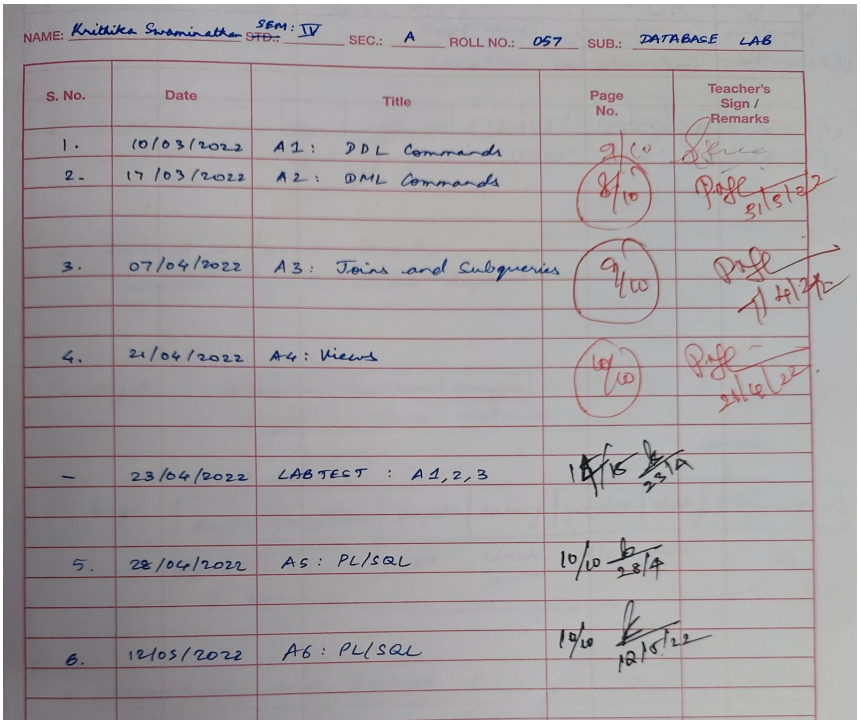
Assignment 6 – PL/SQL: Procedures and Functions

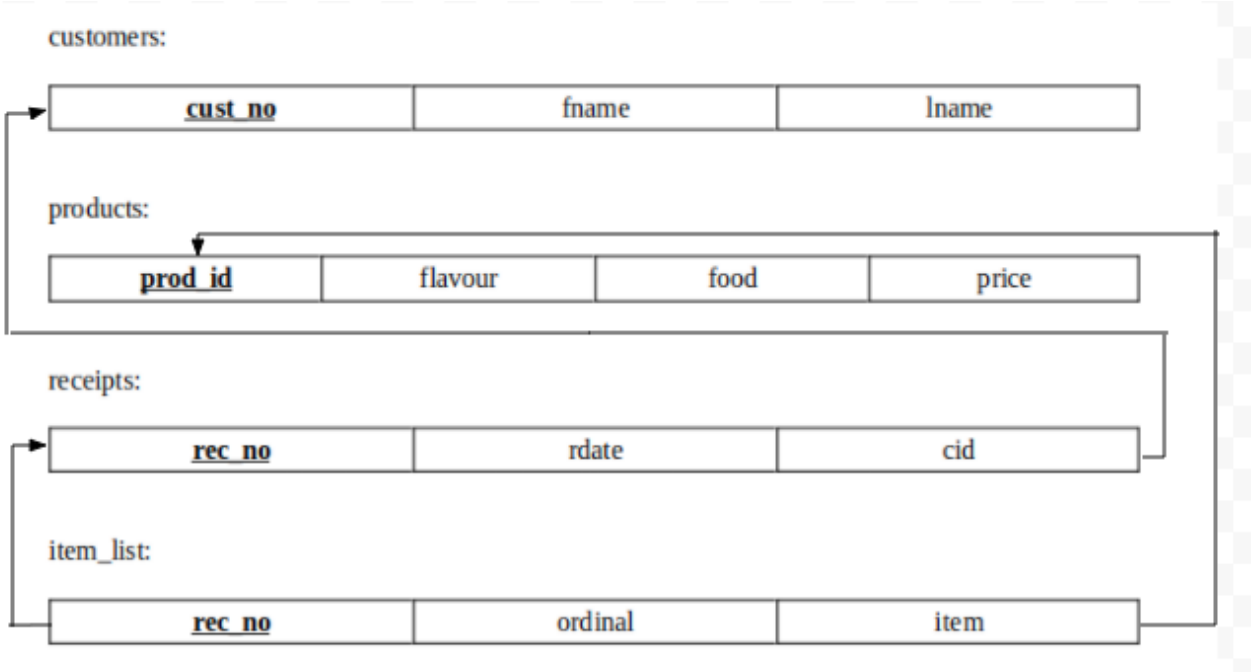
Validation:



NAME: Krithika Swaminathan STD.: SEM: IV SEC.: A ROLL NO.: 057 SUB.: DATABASE LAB

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	10/03/2022	A1: DDL Commands	9/10	
2.	17/03/2022	A2: DML Commands	8/10	Prof 31/3/22
3.	07/04/2022	A3: Joins and Subqueries	9/10	Prof 7/4/22
4.	21/04/2022	A4: Views	10/10	Prof 21/4/22
–	23/04/2022	LAB TEST : A1,2,3	15/15 23/4	
5.	28/04/2022	A5: PL/SQL	10/10 28/4	
6.	12/05/2022	A6: PL/SQL	10/10 12/5/22	

Schema diagram:



Data file:

```
SQL> @C:/Krithika/DBL/a6data.sql;
SQL> REM Population of Bakery Database

SQL> drop table item_list;
Table dropped.

