```
REM Display table contents
SQL>
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> select * from Orders;

ORDER CUST ORDER_DAT DELV_DATE
----- ---- --------- ---------
OP100 c001 28-JUN-15 30-JUN-15
OP200 c002 28-JUN-15 30-JUN-15
OP300 c003 29-JUN-15 01-JUL-15
OP400 c004 29-JUN-15 01-JUL-15
OP500 c001 29-JUN-15 01-JUL-15
OP600 c002 29-JUN-15 01-JUL-15

6 rows selected.

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> REM --------------------------------------------------------------------------------------------------------
>
SQL> REM 1. Check whether the given pizza type is available. If not display appropriate message.
SQL>
SQL> SET SERVEROUTPUT ON;
SQL>
SQL> DECLARE
  2   ptype pizza.pizza_type%TYPE;
  3   temp pizza.pizza_type%TYPE;
  4  BEGIN
  5   ptype:='&ptype';
  6   SELECT pizza_type INTO temp FROM Pizza WHERE pizza_type = pType;
  7   IF SQL%FOUND THEN
  8    dbms_output.put_line('Available');
  9   END IF;
 10    EXCEPTION WHEN NO_DATA_FOUND THEN
 11     dbms_output.put_line('Unavailable');
 12  END;
 13  /
Enter value for ptype: pan
old   5:  ptype:='&ptype';
new   5:  ptype:='pan';
Available

PL/SQL procedure successfully completed.

SQL> REM DEMO FOR PIZZA NOT EXISTING
SQL> /
Enter value for ptype: cheesy
old   5:  ptype:='&ptype';
new   5:  ptype:='cheesy';
Unavailable

PL/SQL procedure successfully completed.

SQL> REM --------------------------------------------------------------------------------------------------------
>
SQL> REM 2. For the given customer name and a range of order date, find whether a customer had placed any orde
r, if so
SQL> REM display the number of orders placed by the customer along with the order number(s).
SQL>
SQL>
SQL> DECLARE
```

```
  2   ordercount NUMBER;
  3   customer_name VARCHAR2(15) := '&customer_name';
  4   start_date DATE := '&start_date';
  5   end_date DATE := '&end_date';
  6   ord VARCHAR2(5);
  7   CURSOR cur IS
  8    SELECT order_no
  9    FROM customer,orders
 10     WHERE
 11      (customer.cust_name LIKE customer_name)
 12      AND (orders.order_date BETWEEN start_date AND end_date)
 13      AND(customer.cust_id=orders.cust_id);
 14  BEGIN
 15   ordercount := 0;
 16   dbms_output.put_line('ORDER NUMBERS FOR THE ORDERS PLACED BY '|| customer_name ||' BETWEE
N '|| start_date ||' AND '|| end_date);
 17   OPEN cur;
 18   LOOP
 19    FETCH cur INTO ord;
 20    EXIT WHEN cur%NOTFOUND;
 21    ordercount := ordercount+1;
 22    dbms_output.put_line(ord);
 23   END LOOP;
 24   CLOSE cur;
 25   dbms_output.put_line('TOTAL NUMBER OF ORDERS PLACED ARE: '||ordercount);
 26  END;
 27  /
Enter value for customer_name: Ashok
old   3:  customer_name VARCHAR2(15) := '&customer_name';
new   3:  customer_name VARCHAR2(15) := 'Ashok';
Enter value for start_date: 27-JUN-15
old   4:  start_date DATE := '&start_date';
new   4:  start_date DATE := '27-JUN-15';
Enter value for end_date: 30-JUN-15
old   5:  end_date DATE := '&end_date';
new   5:  end_date DATE := '30-JUN-15';
ORDER NUMBERS FOR THE ORDERS PLACED BY Ashok BETWEEN 27-JUN-15 AND 30-JUN-15

OP200
OP600
TOTAL NUMBER OF ORDERS PLACED ARE: 2

PL/SQL procedure successfully completed.

SQL> REM DEMO FOR EXISTING CUSTOMER WHO HASNT PLACED ORDER IN GIVEN PERIOD
SQL> /
Enter value for customer_name: Ashok
old   3:  customer_name VARCHAR2(15) := '&customer_name';
new   3:  customer_name VARCHAR2(15) := 'Ashok';
Enter value for start_date: 26-JUN-15
old   4:  start_date DATE := '&start_date';
new   4:  start_date DATE := '26-JUN-15';
```

```
Enter value for end_date: 27-JUN-15
old   5:  end_date DATE := '&end_date';
new   5:  end_date DATE := '27-JUN-15';
ORDER NUMBERS FOR THE ORDERS PLACED BY Ashok BETWEEN 26-JUN-15 AND 27-JUN-15

TOTAL NUMBER OF ORDERS PLACED ARE: 0

PL/SQL procedure successfully completed.

