```
@E:184_A62.sql
SQL> REM All tables were created using another file
SQL> REM Display all tables
SQL>
SQL> SELECT * FROM CUSTOMER;

CUST CUST_NAME       ADDRESS                                              PHONE

---- -------------- -------------------------------------------------- ----------
c001 Hari            32 RING ROAD,ALWARPET                              9001200031

c002 Ashok           42 bull ROAD,numgambakkam                          9444120003

c003 Raj             12a RING ROAD,ALWARPET                             9840112003

c004 Raghu           P.H ROAD,Annanagar                              9845712993

c005 Sindhu          100 feet ROAD,vadapalani                        9840166677

c006 Brinda          GST ROAD, TAMBARAM                                 9876543210


6 rows selected.

SQL> SELECT * FROM PIZZA;

PIZZ PIZZA_TYPE UNIT_PRICE
---- ---------- ----------
p001 pan               130
p002 grilled           230
p003 italian           200
p004 spanish             260

SQL> SELECT * FROM ORDERS;

ORDER CUST ORDER_DAT DELV_DATE  TOTAL_AMT   DISCOUNT   BILL_AMT

----- ---- --------- --------- ---------- ---------- ----------
OP100 c001 28-JUN-15 30-JUN-15          0          0          0
OP200 c002 28-JUN-15 30-JUN-15          0          0          0
OP300 c003 29-JUN-15 01-JUL-15          0          0          0
OP400 c004 29-JUN-15 01-JUL-15          0          0          0
OP500 c001 29-JUN-15 01-JUL-15          0          0          0
OP600 c002 29-JUN-15 01-JUL-15          0          0          0

6 rows selected.

SQL> SELECT * FROM ORDER_LIST;

ORDER PIZZ        QTY
----- ---- ----------
OP100 p001          3
```

```
OP100 p002          2
OP100 p003          1
OP100 p004          5
OP200 p003          2
OP200 p001          6
OP200 p004          8
OP300 p003          3
OP400 p001          3
OP400 p004          1
OP500 p003          6

ORDER PIZZ        QTY
----- ---- ----------
OP500 p004          5
OP500 p001
OP600 p002          3

14 rows selected.

SQL>
SQL> REM ******************************************************************************
*************
SQL>
SQL> REM 1.Write a stored function to display the total number of pizzas ordered by the given order number
SQL>
SQL> SET SERVEROUTPUT ON;
SQL> create or replace function totalqty(oid orders.order_no%type) return int is
  2       tot_qty number(3) :=0;
  3       temp_ordno orders.order_no%type;
  4       cursor getOrder is select order_no from orders where order_no=oid;
  5  begin
  6       open getOrder;
  7       fetch getOrder into te mp_ordno;
  8       if getOrder%notfound then
  9        dbms_output.put_line('Order number not found');
 10        else
 11         select sum(qty) into tot_qty from order_list where order_no=oid;
 12        end if;
 13        close getOrder;
 14        return tot_qty;
 15  END;
 16  /

Function created.

SQL> REM ******************************************************************************
*************
SQL>
SQL> REM Function demo for existing order no
SQL> select totalqty('OP100') from dual;

TOTALQTY('OP100')
```

```
----------------
           11

SQL>
SQL> REM order 'OP100' details for verification using sql command
SQL> select order_no,sum(qty) from order_list where order_no='OP100' group by order_no;

ORDER   SUM(QTY)
----- ----------
OP100        11

SQL>
SQL> REM ***********************************************************************************
*************
SQL>
SQL> REM Function demo for NOT existing order_no
SQL> select totalqty('OP700') from dual;

TOTALQTY('OP700')
----------------
            0

Order number not found
SQL> REM ***********************************************************************************
*************
SQL>
SQL> REM ***********************************************************************************
*************
SQL>
SQL> REM 2. Write a PL/SQL block to calculate the total amount, discount and billable amount
SQL> REM (Amount to be paid) as given below:
SQL> REM total amount > 2000 and total amount < 5000:  Discount=5%
SQL> REM For total amount > 5000 and total amount < 10000:  Discount=10%
SQL> REM For total amount > 10000:  Discount=20%
SQL> REM Calculate the billable amount (after the discount) and update the same in orders
SQL> REM table.
