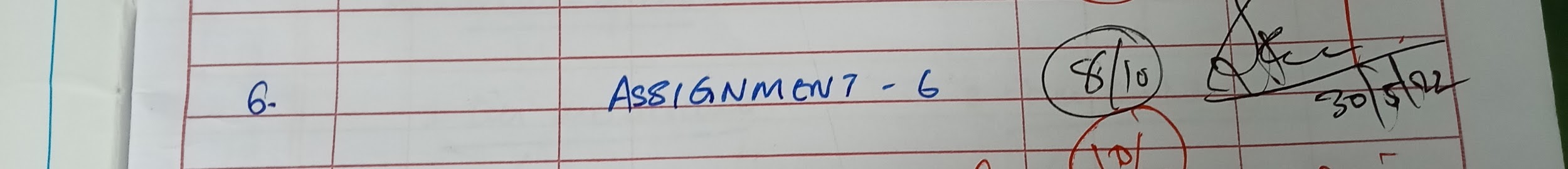
**UCS 1412 - DATABASE LAB**

**ASSIGNMENT 6- Procedures and Functions**

NAME: DIKSHITHA VANI V

ROLL NO: 205001032

INDEX PAGE:

****

**SQL SPOOL FILE:**

SQL> @ Z:\csea\_a\_1032\assig6\assig6.sql

SQL> set serveroutput on;

SQL> REM \* Write a PL/SQL stored procedure for the following: \*

SQL> REM \*\*\* 1. For the given receipt number, calculate the Discount as follows:

SQL> REM For total amount > $10 and total amount < $25: Discount=5%

SQL> REM For total amount > $25 and total amount < $50: Discount=10%

SQL> REM For total amount > $50: Discount=20%

SQL> REM Calculate the amount (after the discount) and update the same in Receipts table.

SQL> REM Print the receipt \*\*\*

SQL>

SQL> REM \* PROCEDURE TO FIND AMOUNT TO BE PAID AFTER DISCOUNT \*

SQL>

SQL> create or replace procedure discount (rno\_cust IN receipts.rno%type) is

2 dis products.price%type;

3 amount products.price%type;

4 total products.price%type;

5 Sno int;

6 namef customers.fname%type;

7 namel customers.lname%type;

8 dater Receipts.rdate%type;

9 Cursor c1 is select food,flavor,price from products join item\_list on item=pid and rno=rno\_cust;

10 record c1%rowtype;

11 Begin

12 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

13 select fname,lname,rdate into namef,namel,dater from customers c join receipts r on r.cid=c.cid and rno=rno\_cust;

14 dbms\_output.put\_line('Receipt number:'||rno\_cust||' '||'Customer Name:'||namef||' '||namel);

15 dbms\_output.put\_line('Date:'||dater);

16 dbms\_output.put\_line('sno'||chr(9)||'food'||chr(9)||'flavor'||chr(9)||chr(9)||'price');

17 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

18 sno:=0;

19 open c1;

20 loop

21 fetch c1 into record;

22 if c1%NOTFOUND then

23 exit;

24 end if;

25 sno:=sno+1;

26 dbms\_output.put\_line(sno||chr(9)||record.food||chr(9)||record.flavor||chr(9)||chr(9)||record.price);

27 end loop;

28 close c1;

29 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

30 select sum(price) into amount from item\_list join products on item=pid where rno=rno\_cust;

31 if amount>10 and amount<25 then

32 dis:=amount\*(5)/100;

33 elsif amount<50 then

34 dis:=amount\*(10)/100;

35 else

36 dis:=amount\*(20)/100;

37 end if;

38 total:=amount-dis;

39 dbms\_output.put\_line('Amount:'||amount);

40 dbms\_output.put\_line('Discount is:'||dis );

41 dbms\_output.put\_line( 'TOTAL:'||total);

42 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

43 end;

44 /

Procedure created.

