```
SQL> --*****************************************************
SQL> --UCS1412                               B.Senthil Kumar
SQL> --Database LabAsst. Prof
SQL> --        Computer Science Department
SQL> --                 SSN College of Engineering
SQL> --                        senthil@ssn.edu.in
SQL> --*****************************************************
SQL> --           PIZZA ORDERING DATASET
SQL> --                 Version 1.0
SQL> --                February 05, 2015
SQL> --*****************************************************
SQL> --Sources:
SQL> --         This dataset is prepared for the assignment
SQL> --on DML, PL/SQL blocks in Database Programming.
SQL> --This is a test dataset - pizza ordered on 28 & 29th Jun 2015.
SQL> --Do NOT MODIFY the instances.
SQL> --
SQL> --******************************************************
SQL>
SQL>
SQL> REM customer(cust_id, cust_name, address, phone)
SQL> REM pizza (pizza_id, pizza_type, unit_price)
SQL> REM orders(order_no, cust_id, order_date ,delv_date, total_amt)
SQL> REM order_list(order_no, pizza_id, qty)
SQL>
SQL> DROP TABLE order_list;

Table dropped.

SQL> DROP TABLE orders;

Table dropped.

SQL> DROP TABLE pizza;

Table dropped.

SQL> DROP TABLE customer;

Table dropped.

SQL>
SQL>
SQL> CREATE TABLE customer(
  2  cust_id VARCHAR(7),
  3  cust_name VARCHAR(25),
  4  address VARCHAR(75),
  5  phone NUMBER(10),
  6  CONSTRAINT pk_customer PRIMARY KEY(cust_id));

Table created.

SQL>
SQL> CREATE TABLE pizza(
  2  pizza_id VARCHAR(6),
  3  pizza_type VARCHAR(15),
  4  unit_price NUMBER(5),
  5  CONSTRAINT pk_pizza PRIMARY KEY(pizza_id));

Table created.

SQL>
SQL> CREATE TABLE orders(
  2  order_no VARCHAR(6),
  3  cust_id VARCHAR(6),
  4  order_date DATE,
  5  delv_date DATE,
  6  CONSTRAINT pk_orders PRIMARY KEY(order_no),
  7  CONSTRAINT fk_custid FOREIGN KEY(cust_id) REFERENCES customer(cust_id));

Table created.

SQL>
SQL> CREATE TABLE order_list(
  2  order_no VARCHAR(6),
  3  pizza_id VARCHAR(6),
  4  qty NUMBER,
  5  CONSTRAINT pk_orderlist PRIMARY KEY(order_no, pizza_id),
  6  CONSTRAINT fk_orderno FOREIGN KEY(order_no) REFERENCES orders(order_no),
  7  CONSTRAINT fk_pizzaid FOREIGN KEY(pizza_id) REFERENCES pizza(pizza_id));

Table created.

SQL>
SQL> DESC customer;
 Name
 Null?    Type

 -----------------------------------------------------------------------------
 -------- --------------------------------------------------------
 CUST_ID
 NOT NULL VARCHAR2(7)
 CUST_NAME
 VARCHAR2(25)
 ADDRESS
 VARCHAR2(75)
 PHONE
 NUMBER(10)

SQL> DESC pizza;
 Name
 Null?    Type

 -----------------------------------------------------------------------------
 -------- --------------------------------------------------------
 PIZZA_ID
 NOT NULL VARCHAR2(6)
 PIZZA_TYPE
 VARCHAR2(15)
 UNIT_PRICE
 NUMBER(5)

SQL> DESC orders;
 Name
 Null?    Type

 -----------------------------------------------------------------------------
 -------- --------------------------------------------------------
 ORDER_NO
 NOT NULL VARCHAR2(6)
 CUST_ID
 VARCHAR2(6)
 ORDER_DATE
 DATE
 DELV_DATE
 DATE

SQL> DESC order_list;
```

```
 Name
 Null?    Type

 -----------------------------------------------------------------------------
 -------- --------------------------------------------------------
 ORDER_NO
 NOT NULL VARCHAR2(6)
 PIZZA_ID
 NOT NULL VARCHAR2(6)
 QTY
 NUMBER

SQL>
SQL>
SQL> REM
------------------------------------------------------------------------------
------------
>
SQL> REM customer(cust_id, cust_name,address,phone)
SQL>
SQL> insert into customer values('c001','Hari','32 RING
ROAD,ALWARPET',9001200031);

1 row created.

SQL> insert into customer values('c002','Ashok','42 bull
ROAD,numgambakkam',9444120003);

1 row created.

