List-5

1. Write a PL/SQL program to perform the following:
Insert <99,'John',9,25.6> into the Sailors table, Update sname 'John' to 'Joe' in the Sailors table, Delete tuple from the Sailorstable whose sname is 'Joe'

Program:

```
DECLARE
s_id number(2):=99;
s_name varchar2(20):='john';
s_rating number(2):=9;
s_age number(3,1):=25.6;
BEGIN
INSERT INTO Sailors VALUES(s_id,s_name,s_rating,s_age);
end;
```

Query: select * from sailors;

Program:

```
begin
update sailors set sname='joe' where sname='john';
end;
```

Query: select * from sailors;

Program:

```
begin
delete from sailors where sname='joe';
end;
```

Query: select * from sailors;

2. Retrieve tuple from the Sailors table whose sname is 'Lubber' using PL/SQL program

Program:

```
declare
s_id sailors.sid%type;
s_sname sailors.sname%type;
s_rating sailors.rating%type;
s_age sailors.age%type;
begin
```

```
select sid,sname,rating,age into s_id,s_sname,s_rating,s_age from sailors where
sname='lubber';
dbms_output.put_line('sid is '||s_id||' sname is'||s_sname||' rating'||s_rating||' age i
s'||s_age);
end;
```

3. Write a PL/SQL program to show handling of Pre-defined exception.

Program:

```
declare
a number:=4;
c number;
begin
c:=a/0;
exception
when ZERO_DIVIDE then
dbms_output.put_line('divide by zero exception');
end;
```

4. Write a PL/SQL program to show handling of User-defined exception.

Program:

```
declare
a number(2);
b number(2);
c number(2);
ex exception;
begin
a:=&a;
b:=&b;
if(b=0) then
raise ex;
else
c:=a/b;
dbms_output.put_line(c);
end if;
exception
when ex then
dbms_output.put_line('divide by zero');
end;
```

5. Write a PL/SQL program to print the following values

0 if a=0
1 if a>0
-1 if a<0

Program:

```
declare
a number;
begin
a :=&a;
if a=0 then
dbms_output.put_line(&#39;0&#39;);
elsif a>0 then
dbms_output.put_line(&#39;1&#39;);
else
dbms_output.put_line(&#39;-1&#39;);
end if;
end;
```

6. **Write a PL/SQL program to retrieve records from the sailors table using cursor.**

Program:

```
s_id number(2);
s_name varchar2(20);
s_rating number(2);
s_age number(3,1);
cursor c_sailors is select sid,sname,rating,age from sailors;
begin
open c_sailors;
loop
fetch c_sailors into s_id,s_name,s_rating,s_age;
dbms_output.put_line(s_id||' '||s_name||' '||s_rating||' '||s_age);
exit when c_sailors%notfound;
end loop;
close c_sailors;
end;
```

7. **Write a PL/SQL procedure to find the rating of a sailor with the given sailor id.**

Procedure:

```
create or replace procedure findrating(s_id in number, s_rating out number) is
begin
```

```
select rating into s_rating from sailors where sid=s_id;
end findrating;
```

Program:

```
declare
id sailors.sid%type;
srating sailors.rating%type;
begin
id:=&id;
findrating(id,srating);
dbms_output.put_line(id||' rating is '||srating);
end;
```

8. Create a function to return sailor name based on the sailor id.

Function:

```
create or replace function findname(s_id in number) return varchar2 is
name sailors.sname%type;
begin
select sname into name from sailors where sid=s_id;
return name;
end findname;
```

Program:

```
declare
id sailors.sid%type;
name sailors.sname%type;
begin
id:=&id;
name:=findname(id);
dbms_output.put_line('Sailor name is '||name);
end;
```

9. Create a trigger to convert sname to upper case before inserting a record.

Trigger:

```
create or replace trigger triuppersname
before
insert on sailors
for each row
begin
:new.sname := upper(:new.sname);
```

```
end;
```

Query: insert into sailors values(96,'uwxyz',7,43);

10. Create a trigger to display the rating difference between the old values and old values.

Trigger:

```
create or replace trigger displaydiff
before insert or update or delete on sailors
for each row
when (new.sid>0)
declare
rating_diff number;
begin
rating_diff := :new.rating - :old.rating;
dbms_output.put_line('Old rating: '|| :old.rating);
dbms_output.put_line('New rating: '|| :new.rating);
dbms_output.put_line('Rating difference: '|| rating_diff);
end;
```

Query: insert into sailors values(96,'abc',8,32);