SQL> @ z:/lab6.sql

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> alter table RECEIPTS

2 add amount float default 0;

Table altered.

SQL>

SQL> CREATE OR REPLACE PROCEDURE

2 calcDiscount(rid IN item\_list.rno%type)

3

4 IS

5

6 cfname customers.fname%type;

7 clname customers.lname%type;

8 rpt\_dt date;

9 prod\_row products%rowtype;

10 total\_amt float(10);

11 discount\_cent number(2);

12 discount\_amt float(10);

13 final\_amt float(10);

14 no\_of\_items number(3);

15

16 cursor c1 is select pid,flavor,food,price

17 from item\_list join products on item=pid

18 where rno=rid;

19

20 BEGIN

21

22 no\_of\_items:=1;

23 select fname,lname into cfname,clname

24 from customers natural join receipts

25 where rno=rid;

26

27 select rdate into rpt\_dt from receipts where rno=rid;

28

29 select sum(price) into total\_amt

30 from item\_list join products on item=pid

31 where rno=rid;

32

33 if total\_amt>10 and total\_amt<=25 then

34 discount\_cent:=5;

35 elsif total\_amt>25 and total\_amt<=50 then

36 discount\_cent:=10;

37 elsif total\_amt>50 then

38 discount\_cent:=20;

39 end if;

40

41 discount\_amt:=(total\_amt\*discount\_cent)/100;

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

43 dbms\_output.put\_line

44 ('Receipt Number: '||rid||' Customer Name:'||cfname||' '||clname);

45 dbms\_output.put\_line('Receipt Date : '||rpt\_dt);

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

47 dbms\_output.put\_line('S.no Flavor Food Price');

48 for prod\_row in c1 loop

49 dbms\_output.put\_line(no\_of\_items||' '||prod\_row.flavor||' '||prod\_row.food||' '||prod\_row.price);

50 no\_of\_items:=no\_of\_items+1;

51 end loop;

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

53 dbms\_output.put\_line(' '||'Total = '||total\_amt);

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

55 dbms\_output.put\_line('Total Amount :$ '||total\_amt);

56 dbms\_output.put\_line('Discount('||discount\_cent||'%) :$ '||discount\_amt);

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

58 final\_amt:=total\_amt-discount\_amt;

59 update RECEIPTS set amount = final\_amt where RECEIPTS.rno = rid;

60 dbms\_output.put\_line('Amount to be paid :$ '||final\_amt);

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

62 dbms\_output.put\_line('Great Offers! Discount up to 25% on DIWALI Festival Day...');

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

64 END;

65 /

Procedure created.

SQL> declare

2 ono item\_list.rno%type;

3 BEGIN

4 ono:='&Reciept\_no';

5 calcDiscount(ono);

6 end;

7 /

Enter value for reciept\_no: 13355

old 4: ono:='&Reciept\_no';

new 4: ono:='13355';

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

Receipt Number: 13355 Customer Name:TOUSSAND SHARRON

Receipt Date : 19-OCT-07

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

S.no Flavor Food Price

1 Opera Cake 15.95

2 Lemon Cookie .79

3 Napoleon Cake 13.49

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

Total = 30.23

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

Total Amount :$ 30.23

Discount(10%) :$ 3.023

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

Amount to be paid :$ 27.21

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

Great Offers! Discount up to 25% on DIWALI Festival Day...

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

PL/SQL procedure successfully completed.

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.

SQL>

SQL> CREATE OR REPLACE PROCEDURE

2 chooseProd (budget IN PRODUCTS.price%type, foodType IN PRODUCTS.food%type)

3

4 IS

5

6 prod\_row PRODUCTS%rowtype;

7 bestpid PRODUCTS.pid%type;

8 bestflav PRODUCTS.flavor%type;

9 bestprice PRODUCTS.price%type;

10 bestfood PRODUCTS.food%type;

11 num number(5);

12 cnt number(2);

13

14 cursor c1 is select pid,food,flavor,price,count(\*)

15 from PRODUCTS join ITEM\_LIST on pid=item

16 where food=foodType and price<=budget

17 group by pid,food,flavor,price

18 order by count(\*) desc;

19

20 cursor c2 is select pid,food,flavor,price,count(\*)

21 from PRODUCTS join ITEM\_LIST on pid=item

22 where food=foodType and price<=budget

23 group by pid,food,flavor,price

24 order by count(\*) desc;

25

26 BEGIN

27

28 open c2;

29 fetch c2 into bestpid, bestfood, bestflav, bestprice,cnt;