SQL> drop table receipts;
Table dropped.

SQL> drop table products;
Table dropped.

SQL> drop table customers;
Table dropped.

SQL>
SQL> create table customers(
  2         cust_no number(2) constraint c_pk primary key,
  3         lname varchar2(20),
  4         fname varchar2(20)
  5         );

Table created.

SQL>
SQL> insert into customers values(1, 'LOGAN', 'JULIET');
1 row created.
.
.
.
SQL> insert into customers values(21, 'JOHN', 'DAVID');
1 row created.

SQL> create table products(
  2         prod_id varchar2(20) constraint prod_pk primary key,
  3         flavour varchar2(20),
  4         food varchar2(20),
  5         price number
  6         );

Table created.

SQL>
SQL> insert into products values('20-BC-C-10','Chocolate','Cake',8.95);
1 row created.
.
.
```

```
.
SQL> insert into products values('51-BLU','Blueberry','Danish',1.15);
1 row created.

SQL> create table receipts(
  2         rec_no number(5) constraint rec_pk primary key,
  3         rdate date,
  4         cid number(2) constraint rec_fk references customers(cust_no)
  5         );

Table created.

SQL>
SQL> INSERT INTO Receipts values(18129, '28-Oct-2007', 15);
1 row created.
.
.
.
SQL> INSERT INTO Receipts values(34378, '23-Oct-2007', 6);
1 row created.

SQL> create table item_list(
  2         rec_no number(5) constraint it_fk1 references receipts(rec_no),
  3         ordinal number(2),
  4         item varchar2(20) constraint it_fk2 references products(prod_id),
  5         constraint item_pk primary key(rec_no,ordinal)
  6         );

Table created.

SQL>
SQL> insert into item_list values(18129, 1,  '70-TU');
1 row created.
.
.
.
SQL> insert into item_list values(34378, 2,  '45-VA');
1 row created.

SQL>
SQL> REM *** End of database population ***

SQL>
SQL> REM ********************* END OF DATA FILE ************************
```

Script file:

```
SQL> @z:/a6plsql.sql;
SQL> REM Assignment 6
SQL>
SQL> REM ------------------------------------------------------------------------
> REM *** ASSIGNMENT QUESTIONS ***
SQL> REM ------------------------------------------------------------------------
> REM Consider the schema used in Assignment 3.
SQL>
SQL>
SQL> REM **_____Write a PL/SQL stored procedure for the following:_____**
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
SQL> REM 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 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>
SQL> create or replace procedure discount
  2  (amt in products.price%type, dis out products.price%type, dispct out products.price%type, sp
out products.price%type) is
  3  begin
  4          dis := 0;
  5          dispct := 0;
  6          if amt>10 and amt<25 then
  7                  dis := (5*amt)/100.00;
  8                  dispct := 5;
  9          elsif amt>25 and amt<50 then
 10                  dis := (10*amt)/100.00;
 11                  dispct := 10;
```

```
 12          elsif amt>50 then
 13                  dis := (20*amt)/100.00;
 14                  dispct := 20;
 15          end if;
 16          sp := amt - dis;
 17  end;
 18  /
```

Procedure created.

