

## TP2 PL/SQL Solutions:

```
-- Sequences
CREATE SEQUENCE tablearticle START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE tableclient START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE tabledetail START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE tablecommande START WITH 1 INCREMENT BY 1;

-- Tables
CREATE TABLE client (
    codecl INT DEFAULT tableclient.NEXTVAL PRIMARY KEY,
    nom VARCHAR2(50),
    ville VARCHAR2(50)
);

CREATE TABLE ARTICLE (
    codeArt INT DEFAULT tablearticle.NEXTVAL PRIMARY KEY,
    Desi VARCHAR2(50),
    PU NUMBER,
    Qdisp INT
);

CREATE TABLE COMMANDE (
    NumCom INT DEFAULT tablecommande.NEXTVAL PRIMARY KEY,
    DateCom DATE,
    codecl INT NOT NULL,
    FOREIGN KEY (codecl) REFERENCES CLIENT(codecl)
);

CREATE TABLE Detail (
    NumComligne INT DEFAULT tabledetail.NEXTVAL PRIMARY KEY,
    NumCom INT NOT NULL,
    codeArt INT NOT NULL,
    QTE INT,
    FOREIGN KEY (codeart) REFERENCES ARTICLE(codeart),
    FOREIGN KEY (numCom) REFERENCES COMMANDE(numCom)
);

-- Insertion CLIENTS
INSERT INTO client (nom, ville) VALUES ('John Doe', 'Kenitra');
INSERT INTO client (nom, ville) VALUES ('Jane Smith', 'Kenitra');
INSERT INTO client (nom, ville) VALUES ('Alice Johnson', 'Casa');
INSERT INTO client (nom, ville) VALUES ('Bob Williams', 'Marrakesh');
INSERT INTO client (nom, ville) VALUES ('Eva Davis', 'Rabat');

-- Insertion ARTICLE
INSERT INTO ARTICLE (Desi, PU, Qdisp) VALUES ('Product1', 25.5, 100);
INSERT INTO ARTICLE (Desi, PU, Qdisp) VALUES ('Product2', 15.75, 50);
INSERT INTO ARTICLE (Desi, PU, Qdisp) VALUES ('Product3', 30.0, 75);
INSERT INTO ARTICLE (Desi, PU, Qdisp) VALUES ('Product4', 10.0, 200);
INSERT INTO ARTICLE (Desi, PU, Qdisp) VALUES ('Product5', 22.3, 120);
```

## TP2 PL/SQL Solutions:

```
-- Insertion COMMANDE
INSERT INTO COMMANDE (DateCom, codecl) VALUES (TO_DATE('2023-11-18', 'YYYY-MM-DD'), 1);
INSERT INTO COMMANDE (DateCom, codecl) VALUES (TO_DATE('2023-11-19', 'YYYY-MM-DD'), 2);
INSERT INTO COMMANDE (DateCom, codecl) VALUES (TO_DATE('2023-11-20', 'YYYY-MM-DD'), 3);
INSERT INTO COMMANDE (DateCom, codecl) VALUES (TO_DATE('2023-11-21', 'YYYY-MM-DD'), 1);
INSERT INTO COMMANDE (DateCom, codecl) VALUES (TO_DATE('2023-11-22', 'YYYY-MM-DD'), 1);

-- Insertion DETAIL
INSERT INTO Detail (NumCom, codeArt, QTE) VALUES (1, 1, 12);
INSERT INTO Detail (NumCom, codeArt, QTE) VALUES (2, 2, 50);
INSERT INTO Detail (NumCom, codeArt, QTE) VALUES (3, 1, 20);
INSERT INTO Detail (NumCom, codeArt, QTE) VALUES (4, 4, 15);
INSERT INTO Detail (NumCom, codeArt, QTE) VALUES (5, 5, 12);

SELECT * FROM DETAIL;

--Q1 :

SET SERVEROUTPUT ON
DECLARE
numClient INTEGER := &num;
nomClient Varchar2(50) := '';
nbrCommande Number := 0;
BEGIN
SELECT nom, count(*) INTO nomClient , nbrCommande FROM CLIENT C1 , COMMANDE C
WHERE C1.codecl = C.codecl AND C1.codecl = numClient
GROUP BY nom;

dbms_output.put_line('Le client : ' || nomClient || ' a commande ' ||
nbrCommande || ' fois.');
```

```
END;

--Q2
--NBRES de Commandes par villes
DECLARE
nbrCommandes NUMBER;
Ville VARCHAR2(50);
CURSOR nbrParVille is
SELECT ville , count(*) FROM Client c1 , Commande C WHERE C1.codecl =
C.codecl
GROUP BY ville;
BEGIN
```

## TP2 PL/SQL Solutions:

```
OPEN nbrParVille;
LOOP
FETCH nbrParVille INTO Ville , nbrCommandes;
EXIT WHEN nbrParVille%notfound;
dbms_output.put_line('La ville : ' || Ville || ' a ' || nbrCommandes || '
commandes.');
```

-- Q3

```
DECLARE
dateD DATE := TO_DATE('&date','YYYY-MM-DD');
Ville Varchar2(50) := '&ville';
nbrCommande Number := 0;
BEGIN
SELECT count(*) INTO nbrCommande FROM CLIENT Cl , COMMANDE C WHERE Cl.codecl =
C.codecl AND Cl.Ville = Ville AND C.DATECOM = dateD;

dbms_output.put_line('Le nombre de commande passe a ' || Ville || ' Le ' ||
dateD || ' est: ' || nbrCommande);

END;
```