SQL> insert into customer values('c003','Raj','12a RING
ROAD,ALWARPET',9840112003);

1 row created.

SQL> insert into customer values('c004','Raghu','P.H
ROAD,Annanagar',9845712993);

1 row created.

SQL> insert into customer values('c005','Sindhu','100 feet
ROAD,vadapalani',9840166677);

1 row created.

SQL> insert into customer values('c006','Brinda','GST ROAD, TAMBARAM',
9876543210);

1 row created.

SQL>
SQL>
SQL>
SQL> REM pizza (pizza_id, pizza_type, unit_price)
SQL>
SQL> insert into pizza values('p001','pan',130);

1 row created.

SQL> insert into pizza values('p002','grilled',230);

1 row created.

SQL> insert into pizza values('p003','italian',200);
```

```
1 row created.

SQL> insert into pizza values('p004','spanish',260);

1 row created.

SQL>
SQL> REM insert into pizza values('p005','supremo',250);
SQL>
SQL>
SQL>
SQL> REM orders(order_no, cust_id, order_date ,delv_date)
SQL>
SQL> insert into orders values('OP100','c001','28-JUN-2015','30-JUN-2015');

1 row created.

SQL> insert into orders values('OP200','c002','28-JUN-2015','30-JUN-2015');

1 row created.

SQL> insert into orders values('OP300','c003','29-JUN-2015','01-JUL-2015');

1 row created.

SQL> insert into orders values('OP400','c004','29-JUN-2015','01-JUL-2015');

1 row created.

SQL> insert into orders values('OP500','c001','29-JUN-2015','01-JUL-2015');

1 row created.

SQL> insert into orders values('OP600','c002','29-JUN-2015','01-JUL-2015');

1 row created.

SQL>
SQL>
SQL>
SQL> REM order_list(order_no, pizza_id, qty)
SQL>
SQL> insert into order_list values('OP100','p001',3);

1 row created.

SQL> insert into order_list values('OP100','p002',2);

1 row created.

SQL> insert into order_list values('OP100','p003',1);

1 row created.

SQL> insert into order_list values('OP100','p004',5);

1 row created.

SQL>
SQL> insert into order_list values('OP200','p003',2);

1 row created.
```

```
SQL> insert into order_list values('OP200','p001',6);

1 row created.

SQL> insert into order_list values('OP200','p004',8);

1 row created.

SQL>
SQL> insert into order_list values('OP300','p003',3);

1 row created.

SQL>
SQL> insert into order_list values('OP400','p001',3);

1 row created.

SQL> insert into order_list values('OP400','p004',1);

1 row created.

SQL>
SQL> insert into order_list values('OP500','p003',6);

1 row created.

SQL> insert into order_list values('OP500','p004',5);

1 row created.

SQL> insert into order_list values('OP500','p001',null);

1 row created.

SQL>
SQL> insert into order_list values('OP600','p002',3);

1 row created.

SQL>
SQL> --******************************************************
SQL> set echo on:
SP2-0158: unknown SET option ":"
SQL> set serveroutput on
SQL>
SQL> @z:/Pizza_DB.sql
SP2-0310: unable to open file "z:/Pizza_DB.sql"
SQL>
SQL> REM: ***************************************************************Ex5 -
PL/SQL-CONTROL
STRUCTURES***************************************************************
SQL> REM: PIZZA ORDERING SYSTEM
SQL>
SQL>
SQL> REM: Consider the following relations for Pizza Ordering System:
SQL> REM: CUSTOMER ( cust_id , cust_name, address, phone, cust_friend)
SQL> REM: PIZZA (pizza_id, pizza_type, unit_price)
SQL> REM: ORDERS (order_no, cust_id, order_date ,delv_date, total_amt)
SQL> REM: ORDER_LIST (order_no, pizza_id, qty)
SQL>
SQL>
SQL> REM: Write a PL/SQL block for the following:
SQL> REM: Note: Use implicit/explicit cursor wherever required.
SQL>
SQL> REM: 1. Check whether the given pizza type is available.
SQL> REM:    If not display appropriate message.
SQL>
SQL> DECLARE
  2  pid pizza.pizza_id%TYPE;
  3  ptype pizza.pizza_type%TYPE;
  4  price pizza.unit_price%TYPE;
  5  BEGIN
  6  ptype:=&pizzatype;
  7  SELECT pizza_id, pizza_type, unit_price INTO pid, ptype, price FROM pizza
  WHERE pizza_type=ptype;
  8  dbms_output.put_line('ID: ' ||pid||' Type: ' ||ptype||' Price: '||price);
  9
 10  EXCEPTION
 11  WHEN NO_DATA_FOUND THEN
 12  dbms_output.put_line ('There is no pizza with the type '||ptype);
 13  END;
 14  /
Enter value for pizzatype: 'pan'
old   6: ptype:=&pizzatype;
new   6: ptype:='pan';
ID: p001 Type: pan Price: 130

PL/SQL procedure successfully completed.