```
SQL>
SQL> declare
  2  rid receipts.rec_no%type;
  3  billdate receipts.rdate%type;
  4  custfname customers.fname%type;
  5  custlname customers.lname%type;
  6  rtotal products.price%type;
  7  d products.price%type;
  8  dp products.price%type;
  9  finalamt products.price%type;
 10  cursor c1 is select flavour, food, price
 11          from products join item_list on (prod_id = item)
 12          where rec_no = rid;
 13  countnum integer;
 14  itemfood products.food%type;
 15  itemfl products.flavour%type;
 16  itemprice products.price%type;
 17
 18  begin
 19
 20  rid := &receipt;
 21
 22  select rdate, fname, lname, count(item), sum(price) as total
 23  into billdate, custfname, custlname, countnum, rtotal
 24  from receipts join item_list using (rec_no) join products on (prod_id = item) join customers on
(cid = cust_no)
 25  group by rec_no, rdate, fname, lname having rec_no = rid;
 26
 27  discount(rtotal,d,dp,finalamt);
 28
 29  dbms_output.put_line('->');
 30  dbms_output.put_line('**************************************************');
 31  dbms_output.put_line('Receipt no.: '||rid);
 32  dbms_output.put_line('Customer name: '||custfname||' '||custlname);
 33  dbms_output.put_line('Receipt date: '||billdate);
 34  dbms_output.put_line('**************************************************');
 35  dbms_output.put_line('S.No.  Flavour     Food       Price');
 36  open c1;
 37  for sno in 1..countnum loop
 38          fetch c1 into itemfl, itemfood, itemprice;
```

```
 39          dbms_output.put_line(sno||'    '||itemfl||'    '||itemfood||'    '||itemprice);
 40          end loop;
 41  close c1;
 42  dbms_output.put_line('_____________________________________________________');
 43  dbms_output.put_line('Total = $'||rtotal);
 44  dbms_output.put_line('Discount('||dp||'%): $'||d);
 45  dbms_output.put_line('_____________________________________________________');
 46  dbms_output.put_line('Grand Total = $'||finalamt);
 47  dbms_output.put_line('**************************************************');
 48  dbms_output.put_line('Great Offers! Discounts up to 25% on DIWALI Day...');
 49  dbms_output.put_line('**************************************************');
 50
 51  end;
 52  /
Enter value for receipt: 13355
old  20: rid := &receipt;
new  20: rid := 13355;
->
**************************************************
Receipt no.: 13355
Customer name: SHARRON TOUSSAND
Receipt date: 19-OCT-07
**************************************************
S.No.  Flavour      Food      Price
1    Opera       Cake       15.95
2    Lemon       Cookie       .79
3    Napoleon       Cake       13.49
_____________________________________________________
Total = $30.23
Discount(10%): $3.023
_____________________________________________________
Grand Total = $27.207
**************************************************
Great Offers! Discounts up to 25% on DIWALI Day...
**************************************************

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL> REM Validations:
SQL>
SQL> select rdate, fname, lname, count(item) as count, sum(price) as total
  2  from receipts join item_list using (rec_no) join products on (prod_id = item) join customers on
(cid = cust_no)
  3  group by rec_no, rdate, fname, lname having rec_no = 13355;

RDATE     FNAME              LNAME                COUNT      TOTAL
--------- ------------------- ------------------- ---------- ----------
19-OCT-07 SHARRON            TOUSSAND                    3      30.23
```

```
SQL>
SQL>
SQL> REM 2. Ask the user for the budget and his/her preferred food type. You recommend the best
item(s) within the planned budget for the given food type.
SQL> REM The best item is determined by the maximum ordered product among many customers
for the given food type.
SQL>
SQL> create or replace procedure calcount
  2  (budget in products.price%type, val in products.price%type, qty out integer) is
  3  begin
  4          if val <= budget then
  5                  qty := budget/val;
  6          else
  7                  qty := 0;
  8          end if;
  9  end;
 10  /

Procedure created.

