

PL/SQL

Create a PL/SQL block which displays the list of customers (all the informations)

```
BEGIN

    SELECT * FROM CUSTOMERS

END ;

/
```

Create a Procedure PS_Customer_Prodcuts which displays the list of product names of a given customer (customer_id). If no result returned (No_Data_Found exception raised), display the following message "No products returned or customer not found"

```
CREATE OR REPLACE PROCEDURE PS_Customer_Product(v_customer_id Orders.customer_id
%type) IS

CURSOR cur IS SELECT customer_name FROM CUSTOMER WHERE custoer_id =
v_customer_id

BEGIN

FOR rec IN cur LOOP

    Dbms.output.put_line(rec) ;

END LOOP ;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        Dbms_output.put_line('No customer found !') ;

END ;

/
```

Create a Function FN_Customer_Orders which returns the number of orders of a given customer (customer_id).

```
CREATE OR REPLACE FUNCTION FN_Customer_Orders (v_cust_id CUSTOMER.customer_id
%type) RETURN number IS

nb_orders number ;

BEGIN

SELECT COUNT(*) INTO nb_orders FROM ORDERS WHERE customer_id = v_cust_id ;
```

```
RETURN nb_orders ;  
  
END ;  
  
/
```

Create a trigger TRIG_INS_ORDERS which starts before each INSERT on Orders tables and test if the OrderDate >= SYSDATE. If not the following message is displayed "Order Date must be greater than or equal to today's date"

```
CREATE TRIGGER TRIG_INS_ORDERS  
BEFORE INSERT ON ORDERS  
BEGIN  
    INSERT INTO log(table_name, v_date, action)  
    VALUES ('ORDERS' , SYSDATE ,  
    IF OrderDate>= SYSDATE THEN INSERT ON ORDERS  
    ELSE RAISE excp) ;  
EXCEPTION  
    WHEN excp THEN  
        dbms_output.put_line("Order Date must be greater than or equal to today's  
        date") ;  
  
END ;  
  
/
```