

# TP 1

Up

## TP 1: DEVELOPPEMENT AVEC PL/SQL (1)

### GESTION D'UN PARC DE VEHICULES (PARCVEH)

Auteur :



SUJET



QUESTION



CORRECTION

### **SUJET:** GESTION D'UN PARC DE VEHICULES (PARCVEH)

#### **VOITURE (NOV,MV,KM,PSG)**

prédicat : à une voiture on associe son numéro de voiture NOV qui la distingue des autres voitures, sa marque MV, le nombre de kilomètres qu'elle a parcourus KM, le nombre de places disponibles de passagers PSG.

#### **CH (NCH,CHAUFFEUR)**

prédicat : à un numéro de chauffeur NCH on associe un seul nom du chauffeur CHAUFFEUR.

#### **V-CH (NOV,NCH,NKM)**

prédicat : le chauffeur de tel numéro NCH a conduit la voiture de tel numéro NOV pendant tant de kilomètres NKM depuis que la voiture est en service.

#### **REPARATION (NOREP,NOV,NOG,TYPREP,PX,KMCPT)**

prédicat : la voiture de tel numéro NOV est menée au garage de tel numéro NOG pour une réparation de numéro NOREP et de type TYPERP; elle a alors tant de kilomètres au compteur KMCPT. Cette réparation a coûté tant PX.

### **TRAJET (NOTRAJ,VILLEDEP,VILLEARR,DATETRAJET,NBKM)**

prédicat : un trajet de tel numéro NOTRAJ a été effectué à telle date DATE-TRAJET; les villes de départ et d'arrivée sont respectivement VILLEDEP, VILLEARR; le trajet est de tant de kilomètres NBKM.

### **TR\_NOV (NOTRAJ,NOV,NCH,NBPERSTR)**

prédicat : la voiture de numéro NOV, conduite par le chauffeur de numéro NCH, a transporté tant de personnes (NBPERSTR) pour le trajet de numéro NOTRAJ.

*Un chauffeur peut conduire plusieurs fois la même voiture pour des trajets différents, et il y a un seul chauffeur qui conduit une voiture au cours d'un trajet.*

## **QUESTION:**

1. Ecrivez un bloc PL/SQL pour :



a. Entrez un nom d'une ville,

b. Affichez le nombre moyen, le nombre minimum, le nombre maximum de kilomètres des trajets qui sont partis de cette ville.

2. Ecrivez un bloc PL/SQL pour :



a. Entrez un ans,

b. Affichez le numéro et la marque de la voiture qui a effectué le plus de trajets dans cette année. Affichez aussi son nombre de trajets effectués.

3. Ecrivez un bloc PL/SQL pour afficher une phrase à annoncer aujourd'hui. Par



exemple : 'Aujourd'hui est mardi, 16 Avril 2002' pour le date 16.04.2002

Conseil : Utiliser SYSDATE avec les formats : W : jour de semaine ; DD : jour, MM : mois et YYYY : année



4. Ecrivez un bloc PL/SQL pour :
- Entrez un chiffre,
  - Affichez la table de multiplication de ce chiffre.



5. Ecrivez un bloc PL/SQL pour :
- Entrez un numéro de trajet,
  - Si ce numéro est déjà existé dans la table TRAJET, affichez toutes ses informations. Si non, affichez une phrase comme suit 'Ce numéro n'est pas existé'.



6. Ecrivez un bloc PL/SQL pour :
- Entrez des informations d'une nouvelle réparation (c.à.d NOREP, NOV, NOG, TYPREP, PX, KMCPT),
  - Validez que NOV est déjà existé dans la table VOITURE ou pas.
    - Si oui, créer ce nouveau tuple correspondant de la table REPARATION,
    - Si non, affichez un message d'erreur.

## CORRECTION

### Préparation

- Créer les tables : [CREATE\\_TABLES.SQL](#)



- Insérer les données : [INSERT\\_DATA.SQL](#)



### Proposition

1. Ecrivez un bloc PL/SQL pour :

- Entrez un nom d'une ville,
- Affichez le nombre moyen, le nombre minimum, le nombre maximum de kilomètres des



*trajets qui sont partis de cette ville.*

```

DECLARE
  nom_ville CHAR(20) ;
  nombre_min INTEGER ;
  nombre_max INTEGER ;
  nombre_moyen INTEGER ;
BEGIN
  -- Entrez un nom d'une ville
  nom_ville := '&Entrez_un_nom_ville';
  -- Selectionnez
  SELECT AVG(nbkm), MIN(nbkm), MAX(nbkm)
  INTO nombre_moyen, nombre_min, nombre_max
  FROM trajet
  WHERE villedep = nom_ville ;
  -- Affichez
  DBMS_OUTPUT.PUT_LINE('Ville départ : ' || nom_ville) ;
  DBMS_OUTPUT.PUT_LINE(' - Nombre moyen : ' || TO_CHAR(nombre_moyen)) ;
  DBMS_OUTPUT.PUT_LINE(' - Nombre minimum : ' || TO_CHAR(nombre_min)) ;
  DBMS_OUTPUT.PUT_LINE(' - Nombre maximum : ' || TO_CHAR(nombre_max)) ;
END ;

