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

case 1: SOH < 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
  7  THAN ROL');
  8  END IF;
  9  END;
 10  /

Trigger created.

SQL> INSERT INTO ITEM VALUES(123,'ABC',100,13,12,11,130);

1 row created.

SQL> SELECT * FROM ITEM;

     ICODE IDESC                UNIT_PRICE        EQQ        ROL        SOH        BOQ
---------- -------------------- ---------- ---------- ---------- ---------- ----------
         1 book                       27.5         75         15         50         12
         2 calculator                  715         80         50         12         21
         3 record                       66         23         72         40         31
         4 pen                         5.5         44         32         93         80
         5 bag                        1100         99         32         78         32
         6 notebook                    100         10          5          8          7
       123 ABC                         100         13         12         11        130

7 rows selected.
```

case 2: SOH > 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
  7  THAN ROL');
  8  END IF;
  9  END;
 10  /

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 "AYUSH0412.T", line 4
ORA-04088: error during execution of trigger 'AYUSH0412.T'
```

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
  2  CURSOR A1 IS
  3  SELECT ORDERS.ONUM,ITEMCODE
  4  FROM ORDERS,ORDERING
  5  WHERE ODATE=IN_DATE AND
  6  ORDERS.ONUM=ORDERING.ORDERNUM;
  7  C1 ORDERS.ONUM%TYPE;
  8  C2 ORDERING.ITEMCODE%TYPE;
  9  BEGIN
 10  OPEN A1;
 11  FETCH A1 INTO C1,C2;
 12  LOOP
 13  EXIT WHEN A1%NOTFOUND;
 14  DBMS_OUTPUT.PUT_LINE('ORDER NO.'||C1||'ITEM_NO'||C2);
 15  FETCH A1 INTO C1,C2;
 16  END LOOP;
 17  CLOSE A1;
 18  END;
 19  /

Procedure created.

SQL> SELECT ORDERS.ONUM,ITEMCODE
  2  FROM ORDERS,ORDERING
  3  WHERE ODATE='06-APR-2012' AND
  4  ORDERS.ONUM=ORDERING.ORDERNUM;

      ONUM   ITEMCODE
---------- ----------
        67          5

SQL> SET SERVEROUTPUT ON;
SQL> EXEC P1('06-APR-2012');
ORDER NO.67ITEM_NO5

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
  2  CURSOR A2 IS
  3  SELECT COUNT(*) AS NO_OF_ORDERS
  4  FROM ORDERING O2,ORDERS O,INDENT I
  5  WHERE O2.ORDERNUM=O.ONUM AND
  6  O2.ITEMCODE=I.ITEMCODE AND O.VENDER_NUM=V_NO AND I.ITEMCODE=CODE;
  7  QTY INDENT.QTDMD%TYPE;
  8  BEGIN
  9  OPEN A2;
 10  FETCH A2 INTO QTY;
 11  LOOP
 12  EXIT WHEN A2%NOTFOUND;
 13  DBMS_OUTPUT.PUT_LINE('NO. OF ORDERS='||QTY);
 14  FETCH A2 INTO QTY;
 15  END LOOP;
 16  CLOSE A2;
 17  END;
 18  /

Procedure created.

SQL>
SQL> SELECT COUNT(*) AS NO_OF_ORDERS
  2  FROM ORDERING O2,ORDERS O,INDENT I
  3  WHERE O2.ORDERNUM=O.ONUM AND
  4  O2.ITEMCODE=I.ITEMCODE AND O.VENDER_NUM=21 AND
  5  I.ITEMCODE=1;

NO_OF_ORDERS
------------
           1

SQL> SET SERVEROUTPUT ON;
SQL> EXEC P2(1,21);
NO. OF ORDERS=1

PL/SQL procedure successfully completed.
```