SQL> /
Enter value for pizzatype: 'Nivedhitha D'
old   6: ptype:=&pizzatype;
new   6: ptype:='Nivedhitha D';
There is no pizza with the type Nivedhitha D

PL/SQL procedure successfully completed.

SQL> /
Enter value for pizzatype: 'grilled'
old   6: ptype:=&pizzatype;
new   6: ptype:='grilled';
ID: p002 Type: grilled Price: 230

PL/SQL procedure successfully completed.

SQL> /
Enter value for pizzatype: 'spanish'
old   6: ptype:=&pizzatype;
new   6: ptype:='spanish';
ID: p004 Type: spanish Price: 260

PL/SQL procedure successfully completed.

SQL> /
Enter value for pizzatype: 'italian'
old   6: ptype:=&pizzatype;
new   6: ptype:='italian';
ID: p003 Type: italian Price: 200

PL/SQL procedure successfully completed.

SQL> REM: 2. For the given customer name and a range of order date,
SQL> REM:    find whether a customer had
SQL> REM:    placed any order, if so display the number of orders
SQL> REM:    placed by the customer along
SQL> REM:    with the order number(s).
SQL>
SQL> /*
SQL> SELECT c.cust_id, c.cust_name, o.order_no, o.order_date
SQL> FROM orders o, customer c
SQL> WHERE o.cust_id=c.cust_id
SQL> AND c.cust_name='Hari'
SQL> AND o.order_date BETWEEN TO_DATE('01-01-2015','dd-mm-yyyy') AND
TO_DATE('01-10-2015','dd-mm-yyyy');
SQL> */
SQL>
SQL> DECLARE
  2  CURSOR c_orders(cname VARCHAR2, sdate DATE, edate DATE)
  3  IS
  4  SELECT o.order_no
  5  FROM orders o, customer c
  6  WHERE o.cust_id=c.cust_id
  7  AND c.cust_name=cname
  8  AND o.order_date BETWEEN sdate AND edate;
  9
 10  r_order c_orders%ROWTYPE;
 11
 12  name customer.cust_name%TYPE;
 13  stdate orders.order_date%TYPE;
 14  endate orders.order_date%TYPE;
 15
 16  BEGIN
 17  name:=&name;
 18  stdate:=&stdate;
 19  endate:=&endate;
 20
 21  OPEN c_orders(name, stdate, endate);
 22  LOOP
 23  FETCH c_orders INTO r_order;
 24  EXIT WHEN c_orders%NOTFOUND;
 25
 26  dbms_output.put_line('Order Number: '||r_order.order_no);
 27  END LOOP;
 28
 29  dbms_output.put_line('Number of Orders Placed: '||c_orders%ROWCOUNT);
 30  CLOSE c_orders;
 31
 32  EXCEPTION
 33  WHEN NO_DATA_FOUND THEN
 34  dbms_output.Put_line ('The customer '||name||' did not place any orders
 !');
 35  END;
 36  /
Enter value for name: 'Hari'
old  17: name:=&name;
new  17: name:='Hari';
Enter value for stdate: '01-JAN-2015'
old  18: stdate:=&stdate;
new  18: stdate:='01-JAN-2015';
Enter value for endate: '01-OCT-2015'
old  19: endate:=&endate;
new  19: endate:='01-OCT-2015';
Order Number: OP100
Order Number: OP500
Number of Orders Placed: 2

PL/SQL procedure successfully completed.

SQL> REM: 3. Display the customer name along with the details of pizza type
SQL> REM:    and its quantity ordered for the given order number.