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

31 dbms\_output.put\_line

32 ('Budget:$ '||budget||' Food Type:'||foodType);

33 dbms\_output.put\_line('Item ID Flavor Food Price');

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

35 for prod\_row in c1 loop

36 dbms\_output.put\_line(prod\_row.pid||' '||prod\_row.flavor||' '||foodType||' '||prod\_row.price);

37 end loop;

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

39 dbms\_output.put\_line(bestpid||' with '||bestflav||' flavor is the best type in '||foodType||' type!');

40 num := trunc(budget/bestprice);

41 dbms\_output.put\_line('You are entitled to purchase '||num||' '||foodType||' chocolates for the given budget !!!');

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

43 END;

44 /

Procedure created.

SQL> declare

2 budg PRODUCTS.price%type;

3 fooType PRODUCTS.food%type;

4 BEGIN

5 fooType := '&food\_type';

6 budg := '&budget';

7 chooseProd(budg,fooType);

8 end;

9 /

Enter value for food\_type: Meringue

old 5: fooType := '&food\_type';

new 5: fooType := 'Meringue';

Enter value for budget: 10

old 6: budg := '&budget';

new 6: budg := '10';

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

Budget:$ 10 Food Type:Meringue

Item ID 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 flavor is the best type in Meringue type!

You are entitled to purchase 8 Meringue chocolates for 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

SQL> REM the Item list. However, if there is already a receipt with that receipt number, then

SQL> REM keep adding 1 to the maximum ordinal number. Else before inserting into the Item list

SQL> REM with ordinal as 1, ask the user to give the customer name who placed the order and insert

SQL> REM this information into the Receipts.

SQL>

SQL> alter table receipts

2 drop column amount;

Table altered.

SQL>

SQL> CREATE OR REPLACE PROCEDURE

2 insertRow(rec\_no IN RECEIPTS.rno%type, item\_no IN ITEM\_LIST.item%type, o OUT ITEM\_LIST.ordinal%type)

3

4 IS

5

6 ord ITEM\_LIST.ordinal%type;

7 ordc ITEM\_LIST.ordinal%type;

8

9 cursor c1 is select ordinal

10 from ITEM\_LIST

11 where rno=rec\_no;

12 BEGIN

13

14 ordc:=1;

15 o:=1;

16 open c1;

17 loop

18 fetch c1 into ord;

19 if c1%FOUND then

20 o:=o+1;

21 else

22 exit;

23 end if;

24 end loop;

25

26 END;

27 /

Procedure created.

SQL> declare rc\_no RECEIPTS.rno%type;ite\_no ITEM\_LIST.item%type;cust\_id CUSTOMERS.cid%type;date\_in RECEIPTS.rdate%type;o ITEM\_LIST.ordinal%type;

2

3 BEGIN

4 rc\_no:='&Receipt\_no';

5 ite\_no:='&Item';

6 insertRow(rc\_no,ite\_no,o);

7 if o = 1 then

8 cust\_id := '&customer\_id';

9 date\_in := '&date';

10 insert into Receipts values(rc\_no, date\_in, cust\_id);

11 end if;

12 INSERT into item\_list values(rc\_no, o, ite\_no);

13 dbms\_output.put\_line('Inserted '||rc\_no||' '||o||' '||ite\_no);

14 end;

15 /

Enter value for receipt\_no: 10001

old 4: rc\_no:='&Receipt\_no';

new 4: rc\_no:='10001';

Enter value for item: 51-BLU

old 5: ite\_no:='&Item';

new 5: ite\_no:='51-BLU';

Enter value for customer\_id: 2

old 8: cust\_id := '&customer\_id';

new 8: cust\_id := '2';

Enter value for date: 15-APR-2007

old 9: date\_in := '&date';

new 9: date\_in := '15-APR-2007';

Inserted 10001 1 51-BLU

PL/SQL procedure successfully completed.

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

SQL> REM maximum for the given food and flavor.

SQL>

SQL>

SQL> create or replace function maxcustomer(p IN products.pid%type) return varchar2 as c customers.cid%type;

2

3 m int;

4 n1 customers.fname%type;

5 n2 customers.lname%type;

6 name varchar2(40);

7 BEGIN

8 select max(count(\*)) into m from receipts r join item\_list i on i.rno = r.rno

9 where i.item = p

10 group by r.cid;

11 select r.cid into c from receipts r join item\_list i on i.rno = r.rno

12 where i.item = p

13 group by r.cid

14 having count(\*) = m;

15 select c1.fname into n1 from customers c1 where c1.cid = c;

16 select c1.lname into n2 from customers c1 where c1.cid = c;

17 name := n1||n2;

18 return name;

19 end maxcustomer;

20 /

Function created.

