

Enoncé :

Soit le schéma de la base de données suivant:

Table **client** : CLI (NumCli, NomCli, AdrClt, TelClt, Age)

Table **commande** : COM (NumCom, #NumCli, FraisPort, DateCom);

Table **détail commande** : DET (#NumCom, #NumPro, Qte, Remise)

Table **produit** : PRO (NumPro, #NumFou, NomPro, TypePro, PrixUnit)

Table **fournisseur** : FOU (NumFou, NomFou, AdrFour, TelFour)

Questions :

1) Parmi les déclarations de variables suivantes, déterminer celles qui sont correctes et justifier votre réponse.

a) DECLARE
v_id NUMBER(4) ;

Correcte.

b) DECLARE
v_x,v_y,v_z VARCHAR2(10) ;

Incorrecte, parce que on ne peut déclarer plusieurs variables à la fois.

c) DECLARE
v_date_naissance DATE NOT NULL ;

Incorrecte, parce que on doit attribuer une valeur.

d) DECLARE
v_en_stock BOOLEAN :=1 ;

Incorrecte, il devrait être BOOLEAN: = true OR BOOLEAN: = false;

e) DECLARE
emp_record emp_record_type ;

Incorrecte, parce que dans la création d'une table PL / SQL il faut d'abord, déclarez le type de emp_record_type, puis déclarez un identificateur en utilisant ce type.

2) Ecrire un bloc PL/SQL permettant d'afficher le prix du produit numéro 1212.

```
declare
prix_prod integer;
begin
select PrixUnit into prix_prod from pro
where NumPro=1212;
```

```
dbms_output.put_line (prix_prod);
```

```
end ;
```

```
/
```

- 3) Modifier le bloc PL/SQL précédent pour qu'il puisse afficher un message suivant "Le prix du produit 1212 est".

```
declare
```

```
prix_prod integer;
```

```
begin
```

```
select PrixUnit into prix_prod from pro
```

```
where NumPro=1212;
```

```
dbms_output.put_line ('Le prix du produit 1212 est ' || prix_prod);
```

```
end ;
```

```
/
```

- 4) Écrire un bloc PL/SQL qui, étant donné le numéro d'un client, donne le nombre de produits commandés par ce client.

```
accept num prompt 'saisir le num cli'
```

```
declare
```

```
nbre_prod integer ;
```

```
begin
```

```
select count(NumPro) into nbre_prod from com,det
```

```
where com.NumCom=det.NumCom
```

```
and NumCli=&num;
```

```
dbms_output.put_line(nbre_prod);
```

```
End;
```

```
/
```

- 5) Ecrire un bloc PL/SQL qui affiche toutes les informations relatives à un client. Utiliser dans une première version des variables scalaires et dans une deuxième version une variable composite.

a)

```
Accept num prompt 'saisir un numéro de client :'
```

```
Declare
```

```
V_numCli cli.NumCli%TYPE ;
```

```
V_nomCli cli.NomCli%TYPE ;
```

```
V_AdrCli cli.Adr%TYPE ;
```

```
V_AgeCli cli.Age%TYPE ;
```

```
V_TelCli cli.Tel%TYPE ;
```

```
Begin
```

```
Select NumCli,NomCli,Adr,Tel,Age into V_numCli ,V_nomCli,V_AdrCli,V_AgeCli ,V_TelCli
```

From CLI

Where NumCli=#

Dbms_output.put_line(V_NumCli || ',' || V_nomCli || ',' || V_AdrCli || ',' || V_AgeCli || ',' || V_TelCli);

End;

/

b)

accept num prompt 'donner un numéro de client'

Declare

Client CLI%ROWTYPE;

Begin

Select NumCli,NomCli,Adr,Age,tel into Client From CLI

Where NumCli=#

Dbms_output.put_line(client.NumCli || ',' || client.NomCli || ',' || client.Adr || ',' || client.Age || ',' || client.tel);

End;

/

6) Ecrire un bloc PL/SQL qui permet d'ajouter un nouveau produit dans la table PRODUIT.

```

ACCEPT V_NumPro PROMPT 'Entrer le numéro du produit :'
ACCEPT V_NumFou PROMPT 'Entrer le numéro du fournisseur :'
ACCEPT V_NomPro PROMPT 'Entrer le nom du produit :'
ACCEPT V_TypePro PROMPT 'Entrer le type du produit :'
ACCEPT V_PrixUnit PROMPT 'Entrer le prix unitaire du produit :'

BEGIN
INSERT INTO PRO VALUES (&V_NumPro, &V_NumFou, &V_NomPro, &V_TypePro,
&V_PrixUnit);
END ;
/

```

- 7) Ecrire un bloc PL/SQL qui affiche "produit cher" si le prix du produit est supérieur à 50D ou "produit pas cher" si le prix est inférieur à 50D. Le numéro du produit est saisi au clavier.

Accept num prompt 'donner un numéro de produit'

```

Declare
v_prix pro.PrixUnit%TYPE ;

begin
select PrixUnit into v_prix
from pro
where NumPro=&num;

if v_prix>50 then dbms_output.put_line('prix cher');
else dbms_output.put_line('prix pas cher');
end if;
end;
/

```

- 8) Créer un bloc PL/SQL permettant de mettre à jour la remise d'un produit en fonction de sa quantité.
- Si la quantité est inférieure à 50, passer la remise à 15%.
 - Si la quantité est comprise entre 50 et 100, mettre la remise à 10%.
 - Si la quantité est supérieure à 100, mettre la remise à 5%.

```

ACCEPT NumP PROMPT 'Entrer un numéro de produit :'
DECLARE
cursor C1 is select Qte from det where NumPro= &NumP;
v_Qte DET.Qte%TYPE;
v_remise DET.Remise%TYPE;

```

```

BEGIN
Open C1;
loop
Fetch C1 into v_Qte;
Exit when C1%notfound;

IF v_Qte < 50 THEN v_remise:= 15;
ELSIF v_Qte <=100 THEN v_remise:= 10;
ELSIF v_Qte> 100 THEN v_remise := 5;
END IF;
UPDATE DET
SET remise = v_remise
Where NumPro= &NumP;
End loop;
Close C1;

END;

```

9) Créer un bloc PL/SQL qui affiche le nom de tous les clients qui habitent à Tunis.

```

Declare
cursor C1 is select nomCli from CLI where ADR='tunis';
v_nom CLI.NomCli%type;
begin
open C1;
loop
fetch C1 into v_nom;
exit when C1%notfound;
dbms_output.put_line('Noms des clients :'||v_nom);
end loop ;
close C1 ;
end;

```

10) Créer un bloc PL/SQL qui permet d'afficher le nom et le prix de n produits les plus chers de la table PRODUIT.

```

ACCEPT p_nbre Prompt ' Saisir le nombre de produit ayant les prix les plus élevés:'
DECLARE
CURSOR produit_cursor IS      /*declarer un curseur*/
SELECT NomPro, PrixUnit
FROM PRO
WHERE PrixUnit IS NOT NULL
ORDER BY PrixUnit DESC;      /*trier les prix par ordre décroissant*/
V_nom PRO.NomPro%TYPE ;
V_prix PRO.PrixUnit%TYPE ;

```

```
BEGIN
  OPEN produit_cursor ;
  For i IN 1..&p_nbre
  LOOP
    FETCH produit_cursor INTO V_nom,V_prix;
    Dbms_output.put_line(V_nom||','||V_prix);
  END LOOP;
  CLOSE produit_cursor;
END;
/
```