SQL> REM Bill Amount = Total – Discount.
SQL>
SQL> REM ***********************************************************************************
*************
SQL>
SQL> REM Note:This prcedure is to calculate for a specific order
SQL>
SQL> create or replace procedure calcAmount(oid orders.order_no%type) is
  2        totamt orders.total_amt%type;
  3        dis orders.discount%type := 0.0;
  4        billamt orders.bill_amt%type;
  5        temp orders.order_no%type;
  6        cursor getOrder is select order_no from orders where order_no=oid;
  7        cursor getList is select sum(p.unit_price*o.qty) from order_list o,pizza p
  8         where p.pizza_id=o.pizza_id and order_no=oid;
  9  begin
```

```
 10        open getOrder;
 11        fetch getOrder into temp;
 12        if getOrder%notfound then
 13         dbms_output.put_line('Order number not found');
 14        end if;
 15        close getOrder;
 16        open getList;
 17        loop
 18         fetch getList into totamt;
 19         exit when getList%notfound;
 20         if totamt>2000 and totamt<5000 then
 21          dis := totamt*0.05;
 22         elsif totamt>5000 and totamt<10000 then
 23          dis := totamt*0.1;
 24         elsif totamt > 10000 then
 25          dis := totamt*0.2;
 26         else
 27          dis:= 0.0;
 28         end if;
 29         billamt := totamt - dis;
 30         update orders set total_amt=totamt,discount=dis,bill_amt=billamt where order_no=oid;
 31        end loop;
 32        close getList;
 33  end;
 34  /

Procedure created.

SQL> REM ***********************************************************************************
*************
SQL>
SQL> REM Orders table before calling the procedure
SQL> select * from orders;

ORDER CUST ORDER_DAT DELV_DATE  TOTAL_AMT   DISCOUNT   BILL_AMT

----- ---- --------- --------- ---------- ---------- ----------
OP100 c001 28-JUN-15 30-JUN-15          0          0          0
OP200 c002 28-JUN-15 30-JUN-15          0          0          0
OP300 c003 29-JUN-15 01-JUL-15          0          0          0
OP400 c004 29-JUN-15 01-JUL-15          0          0          0
OP500 c001 29-JUN-15 01-JUL-15          0          0          0
OP600 c002 29-JUN-15 01-JUL-15          0          0          0

6 rows selected.

SQL>
SQL> REM ***********************************************************************************
************
SQL>
SQL> REM Calling the procedure created
SQL>
SQL> REM Procedure demo for order which does  exist
SQL> call calcAmount('OP100');

Call completed.

SQL>
SQL> REM SQL QUERY FOR VERIFICATION
SQL> SELECT sum(p.unit_price*o.qty) from order_list o,pizza p where p.pizza_id=o.pizza_id and order_no='OP10
0';

SUM(P.UNIT_PRICE*O.QTY)
----------------------
                  2350

SQL>
SQL> REM Procedure demo for order which does not exist
SQL> call calcAmount('OP700');
Order number not found

Call completed.

SQL>
SQL> REM ************************************************************************************
*************
SQL>
SQL> REM Orders table after calling the procedure
SQL> select * from orders;

ORDER CUST ORDER_DAT DELV_DATE  TOTAL_AMT   DISCOUNT   BILL_AMT

----- ---- --------- --------- ---------- ---------- ----------
OP100 c001 28-JUN-15 30-JUN-15       2350      117.5     2232.5

OP200 c002 28-JUN-15 30-JUN-15          0          0          0
OP300 c003 29-JUN-15 01-JUL-15          0          0          0
OP400 c004 29-JUN-15 01-JUL-15          0          0          0
OP500 c001 29-JUN-15 01-JUL-15          0          0          0
OP600 c002 29-JUN-15 01-JUL-15          0          0          0

6 rows selected.

