

EXERCISE-17

TRIGGER

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

These two variables retain the new and old values of the column updated in the database. The values in these variables can be used in the database triggers for data manipulation

SYNTAX

```
create or replace trigger triggername [before/after] {DML statements}
on [tablename] [for each row/statement]
begin
```

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-delete
before delete on dept
for each row
```

Declare

```
    v_count number;
```

Begin

```
    Select count(*) into v_count from emp
    where deptno :=旧deptno;
```

If v_count > 0 then

```
        RAISE_APPLICATION_ERROR (-20001,
        'Record not exist')
```

End If;

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 or replace trigger

check_duplicate_roll_no

Before insert or update on student

for each row

declare

v_count number;

Begin

Select count(*) into v_count from student
where roll_no.=New.roll_no;

if v_count>0 then

raise Application_Error(-20002, 'Puplicate
Roll no. not Allowed');

End if;

End;

Program 3

Write a code in PL/SQL to create a trigger that restricts the insertion of new rows if the total of a column's values exceeds a certain threshold.

```
Create table account (acc_no number primary key,  
cust_name varchar2(30), balance number);  
Create or replace Trigger check_tot_Bal ;  
Before insert or update on account  
Before each row  
Declare  
    v_told number;  
    v_threshold constant number := 100000;  
Begin  
    select NVL(sum(balance),0) into v_told from account;  
    v_told := v_told + Newbalance;  
    if v_total > v_threshold then  
        RAISE_APPLICATION_ERROR (-20003, 'Total Bal  
        exceeds allowed unit');  
    End if;  
End;
```

Program 4

Write a code in PL/SQL to design a trigger that captures changes made to specific columns and logs them in an audit table.

Create table employee (emp-id number primary key,
emp-name varchar2(50), salary number);

Create table audit-log (log-id number generated
always as identity Primary key, emp-id number,
old-sal number, new-sal number, changed-on DATE
changed-by varchar2(30));

Create or replace trigger salary_audit trigger
after update of emp-name
for each row

Begin

Insert into audit log values (:old.emp-id,
old.salary, :New.salary, SYSDATE, user);
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 emp-audit-trigger after
insert or update or delete on employee

Begin

Insert into audit-log values ('EMPLOYEE', ORA-
A-SYSEVENT, USER, SYSDATE);

End ;

Create or replace trigger dept-audit-trigger
After insert or update or delete on department.

Begin

Insert into audit-log values ('DEPARTMENT',
OR A-SYSEVENT, USER, SYSDATE);

END;

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 ( sale_id number primary key,
Sale_amt number, over_total number);

Create or replace trigger update_over_tot before
insertion sales
for each row
declare
    v_total number;
Begin
    select NVL(SUM(sale_amt), 0) into v_total
    from sales;
    :NEW.running_tot := v_total + :NEW.
    sale_amt;
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
declare

v_available_qty number;

Begin

Select stock_qty into v_availability from
items where item_id = :new.item_id ;

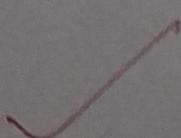
If :new.order_qty > v_availability_qty then

RAISE_APPLICATION_ERROR (-20005, 'Insufficient
Stock, cannot place order');

END IF ;

update items set stock_qty = stock_qty - new
order_qty where item_id = new.item_id ;

END ;



✓

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