SQL> REM DEMO FOR NON EXISTING CUSTOMER
SQL> /
Enter value for customer_name: Vignesh
old   3:  customer_name VARCHAR2(15) := '&customer_name';
new   3:  customer_name VARCHAR2(15) := 'Vignesh';
Enter value for start_date: 27-JUN-15
old   4:  start_date DATE := '&start_date';
new   4:  start_date DATE := '27-JUN-15';
Enter value for end_date: 30-JUN-15
old   5:  end_date DATE := '&end_date';
new   5:  end_date DATE := '30-JUN-15';
ORDER NUMBERS FOR THE ORDERS PLACED BY Vignesh BETWEEN 27-JUN-15 AND 30-JUN-15

TOTAL NUMBER OF ORDERS PLACED ARE: 0

PL/SQL procedure successfully completed.

SQL> REM ----------------------------------------------------------------------------------------------------------
>
SQL> REM 3. Display the customer name along with the details of pizza type and its quantity ordered for the given order number.
SQL> REM Also find the total quantity ordered for the given order number
SQL>
SQL> SET SERVEROUTPUT ON;
SQL> declare
  2  oid varchar2(5);
  3  total int;
  4  customer_name customer.cust_name%type;
  5  customer_id customer.cust_id%type;
  6  cursor c2 is select o.order_no,p.pizza_type,o.qty from order_list o,pizza p where o.pizza_id = p.pizza_id;
  7  begin
  8   total :=0;
  9   oid := '&oid';
 10   select cust_id into customer_id from orders where order_no = oid;
 11   select cust_name into customer_name from customer where cust_id = customer_id;
 12   dbms_output.put_line('Customer name : '||customer_name);
 13   dbms_output.put_line('Ordered Following Pizza');
 14   dbms_output.put_line('PIZZA TYPE QTY');
 15   for cur in c2 loop
 16    if(cur.order_no = oid) then
 17     dbms_output.put_line(cur.pizza_type||'  '||cur.qty);
 18     total:=total+cur.qty;
 19    end if;
 20   end loop;
 21   dbms_output.put_line('------------------------------');
 22   dbms_output.put_line('Total Qty : '||total);
 23   EXCEPTION
 24   when no_data_found then
 25   dbms_output.put_line('Order id Not Available');
 26   end;
 27  /
Enter value for oid: OP100
old   9:  oid := '&oid';
new   9:  oid := 'OP100';
Customer name : Hari
Ordered Following Pizza
PIZZA TYPE QTY
pan  3
grilled  2
italian  1
spanish  5
------------------------------
Total Qty : 11

PL/SQL procedure successfully completed.

SQL> REM DEMO FOR NON EXISTING ORDER
SQL> /
Enter value for oid: OP900
old   9:  oid := '&oid';
new   9:  oid := 'OP900';
Order id Not Available

PL/SQL procedure successfully completed.

SQL> REM ----------------------------------------------------------------------------------------------------------
>
SQL> REM 4. Display the total number of orders that contains one pizza type, two pizza type and so on.
SQL>
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2   CURSOR counter IS SELECT order_no FROM Orders;
  3   tCount INT;
  4   one INT := 0;
  5   two INT := 0;
  6   three INT := 0;
  7   allTypes INT := 0;
  8   orderNum CHAR(5);
  9
 10  BEGIN
 11   OPEN counter;
 12   LOOP
 13    FETCH counter INTO orderNum;
 14    EXIT WHEN counter%NOTFOUND;
 15    SELECT COUNT(*) INTO tCount FROM Order_List
 16    WHERE order_no = orderNum;
 17    IF tCount = 1 THEN
 18     one := one + 1;
 19    ELSIF tCount = 2 THEN
 20     two := two + 1;
 21    ELSIF tCount = 3 THEN
 22     three := three + 1;
 23    ELSE
 24     allTypes := allTypes + 1;
 25    END IF;
 26   END LOOP;
 27   CLOSE counter;
 28   dbms_output.put_line('Only ONE pizza type ' || one);
 29   dbms_output.put_line('Two pizza types     ' || two);
 30   dbms_output.put_line('Three pizza types   ' || three);
 31   dbms_output.put_line('ALL pizza types     ' || allTypes);
 32  END;
 33  /
Only ONE pizza type 2
Two pizza types     1
Three pizza types   2
ALL pizza types     1

PL/SQL procedure successfully completed.

SQL>
SQL> REM -----------------------------------------------------------------------------------------------------
>
```