SQL>
SQL> REM ************************************************************************************
*************
SQL>
SQL> REM Note:This prcedure is to calculate for all orders
SQL>
SQL> create or replace procedure calcAllAmt is
          totamt orders.total_amt%type;
          dis orders.discount%type := 0.0;
          billamt orders.bill_amt%type;
          oid orders.order_no%type;
```

```
        cursor getList is select o.order_no,sum(p.unit_price*o.qty) from order_list o,pizza p
            where p.pizza_id=o.pizza_id group by o.order_no;
begin
        open getList;
        loop
         fetch getList into oid,totamt;
         exit when getList%notfound;
         if totamt between 2000 and 5000 then
          dis := totamt*0.05;
         elsif totamt between 5000 and 10000 then
          dis := totamt*0.1;
         elsif totamt > 10000 then
          dis := totamt*0.2;
         else
          dis:= 0.0;
         end if;
         billamt := totamt - dis;
         update orders set total_amt=totamt,discount=dis,bill_amt=billamt where order_no=oid;
        end loop;
        close getList;
 end;
 /

Procedure created.

SQL> REM ****************************************************************************
************
SQL>
SQL> REM Orders table before calling the procedure
SQL> select * from orders;

ORDER CUST ORDER_DAT DELV_DATE  TOTAL_AMT   DISCOUNT   BILL_AMT

----- ---- --------- --------- ---------- ---------- ----------
OP100 c001 28-JUN-15 30-JUN-15       2350      117.5     2232.5

OP200 c002 28-JUN-15 30-JUN-15          0          0          0
OP300 c003 29-JUN-15 01-JUL-15          0          0          0
OP400 c004 29-JUN-15 01-JUL-15          0          0          0
OP500 c001 29-JUN-15 01-JUL-15          0          0          0
OP600 c002 29-JUN-15 01-JUL-15          0          0          0

6 rows selected.

SQL>
SQL> REM ****************************************************************************
************
SQL>
SQL> REM Calling the procedure created
SQL>
SQL> call calcAllAmt();
```

```
Call completed.

SQL>
SQL>
SQL> REM ***********************************************************************************
*************
SQL>
SQL> REM Orders table after calling the procedure
SQL> select * from orders;

ORDER CUST ORDER_DAT DELV_DATE  TOTAL_AMT   DISCOUNT   BILL_AMT

----- ---- --------- --------- ---------- ---------- ----------
OP100 c001 28-JUN-15 30-JUN-15       2350      117.5     2232.5

OP200 c002 28-JUN-15 30-JUN-15       3260        163       3097

OP300 c003 29-JUN-15 01-JUL-15        600          0        600
OP400 c004 29-JUN-15 01-JUL-15        650          0        650
OP500 c001 29-JUN-15 01-JUL-15       2500        125       2375

OP600 c002 29-JUN-15 01-JUL-15        690          0        690

6 rows selected.

SQL>
SQL> REM ***********************************************************************************
*************
SQL>
SQL> REM 3. For the given order number, write a PL/SQL block to print the order as shown in the question:
SQL> REM  Hint: Use the PL/SQL blocks created in 1 and 2.
