Question –2

```
create or replace TRIGGER TRG_SALESORDERDETAIL
   BEFORE INSERT OR UPDATE
   ON SALESORDERDETAIL
   FOR EACH ROW
BEGIN
select :new.soquantity*:new.price into :new.subtotal from dual;
END;


create or replace PROCEDURE salesordertotal AS
     CURSOR C1 IS SELECT * FROM salesorders WHERE SOTOTAL IS NULL
FOR UPDATE OF SOTOTAL;

    v_customerscore customers.customerscore%type;
    v_discount salesdiscount.salesdiscount%type;
    v_sono salesorders.sono%type;
    v_sono_temp salesorders.sono%type;
    v_ordertotal salesorders.sototal%type;
  BEGIN

        FOR temp_rec IN C1 LOOP
   v_sono_temp:=temp_rec.sono;
   select distinct sono into v_sono from salesorderdetail where
sono=v_sono_temp;
   select customerscore into v_customerscore from customers where
customerid=temp_rec.customerid;
   select salesdiscount into v_discount from salesdiscount where
v_customerscore >= mincustomerscore and
v_customerscore<=maxcustomerscore;
   select sum(subtotal)*(1-v_discount) into v_ordertotal from
SALESORDERDETAIL where sono= v_sono;
   update salesorders set sototal=v_ordertotal where current of
c1 ;
   end loop;

   exception when no_data_found then
   dbms_output.put_line('Error with Sales Order '||v_sono_temp||',
no matching subtotals');
END salesordertotal;
/
```