Université Henri Poincaré / ESIAL 1 / IUP GMI 1

Module Bases de Données

TP 3 : Vues – PL/SQL

Sauf mention contraire, on considère les tables de HR

1) a- Créer une vue `SALARIES_SOUTHLAKE` contenant certaines informations sur les salariés actuellement affectés dans les services localisés dans la ville de `SOUTHLAKE`*.

Le schéma de la vue est le suivant :

```
SALARIES_SOUTHLAKE (NO_SALARIE, NOM, PRENOM, NO_POSTE, NOM_SERVICE,
ANCIENNETE)
```

L'ancienneté dans le service est exprimée en mois**.

* : les fonctions `UPPER` ou `LOWER` permettent de s'affranchir des problèmes de majuscule/minuscule dans la comparaison de chaînes.
** : On pourra utiliser les fonctions `FLOOR` et `MONTHS_BETWEEN` pour calculer l'ancienneté (cf. Cours ou Documentation).

b- Afficher le contenu actuel la vue SALARIES_SOUTHLAKE, limité aux attributs NOM, PRENOM et ANCIENNETE, par ordre d'ancienneté décroissant.

c- Retrouver, pour chaque salarié de Southlake, le nom et le prénom de son supérieur hiérarchique.

2) Créer une fonction CHARGES qui retourne la valeur des charges pour un salaire donné en paramètre. Les charges représentent 45% du salaire.

3) Utiliser la fonction CHARGES pour afficher tous les montants des charges des salaires des employés actuellement en poste.

4)

a- Re-créer la table `LOCALISATION` et la peupler à partir de `HR.LOCATIONS` (cf. TP2).

b- Ecrire une procédure qui insère un nouveau tuple dans `LOCALISATION` : le numéro de localisation est le successeur du numéro de localisation le plus élevé ; la rue et la ville sont donnés en paramètre. La `RUE` aura comme valeur par défaut `'INCONNU'` ; la valeur par défaut de `VILLE` est 'Paris' (cf. schéma de la table `LOCALISATION`).

c- Exécuter la procédure pour ajouter un local à Paris, rue du louvre et à Nancy rue Saint-Jean. Vérifier que les ajouts se sont bien passés.

Rappel et complément : l'appel d'une procédure peut se faire soit à partir d'un bloc PL/SQL selon la syntaxe :

```
NOM_PROCEDURE [(PARAMETRES)] ;
```

ou en dehors d'un bloc PL/SQL comme une commande du langage SQL selon la syntaxe :

```
EXECUTE NOM_PROCEDURE [(PARAMÈTRES)]
```

Corrigé

1)

a- Créer une vue SALARIES_SOUTHLAKE contenant certaines informations sur les salariés actuellement affectés dans les services localisés dans la ville de Southlake.

```
CREATE OR REPLACE VIEW SALARIES_SOUTHLAKE (NO_SALARIE, NOM, PRENOM,
NO_POSTE, NOM_SERVICE, ANCIENNETE)

AS (SELECT EMPLOYEE_ID, LAST_NAME, FIRST_NAME, JOB_ID,
DEPARTMENT_NAME, FLOOR(MONTHS_BETWEEN(CURRENT_DATE , HIRE_DATE))
FROM HR.EMPLOYEES E, HR.DEPARTMENTS D, HR.LOCATIONS L
WHERE E.DEPARTMENT_ID=D.DEPARTMENT_ID
AND D.LOCATION_ID=L.LOCATION_ID
AND UPPER(L.CITY)='SOUTHLAKE');
/
```

d- Afficher le contenu actuel de la vue SALARIES_SOUTHLAKE, limité aux attributs NOM, PRENOM et ANCIENNETE, par ordre d'ancienneté décroissant.

```
SELECT      NOM, PRENOM,ANCIENNETE
FROM        SALARIES_SOUTHLAKE
ORDER BY    ANCIENNETE DESC;
/
```