SQL> REM:    Also find the total quantity ordered for the given
SQL> REM:    order number as shown below:
SQL>
SQL> REM: SQL> /
SQL> REM: Enter value for oid: OP100
SQL> REM: old 11:oid:='&oid';
SQL> REM: new 11:oid:='OP100';
SQL> REM: Customer name: Hari
SQL> REM: Ordered Following Pizza
SQL> REM: PIZZA TYPE QTY
SQL> REM: Pan3
SQL> REM: Grilled2
SQL> REM: Italian1
SQL> REM: Spanish5
SQL> REM: ----------------------------------
> REM: Total Qty: 11
SQL>
SQL> /*
SQL> SELECT c.cust_name, p.pizza_type, ol.qty
SQL> FROM orders o, customer c, pizza p, order_list ol
SQL> WHERE c.cust_id=o.cust_id
SQL> AND o.order_no=ol.order_no
SQL> AND ol.pizza_id=p.pizza_id
SQL> AND o.order_no='OP100';
SQL>
SQL> SELECT SUM(ol.qty) as total_qty
SQL> FROM orders o, customer c, pizza p, order_list ol
SQL> WHERE c.cust_id=o.cust_id
SQL> AND o.order_no=ol.order_no
SQL> AND ol.pizza_id=p.pizza_id
SQL> AND o.order_no='OP100';
SQL> */
SQL>
SQL> DECLARE
  2  CURSOR c1(ordernum VARCHAR2)
  3  IS
  4  SELECT c.cust_name, p.pizza_type, ol.qty
  5  FROM orders o, customer c, pizza p, order_list ol
  6  WHERE c.cust_id=o.cust_id
  7  AND o.order_no=ol.order_no
  8  AND ol.pizza_id=p.pizza_id
  9  AND o.order_no=ordernum;
 10
 11  CURSOR c2(ordernum VARCHAR2)
 12  IS
 13  SELECT SUM(ol.qty) as total_qty
 14  FROM orders o, customer c, pizza p, order_list ol
 15  WHERE c.cust_id=o.cust_id
 16  AND o.order_no=ol.order_no
 17  AND ol.pizza_id=p.pizza_id
 18  AND o.order_no=ordernum;
 19
 20  r_line1 c1%ROWTYPE;
 21  r_line2 c2%ROWTYPE;
 22  ordernum orders.order_no%TYPE;
 23
 24  BEGIN
 25  ordernum:=&orderno;
 26
 27  dbms_output.put_line('Ordered Following Pizza');
 28  dbms_output.put_line('PIZZA TYPE      QTY');
 29
 30  OPEN c1(ordernum);
 31  LOOP
 32  FETCH c1 INTO r_line1;
 33  EXIT WHEN c1%NOTFOUND;
 34
 35  dbms_output.Put_line (RPAD(r_line1.pizza_type, 10)||LPAD(r_line1.qty,
 9));
 36  END LOOP;
 37  CLOSE c1;
 38
 39  OPEN c2(ordernum);
 40  FETCH c2 INTO r_line2;
 41  dbms_output.put_line('------------------------');
 42  dbms_output.put_line('Total Qty: '||r_line2.total_qty);
 43  CLOSE c2;
 44
 45  EXCEPTION
 46  WHEN NO_DATA_FOUND THEN
 47  dbms_output.Put_line ('INVALID order number !');
 48  END;
 49  /
Enter value for orderno: 'OP100'
old  25: ordernum:=&orderno;
new  25: ordernum:='OP100';
Ordered Following Pizza
PIZZA TYPE      QTY
pan               3
grilled           2
italian           1
spanish           5
------------------------
Total Qty: 11

PL/SQL procedure successfully completed.

SQL> REM: 4. Display the total number of orders that contains one pizza type,
SQL> REM: two pizza type and so on.
SQL>
SQL> REM: Number of Orders that contains
SQL> REM: Only ONE Pizza type 8
SQL> REM: Two Pizza types 3
SQL> REM: Three Pizza types 2
SQL> REM: ALL Pizza types 1
SQL>
SQL> DECLARE
  2  typecount NUMBER;
  3  num NUMBER;
  4  no_ord NUMBER;
  5  CURSOR c4 IS
  6  SELECT COUNT(*) AS num
  7  FROM order_list
  8  GROUP BY order_no;
  9  BEGIN
 10  SELECT COUNT(*) INTO typecount FROM PIZZA;
 11  DBMS_OUTPUT.PUT_LINE('Number of Orders that contains');
 12  FOR i in 1..typecount LOOP
 13  no_ord := 0;
 14  FOR x IN c4 LOOP
 15  IF i=x.num THEN
 16  no_ord := no_ord+1;
 17  END IF;
 18  END LOOP;
 19
 20  IF i=typecount THEN
 21  DBMS_OUTPUT.PUT_LINE('All Pizza Types'|| CHR(9) || no_ord);
 22            ELSE
 23            DBMS_OUTPUT.PUT_LINE(i||' Pizza Types'|| CHR(9) || no_ord);
```

```
           END IF;
END LOOP;
END;
/
Number of Orders that contains
1 Pizza Types   2
2 Pizza Types   1
3 Pizza Types   2
All Pizza Types 1

PL/SQL procedure successfully completed.

SQL> spool off
```