```

2. *Ecrivez un bloc PL/SQL pour :*

a. *Entrez un ans,*

b. *Affichez le numéro et la marque de la voiture qui a effectué le plus de trajets dans cette*



*année. Affichez aussi son nombre de trajets effectués.*

```

DECLARE
  un_ans CHAR(4);
  le_plus_trajets INTEGER ;
  m_nombre_voitures INTEGER;
  m_nov CHAR(12) ;
  m_mv CHAR(20) ;
BEGIN
  -- Entrez un ans
  un_ans := TRIM('&Entrez_un_ans');
  -- 1. Calculez le plus trajets
  SELECT MAX(COUNT(tv.notraj)) INTO le_plus_trajets
  FROM tr_nov tv , trajet tr
  WHERE (tv.notraj = tr.notraj) AND
  (to_char(tr.datetrajet,'YYYY') = un_ans)

```

```

GROUP BY tv.nov ;
DBMS_OUTPUT.PUT_LINE('L'annee : '|| un_ans) ;
IF le_plus_trajets <> 0 THEN
DBMS_OUTPUT.PUT_LINE(' - Le plus trajets : '|| TO_CHAR(le_plus_trajets)) ;
-- 2. Calculez le nombre des voitures:
SELECT COUNT(v.nov) INTO m_nombre_voitures FROM voiture v
WHERE v.nov IN (
SELECT tv.nov
FROM tr_nov tv, trajet tr
WHERE (tv.notraj = tr.notraj)
AND (to_char(tr.datetrajet,'YYYY') = un_ans)
GROUP BY tv.nov
HAVING COUNT(tv.notraj) >= le_plus_trajets) ;
IF m_nombre_voitures = 1 THEN
-- 3. Selectionnez le numéro de la voiture:
SELECT tv.nov INTO m_nov
FROM tr_nov tv, trajet tr
WHERE (tv.notraj = tr.notraj)
AND (to_char(tr.datetrajet,'YYYY') = un_ans)
GROUP BY tv.nov
HAVING COUNT(tv.notraj) >= le_plus_trajets ;
-- 4. Selectionnez la marque de cette voiture:
SELECT mv INTO m_mv
FROM voiture WHERE nov = m_nov;
-- 5. Affichez
DBMS_OUTPUT.PUT_LINE(' - Voiture No: '|| m_nov) ;
DBMS_OUTPUT.PUT_LINE(' - Marque : '|| m_mv) ;
ELSE -- nombre de voitures <> 1
DBMS_OUTPUT.PUT_LINE(' Il y a '||to_char(m_nombre_voitures)||' voitures');
END IF ;
ELSE -- le_plus_trajets = 0
DBMS_OUTPUT.PUT_LINE(' Il n"y a pas de trajet') ;
END IF ;
END ;
/

```

3. *Ecrivez un bloc PL/SQL pour afficher une phrase à annoncer aujourd'hui. Par exemple : 'Aujourd'hui est mardi, 16 Avril 2002' pour le date 16.04.2002*

Conseil : Utiliser SYSDATE avec les formats : W : jour de semaine ; DD : jour, MM : mois

et YYYY : année



## DECLARE

```
jour_de_semaine NUMBER(2) ;  
jour_en_caractere CHAR(10) ;  
m_mois NUMBER(2) ;  
mois_en_caractere CHAR(12) ;
```

## BEGIN

```
SELECT to_char(sysdate,'D') INTO jour_de_semaine FROM dual ;  
IF jour_de_semaine = 1 THEN  
    jour_en_caractere := 'dimanche';  
ELSIF jour_de_semaine = 2 THEN  
    jour_en_caractere := 'lundi' ;  
ELSIF jour_de_semaine = 3 THEN  
    jour_en_caractere := 'mardi' ;  
ELSIF jour_de_semaine = 4 THEN  
    jour_en_caractere := 'mercredi' ;  
ELSIF jour_de_semaine = 5 THEN  
    jour_en_caractere := 'jeudi' ;  
ELSIF jour_de_semaine = 6 THEN  
    jour_en_caractere := 'vendredi' ;  
ELSE -- jour_de_semaine = 7  
    jour_en_caractere := 'samedi' ;  
END IF ;  
-- Calculez mois_en_caractere  
SELECT to_char(sysdate,'MM') INTO m_mois FROM dual ;  
IF m_mois = 1 THEN  
    mois_en_caractere := 'janvier';  
ELSIF m_mois = 2 THEN  
    mois_en_caractere := 'fevrier';  
ELSIF m_mois = 3 THEN  
    mois_en_caractere := 'mars';  
ELSIF m_mois = 4 THEN  
    mois_en_caractere := 'avril';  
ELSIF m_mois = 5 THEN  
    mois_en_caractere := 'mai';  
ELSIF m_mois = 6 THEN  
    mois_en_caractere := 'juin';  
ELSIF m_mois = 7 THEN  
    mois_en_caractere := 'juillet';  
ELSIF m_mois = 8 THEN  
    mois_en_caractere := 'aout';  
ELSIF m_mois = 9 THEN  
    mois_en_caractere := 'septembre';  
ELSIF m_mois = 10 THEN  
    mois_en_caractere := 'octobre';
```

```

ELSIF m_mois = 11 THEN
  mois_en_caractere := 'novembre';
ELSE -- m_mois = 12
  jour_en_caractere := 'decembre' ;
END IF ;
-- Affichez
DBMS_OUTPUT.PUT_LINE('Aujourd'hui est '||TRIM(jour_en_caractere)||',
  ||to_char(sysdate,'DD')||' '||TRIM(mois_en_caractere) ||' '
  ||to_char(sysdate,'YYYY')) ;
END ;
/

