Trigger

1) ;new /

Trigger are invoked by oracle engine automatically when a specified events occur. This are stored Programs in database & are invoked repitedly when specified condition matches.

Trigger can be return in responce of the following events.
DML Commands like Insert, update, delete.

Syntax:-
create or replace trigger trigger-name
before/after insert/update/delete
on table-name for each row
begin

end;
/

Types of triggers:-

① Statement trigger :-
execute only once for the triggering event.

② Row trigger :-
execute once for each row effected by the triggering event.

Imp Bind Variables:-

① :new:-
This variables holds new value of database columns at the time of Insert & update command

② :old:-
This variables holds old value of database columns at the time of delete command.

4 marks question
Example

① Game(Gname, no-of-players, coach-name)
write a trigger which will fire before insert or update on game having no-of-players less than or equal to 0 rise user define execption & give appropriate msg.

→ create or replace trigger t1
before Insert or update
on Game for each row.
begin
if :new.no-of-players<=0 then
raise-application-error(-20005,'you cannot insert less then zero value')
end if;
end;
/

② student(rno, sname, class, time-table)
Lab(Lno, Labname, capacity, equipment)
write a trigger which will fire before delete on lab.

→
```
create or replace trigger t2 before delete on
Lab for each row
begin
raise_application_error(-20005,'you cannot delete record
                                on lab table');
end;
/
```

③ Library(Lno, Lname, Location, Librarian, no-of-books)
book(bid, bname, author, Price, Publication)
Relation between Library & book is one to many.

Q. write a trigger which will fire before insert or update on book having price less than equal to zero (0).

→

```
create or replace trigger l1 before insert or
update on book for each row.
begin
if:new.price <= 0 then
raise_application_error(-20004,'you cannot insert
                        less than zero value
                                or
end if;                 (price should not be less than
end;                      or equal to zero);
/
```

④ employee(eno, ename, city, deptname)
Project(Pno, Pname, status)
Emp-Project(eno, Pno, no-of-days)

Q. write a trigger that restrict insertion or updation of record having no of days less than zero.

→ create or replace trigger t3 before insert or update on emp_project for each row.
begin
if:new.no-of-days < 0 then
raise_application_error(-20001, 'you cannot insert less than zero value');
end if;
end;
/

⑤ employee(eno, ename, salary, commission, designation, dno)
dept(dno, dname, location)

Q. define a trigger that will take care of the constraint that employee salary should not be less than zero.

→

create or replace trigger e1 before insert or update on employee for each row.
begin
if:new.salary < 0 then
raise_application_error(-20012, 'salary should not be less than zero');
end if;
end;
/

Imp * 4marks

Package :-

A PLSQL Package is a collection of releted Procedures, function, variables and other elements that are group for modularity and Reusability.

A PLSQL Package consist of two Parts.

1. A Package specification.
2. A package body.

1. Package specification :-

It includes declaration of procedures, functions, variables, cursours.

Syntax:-

```
create or replace Package pkg-name is/as
// PLSQL Package specifications.
```

example:-

```
* create or replace Package mypack as
Procedure myproc (p in int);
function myfun (a in int);
end mypack;
```

2. Package body:-

Package body contains the implementation of the details of package. It includes the coading of the Procedure or function which are declared in the Package specification.

Syntax:-

```
create or replace Package body Pkg-name as
//Implementation code of Procedure & functions
end Pkg-name;
```

example:-

* create or replace Package body mypack as
Procedure myproc (p in int) is
begin
// Implementation code;
end myProc;
function myfun (a in int) return int is
begin
// Implementation code;
end myfun;
end mypack;

Imp Vimp question in exam * write a package which will consist of one Procedu and one function. write a procedure which wil display 1st n numbers using for loop. write a function which will return cube of a given number.

→ Specification
create or replace Package mypack as
Procedure firstnnum ( );
function cubeofn (n in int);
end mypack;

Package body:- (optional)
create or replace Package body mypack as
Procedure firstnnum ( ) is
declare
n int;
b int;
begin
n:=&n; (accepting value for user)
for b in 1..n loop
dbms_output.Put_line(b);
end loop;
end firstnnum;

```
function cubeofn(n in int) return int is
begin cint;
begin
c:= n*n*n;
return c;
end cubeofn;
end mypack;

How can be calling
mypack.firstnnum();
mypack.cubeofn(3);
```

write a Package which consist of one Procedure & one function.

Consider relation

Person(Pno, Pname, Padd, Pcity, Phoneno)

Procedure of a Package will display details of given Person. function of a package will count number of Person from Pune city.

→

```
create or replace Package mypack as
Procedure detailspno();
function countPno (n in int);
end mypack;
Package body:
create or replace Package body mypack as
Procedure detailspno () is
declare
P Person%rowtype;
begin
Select * into P from Person;
dbms_output.Put_line(p.pno||" '||P.Phame||' '||P.add||' '||
                     P.city||' '||P.Phno);
```

```
end person details;

function countPno(n in int) return int is
declare
c int ;
begin
select count(Pno) into c from person where
city="Pune";
return c;
end countPno;
end myPack;
```