

Ex.No.: 13		<p style="text-align: center;">WORKING WITH TRIGGER</p> <p style="text-align: center;"><u>TRIGGER</u></p>
Date:		

DEFINITION

A trigger is a statement that is executed automatically by the system as a side effect of a modification to the database. The parts of a trigger are,

- **Trigger statement:** Specifies the DML statements and fires the trigger body. It also specifies the table to which the trigger is associated.
- **Trigger body or trigger action:** It is a PL/SQL block that is executed when the triggering statement is used.
- **Trigger restriction:** Restrictions on the trigger can be achieved

The different uses of triggers are as follows,

- *To generate data automatically*
- *To enforce complex integrity constraints*
- *To customize complex securing authorizations*
- *To maintain the replicate table*
- *To audit data modifications*

TYPES OF TRIGGERS

The various types of triggers are as follows,

- **Before:** It fires the trigger before executing the trigger statement.
- **After:** It fires the trigger after executing the trigger statement
- **For each row:** It specifies that the trigger fires once per row
- **For each statement:** This is the default trigger that is invoked. It specifies that the trigger fires once per statement.

VARIABLES USED IN TRIGGERS

- :new
- :old

Program 1

Write a code in PL/SQL to develop a trigger that enforces referential integrity by preventing the deletion of a parent record if child records exist.

```
Create OR Replace Trigger prevent parent - deletion  
Before Delete ON PARENT  
For EACH ROW  
DECLARE  
    child - count NUMBER;  
BEGIN  
    SELECT COUNT(*) IN child - count FROM child  
WHERE parent_id  
IF child - count > 0 THEN RAISE - APPLICATION - ERROR  
END;
```

Program 2

Write a code in PL/SQL to create a trigger that checks for duplicate values in a specific column and raises an exception if found.

```
CREATE TABLE SampleTable(  
    id NUMBER(5) PRIMARY key,  
    name VARCHAR(50) NULL,  
    email VARCHAR2(100) unique);  
CREATE OR REPLACE TRIGGER check - duplicate - email  
BEFORE INSERT OR UPDATE ON SampleTable  
FOR EACH ROW  
DECLARE  
    duplicate - count NUMBER  
BEGIN  
    SELECT COUNT(*) INTO duplicate - count  
END IF;  
END;
```

Program 5

Write a code in PL/SQL to implement a trigger that records user activity (inserts, updates, deletes) in an audit log for a given set of tables.

```
CREATE OR REPLACE TRIGGER record-activity
AFTER INSERT OR UPDATE OR DELETE ON employee FOR
EACH ROW
BEGIN
    INSERT INTO Audit log VALUES (audit_seq.
                                NEXTVAL,
CASE WHEN INSERTING THEN 'INSERT' WHEN
UPDATING
THEN 'UPDATE' 'employees', NULL(:OLD emp-id
:NEW-emp-id), SYSDATE, USER);
END;
```

Program 7

Write a code in PL/SQL to implement a trigger that automatically calculates and updates a running total column for a table whenever new rows are inserted.

```
CREATE TABLE Sales (
    Sales-id NUMBER PRIMARY KEY,
    amount NUMBER (10,2);
    running-total number (10,2);
CREATE OR REPLACE TRIGGER Update-
running total
FOR EACH ROW
BEGIN
    SELECT NULL(MAX(running-total,0))+
NEW-amount INTO :New running
END;
```

Program 8

Write a code in PL/SQL to create a trigger that validates the availability of items before allowing an order to be placed, considering stock levels and pending orders.

```

CREATE OR REPLACE TRIGGER validate-stock-
before-order
BEFORE INSERT ON orders
FOR each ROW
BEGIN
  IF = New.order-quantity(SELECT stock
quantity FROM items
WHERE item-id:= NEW.item-id
END IF;
END;
  
```

Evaluation Procedure	Marks awarded
PL/SQL Procedure(5)	5
Program/Execution (5)	5
Viva(5)	4
Total (15)	14
Faculty Signature	