Table of Contents

What is Trigger ?
Trigger fires in following operations
Types of DML triggers
Creating a trigger
Syntax
Trigger example 1
Trigger example 2
Trigger example 3

What is Trigger ?

A trigger is defined as an action taken by database when some database related events occur.

Triggers are implicitly fired by ORACLE when triggering event occur,no matter which user is using or used by which application.

The code within the trigger called the trigger body.

Trigger fires in following operations

1. DML statements (INSERT, UPDATE, DELETE).
2. DDL statements (CREATE or ALTER)
3. Database events such as logon/logoff, errors, startup /shutdown, etc.

Types of DML triggers

1. BEFORE INSERT row
2. BEFORE INSERT statement

3. AFTER INSERT row
4. AFTER INSERT statement
5. BEFORE UPDATE row
6. BEFORE UPDATE statement
7. AFTER UPDATE row
8. AFTER UPDATE statement
9. BEFORE DELETE row
10. BEFORE DELETE statement
11. AFTER DELETE row
12. AFTER DELETE statement

Creating a trigger

Trigger can be created using the create trigger command.

Syntax

CREATE (OR REPLACE) TRIGGER trigger_name[BEFORE|AFTER][DELETE|INSERT|UPDATE[OF column_name]]ON table_name[FOR EACH ROW][WHEN condition][PL/SQL BLOCK];

Trigger example 1

Program of trigger for converting lowercase to uppercase before insert.

Name it as trig.sql.

```
CREATE or REPLACE trigger upper1 BEFORE INSERT on STUDENTFor each
ROWBEGIN:new.stdname:=upper (:new.stdname);END;
```

Than compile above trigger using:

```
SQL>GET TRIG.SQL;
```

After compilation this message will display:

Trigger is created successfully.

Trigger example 2

Program of trigger for after insert. To increment the serial number automatically after inserting the new student record.

```
CREATE or REPLACE trigger after_trigger1 AFTER INSERT ON studentFor
each ROWBEGINUPDATE student SET sno - sno+1 where stdname =
new.stdname;END;
```

Trigger example 3

Program of trigger for after delete.

To decrement the serial number automatically after deleting the student record from table student.

```
CREATE or REPLACE trigger after_delete1 AFTER DELETE ON studentFor
each ROWBEGINUPDATE student SET sno = sno-1 where
```

```
  stdname=:old.stdname;END;
```