1.To write a database trigger before delete for each row, not allowing deletion and giving an appropriate message, for the employee table.

PROGRAM

```
create or replace trigger emply_trigg before delete on employee
for each row
Begin
raise_application_error(-20768,'Deletion not allowed on employee')
End;
/
```

Trigger Created.

2. Create a trigger that may insert a tuple into EMPLOYEE(empno,ename) when a tuple is updated into MANAGER(managerid,mname) the trigger has to check whether the new tuple has a first component 50 or less, and if so insert the tuple into EMPLOYEE.

Tables Used

Manager

MANAGERID	MNAME
41	Nandana
101	Alan
1	Sameer

Employee

EMPNO	ENAME

PROGRAM

```
create or replace trigger tr2
after update on manager
referencing new as newrow
for each row when (newrow.managerid<=50)
begin
insert into employee values
(:newrow.empno, :newrow.ename);
end tr2;
/
```

Trigger Created.

3.Create a trigger that may insert a tuple into table5 when a tuple is inserted into table4, the trigger has to check whether the new tuple has first component 10 or less, and if so insert the tuple into table5.

Tables Used

TABLE4

NO	NAME
5	Rajesh
13	Rejeev
10	Reneesh

TABLE5

NO	NAME

PROGRAM

```
create or replace trigger tr1
after insert on table4
referencing new as newrow
for each row when (newrow.no<=10)
begin
insert into table5 values(:newrow.no, :newrow.name);
end tr1;
/
```

Trigger Created.

4.PROCEDURE PROGRAM I

AIM: Create a procedure to evaluate student grade by accepting student name.

ALGORITHM

Step 1: Start procedure.
Step 2: Define procedure sgrade taking parameter name and program variable mk(mark).
Step 3: Select mark of student into an integer(mk) whose name had been passed.
Step 4: If mk greater than 90 then
Print “Grade:=X”.
Else if mk greater than 80 then
Print “Grade:=A”.

Else if mk greater than 60 then
Print “Grade:=B”.
Else if mk greater than 50 then
Print “Grade:=C”.
Else Print “Grade:=F”.

Step 5: Stop procedure.

PROGRAM

```
create or replace procedure sgrade(n_name IN varchar2)
IS
mk number(3);
Begin
select mark into mk from student where name=n_name;
if mk>90 then
dbms_output.put_line('Grade:=X');
else if mk>80 then
dbms_output.put_line('Grade:=A');
else if mk>60 then
dbms_output.put_line('Grade:=B');
else if mk>50 then
dbms_output.put_line('Grade:=C');
else
dbms_output.put_line('Grade:=F');
end if;
end if;
end if;   end if;
End sgrade;
/
Procedure created.
SQL> execute sgrade('rajesh');
```

Grade:=B