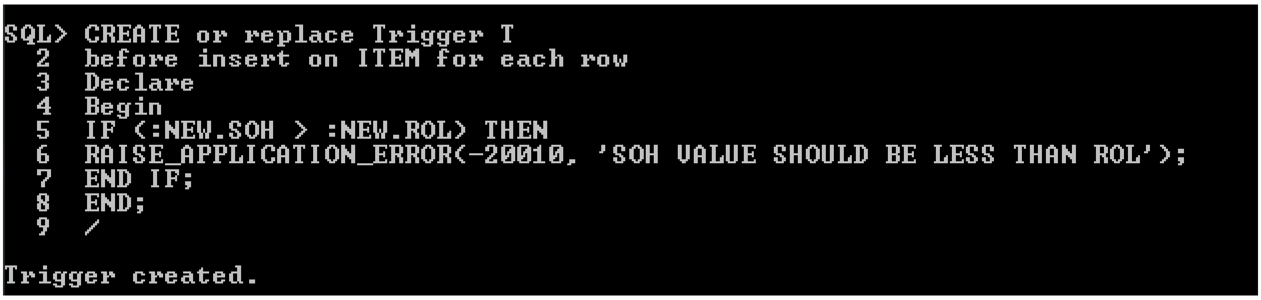
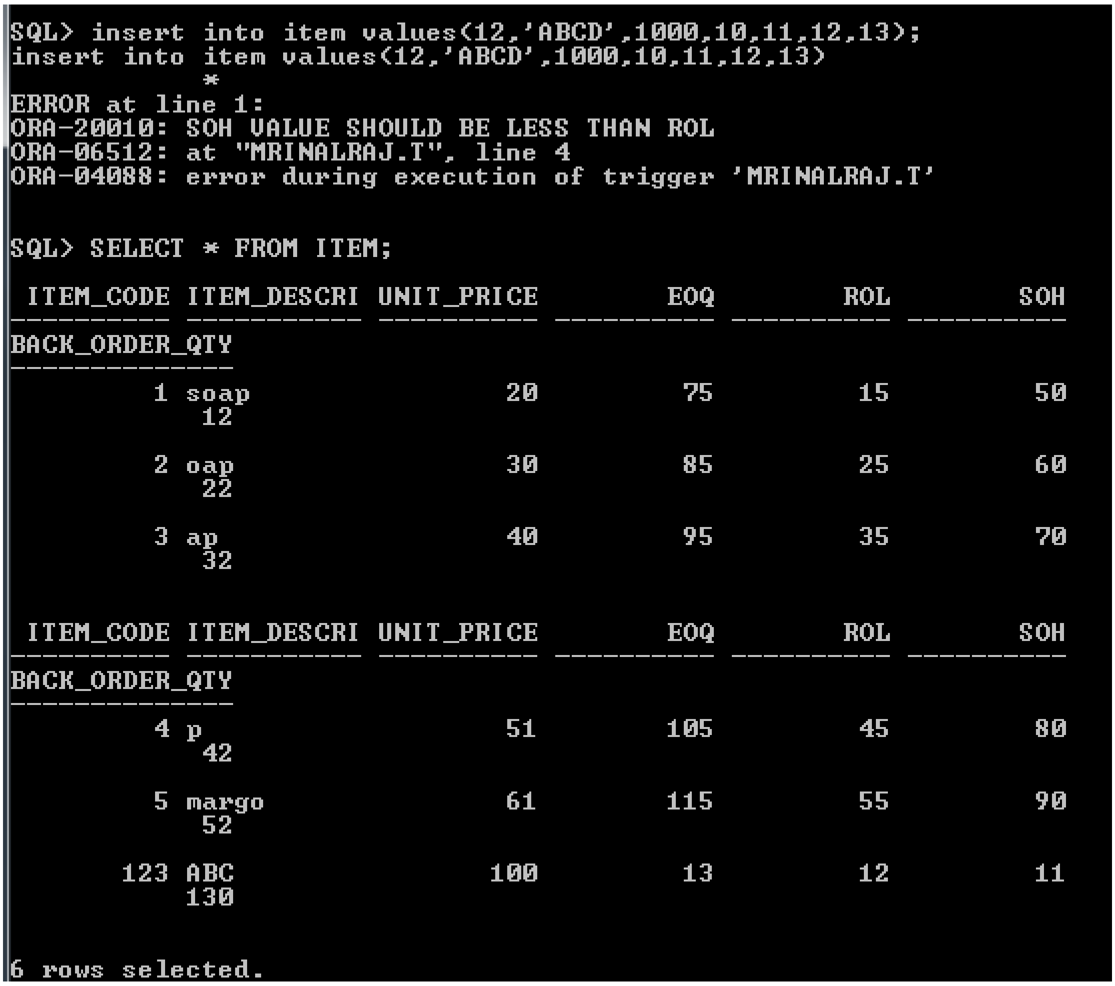
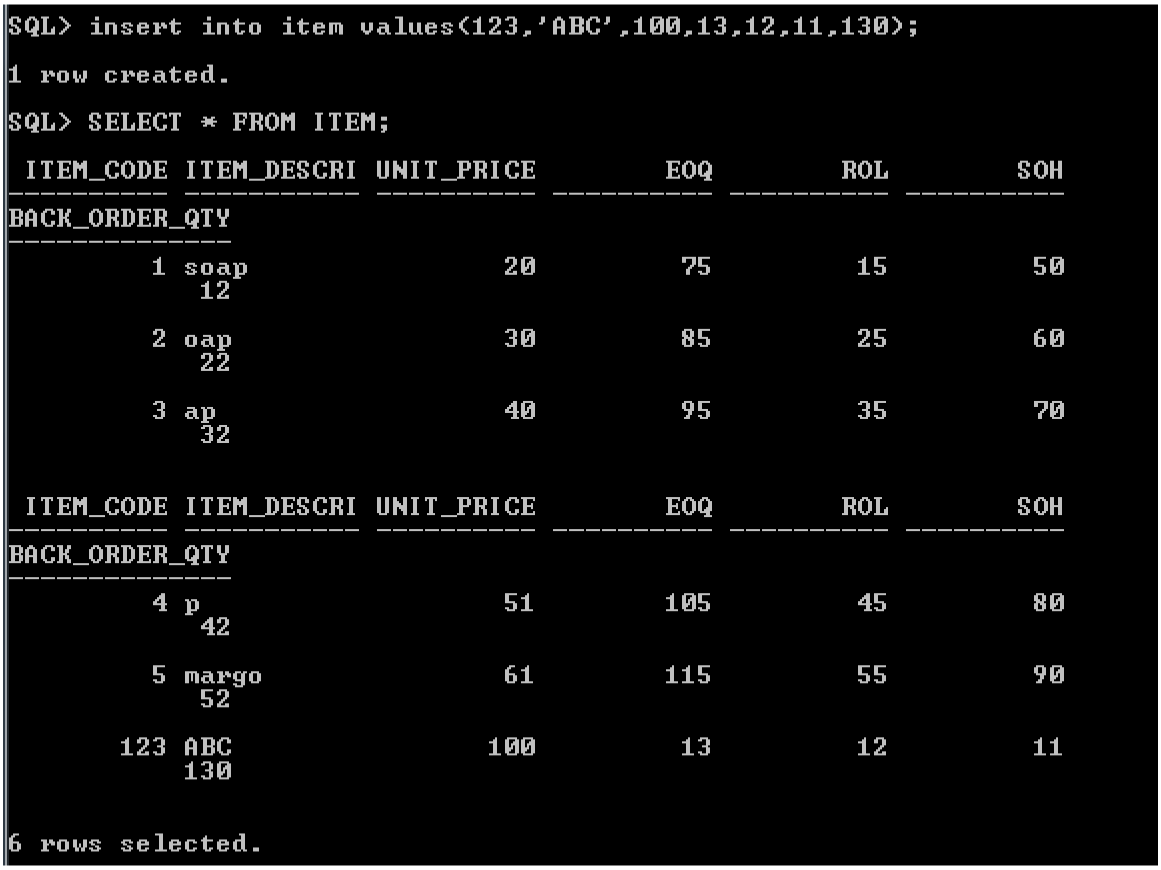
1. Write a trigger to notify back order quantity (ROL-SOH) with suitable message Whenever SOH crosses ROL.



```
SQL> CREATE or replace Trigger T
  2  before insert on ITEM for each row
  3  Declare
  4  Begin
  5  IF (:NEW.SOH > :NEW.ROL) THEN
  6  RAISE_APPLICATION_ERROR(-20010, 'SOH VALUE SHOULD BE LESS THAN ROL');
  7  END IF;
  8  END;
  9  /

Trigger created.
