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

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

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

declare

cursor customers is select \* from customer ;

begin

for enreg in customers loop

dbms\_output.put\_line(enreg .costumer\_name || ' ' || enreg .costumer\_phone) ;

end loop ;

end ;

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

1. 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\_Prodcuts (v\_pro\_id Orders.costumer\_id%type) is

cursor name IS SELECT product\_name FROM product WHERE product\_id in (SELECT product\_id from Ordres WHERE costumer\_id=v\_pro\_id);

BEGIN

FOR enreg IN cur LOOP

dbms\_output.put\_line(enreg.product\_name);

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('No products returned or customer not found')

END;

call PS\_Customer\_Prodcuts()

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

1. Create a Function FN\_Customer\_Orders which returns the number of orders of a given customer (customer\_id).

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

|  |
| --- |
| CREATE OR REPLACE FUNCTION FN\_CUSTO\_ORDERS (v\_cust\_id Orders.costumer\_id%type) RETURN number is |
| order\_nb number := 0 ; |
| BEGIN |
| SELECT COUNT(\*) INTO order\_nb FROM Orders WHERE costumer\_id = v\_cust\_id; |
| RETURN order\_nb ; |
| END ; |
|  |
| ---------------------------------------------------------------------------------------------------------   1. 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 |
| FOR EACH ROW |
| begin |
| if :NEW.OrderDate >= SYSDATE then |
| dbms\_output.put\_line('Order Date must be greater than or equal to today's date'); |
| end if ; |
| end ; |