SQL>

SQL> declare

2 rno\_cust number;

3 begin

4 rno\_cust:=&rno;

5 discount(rno\_cust);

6 end;

7 /

Enter value for rno: 13355

old 4: rno\_cust:=&rno;

new 4: rno\_cust:=13355;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Receipt number:13355 Customer Name:TOUSSAND SHARRON

Date:19-OCT-07

sno food flavor price

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1 Cake Opera 15.95

2 Cookie Lemon .79

3 Cake Napoleon 13.49

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Amount:30.23

Discount is:3.02

TOTAL:27.21

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PL/SQL procedure successfully completed.

SQL>

SQL> REM \*\*\* 2. Ask the user for the budget and his/her preferred food type. You recommend the best

SQL> REM item(s) within the planned budget for the given food type. The best item is

SQL> REM determined by the maximum ordered product among many customers for the given

SQL> REM food type. Print the recommended product that suits your budget: \*\*\*

SQL>

SQL> REM \*PROCEDURE TO FIND RECOMMENATION \*

SQL>

SQL> create or replace procedure recommend(budget IN Number,selected\_food varchar2) is

2 cursor c1 is select item,flavor,food,price,count(\*) as freq from item\_list join products on pid=item where food=selected\_food group by(item,flavor,food,price);

3 item\_dis products.pid%type;

4 flav\_dis products.flavor%type;

5 price\_dis products.price%type;

6 count\_ int;

7 record c1%rowtype;

8 begin

9 count\_:=0;

10 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

11 dbms\_output.put\_line('Budget:'||budget||' '||'Food type :'||selected\_food);

12 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

13 dbms\_output.put\_line('ItemID'||chr(9)||chr(9)||chr(9)||'flavor'||chr(9)||chr(9)||'food'||chr(9)||chr(9)||'price');

14 open c1;

15 loop

16 fetch c1 into record;

17 if c1%NOTFOUND then

18 exit;

19 end if;

20 If length(record.flavor)>=9 then

21 dbms\_output.put\_line(record.item||chr(9)||record.flavor||chr(9)||chr(9)||record.food||chr(9)||chr(9)||record.price);

22 else

23 dbms\_output.put\_line(record.item||chr(9)||chr(9)||record.flavor||chr(9)||chr(9)||record.food||chr(9)||chr(9)||record.price);

24 end if;

25 if(record.freq>count\_) then

26 item\_dis:=record.item;

27 flav\_dis:=record.flavor;

28 price\_dis:=record.price;

29 count\_:=record.freq;

30 end if;

31 end loop;

32 count\_:=0;

33 count\_:=floor(budget/price\_dis);

34 dbms\_output.put\_line('---------------------------------------------------------------------------');

35 dbms\_output.put\_line('');

36 dbms\_output.put\_line(item\_dis||' with '||flav\_dis||' is the best item in'||selected\_food ||'type ! You are entitled to purchase '||count\_||' '||selected\_food||flav\_dis||'for the given budget.');

37 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

38 end;

39 /

Procedure created.

SQL>

SQL>

SQL> declare

2 budget Number;

3 selected\_food varchar2(15);

4 begin

5 budget:=&budget;

6 selected\_food:=&food;

7 recommend(budget,selected\_food);

8 end;

9 /

Enter value for budget: 10

old 5: budget:=&budget;

new 5: budget:=10;

Enter value for food: 'Meringue'

old 6: selected\_food:=&food;

new 6: selected\_food:='Meringue';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Budget:10 Food type :Meringue

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ItemID flavor food price

70-M-CH-DZ Chocolate Meringue 1.25

70-M-VA-SM-DZ Vanilla Meringue 1.15

---------------------------------------------------------------------------

70-M-CH-DZ with Chocolate is the best item inMeringuetype ! You are entitled to

purchase 8 MeringueChocolatefor the given budget.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PL/SQL procedure successfully completed.

SQL>

SQL> REM \*\*\*3. Take a receipt number and item as arguments, and insert this information into the Item list.