```

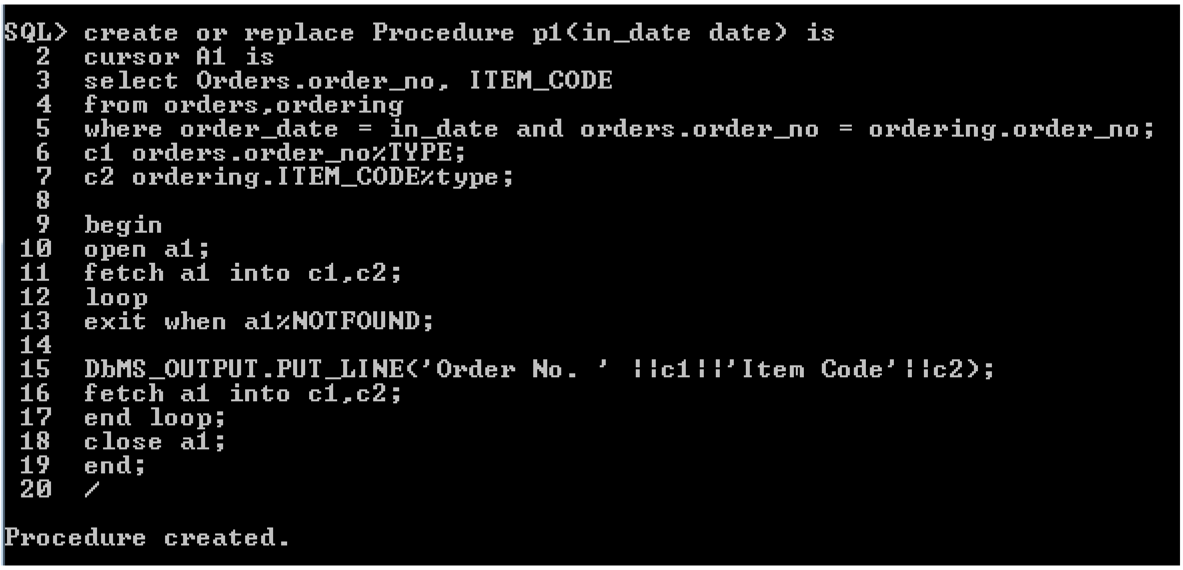


```
SQL> insert into item values(12,'ABCD',1000,10,11,12,13);
insert into item values(12,'ABCD',1000,10,11,12,13)
            *
ERROR at line 1:
ORA-20010: SOH VALUE SHOULD BE LESS THAN ROL
ORA-06512: at "MRINALRAJ.T", line 4
ORA-04088: error during execution of trigger 'MRINALRAJ.T'


SQL> SELECT * FROM ITEM;

 ITEM_CODE ITEM_DESCRI UNIT_PRICE        EOQ        ROL        SOH
---------- ----------- ---------- ---------- ---------- ----------
BACK_ORDER_QTY
--------------
         1 soap                20         75         15         50
            12

         2 oap                 30         85         25         60
            22

         3 ap                  40         95         35         70
            32


 ITEM_CODE ITEM_DESCRI UNIT_PRICE        EOQ        ROL        SOH
---------- ----------- ---------- ---------- ---------- ----------
BACK_ORDER_QTY
--------------
         4 p                   51        105         45         80
            42

         5 margo               61        115         55         90
            52

       123 ABC                100         13         12         11
           130


6 rows selected.
```

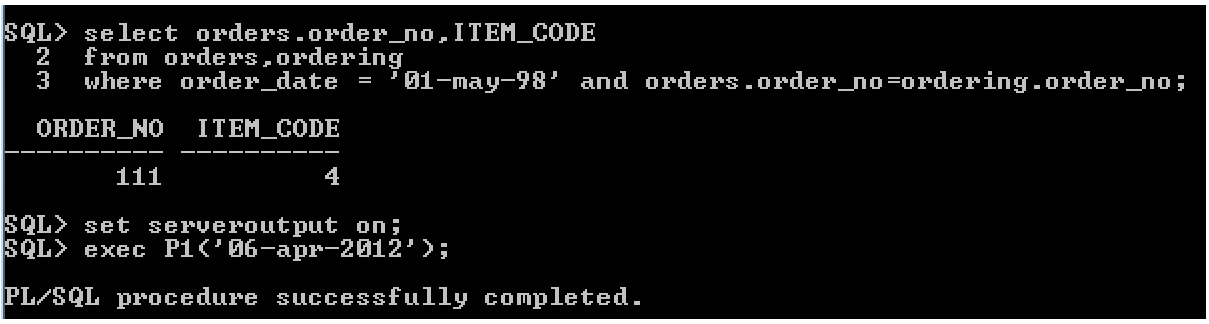


```
SQL> insert into item values(123,’ABC’,100,13,12,11,130);

1 row created.