SQL>
SQL> declare
  2  budget products.price%type;
  3  val products.price%type;
  4  pfood products.food%type;
  5  qty integer(3);
  6  pid products.prod_id%type;
  7  psample products%rowtype;
  8  cursor c1 is select distinct(p.prod_id), p.food, p.flavour, p.price
  9          from products p join item_list i on (p.prod_id = i.item)
 10          where p.price <= budget and p.food = pfood
 11          group by p.prod_id, p.food, p.flavour, p.price
 12          order by count(*) desc;
 13  countnum integer;
 14  pfl products.flavour%type;
 15
 16  begin
 17  budget := '&budget';
 18  pfood := '&food';
 19
 20  begin
 21  select p1.prod_id, p1.price, p1.flavour into pid, val, pfl
 22          from products p1 join item_list i on (p1.prod_id = i.item)
 23          where p1.price <= budget and p1.food = pfood
 24          group by p1.prod_id, p1.food, p1.flavour, p1.price
 25          having count(*) >= all (
 26                  select count(*)
 27                  from products p2 join item_list i on p2.prod_id = i.item
 28                  where p2.price <= budget and p2.food = pfood
```

```
 29                 group by p2.prod_id, p2.food, p2.flavour, p2.price);
 30  EXCEPTION
 31  when no_data_found then
 32          dbms_output.put_line('No recommendations found');
 33          return;
 34  end;
 35
 36  select count(count(*)) into countnum
 37          from products p join item_list i on (p.prod_id = i.item)
 38          where p.price <= budget and p.food = pfood
 39          group by p.prod_id, p.food, p.flavour, p.price;
 40
 41
dbms_output.put_line('*****************************************************************'
);
 42  dbms_output.put_line('S.No.  Prod_ID   Food       Flavour         Price');
 43
dbms_output.put_line('*****************************************************************'
);
 44
 45  open c1;
 46  for sno in 1..countnum loop
 47          fetch c1 into psample;
 48          dbms_output.put_line(sno||' '||psample.prod_id||'    '||psample.food||'   '||psample.flavour||'
'||psample.price);
 49          end loop;
 50  close c1;
 51
dbms_output.put_line('*****************************************************************
*');
 52
 53  calcount(budget,val,qty);
 54  dbms_output.put_line(pid||' with '||pfl||' flavour is the best item in '||pfood||' type!');
 55  dbms_output.put_line('You are entitled to purchase '||qty||' '||pfood||' '||pfl||'s for the given
budget!!!');
 56
dbms_output.put_line('*****************************************************************
*');
 57
 58  end;
 59  /
Enter value for budget: 10
old  17: budget := '&budget';
new  17: budget := '10';
Enter value for food: Meringue
old  18: pfood := '&food';
new  18: pfood := 'Meringue';
*************************************************************
S.No.  Prod_ID       Food        Flavour       Price
*************************************************************
```

```
1 70-M-CH-DZ    Chocolate   Meringue   1.25
2 70-M-VA-SM-DZ    Vanilla   Meringue   1.15
**************************************************************
70-M-CH-DZ with Chocolate flavour is the best item in Meringue type!
You are entitled to purchase 8 Meringue Chocolates 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
Item list.
SQL> REM However, if there is already a receipt with that receipt number, then keep adding 1 to
the maximum ordinal number.
SQL> REM Else, before inserting into the Item list with ordinal as 1, ask the user to give the
customer name who placed the order and insert this information into the Receipts.
SQL>
SQL> create or replace procedure insertitem
  2  (rid in receipts.rec_no%type, ord in item_list.ordinal%type, pid in products.prod_id%type) is
  3  begin
  4          insert into item_list values (rid,ord,pid);
  5  end;
  6  /

Procedure created.

SQL>
SQL> create or replace procedure insertreceipt
  2  (rid in receipts.rec_no%type, rdt in receipts.rdate%type, rcid in customers.cust_no%type) is
  3  begin
  4          insert into receipts values (rid,rdt,rcid);
  5  end;
  6  /

Procedure created.

SQL>
SQL> create or replace procedure findcust
  2  (cfname in customers.fname%type, clname in customers.lname%type, foundcid out
customers.cust_no%type) is
  3  begin
  4          begin
  5                  select c.cust_no into foundcid
  6                  from customers c
  7                  where c.fname = cfname and c.lname = clname;
  8          EXCEPTION
  9          when no_data_found then
 10                  dbms_output.put_line('Customer ID not found!');
 11                  foundcid := 0;
```

```
 12          end;
 13  end;
 14  /
```

Procedure created.

