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 departments
for each row
declare
    v_count number;
begin
  select count(*) into v_count
    from employees
  where department_id = :old.department_id;

  if v_count > 0 then
      raise_application_error(-20001,
   'cannot delete department: Employee exist
      in this department');
  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 prevent_duplicate_
email
before insert or update on employees
for each row
declare
  v_count number;
begin
    select count(*) into v_count
    from employees
    where email = :New.email
if v_count > 0 then
        raise_application_error(-20002,
    'duplicate email detected! each employee
must have a unique email');
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 salary_limit_check
before insert on employees
for each row
declare
   v_limit constant number := 100000;
begin
   select nvl(sum(salary),0) + :new.salary
      into v_total_salary
      from employees;
   if v_total_salary > v_limit then
      raise_application_error(-20003,
         'Insertion rejected: Total salary (' ||
   of ' || v_limit || ' exceeded');
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 emp trigger employee_audit-
        trigger

after update of salary, job_id on employees
for each row
begin
  if :old.salary != New salary then
        insert into employee_audit(
employee_id, column_name, old_value, new_value,
changed_by, changed_on) values (:old.employee_id
('salary', to_char(:old.salary), to_char
  (:new.salary));
  new, sysdate
  );
end if;
if :old.job_id != new job_id then
        :old employee_id, 'job_id',
        :old.job_id : new.job_id;
  new, sysdate

  );
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 employee_activity_audit
after insert or update or delete on employees
  for each row
declare
    v_action varchar2(10);
begin
  if inserting then
      v_action := 'insert';
  elsif updating then
      v_action := 'update';
  else if deleting then
      v_action := 'delete';
    end if;

    insert into audit_log (
table_name, operation, user_name, operation_date, primary_key
value ) values (
  'employees', v_action, user, sysdate);
    end
  when inserting then to_char(:new.employee_id)
  when updating then to_char(:new.employee_id)

  ): end
end;
/
```