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 (*) into child-count from child where
                                              parent_id .
```

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 Sample Table (
      id NUMBER (5) primary key
   name VARCHAR (50) NULL
   email VARCHAR (100) UNIQUE
 );
 CREATE OR REPLACE TIGIGER check_duplicate_email
  BEFORE INSERT OR UPDATE ON sample Table
  FOR EACH ROW
 DECLARE
        duplicate-count NUMBER ;
 BEGIN SELECT COUNT (*) INTO duplicate_count
  END IF ;
 END;
```

#B6
121

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  OR  REPLACE  TIGGER restrict_total_sales
BEFORE  INSERT  ON sales
For Each Row
REGIN
    IF (SELECT SUM (amount) FROM Sales) + : NewAmount >10000
   RAISE _ APRLICATION_ERROR (-20002), 'Totalexceeds
                              thress hold.');

 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 OR REPLACE  TRIGGER log.salary_changes
AFTER  UPDATE  OF SALARY ON employees
FOR  EACH  ROW
  BEGIN
INSERT  INTO EmployeeAudit VALES (audit_seq.
                         NEXTUAL, : OLD.
emp_id : OLD : salary, : NEW.salary, SYSDATE);
 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_user_activity
AFTER INSERT OR UPDATE OR DELETE ON Employees
                    FOR EACH ROW
BEGIN
   INSERT INTO Auditlog VALUES (audit_seq.NXQTVAL,
 CASE WHEN INSERTING THEN 'INSERT' WHEN
      UPDATING THEN 'UPDATE'
'Employees', NUL (:OLD emp-id :NEW emp_id),
      SYSTEM, 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 (
     Salesid NUMBER PRIMARY KEY,
amount NUMBER (10,2);
     running_total number (10,2)
 );
CREATE OR REPLACE TRIGGER update_running total
         FOR EACH ROW

BEGIN
   SELECT NVL (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_stack_before_order
 BEFORE INSERT ON order
     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)	
Program/Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	