```
SQL>
SQL> declare
  2  cfname customers.fname%type;
  3  clname customers.lname%type;
  4  fcid customers.cust_no%type;
  5  rid receipts.rec_no%type;
  6  ord item_list.ordinal%type;
  7  pid products.prod_id%type;
  8  rdt receipts.rdate%type;
  9  item_row item_list%rowtype;
 10  cursor c1 is select * from item_list i
 11          where i.rec_no = rid
 12          order by i.ordinal desc;
 13  maxord item_list.ordinal%type;
 14  flag number(1) := 0;
 15
 16  begin
 17          rid := '&receipt';
 18          pid := '&product';
 19          open c1;
 20          fetch c1 into item_row;
 21          if c1%rowcount > 0 then
 22                  flag := 1;
 23          end if;
 24          close c1;
 25
 26          if (flag = 1) then
 27                  ord := item_row.ordinal+1;
 28
 29          else
 30                  cfname := '&firstname';
 31                  clname := '&lastname';
 32                  rdt := '&date';
 33                  findcust(cfname,clname,fcid);
 34                  if (fcid = 0) then
 35                          dbms_output.put_line('Cannot add a new customer. Exiting...');
 36                          return;
 37                  end if;
 38                  insertreceipt(rid,rdt,fcid);
 39                  ord := 1;
 40          end if;
 41
 42          insertitem(rid,ord,pid);
 43          dbms_output.put_line('Successfully inserted record!');
```

```
 44
 45  EXCEPTION
 46          when no_data_found then
 47                  dbms_output.put_line('Could not insert record!');
 48  end;
 49  /
Enter value for receipt: 8888
old  17:          rid := '&receipt';
new  17:          rid := '8888';
Enter value for product: 51-BC
old  18:          pid := '&product';
new  18:          pid := '51-BC';
Enter value for firstname: DAVID
old  30:                  cfname := '&firstname';
new  30:                  cfname := 'DAVID';
Enter value for lastname: JOHN
old  31:                  clname := '&lastname';
new  31:                  clname := 'JOHN';
Enter value for date: 11-OCT-07
old  32:                  rdt := '&date';
new  32:                  rdt := '11-OCT-07';
Successfully inserted record!

PL/SQL procedure successfully completed.

SQL> /
Enter value for receipt: 34378
old  17:          rid := '&receipt';
new  17:          rid := '34378';
Enter value for product: 51-BLU
old  18:          pid := '&product';
new  18:          pid := '51-BLU';
Enter value for firstname: JULIET
old  30:                  cfname := '&firstname';
new  30:                  cfname := 'JULIET';
Enter value for lastname: LOGAN
old  31:                  clname := '&lastname';
new  31:                  clname := 'LOGAN';
Enter value for date: 12-0CT-07
old  32:                  rdt := '&date';
new  32:                  rdt := '12-0CT-07';
Successfully inserted record!

PL/SQL procedure successfully completed.

SQL>
SQL> REM Validations:
SQL>
SQL> select * from item_list where rec_no = 8888;
```

```
   REC_NO    ORDINAL ITEM
---------- ---------- --------------------
     8888          1 51-BC

SQL> select * from item_list where rec_no = 34378;

   REC_NO    ORDINAL ITEM
---------- ---------- --------------------
    34378          1 90-CHR-11
    34378          2 45-VA
    34378          3 51-BLU

SQL>
SQL>
SQL> REM 4. Write a stored function to display the customer name who ordered maximum for the
SQL> given food and flavor.
SP2-0734: unknown command beginning "given food..." - rest of line ignored.
SQL>
SQL> create or replace function maxorders
  2  (pid in products.prod_id%type) return varchar2 as
  3  cid customers.cust_no%type;
  4  max int;
  5  name1 customers.fname%type;
  6  name2 customers.lname%type;
  7  name varchar2(40);
  8
  9  begin
 10          select max(count(*)) into max
 11                  from receipts r join item_list i on (i.rec_no = r.rec_no)
 12                  where i.item = pid
 13                  group by r.cid;
 14          select r.cid into cid from receipts r join item_list i on (i.rec_no = r.rec_no)
 15                  where i.item = pid
 16                  group by r.cid having count(*)=max;
 17          select c1.fname, c1.lname into name1, name2 from customers c1 where c1.cust_no = cid;
 18
 19          name := name1||' '||name2;
 20          return name;
 21  EXCEPTION
 22          when no_data_found then
 23                  dbms_output.put_line('The specified product does not exist!');
 24  end maxorders;
 25  /

Function created.