SQL> SELECT * FROM ITEM;

 ITEM_CODE ITEM_DESCRI UNIT_PRICE        EOQ        ROL        SOH
---------- ----------- ---------- ---------- ---------- ----------
BACK_ORDER_QTY
--------------
         1 soap                20         75         15         50
            12

         2 oap                 30         85         25         60
            22

         3 ap                  40         95         35         70
            32


 ITEM_CODE ITEM_DESCRI UNIT_PRICE        EOQ        ROL        SOH
---------- ----------- ---------- ---------- ---------- ----------
BACK_ORDER_QTY
--------------
         4 p                   51        105         45         80
            42

         5 margo               61        115         55         90
            52

       123 ABC                100         13         12         11
           130


6 rows selected.
```

2. Write a Stored procedure to display the details of ITEM which are ordered on specific Order-Date.

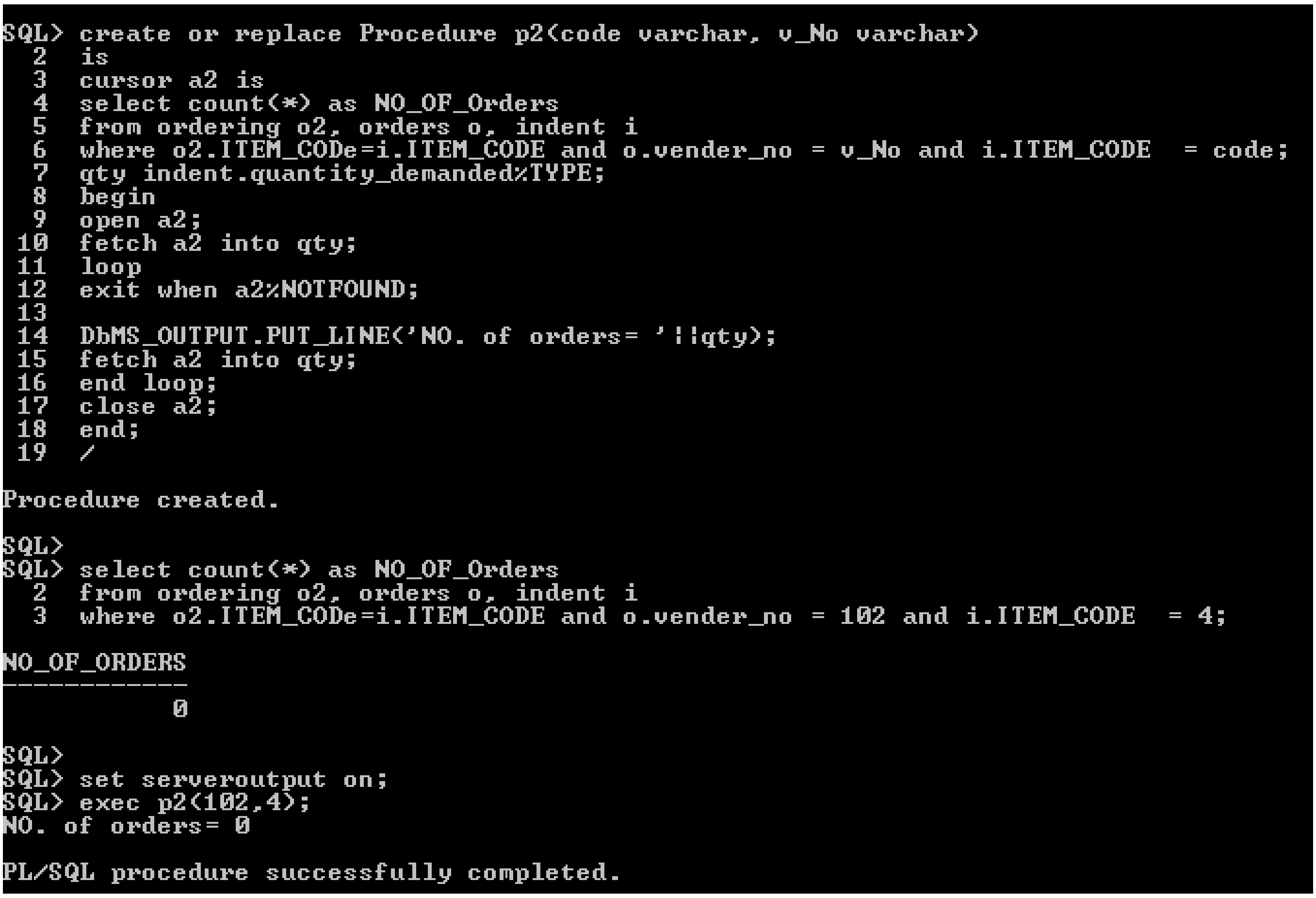


```
SQL> create or replace Procedure p1(in_date date) is
     cursor A1 is
     select Orders.order_no, ITEM_CODE
     from orders,ordering
     where order_date = in_date and orders.order_no = ordering.order_no;
     c1 orders.order_no%TYPE;
     c2 ordering.ITEM_CODE%type;

     begin
     open a1;
     fetch a1 into c1,c2;
     loop
     exit when a1%NOTFOUND;

     DbMS_OUTPUT.PUT_LINE(’Order No. ’ ||c1||’Item Code’||c2);
     fetch a1 into c1,c2;
     end loop;
     close a1;
     end;
     /

Procedure created.
