SQL> @D:/Ex06.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 as shown below:

SQL> REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL> REM Receipt Number:13355 Customer Name: TOUSSAND SHARRON

SQL> REM Receipt Date :19­Oct­2007

SQL> REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL> REM Sno Flavor Food Price

SQL> REM 1. Opera Cake 15.95

SQL> REM 2. Lemon Cookie 0.79

SQL> REM 3. Napoleon Cake 13.49

SQL> REM ­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­

SQL> REM Total = $ 30.23

SQL> REM ­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­

SQL> REM Total Amount :$ 30.23

SQL> REM Discount(10%) :$ 3.02

SQL> REM ­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­

SQL> REM Amount to be paid :$ 27.21

SQL> REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

SQL> REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL>

SQL> create or replace procedure discount(cp IN products.price%type, dis OUT products.price%type,dp OUT products.price%type,sp OUT products.price%type) is

2 begin

3 dis := 0;

4 dp := 0;

5 if cp>10 and cp<25 then

6 dis := (5\*cp)/100.00;

7 dp := 5;

8 else

9 if cp>25 and cp<50 then

10 dis := (10\*cp)/100.00;

11 dp := 10;

12 else

13 if cp>50 then

14 dis := (20\*cp)/100.00;

15 dp := 20;

16 end if;

17 end if;

18 end if;

19 sp := cp -dis;

20 end;

21 /

Procedure created.

SQL> declare

2 sel receipts.rno%type;

3 billdate receipts.r\_date%type;

4 custlname customers.lname%type;

5 custfname customers.fname%type;

6 cursor c1 is select p.food ,p.flavor,sum(p.price)

7 from products p join item\_list i on i.item=p.pid

8 where i.rno = sel

9 group by p.food ,p.flavor;

10 cp products.price%type;

11 d products.price%type;

12 dp products.price%type;

13 sp products.price%type;

14 counts integer;

15 food\_sel products.food%type;

16 flavor\_sel products.flavor%type;

17 lprice products.price%type;

18 begin

19 sel := &receipt;

20 select sum(p.price) into cp from products p join item\_list i on p.pid = i.item

21 where i.rno = sel;

22 select count(count(\*)) into counts from products p join item\_list i on p.pid = i.item

23 where i.rno = sel

24 group by p.food,p.flavor;

25 open c1;

26 select c.lname,c.fname,r.r\_date into custfname,custlname,billdate from receipts r join customers c on c.cid = r.cid

27 where r.rno=sel;

28

29

30 dbms\_output.put\_line('Customer name: '||custfname||' '||custlname);

31 dbms\_output.put\_line('Receipt No.: '||sel);

32 dbms\_output.put\_line('Receipt date: '||billdate);

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

34 dbms\_output.put\_line('SNO FOOD FLAVOR ');

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

36 for a in 1..counts loop

37 fetch c1 into food\_sel,flavor\_sel,lprice;

38 dbms\_output.put\_line(' '||a||' '||flavor\_sel||' '||food\_sel);

39 end loop;

40 discount(cp,d,dp,sp);

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

42 dbms\_output.put\_line('Total = $ '||cp);

43 dbms\_output.put\_line('Discount ('||dp||'%) = $ '||d);

44 dbms\_output.put\_line('Grand Total = $ '||sp);

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

46 dbms\_output.put\_line('Upto 20% discount available!');

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

48 end;

49 /

Enter value for receipt: 79287

old 19: sel := &receipt;

new 19: sel := 79287;

Customer name: HELING RUPERT

Receipt No.: 79287

Receipt date: 30-OCT-07

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

SNO FOOD FLAVOR

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

1 Vanilla Eclair

2 Blueberry Danish

3 Lemon Tart

4 Pecan Tart

5 Apple Tart

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

Total = $ 14.65

Discount (5%) = $ .73

Grand Total = $ 13.92

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

Upto 20% discount available!

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

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.

SQL> REM Print the recommended product that suits your budget as below:

SQL> REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL> REM Budget: $10 Food type: Meringue

SQL> REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL> REM Item ID Flavor Food Price

SQL> REM 70­M­CH­DZ Chocolate Meringue 1.25

SQL> REM 70­M­VA­SM­DZ Vanilla Meringue 1.15

SQL> REM ­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­

SQL> REM 70­M­CH­DZ with Chocolate flavor is the best item in Meringue type!

SQL> REM REM You are entitled to purchase 8 Meringue chocolates for the given

SQL> REM budget !!!

SQL> REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL>

SQL> create or replace procedure calcount(budget in products.price%type, val in products.price%type, qty out integer) is

2 begin

3 if val <= budget then

4 qty := budget/val;

5 else

6 qty := 0;

7 end if;

8 end;

9 /

Procedure created.