SQL>
SQL> declare
  2  name varchar2(40);
  3  pid products.prod_id%type;
```

```
 4  pfood products.food%type;
 5  pfl products.flavour%type;
 6
 7  begin
 8          pfood := '&food';
 9          pfl := '&flavour';
10          select p1.prod_id into pid from products p1
11                  where p1.food = pfood and p1.flavour = pfl;
12          name := maxorders(pid);
13          dbms_output.put_line('Name: '||name);
14  EXCEPTION
15          when no_data_found then
16                  dbms_output.put_line('The specified product does not exist!');
17  end;
18  /
Enter value for food: Danish
old   8:          pfood := '&food';
new   8:          pfood := 'Danish';
Enter value for flavour: Blueberry
old   9:          pfl := '&flavour';
new   9:          pfl := '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> update the same, for the given receipt number.
  2
SQL> create or replace function discountfn
  2  (amt in products.price%type, dis out products.price%type, dispct out products.price%type)
  3  return products.price%type is
  4  sp products.price%type;
  5
  6  begin
  7          dis := 0;
  8          dispct := 0;
  9          if amt>10 and amt<25 then
 10                  dis := (5*amt)/100.00;
 11                  dispct := 5;
 12          elsif amt>25 and amt<50 then
 13                  dis := (10*amt)/100.00;
 14                  dispct := 10;
 15          elsif amt>50 then
 16                  dis := (20*amt)/100.00;
 17                  dispct := 20;
 18          end if;
 19          sp := amt - dis;
 20          return sp;
```

```
 21  end;
 22  /
```

Function created.

```
SQL>
SQL> declare
  2  rid receipts.rec_no%type;
  3  billdate receipts.rdate%type;
  4  custfname customers.fname%type;
  5  custlname customers.lname%type;
  6  rtotal products.price%type;
  7  d products.price%type;
  8  dp products.price%type;
  9  finalamt products.price%type;
 10  cursor c1 is select flavour, food, price
 11          from products join item_list on (prod_id = item)
 12          where rec_no = rid;
 13  countnum integer;
 14  itemfood products.food%type;
 15  itemfl products.flavour%type;
 16  itemprice products.price%type;
 17
 18  begin
 19
 20  rid := '&receipt';
 21
 22  select rdate, fname, lname, count(item), sum(price) as total
 23  into billdate, custfname, custlname, countnum, rtotal
 24  from receipts join item_list using (rec_no) join products on (prod_id = item) join customers on
(cid = cust_no)
 25  group by rec_no, rdate, fname, lname having rec_no = rid;
 26
 27  finalamt := discountfn(rtotal,d,dp);
 28
 29  dbms_output.put_line('->');
 30  dbms_output.put_line('**************************************************');
 31  dbms_output.put_line('Receipt no.: '||rid);
 32  dbms_output.put_line('Customer name: '||custfname||' '||custlname);
 33  dbms_output.put_line('Receipt date: '||billdate);
 34  dbms_output.put_line('**************************************************');
 35  dbms_output.put_line('S.No.  Flavour     Food       Price');
 36  open c1;
 37  for sno in 1..countnum loop
 38          fetch c1 into itemfl, itemfood, itemprice;
 39          dbms_output.put_line(sno||'     '||itemfl||'     '||itemfood||'    '||itemprice);
 40          end loop;
 41  close c1;
 42  dbms_output.put_line('_____________________________________________________');
 43  dbms_output.put_line('Total = $'||rtotal);
```

```
 44  dbms_output.put_line('Discount('||dp||'%): $'||d);
 45  dbms_output.put_line('_____');
 46  dbms_output.put_line('Grand Total = $'||finalamt);
 47  dbms_output.put_line('**************************************************');
 48  dbms_output.put_line('Great Offers! Discounts up to 25% on DIWALI Day...');
 49  dbms_output.put_line('**************************************************');
 50
 51  end;
 52  /
Enter value for receipt: 13355
old  20: rid := '&receipt';
new  20: rid := '13355';
->
**************************************************
Receipt no.: 13355
Customer name: SHARRON TOUSSAND
Receipt date: 19-OCT-07
**************************************************
S.No.  Flavour       Food       Price
1      Opera         Cake        15.95
2      Lemon         Cookie        .79
3      Napoleon      Cake         13.49
_____
Total = $30.23
Discount(10%): $3.023
_____
Grand Total = $27.207
**************************************************
Great Offers! Discounts up to 25% on DIWALI Day...
**************************************************

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL> REM ************************ END OF FILE ******************************
```