NOM	PRENOM	ANCIENNETE
Hunold	Alexander	170
Ernst	Bruce	154
Austin	David	80
Pataballa	Valli	73
Lorentz	Diana	61

c- Retrouver, pour chaque salarié de Southlake, le nom et le prénom de son supérieur hiérarchique.

```
SELECT      S.NOM, S.PRENOM, CH.FIRST_NAME AS PRENOM_CHEF,
            CH.LAST_NAME AS NOM_CHEF
FROM        SALARIES_SOUTHLAKE S, HR.DEPARTMENTS, HR.EMPLOYEES CH
WHERE       S.NOM_SERVICE=D.DEPARTMENT_NAME
AND         D.MANAGER_ID=CH.EMPLOYEE_ID;
```

```
NOM             PRENOM          NOM_CHEF        PRENOM_CHEF
--------------- --------------- -----------     -----------
AUSTIN          DAVID           HUNOLD          ALEXANDER

ERNST           BRUCE           HUNOLD          ALEXANDER
```

HUNOLD	ALEXANDER	HUNOLD	ALEXANDER
LORENTZ	DIANA	HUNOLD	ALEXANDER
PATABALLA	VALLI	HUNOLD	ALEXANDER

2) Créer une fonction CHARGES qui retourne la valeur des charges pour un salaire donné en paramètre. Les charges représentent 45% du salaire.

```
CREATE OR REPLACE FUNCTION CHARGES (SALAIRE IN FLOAT )
RETURN FLOAT
AS
BEGIN
RETURN (SALAIRE*0.45);
END;
```

3) Utiliser la fonction CHARGES pour afficher tous les montants des charges des salaires des employés actuellement en poste.

```
SELECT EMPLOYEE_ID, CHARGES(SALARY) AS "SALAIRE BRUT"
FROM HR.EMPLOYEES;
```

EMPLOYEE_ID	SALAIRE BRUT
100	10800
101	7650
102	7650
103	4050
104	2700
....	

107 tuples sélectionnés

4)

a- Re-créer la table LOCALISATION et la peupler (cf. TP2).

```
CREATE TABLE LOCALISATION
(noloc number      PRIMARY KEY,
rue varchar2(40)   NULL,
ville varchar2(30)       DEFAULT 'Paris') ;

INSERT INTO LOCALISATION
(SELECT location_id noloc, street_address rue, city ville
 FROM HR.LOCATIONS) ;
```

b- Ecrire une procédure qui insère un nouveau tuple dans LOCALISATION : le numéro de localisation est le successeur du numéro de localisation le plus élevé ; la rue et la ville sont donnés en paramètre. La RUE aura comme valeur par défaut 'INCONNU' ; la valeur par défaut de VILLE est 'Paris' (cf. schéma de la table LOCALISATION.

```
CREATE OR REPLACE PROCEDURE AJOUT_LOCAL
      (RUE IN LOCALISATION.RUE%TYPE DEFAULT 'INCONNU',
       VILLE IN LOCALISATION.VILLE%TYPE DEFAULT 'PARIS')
AS
NO_MAX INTEGER ;
BEGIN
SELECT MAX(NOLOC) INTO NO_MAX FROM LOCALISATION;
INSERT INTO LOCALISATION VALUES (NO_MAX+1, RUE, VILLE);
COMMIT;
END  AJOUT_LOCAL;
```

c- Exécuter la procédure pour ajouter un local à PARIS, RUE DU LOUVRE et à NANCY, RUE SAINT-JEAN. Vérifier que les ajouts se sont bien passés.

```
BEGIN
AJOUT_LOCAL (RUE=>'RUE DU LOUVRE') ;
AJOUT_LOCAL (RUE=> 'RUE SAINT-JEAN', VILLE=> 'NANCY') ;
SELECT * FROM LOCALISATION
ORDER BY NOLOC DESC;
END ;
```