SQL> declare

2 budget products.price%type;

3 val products.price%type;

4 pfood products.food%type;

5 qty INTEGER(3);

6 psel products.pid%type;

7 psample products%rowtype;

8 cursor c1 is select p.pid,p.food,p.flavor,p.price

9 from products p join item\_list i on p.pid = i.item

10 where p.price <= budget and p.food = pfood

11 group by p.pid,p.food,p.flavor,p.price

12 order by count(\*) desc;

13 cts integer;

14 fsel products.flavor%type;

15 begin

16 budget := &budget;

17 pfood := '&food';

18

19 open c1;

20 begin

21 select p1.pid,p1.price,p1.flavor into psel,val,fsel

22 from products p1 join item\_list i on p1.pid = i.item

23 where p1.price <= budget and p1.food = pfood

24 group by p1.pid,p1.food,p1.flavor,p1.price

25 having count(\*)>=ALL(select count(\*)

26 from products p2 join item\_list i on p2.pid = i.item

27 where p2.price <= budget and p2.food = pfood

28 group by p2.pid,p2.food,p2.flavor,p2.price);

29 EXCEPTION

30 when no\_data\_found then

31 dbms\_output.put\_line('No Recomendations found');

32 return;

33 end;

34

35 select count(count(\*)) into cts from products p join item\_list i on p.pid = i.item

36 where p.price <= budget and p.food = pfood

37 group by p.pid,p.food,p.flavor,p.price;

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

39

40 dbms\_output.put\_line('SNO PID FOOD FLAVOR PRICE');

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

42

43 for a in 1..cts loop

44 fetch c1 into psample;

45 dbms\_output.put\_line(a||' '||psample.pid||' '||psample.food||' '||psample.flavor||' '||psample.price);

46 end loop;

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

48

49 calcount(budget,val,qty);

50 dbms\_output.put\_line(psel||' with '||fsel||' flavor is the best item in '||pfood||' type!');

51

52 dbms\_output.put\_line(' You are entitled to purchase '||qty||' '||pfood||' '||fsel||' for the given budget !!!');

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

54

55 end;

56 /

Enter value for budget: 20

old 16: budget := &budget;

new 16: budget := 20;

Enter value for food: Tart

old 17: pfood := '&food';

new 17: pfood := 'Tart';

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

SNO PID FOOD FLAVOR PRICE

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

1 90-APP-11 Apple Tart 3.25

2 90-APR-PF Apricot Tart 3.25

3 90-BLK-PF Blackberry Tart 3.25

4 90-BER-11 Berry Tart 3.25

5 90-CHR-11 Cherry Tart 3.25

6 90-CH-PF Chocolate Tart 3.75

7 90-BLU-11 Blueberry Tart 3.25

8 90-PEC-11 Pecan Tart 3.75

9 90-ALM-I Almond Tart 3.75

10 90-LEM-11 Lemon Tart 3.25

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

90-APP-11 with Apple flavor is the best item in Tart

type!

You are entitled to purchase 6 Tart Apple for the

given budget !!!

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

PL/SQL procedure successfully completed.

SQL>

SQL>

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

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

SQL> REM 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

SQL> REM insert this information into the Receipts.

SQL>

SQL> create or replace procedure insertitem(rec IN receipts.rno%type,ordi IN item\_list.ordinal%type,prodid IN products.pid%type) is

2 begin

3 insert into item\_list values(rec,ordi,prodid);

4 end;

5 /

Procedure created.

SQL>

SQL> create or replace procedure insertreceipts(rec IN receipts.rno%type,rdt IN receipts.r\_date%type,rcid IN receipts.cid%type) is

2 begin

3 insert into receipts values(rec,rdt,rcid);

4 end;

5 /

Procedure created.

SQL>

SQL> create or replace procedure findcid(cfname IN customers.fname%type,clname IN customers.lname%type, fcid OUT customers.cid%type) is

2 begin

3 begin

4 select c.cid into fcid

5 from customers c

6 where c.fname= cfname and c.lname= clname;

7 EXCEPTION

8 WHEN no\_data\_found then

9 DBMS\_OUTPUT.PUT\_LINE('customer ID not found');

10 fcid := 0;

11 end;

12 end;

13 /

Procedure created.

