.

```
DELIMITER //
Create Trigger BIE
BEFORE INSERT
ON employee FOR EACH ROW
BEGIN
IF NEW.working_hours < 0
THEN SET NEW.working_hours = 0;
END IF;
END //
```

This trigger is automatically invoked if someone tries to insert working_hours < 0.

5. What is the difference bet" stored procedures & triggers?

Stored procedures are pieces of code written in PL/SQL to do some specific task. They can be invoked explicitly by the user. It is like a java program. It can take some input as a parameter then do some processing & can return values. On the other hand, triggers are stored procedures that run automatically when various events happen (eg- update, insert, delete). Triggers are more like an event handler, they run at the specific event. Triggers can't take input & can't return values.

	Triggers	Stored Procedures
Basic difference	Trigger is a stored procedure that runs automatically when various events happen (update, insert, delete)	Stored procedures are pieces of code written in PL/SQL to do some specific task.
Running Methodology	It can execute automatically based on the events.	It can be invoked explicitly by the user.
Parameter	It cannot take input as parameters.	It can take input as parameter.

	Triggers	Stored Procedures
Transaction statements	We can't use transaction statements inside a trigger.	We can use transaction statements like - begin transaction, commit transaction & rollback
Return	Triggers cannot return values.	Stored procedures can return values.

6. How to view & drop all triggers in a given database?

The SHOW/LIST trigger is much needed when we have many databases that contain various tables. This statement returns all the triggers in all the databases:-

```
SHOW TRIGGERS;
```

To view the trigger information in a specific database, MySQL allows us to use the FROM or IN clause followed by the database name.

```
mysql> SHOW TABLES IN database_name;
```

or

```
mysql> SHOW TABLES FROM database_name;
```

We can drop an existing trigger from the database by using the DROP TRIGGER statement -

```
DROP TRIGGER [IF EXISTS] [schema_name].trigger_name;
```

eg → `DROP TRIGGER IF EXISTS employeedb.before_update_salaries;`