Activity 2 PL/SQL Exercise - Stored procedures, functions

1. **Stored Function**

Create stored function is called get_cleaners_location. This function takes as input a cleaner’s number and returns the cleaner’s depot address. Call the function from within an SQL statement to select the cleaner’s name and location for a particular cleaner.

```sql
create or replace function get_cleaners_location (cleaner_num in cleaner.cno%type) return depot.daddress %type as dlocation depot.daddress %type;
begin
    select daddress into dlocation from cleaner c, depot d where cno= cleaner_num and d.dno=c.dno;
    return (dlocation);
end;
/
```

Function created.

```sql
select cname, get_cleaners_location (cno) "Address" from cleaner where cno='110';
```

<table>
<thead>
<tr>
<th>CNAME</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Camden Road</td>
</tr>
</tbody>
</table>

2. **Stored Procedures and cursors**

Examine the following three example stored procedures to print out the same and salary of all bus drivers.

2.a The following example shows a cursor ‘for loop’, with the cursor defined inside the loop.

Create or replace procedure DisplayBusDrivers as
```sql
begin
    for driver in (select * from BusDriver)
    loop
        dbms_output.put_line(driver.bdname || ' ' || driver.bdsalary);
    end loop;
end;
```

We now use the ‘Execute’ statement to execute the stored procedure

```sql
Execute DisplayBusDrivers;
```

Jane Brown 1800
Sally Smith 1750
James Bond 1500
Maggie May 2200
Jack Jones 1400
2.b The next example has the cursor defined in the declaration section, and used in the loop.

Create or replace procedure DisplayBusDrivers2 as
cursor drivercursor is
    select * from busdriver;
begin
    for driver in drivercursor
    loop
        dbms_output.put_line(driver.bdname || ' ' || driver.bdsalary);
    end loop;
end;
Execute DisplayBusDrivers2;

2.c The example below demonstrates explicit open and fetch statements. Notice that the cursor needs to be declared.

Create or replace procedure DisplayBusDrivers3 as
cursor drivercursor is
    select * from busdriver;
driver busdriver%rowtype;
begin
    open drivercursor;
    loop
        fetch drivercursor into driver;
        exit when drivercursor %notfound;
        dbms_output.put_line(driver.bdname || ' ' || driver.bdsalary);
    end loop;
end;
Execute DisplayBusDrivers3;

Write similar code as the three examples above but this time output the cleaners’ name and depot name. Use the Execute statement to test the stored procedures.

Answer 2.a

Create or replace procedure DisplayCleaners as
begin
    for cl in (select cname, dname from cleaner c, depot d where c.dno=d.dno)
    loop
        dbms_output.put_line(cl.cname || ' ' || cl.dname);
    end loop;
end;
Execute DisplayCleaners;

John Holloway
Jean Holloway
Betty Hornsey
Vince Hornsey
Jay Hornsey
Doug Hornsey
PL/SQL procedure successfully completed.

**Answer 2b**

Create or replace procedure DisplayCleaners2 as
cursor cleanercursor is
select cname, dname from cleaner c, depot d where c.dno=d.dno;
beginn
for cl in cleanercursor
loop
    dbms_output.put_line(cl.cname || ' ' || cl.dname);
end loop;
end;
Procedure created.
Execute DisplayCleaners2;
Output as above
PL/SQL procedure successfully completed.

**Answer 2c**

Create or replace procedure DisplayCleaners3 as
cursor cleanercursor is
    select cname, dname from cleaner c, depot d where c.dno=d.dno;
cl cleanercursor%rowtype;
beginn
    open cleanercursor;
    loop
        fetch cleanercursor into cl;
        exit when cleanercursor%notfound;
        dbms_output.put_line(cl.cname || ' ' || cl.dname);
    end loop;
end;
Create or replace procedure DisplayBusDrivers3 as
cursor drivercursor is
    select * from busdriver;
driver busdriver%rowtype;
beginn
    open drivercursor;
    loop
        fetch drivercursor into driver;
        exit when drivercursor %notfound;
        dbms_output.put_line(driver.bdname || ' ' || driver.bdsalary);
    end loop;
end;
John Holloway
Jean Holloway
Betty Hornsey
Vince Hornsey
Jay Hornsey
Doug Hornsey
PL/SQL procedure successfully completed.