-- Q4

```
DECLARE
nbrDeClient NUMBER;
nbrDeCommande NUMBER;
BEGIN

SELECT count(Cl.codecl) INTO nbrDeClient FROM CLIENT Cl WHERE Cl.codeCl NOT
IN (SELECT codecl FROM COMMANDE) ;
dbms_output.put_line('Le nombre de commande ayant passe 0 commandes:
' || nbrDeClient );
END;
```

-- Q5

```
DECLARE
nbrDArticle NUMBER;
CommandeID INTEGER :=&CommandeID;
BEGIN

SELECT count(DISTINCT codeArt) INTO nbrDArticle FROM COMMANDE C , Detail D
WHERE C.NumCom = D.NumCom AND C.NumCom = CommandeID;
dbms_output.put_line('Le nombre d article dans la commande ' || CommandeID || '
est: ' || nbrDArticle );
```

## TP2 PL/SQL Solutions:

```
END;

--Q6

DECLARE
QTE NUMBER;
ArticleID INTEGER :=&article;
BEGIN

SELECT SUM(QTE) INTO QTE FROM ARTICLE A , Detail D WHERE A.CodeArt = D.CodeArt
AND A.codeArt = ArticleID;
dbms_output.put_line('La QTE TOTAL de 1 article ' || ArticleID || ' est:
' || QTE );

END;

--Q7

DECLARE
QTE NUMBER;
ArticleID INTEGER :=&ArticleID;
CommandeID INTEGER :=&CommandeID;

BEGIN

SELECT SUM(QTE) INTO QTE FROM ARTICLE A , Detail D, Commande C WHERE C.NumCom
= D.NumCom AND A.CodeArt = D.CodeArt AND A.codeArt = ArticleID AND C.NumCom =
CommandeID;
dbms_output.put_line('La QTE TOTAL de 1 article ' || ArticleID || ' dans la
commande ' || CommandeID || ' est: ' || QTE );

END;

--Q8
DECLARE
ArticleID INTEGER :=&ArticleID;
QDisp INTEGER;
CURSOR DataReturned IS
    SELECT QDisp FROM ARTICLE A WHERE A.codeArt = ArticleID ;
BEGIN

OPEN DataReturned;
FETCH DataReturned INTO QDisp;

IF DataReturned%FOUND THEN
    dbms_output.put_line('Yes!');
ELSE
    dbms_output.put_line('NO. ');
END IF;
```

## TP2 PL/SQL Solutions:

```
CLOSE DataReturned;
END;

--Q9
DECLARE
CommandeID INTEGER := &CommandeID;
Total NUMBER;

BEGIN

SELECT SUM(Art.PU * D.Qte) INTO Total FROM DETAIL D , ARTICLE ART
    WHERE ART.CodeArt = D.CodeArt AND D.NumCom = CommandeID;
dbms_output.put_line('Le cout TOTAL de la commande ' || CommandeID || ' est:
' || Total );

END;

--Q10
DECLARE
NumDeCommandes NUMBER;
CURSOR commandInf IS
    SELECT DISTINCT D.NumCom FROM DETAIL D
    WHERE D.NumCom NOT IN (SELECT D.NumCom FROM DETAIL D , ARTICLE ART WHERE
ART.CodeArt = D.CodeArt
    GROUP BY D.NumCom HAVING SUM(Art.PU * D.Qte) >= 1000);
BEGIN
OPEN commandInf;
LOOP
FETCH commandInf INTO NumDeCommandes;
EXIT WHEN commandInf%NOTFOUND;
dbms_output.put_line('La commande ' || NumDeCommandes || ' coute moins de
1000');
END LOOP;
CLOSE commandInf;
END;

--Q11
DECLARE
ClientID INTEGER := &IdClient;
total NUMBER := 0;
Actuel NUMBER;

BEGIN
FOR C IN (SELECT C.NumCom FROM Commande C, Client Cl WHERE cl.codecl =
C.codecl AND C.codecl = ClientID) LOOP
SELECT SUM(Art.PU * D.Qte) INTO Actuel FROM DETAIL D , ARTICLE ART
    WHERE ART.CodeArt = D.CodeArt AND D.NumCom = C.NumCom ;
```

## TP2 PL/SQL Solutions:

```
total := total + Actuel;

END LOOP;
dbms_output.put_line('Le chiffre d affaire de ' || ClientID || ' est: ' ||
total);
END;

--Q12
DECLARE
    numCommande NUMBER := &numCommande;
    codeArticle NUMBER;
    montantPuMax NUMBER := 0;
    codeArticleMax NUMBER;

    CURSOR CommandeCursor IS
        SELECT d.codeart, a.pu
        FROM article a, detail d
        WHERE a.codeart = d.codeart
        AND d.numcom = numCommande;

BEGIN
    FOR record IN CommandeCursor LOOP
        IF montantPuMax <= record.pu OR montantPuMax = 0 THEN
            montantPuMax := record.pu;
            codeArticleMax := record.codeart;
        END IF;
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Montant Pu Max: ' || montantPuMax);
    DBMS_OUTPUT.PUT_LINE('Code Article Max: ' || codeArticleMax);

END;
```