```

4. *Ecrivez un bloc PL/SQL pour :*

a. *Entrez un chiffre,*



b. *Affichez la table de multiplication de ce chiffre.*

```

DECLARE
  ce_chiffre INTEGER(1); -- Chiffre à entrer
  ii INTEGER(1) ;
BEGIN
  -- Entrer un chiffre
  ce_chiffre := &ENTER_UN_CHIFFRE ;
  DBMS_OUTPUT.PUT_LINE('Table de multiplication de '|| ce_chiffre) ;
  -- Boucle FOR:
  FOR ii IN 1..9 LOOP
    DBMS_OUTPUT.PUT_LINE(ce_chiffre||' x '||ii||
      ' = '||to_char(ce_chiffre*ii,'99')) ;
  END LOOP ;
END ;
/

```

5. *Ecrivez un bloc PL/SQL pour :*

a. *Entrez un numéro de trajet,*

b. *Si ce numéro est déjà existé dans la table TRAJET, affichez toutes ses informations. Si*



non, affichez une phrase comme suit 'Ce numéro n'est pas existé'.

```

DECLARE
  m_notraj INTEGER;
  m_nombre INTEGER ;
  m_villedep CHAR(20) ;
  m_villearr CHAR(20) ;
  m_datetrajet DATE ;
  m_nbkm NUMBER(3,0) ;
BEGIN
  -- Entrez un numero de trajet
  m_notraj := TRIM('&Entrez_un_numero_trajet');
  -- Validez ce numero de trajet
  SELECT COUNT(notraj) INTO m_nombre
  FROM trajet WHERE notraj = m_notraj ;
  -- Ce numero déjà existé
  IF m_nombre = 1 THEN
    -- Selectionnez les informations de ce numéro
    SELECT villedep, villearr, datetrajet, nbkm
    INTO m_villedep, m_villearr, m_datetrajet, m_nbkm
    FROM trajet WHERE notraj = m_notraj ;
    -- Affichez
    DBMS_OUTPUT.PUT_LINE(' - Trajet No: '|| m_notraj) ;
    DBMS_OUTPUT.PUT_LINE(' - Ville depart : '|| m_villedep) ;
    DBMS_OUTPUT.PUT_LINE(' - Ville arrivé : '|| m_villearr) ;
    DBMS_OUTPUT.PUT_LINE(' - Date      : '|| m_datetrajet) ;
    DBMS_OUTPUT.PUT_LINE(' - Nombre de km : '|| m_nbkm) ;
  ELSE -- nombre <> 1
    DBMS_OUTPUT.PUT_LINE('Ce numero n'existe pas ou il y a des erreurs');
  END IF ;
END ;
/

```

6. *Ecrivez un bloc PL/SQL pour :*

a. *Entrez des informations d'une nouvelle réparation (c.à.d NOREP, NOV, NOG, TYPREP, PX, KMCPT),*

b. *Validez que NOV est déjà existé dans la table VOITURE ou pas.*

i. *Si oui, créer ce nouveau tuple correspondant de la*

*table REPARATION,*

ii. *Si non, affichez un message d'erreur.*

```
DECLARE
  m_nombre INTEGER ;
  m_norep NUMBER(4) ;
  m_nov CHAR(12) ;
  m_nog NUMBER(3) ;
  m_typerep CHAR(12) ;
  m_px NUMBER(12,2) ;
  m_kmcpt NUMBER(10) ;
BEGIN
  -- Entrez les informations d'une nouvelle reparation
  m_norep := &Entrez_numero_reparation ;
  m_nov := '&Entrez_numero_voiture' ;
  m_nog := &Entrez_numero_garage ;
  m_typerep := '&Entrez_type_reparation' ;
  m_px := &Entrez_prix_reparation ;
  m_kmcpt := &Entrez_nombre_kilometres ;
  -- Validez le NOV est deja existe dans VOITURE
  SELECT COUNT(m_nov) INTO m_nombre
  FROM voiture WHERE nov = m_nov ;
  IF m_nombre = 1 THEN
    -- Insérer cette reparation
    INSERT INTO reparation VALUES
    (m_norep, m_nov, m_nog, m_typerep, m_px, m_kmcpt) ;
    -- Affichez le resultat
    DBMS_OUTPUT.PUT_LINE(' - Insere la reparation No '|| m_norep) ;
  ELSE -- nombre <> 1
    DBMS_OUTPUT.PUT_LINE('Ce numero de voiture n'existe pas ou il y a des erreurs');
  END IF ;
END ;
```