SQL> @D:/ex5.sql

SQL> --\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL> --UCS1412

SQL> --Database Lab

SQL> -- Computer Science Department

SQL> -- SSN College of Engineering

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> 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> set serveroutput on;

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> REM: ------------------------------------------------------------------------------------------

>

SQL> REM: customer(cust\_id, cust\_name,address,phone)

SQL> create table customer(cust\_id varchar2(20) constraint pk1 primary key,cust\_name varchar2(20),address varchar2(30),phone number(10));

Table created.

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> create table pizza(pizza\_id varchar2(20) constraint pk2 primary key ,pizza\_type varchar2(20),unit\_price number(5));

Table created.

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> create table orders(order\_no varchar2(20) constraint pk3 primary key,cust\_id varchar2(20) constraint foriegn1 references customer(cust\_id), order\_date date,delv\_date date);

Table created.

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> create table order\_list(order\_no varchar2(20) constraint foriegn2 references orders(order\_no),pizza\_id varchar2(20) constraint foriegn3 references pizza(pizza\_id),qty number(2));

Table created.

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> REM:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL> REM:\*\*\*CREATION AND INSERTION DONE\*\*\*\*\*

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

SQL>

SQL>

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

SQL> REM:\*\*\*\*DISPALYING OF PROCEDURES\*\*\*\*\*\*\*\*\*\*\*

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

SQL>

SQL>

SQL> REM:1)Check whether the given pizza type is available. If not display appropriate message

SQL>

SQL> DECLARE

2 pty pizza.pizza\_type%type;

3 BEGIN

4 pty:='&pizza\_type';

5 update pizza set unit\_price=unit\_price+0 where pizza\_type=pty;

6 if SQL%FOUND then

7 dbms\_output.put\_line(pty ||' is available');

8 else

9 dbms\_output.put\_line(pty ||' is not available');

10 end if;

11 END;

12 /

Enter value for pizza\_type: pan

old 4: pty:='&pizza\_type';

new 4: pty:='pan';

pan is available

PL/SQL procedure successfully completed.

SQL>

SQL> REM:2)For the given customer name and a range of order date, find whether a customer had

SQL> REM:placed any order, if so display the number of orders placed by the customer along

SQL> REM:with the order number(s).

SQL>

SQL> declare

2 customer\_name varchar2(20);

3 date1 date;

4 date2 date;

5 id customer.cust\_id%type;

6 no\_of\_orders int;

7 cursor c1 is select \* from orders;

8 begin

9 customer\_name := '&name';

10 date1 := '&date1';

11 date2 := '&date2';

12 no\_of\_orders := 0;

13 select cust\_id into id from customer where cust\_name = customer\_name;

14 dbms\_output.put\_line('Customer Name = '|| customer\_name);

15 dbms\_output.put\_line('Orders : ');

16 for cur in c1 loop

17 if(cur.cust\_id = id and cur.order\_date >= date1 and cur.order\_date <=date2 ) then

18 dbms\_output.put\_line(cur.order\_no);

19 no\_of\_orders:=no\_of\_orders+1;

20 end if;

21 end loop;

22 if (no\_of\_orders =0) then

23 dbms\_output.put\_line('No Orders');

24 else

25 dbms\_output.put\_line('Total Order = '|| no\_of\_orders);

26 end if;

27 EXCEPTION

28 when no\_data\_found then

29 dbms\_output.put\_line('Customer Not Available');

30 end;

31 /

Enter value for name: Raj

old 9: customer\_name := '&name';

new 9: customer\_name := 'Raj';

Enter value for date1: 25-JUN-2015

old 10: date1 := '&date1';

new 10: date1 := '25-JUN-2015';

Enter value for date2: 30-JUN-2015

old 11: date2 := '&date2';

new 11: date2 := '30-JUN-2015';

Customer Name = Raj

Orders :

OP300

Total Order = 1

PL/SQL procedure successfully completed.

SQL> REM :3)Display the customer name along with the details of pizza type and its quantity

SQL> REM :ordered for the given order number.

SQL> declare

2 oid varchar2(20);

3 total int;

4 cus\_name customer.cust\_name%type;

5 cus\_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 cus\_id from orders where order\_no= oid;

11 select cust\_name into cus\_name from customer where cust\_id = cus\_id;

12 dbms\_output.put\_line('Customer name : '||cus\_name);

13 dbms\_output.put\_line('PIZZA TYPE QTY');

14 for cur in c2 loop

15 if(cur.order\_no=oid) then

16 dbms\_output.put\_line(cur.pizza\_type||' '||cur.qty);

17 total:=total+cur.qty;

18 end if;

19 end loop;

20 dbms\_output.put\_line('------------------------------');

21 dbms\_output.put\_line('Total Qty : '||total);

22 EXCEPTION

23 when no\_data\_found then

24 dbms\_output.put\_line('Order id Not Available');

25 end;

26 /

Enter value for oid: OP200

old 9: oid:= '&oid';

new 9: oid:= 'OP200';

Customer name : Ashok

PIZZA TYPE QTY

italian 2

pan 6

spanish 8

------------------------------

Total Qty : 16

PL/SQL procedure successfully completed.

SQL>

SQL> REM:3)Display the total number of orders that contains one pizza type, two pizza type and so on.

SQL> declare

2 cursor c4 is select distinct order\_no from order\_list;

3 cursor c5 is select \* from order\_list;

4 one int(1);

5 two int(1);

6 three int(1);

7 four int(1);

8 ct int(1);

9 begin

10 one:=0;

11 three:=0;

12 two:=0;

13 four:=0;

14 for cur in c4 loop

15 ct:=0;

16 for cust in c5 loop

17 if(cur.order\_no = cust.order\_no and cust.qty is not NULL) then

18 ct:=ct+1;

19 end if;

20 end loop;

21 if(ct > 3) then

22 four:=four+1;

23 elsif(ct > 2) then

24 three:=three+1;

25 elsif(ct > 1) then

26 two:=two+1;

27 elsif(ct = 1) then

28 one:=one+1;

29 end if;

30 end loop;

31 dbms\_output.put\_line('No. of orders that contains');

32 dbms\_output.put\_line('Only one Pizza type : '||one);

33 dbms\_output.put\_line('Two Pizza type : '||two);

34 dbms\_output.put\_line('Three Pizza type : '||three);

35 dbms\_output.put\_line('ALL Pizza type : '||four);

36 end;

37 /

No. of orders that contains

Only one Pizza type : 2

Two Pizza type : 2

Three Pizza type : 1

ALL Pizza type : 1

PL/SQL procedure successfully completed.

SQL> /

No. of orders that contains

Only one Pizza type : 2

Two Pizza type : 2

Three Pizza type : 1

ALL Pizza type : 1

PL/SQL procedure successfully completed.

SQL> spool off;