SQL>
SQL> SET SERVEROUTPUT ON;
SQL> CREATE OR REPLACE PROCEDURE displayBILL(ord_num IN VARCHAR2) IS
  2  custdata CUSTOMER%ROWTYPE;
  3  order_date DATE;
  4  sno NUMBER;
  5  totqty NUMBER(5);
  6  totamt NUMBER(7,2);
  7  discounts NUMBER(7,2);
  8  bills NUMBER(7,2);
  9  CURSOR pizcur IS
 10        SELECT pizza.pizza_type,order_list.qty,pizza.unit_price FROM PIZZA ,ORDER_LIST
 11        WHERE order_list.order_no=ord_num
 12        AND order_list.pizza_id=pizza.pizza_id;
 13
 14  BEGIN
 15  totqty := totalqty(ord_num);
 16  calcAmount(ord_num);
 17
 18  SELECT customer.cust_id,customer.cust_name,customer.address,customer.phone INTO custdata FROM CUST
OMER,orders WHERE orders.order_no=ord_num and orders.cust_id=customer.cust_id;
```

```
 19  SELECT order_date,total_amt INTO order_date,totamt FROM ORDERS WHERE order_no=ord_num;
 20  DBMS_OUTPUT.PUT_LINE('*************************************************************************');
 21  DBMS_OUTPUT.PUT_LINE('ORDER NUMBER:'||ord_num||'      '||'CUSTOMER NAME: '||custdata.cust_name);
 22  DBMS_OUTPUT.PUT_LINE('ORDER DATE:'||order_date||'      '||'PHONE: '||custdata.phone);
 23  DBMS_OUTPUT.PUT_LINE('*************************************************************************');
 24  DBMS_OUTPUT.PUT_LINE('SNO'||'   '||'PIZZA TYPE'||'      '||'QTY'||'     '||'PRICE'||'   '||'AMOUNT');
 25  sno := 1;
 26  FOR x in pizcur LOOP
 27  DBMS_OUTPUT.PUT_LINE(sno||'.'||'      '||x.pizza_type||'      '||'    '||x.qty||'     '||x.unit_price||'      '||x.qty*x.unit_price);
 28  sno := sno+1;
 29  END LOOP;
 30  DBMS_OUTPUT.PUT_LINE('------------------------------------------------------------------------');
 31  DBMS_OUTPUT.PUT_LINE('  '||'TOTAL = '||'      '||totqty||'    '||'    '||totamt);
 32  SELECT discount,bill_amt INTO discounts,bills FROM ORDERS where orders.order_no=ord_num;
 33  DBMS_OUTPUT.PUT_LINE('------------------------------------------------------------------------');
 34  DBMS_OUTPUT.PUT_LINE('Total Amount        :Rs. '||totamt);
 35  DBMS_OUTPUT.PUT_LINE('Discount (5%)       :Rs. '||discounts);
 36  DBMS_OUTPUT.PUT_LINE('-----------------------------------');
 37  DBMS_OUTPUT.PUT_LINE('Amount to be paid   :Rs. '||bills);
 38  DBMS_OUTPUT.PUT_LINE('-----------------------------------');
 39  DBMS_OUTPUT.PUT_LINE('Great Offers! Discount up to 25% on DIWALI Festival Day... ');
 40  DBMS_OUTPUT.PUT_LINE('**************************************************************');
 41  END;
 42  /

Procedure created.

SQL> REM *************************************************************************************************
SQL>
SQL> REM Calling the procedure created
SQL>
SQL> call displayBILL('OP100');

*************************************************************************

ORDER NUMBER:OP100 CUSTOMER NAME: Hari

ORDER DATE:28-JUN-15 PHONE: 9001200031

*************************************************************************

SNO PIZZA TYPE QTY PRICE AMOUNT

1. pan  3 130 390
2. grilled  2 230 460
3. italian  1 200 200
4. spanish  5 260 1300
```

```
--------------------------------------------------------------------
 TOTAL =  11  2350
--------------------------------------------------------------------
Total Amount       :Rs. 2350
Discount (5%)      :Rs. 117.5
--------------------------------
Amount to be paid   :Rs. 2232.5
--------------------------------
Great Offers! Discount up to 25% on DIWALI Festival Day...

**********************************************************


Call completed.
SQL>
SQL> REM Demo For a order no which doen not exist
SQL> call displayBILL('OP700');
Order number not found
Order number not found

Call completed.
SQL>
SQL> REM ********************************************************************************
*************
SQL> REM ********************************************************************************
*************
SQL>
```