SQL> declare

2 name varchar2(40);

3 p products.pid%type;

4 fo products.food%type;

5 fl products.flavor%type;

6 BEGIN

7 fo:='&food';

8 fl:='&flavor';

9 select p1.pid into p from products p1 where p1.food = fo and p1.flavor = fl;

10 name := maxcustomer(p);

11 dbms\_output.put\_line('Name: '||name);

12 end;

13 /

Enter value for food: Eclair

old 7: fo:='&food';

new 7: fo:='Eclair';

Enter value for flavor: Coffee

old 8: fl:='&flavor';

new 8: fl:='Coffee';

Name: ZEMESTEPHEN

PL/SQL procedure successfully completed.

SQL>

SQL>

SQL>

SQL> REM 5. Implement Question (1) using stored function to return the amount to be paid

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

SQL>

SQL> alter table RECEIPTS

2 add amount float default 0;

Table altered.

SQL>

SQL>

SQL> CREATE OR REPLACE FUNCTION

2 discountCalc(rid IN item\_list.rno%type) return PRODUCTS.price%type as final\_amt PRODUCTS.price%type;

3

4 cfname customers.fname%type;

5 clname customers.lname%type;

6 rpt\_dt date;

7 prod\_row products%rowtype;

8 total\_amt float(10);

9 discount\_cent number(2);

10 discount\_amt float(10);

11 no\_of\_items number(3);

12

13 cursor c1 is select pid,flavor,food,price

14 from item\_list join products on item=pid

15 where rno=rid;

16

17 BEGIN

18

19 no\_of\_items:=1;

20 select fname,lname into cfname,clname

21 from customers natural join receipts

22 where rno=rid;

23

24 select rdate into rpt\_dt from receipts where rno=rid;

25

26 select sum(price) into total\_amt

27 from item\_list join products on item=pid

28 where rno=rid;

29

30 if total\_amt>10 and total\_amt<=25 then

31 discount\_cent:=5;

32 elsif total\_amt>25 and total\_amt<=50 then

33 discount\_cent:=10;

34 elsif total\_amt>50 then

35 discount\_cent:=20;

36 end if;

37

38 discount\_amt:=(total\_amt\*discount\_cent)/100;

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

40 dbms\_output.put\_line

41 ('Receipt Number: '||rid||' Customer Name:'||cfname||' '||clname);

42 dbms\_output.put\_line('Receipt Date : '||rpt\_dt);

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

44 dbms\_output.put\_line('S.no Flavor Food Price');

45 for prod\_row in c1 loop

46 dbms\_output.put\_line(no\_of\_items||' '||prod\_row.flavor||' '||prod\_row.food||' '||prod\_row.price);

47 no\_of\_items:=no\_of\_items+1;

48 end loop;

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

50 dbms\_output.put\_line(' '||'Total = '||total\_amt);

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

52 dbms\_output.put\_line('Total Amount :$ '||total\_amt);

53 dbms\_output.put\_line('Discount('||discount\_cent||'%) :$ '||discount\_amt);

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

55 final\_amt:=total\_amt-discount\_amt;

56 dbms\_output.put\_line('Amount to be paid :$ '||final\_amt);

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

58 dbms\_output.put\_line('Great Offers! Discount up to 25% on DIWALI Festival Day...');

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

60 return final\_amt;

61 END;

62 /

Function created.

SQL>

SQL> declare

2 ono item\_list.rno%type;Amount PRODUCTS.price%type;

3 BEGIN

4 ono:='&Reciept\_no';

5 Amount:=discountCalc(ono);

6 update RECEIPTS set amount = Amount where RECEIPTS.rno = ono;

7 end;

8 /

Enter value for reciept\_no: 51991

old 4: ono:='&Reciept\_no';

new 4: ono:='51991';

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

Receipt Number: 51991 Customer Name:SOPKO RAYFORD

Receipt Date : 17-OCT-07

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

S.no Flavor Food Price

1 Truffle Cake 15.95

2 Apple Pie 5.25

3 Apple Tart 3.25

4 Chocolate Tart 3.75

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

Total = 28.2

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

Total Amount :$ 28.2

Discount(10%) :$ 2.82

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

Amount to be paid :$ 25.38

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

Great Offers! Discount up to 25% on DIWALI Festival Day...

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

PL/SQL procedure successfully completed.

SQL> spool off;