SQL> declare

2 cfname customers.fname%type;

3 clname customers.lname%type;

4 fcid customers.cid%type;

5 rec receipts.rno%type;

6 ordi item\_list.ordinal%type;

7 prodid products.pid%type;

8 rdt receipts.r\_date%type;

9 item\_row item\_list%rowtype;

10 cursor c1 is

11 select \*

12 from item\_list i

13 where i.rno = rec

14 order by i.ordinal desc;

15 maxordi item\_list.ordinal%type;

16 begin

17 rec := &RECEIPT;

18 prodid := '&product';

19 open c1;

20 fetch c1 into item\_row;

21 if c1%rowcount>0 then

22 begin

23 ordi := item\_row.ordinal + 1;

24 insertitem(rec,ordi,prodid);

25 return;

26 end;

27 else

28 begin

29

30 dbms\_output.put\_line('Receipt number not found!!!');

31 dbms\_output.put\_line('CREATE A RECEIPT:');

32 cfname := '&firstname';

33 clname := '&lastname';

34 rdt := '&date';

35 findcid(cfname,clname,fcid);

36 insertreceipts(rec,rdt,fcid);

37 ordi := 1;

38 insertitem(rec,ordi,prodid);

39 return;

40 end;

41 end if;

42 end;

43 /

Enter value for receipt: 70796

old 17: rec := &RECEIPT;

new 17: rec := 70796;

Enter value for product: 51-BC

old 18: prodid := '&product';

new 18: prodid := '51-BC';

Enter value for firstname: abc

old 32: cfname := '&firstname';

new 32: cfname := 'abc';

Enter value for lastname: def

old 33: clname := '&lastname';

new 33: clname := 'def';

Enter value for date: 3-oct-2007

old 34: rdt := '&date';

new 34: rdt := '3-oct-2007';

PL/SQL procedure successfully completed.

SQL>

SQL>

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> create or replace function maxcustomer(p IN products.pid%type) return varchar2 as

2 c customers.cid%type;

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>

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: Danish

old 7: fo:='&food';

new 7: fo:='Danish';

Enter value for flavor: Blueberry

old 8: fl:='&flavor';

new 8: fl:='Blueberry';

Name: RAYFORD SOPKO

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> create or replace function discountfun(cp IN products.price%type, dis OUT products.price%type,dp OUT products.price%type) return products.price%type is

2 sp products.price%type;

3 begin

4 dis := 0;

5 dp := 0;

6 if cp>10 and cp<25 then

7 dis := (5\*cp)/100.00;

8 dp := 5;

9 else

10 if cp>25 and cp<50 then

11 dis := (10\*cp)/100.00;

12 dp := 10;

13 else

14 if cp>50 then

15 dis := (20\*cp)/100.00;

16 dp := 20;

17 end if;

18 end if;

19 end if;

20 sp := cp -dis;

21 return sp;

22 end;

23 /

Function created.

SQL> declare

2 sel receipts.rno%type;

3 billdate receipts.r\_date%type;

4 custlname customers.lname%type;

5 custfname customers.fname%type;

6 cursor c1 is select p.food,p.flavor,sum(p.price)

7 from products p join item\_list i on i.item=p.pid

8 where i.rno = sel

9 group by p.food,p.flavor;

10 cp products.price%type;

11 d products.price%type;

12 dp products.price%type;

13 sp products.price%type;

14 counts integer;

15 food\_sel products.food%type;

16 flavor\_sel products.flavor%type;

17 lprice products.price%type;

18 begin

19 sel := &receipt;

20 select sum(p.price) into cp from products p join item\_list i on p.pid = i.item

21 where i.rno = sel;

22 select count(count(\*)) into counts from products p join item\_list i on p.pid = i.item

23 where i.rno = sel

24 group by p.food,p.flavor;

25 open c1;

26 select c.lname,c.fname,r.r\_date into custfname,custlname,billdate from receipts r join customers c on c.cid = r.cid

27 where r.rno=sel;

28

29

30 dbms\_output.put\_line('Customer name: '||custfname||' '||custlname);

31 dbms\_output.put\_line('Receipt No.: '||sel);

32 dbms\_output.put\_line('Receipt date: '||billdate);

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

34 dbms\_output.put\_line('SNO FOOD FLAVOR PRICE');

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

36 for a in 1..counts loop

37 fetch c1 into food\_sel,flavor\_sel,lprice;

38 dbms\_output.put\_line(' '||a||' '||flavor\_sel||' '||food\_sel||' '||lprice);

39 end loop;

40 sp := discountfun(cp,d,dp);

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

42 dbms\_output.put\_line('Total = $ '||cp);

43 dbms\_output.put\_line('Discount ('||dp||'%) = $ '||d);

44 dbms\_output.put\_line('Grand Total = $ '||sp);

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

46 dbms\_output.put\_line('Upto 20% discount available!');

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

48 end;

49 /

Enter value for receipt: 13355

old 19: sel := &receipt;

new 19: sel := 13355;

Customer name: TOUSSAND SHARRON

Receipt No.: 13355

Receipt date: 19-OCT-07

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

SNO FOOD FLAVOR PRICE

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

1 Napoleon Cake 13.49

2 Opera Cake 15.95

3 Lemon Cookie .79

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

Total = $ 30.23

Discount (10%) = $ 3.02

Grand Total = $ 27.21

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

Upto 20% discount available!

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

PL/SQL procedure successfully completed.

SQL>

SQL> spool off