SQL> REM However, if there is already a receipt with that receipt number, then keep adding 1 to the maximum

SQL> REM ordinal number. Else before inserting into the Item list with ordinal as 1, ask the user to give the

SQL> REM customer name who placed the order and insert this information into the Receipts.

SQL>

SQL> REM \* PROCEDURE TO INSERT INTO ITEM\_LIST \*

SQL> Create or replace procedure p3

2 (rno1 in number,item1 in varchar) is

3 Ord number; my\_ord number;

4 ln customers.lname%type; mycid customers.cid%type;

5 Cursor c1 is select max(ordinal) from item\_list i join receipts r on r.rno=i.rno group by r.rno having r.rno=rno1;

6 Begin

7 Open c1;

8 Fetch c1 into ord;

9 If c1%FOUND then

10 my\_ord:=ord+1;

11 Insert into item\_list values(rno1,my\_ord,item1);

12 Else

13 ln:=&ln;

14 Select cid into mycid from Customers where lname=ln;

15 If sql%FOUND then

16 my\_ord:=1;

17 Insert into receipts values(rno1,NULL,mycid);

18 Insert into item\_list values(rno1,my\_ord,item1);

19 Else

20 dbms\_output.put\_line(' cust not found');

21 End if;

22 End if;

23 Close c1;

24 End;

25 /

Enter value for ln: 'RAYFORD'

old 13: ln:=&ln;

new 13: ln:='RAYFORD';

Procedure created.

SQL>

SQL> REM anonymous block

SQL> begin

2 p3(51997,'70-R');

3 end;

4 /

begin

\*

ERROR at line 1:

ORA-20002: The item to be purchased is already purchased thrice.

ORA-06512: at "1032.CHECK\_RECEIPTS", line 16

ORA-04088: error during execution of trigger '1032.CHECK\_RECEIPTS'

ORA-06512: at "1032.P3", line 11

ORA-06512: at line 2

SQL>

SQL>

SQL> REM \*\*\* 4. Write a stored function to display the customer name who ordered maximum for the

SQL> REM given food and flavor. \*\*\*

SQL>

SQL> REM \* FUNCTION TO DISPLAY THE CUSTOMER WHO ORDERED MAXIMUM OF GIVEN FOOD AND FLAVOR \*

SQL>

SQL> create or replace function maxorder

2 (gn\_food IN products.food%type, gn\_flavor IN products.flavor%type)

3 return Number as

4 maxcount number:=0;

5 item\_id products.pid%type;

6 c\_name varchar2(40);

7 count\_ number:=0;

8 fname\_ customers.fname%type;

9 lname\_ customers.lname%type;

10 begin

11 select pid into item\_id from products where food=gn\_food and flavor=gn\_flavor;

12 select max(count(\*)) into maxcount from receipts inner join item\_list using(rno)

13 where item=item\_id group by cid;

14 declare cursor c1 is select cid from receipts inner join item\_list using(rno) where item=item\_id group by cid having count(\*)=maxcount;

15 c\_id c1%rowtype;

16 begin

17 open c1;

18 loop

19 fetch c1 into c\_id;

20 if c1%NOTFOUND then

21 return count\_;

22 end if;

23 select fname,lname into fname\_,lname\_ from customers where cid=c\_id.cid;

24 c\_name:=lname\_||' '||fname\_;

25 dbms\_output.put\_line('Customer Name: '||c\_name);

26 count\_:=count\_+1;

27 end loop;

28 end;

29 return count\_;

30 end;

31 /

Function created.