```

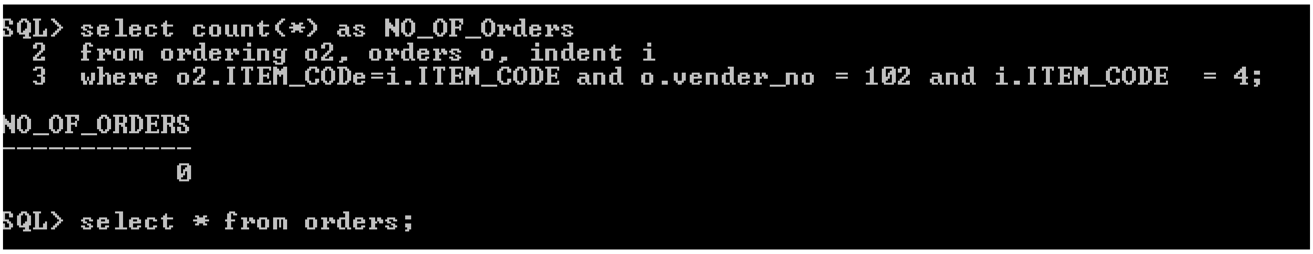


```
SQL> select orders.order_no,ITEM_CODE
     from orders,ordering
     where order_date = ’01-may-98’ and orders.order_no=ordering.order_no;

  ORDER_NO  ITEM_CODE
---------- ----------
       111          4

SQL> set serveroutput on;
SQL> exec P1(’06-apr-2012’);

PL/SQL procedure successfully completed.
```

3. Write a Stored procedure which accepts Item-Code and vendor-no as parameter and displays the number of orders on the Item ordered by the vendor.



```
SQL> create or replace Procedure p2(code varchar, v_No varchar)
     is
     cursor a2 is
     select count(*) as NO_OF_Orders
     from ordering o2, orders o, indent i
     where o2.ITEM_CODe=i.ITEM_CODE and o.vender_no = v_No and i.ITEM_CODE  = code;
     qty indent.quantity_demanded%TYPE;
     begin
     open a2;
     fetch a2 into qty;
     loop
     exit when a2%NOTFOUND;

     DbMS_OUTPUT.PUT_LINE('NO. of orders= '||qty);
     fetch a2 into qty;
     end loop;
     close a2;
     end;
     /

Procedure created.

SQL>
SQL> select count(*) as NO_OF_Orders
     from ordering o2, orders o, indent i
     where o2.ITEM_CODe=i.ITEM_CODE and o.vender_no = 102 and i.ITEM_CODE  = 4;

NO_OF_ORDERS
------------
           0

SQL>
SQL> set serveroutput on;
SQL> exec p2(102,4);
NO. of orders= 0

PL/SQL procedure successfully completed.
```



```
SQL> select count(*) as NO_OF_Orders
     from ordering o2, orders o, indent i
     where o2.ITEM_CODe=i.ITEM_CODE and o.vender_no = 102 and i.ITEM_CODE  = 4;

NO_OF_ORDERS
------------
           0

SQL> select * from orders;
```