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 parent
for each row
declare
    V_Count number;
begin
   select Count (*) into V-Count
   from child
  where child.parent_id = :old.parent_id;
  if V_Count > 0 then
     Raise_application_error (-20001, 'Can't delete');
     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 trigger pdv
before isert or update on employee
for each row
declare
   v_count number
begin
   select count(*) into v_count
 from employee
 where emp-id = :New.emp-id;
 if v-count >0 then
    raise_application_error(-20002, 'duplicate value found');
 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 or replace trigger rtu
before insert on sales
for each row
declare
  v_total number;
  V-limit Constant number := 100000;
begin
  select sum (~~amount~~) into v_total from sales;
 if ( v_total + :NEW.amount ) > V.limit Then
  Raise-application_error (-20003, 'insertion not allowed : total exc
    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 trigger audit_empchange
 after update of salary on employee
for each row
begin
   insert into emp_audit (emp_id, old_salary, new_salary,
  changed_on)
Values (: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 user_activity
after insert or update or delete on employee
  Begin
  insert into audit_log values
  (user, ORA_SYSTEM, 'EMPLOYEE, SYSDATE);

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 or replace trigger update_running_total
after insert on sales
for each row
declare
   v_total. number
begin
   select NVL (sum(amount), 0) into v_total from sales;
  update total_summary set running_total = v_total;
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 check-item-availability
before insert on orders
for each row
declare
    v_stock number
    v-pending number
begin
    select stock into v_stock
    from items
    where item-id = :New.item_id;
select NVL (sum(quantity), 0) into v-pending
from orders
where item-id = new.item-id and status = "pending";

END;
/
```

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