SQL>

SQL> declare

2 count\_ Number;

3 gn\_food products.food%type;

4 gn\_flavor products.flavor%type;

5 begin

6 gn\_food:='&food\_to\_seach';

7 gn\_flavor:='&flavor\_to\_seach';

8 count\_:=maxorder(gn\_food,gn\_flavor);

9 dbms\_output.put\_line('Number of customers with maximum orders= '||count\_);

10 end;

11 /

Enter value for food\_to\_seach: Cake

old 6: gn\_food:='&food\_to\_seach';

new 6: gn\_food:='Cake';

Enter value for flavor\_to\_seach: Lemon

old 7: gn\_flavor:='&flavor\_to\_seach';

new 7: gn\_flavor:='Lemon';

Customer Name: MIGDALIA STADICK

Customer Name: TRAVIS ESPOSITA

Customer Name: MELLIE MCMAHAN

Number of customers with maximum orders= 3

PL/SQL procedure successfully completed.

SQL>

SQL>

SQL> REM \*\*\*5. Implement Question (2) using stored function to return the amount to be paid and

SQL> REM update the same, for the given receipt number. \*\*\*

SQL>

SQL> REM \* FUNCTION TO RETURN THE AMOUNT TO BE PAID AFTER \*

SQL>

SQL>

SQL> create or replace function func\_rec(budget IN Number,selected\_food varchar2) return number is

2 cursor c1 is select item,flavor,food,price,count(\*) as freq from item\_list join products on pid=item where food=selected\_food group by(item,flavor,food,price);

3 item\_dis products.pid%type;

4 flav\_dis products.flavor%type;

5 price\_dis products.price%type;

6 count\_ int;

7 record c1%rowtype;

8 Total Number;

9 begin

10 count\_:=0;

11 total:=0;

12 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

13 dbms\_output.put\_line('Budget:'||budget||' '||'Food type :'||selected\_food);

14 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

15 dbms\_output.put\_line('ItemID'||chr(9)||chr(9)||chr(9)||'flavor'||chr(9)||chr(9)||'food'||chr(9)||chr(9)||'price');

16 open c1;

17 loop

18 fetch c1 into record;

19 if c1%NOTFOUND then

20 exit;

21 end if;

22 If length(record.flavor)>=9 then

23 dbms\_output.put\_line(record.item||chr(9)||record.flavor||chr(9)||chr(9)||record.food||chr(9)||chr(9)||record.price);

24 else

25 dbms\_output.put\_line(record.item||chr(9)||chr(9)||record.flavor||chr(9)||chr(9)||record.food||chr(9)||chr(9)||record.price);

26 end if;

27 if(record.freq>count\_) then

28 item\_dis:=record.item;

29 flav\_dis:=record.flavor;

30 price\_dis:=record.price;

31 count\_:=record.freq;

32 end if;

33 end loop;

34 count\_:=0;

35 count\_:=floor(budget/price\_dis);

36 total:=count\_\*price\_dis;

37 dbms\_output.put\_line('---------------------------------------------------------------------------');

38 dbms\_output.put\_line('');

39 dbms\_output.put\_line(item\_dis||' with '||flav\_dis||' is the best item in'||selected\_food ||'type ! You are entitled to purchase '||count\_||' '||selected\_food||' '||flav\_dis||'for the given budget.');

40 dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

41 Return total;

42 end;

43 /

Function created.

SQL>

SQL>

SQL> declare

2 budget Number;

3 selected\_food varchar2(15);

4 total number;

5 begin

6 budget:=&budget;

7 selected\_food:=&food;

8 total:=func\_rec(budget,selected\_food);

9 dbms\_output.put\_line('The total amount to be paid is:'||total);

10 end;

11 /

Enter value for budget: 99

old 6: budget:=&budget;

new 6: budget:=99;

Enter value for food: 'Meringue'

old 7: selected\_food:=&food;

new 7: selected\_food:='Meringue';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Budget:99 Food type :Meringue

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ItemID flavor food price

70-M-CH-DZ Chocolate Meringue 1.25

70-M-VA-SM-DZ Vanilla Meringue 1.15

---------------------------------------------------------------------------

70-M-CH-DZ with Chocolate is the best item inMeringuetype ! You are entitled to

purchase 79 Meringue Chocolatefor the given budget.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The total amount to be paid is:98.75

PL/SQL procedure successfully completed.

SQL>

SQL> spool off;