

EFREI - L3/L'3/L3Alternance A. Lahlou	Année : 2009/2010 TP-1 Bases de Données Oracle Karim.Lahlou@efrei.fr
--	---

Correction TP-1

Cas Emploi

I – Le Modèle Conceptuel de Données (MCD) :

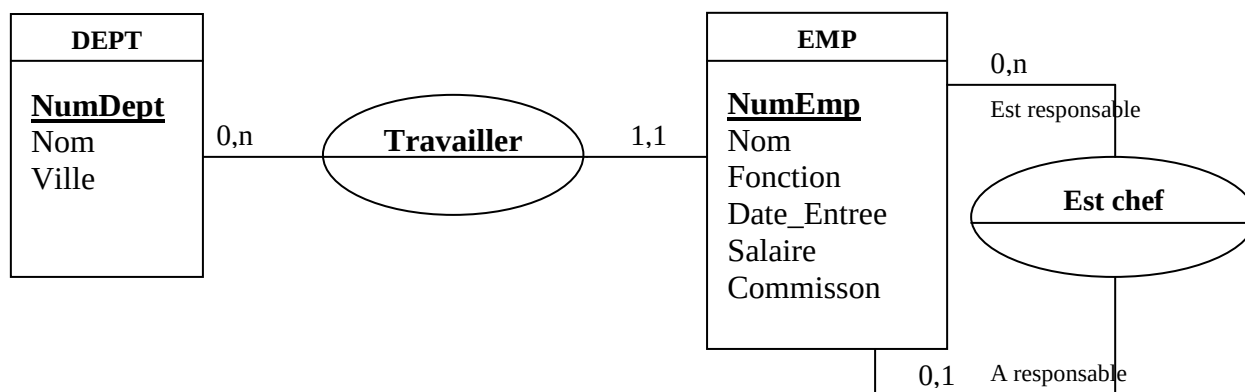
A partir du modèle logique de données relationnel (MLD) suivant, proposer le modèle conceptuel de données (MCD) équivalent :

EMP(NUMEMP, NOM, FONCTION, RESPONSABLE, DATE_ENTREE, SALAIRE, COMMISSION, NUMDEPT)
 DEPT(NUMDEPT, NOM, VILLE)

Remarques :

- dans la relation EMP, NumEmp est l'identifiant de l'employé, Fonction est la fonction qu'il exerce actuellement, Responsable est le numéro de son responsable hiérarchique, NumDept est le numéro du département dans lequel il travaille,
- dans DEPT, NumDept est le numéro de département, Nom son nom et Ville la ville où il est situé.
- Les clés étrangères ne sont pas signalées, à vous de les déduire

Réponse :



II – Le Modèle Logique de Données relationnel (MLD) :

```

CREATE TABLE Emp (
    NumEmp NUMBER(4),
    Nom VARCHAR2(20) NOT NULL,
    Fonction VARCHAR2(15),
    Responsable NUMBER(4),
    Date_Entree DATE,
    Salaire NUMBER(7, 2),
    Commission NUMBER(7, 2),
    NumDept NUMBER(2));
  
```

```
CREATE TABLE Dept(
    NumDept NUMBER(2),
    Nom VARCHAR2(15),
    Ville VARCHAR2(25)) ;
```

III – Les Contraintes d'intégrité :

1. Vérifier la création et le remplissage des tables EMP et DEPT.

```
DESC Emp ;
DESC Dept ;
SELECT * FROM Emp ;
SELECT * FROM Dept ;
```

2. Visualiser la liste des contraintes en affichant le contenu de **USER_CONSTRAINTS** du dictionnaire de données.

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
FROM USER_CONSTRAINTS ;
```

Quelles contraintes peut-on remarquer ? **NOT NULL** sur le nom de l'employé

3. Tester l'insertion d'un employé dont on ne connaît que le nom (DUPONT), le salaire (600 €) et le département (30).

```
INSERT INTO Emp (Nom, Salaire, NumDept) VALUES ('DUPONT', 600, 30) ;
Ok.
```

4. Modifier le schéma de EMP de manière à ce que l'attribut NUMEMP comporte 5 chiffres au maximum.

```
ALTER TABLE Emp MODIFY NumEmp NUMBER(5) ;
```

Peut-on remettre NUMEMP à 4 chiffres ?

Non vu qu'on risque de perdre de l'information en réduisant la taille du type de NumEmp

5. Modifier la table EMP de manière à ce qu'il ne puisse pas y avoir de salaire inférieur à 800€. Tester la contrainte.

```
ALTER TABLE Emp ADD CONSTRAINT Ck_Salaire CHECK (Salaire >=800) ;
```

Impossible car Dupond a un salaire de 600€

Donc supprimer l'employé Dupond

```
DELETE FROM Emp WHERE Nom = 'DUPONT' ;
```

Puis remettre la contrainte.

```
INSERT INTO Emp (NumEmp, Nom, Salaire, NumDept) VALUES (1200,
'DUPONT', 600, 30) ;
```

Insertion impossible

6. Modifier la table EMP de manière à ce que la date d'entrée soit par défaut la date courante. Tester la contrainte.

```
ALTER TABLE Emp MODIFY Date_Entree DEFAULT SYSDATE ;
```

7. Vérifier, en affichant le contenu de DEPT, si NUMDEPT satisfait la contrainte de clé primaire dans la table DEPT.

```
SELECT * FROM Dept ;
```

NumDept satisfait bien la contrainte de clé primaire vu qu'elle est non nulle et unique.

8. Insérer un département ayant la même valeur de NUMDEPT qu'un département existant.

```
INSERT INTO Dept (NumDept, Nom, Ville) VALUES (10,
'FINANCES', 'PARIS') ;
```

9. Tenter d'ajouter une contrainte sur la table DEPT définissant NUMDEPT comme clé primaire.

```
ALTER TABLE Dept ADD CONSTRAINT PK_Dept PRIMARY KEY(NumDept) ;  
Impossible vu qu'il existe dans Dept deux départements avec le même  
NumDept 10 (la clé primaire doit être unique).
```

10. Supprimez le tuple que vous venez d'insérer. Retentez la modification définissant NUMDEPT comme clé primaire de DEPT. Testez cette contrainte en ré-insérant à nouveau le tuple.

```
DELETE FROM Dept WHERE Nom = 'FINANCES' ;  
ALTER TABLE Dept ADD CONSTRAINT PK_Dept PRIMARY KEY(NumDept) ;  
INSERT INTO Dept (NumDept, Nom, Ville) VALUES (10,  
'FINANCES','PARIS') ;  
Insertion impossible car violation de la contrainte de clé primaire.
```

11. Modifier la table EMP de manière à ce que le cumul du salaire et de la commission soit obligatoirement compris entre 800 et 8000 €. Tester la contrainte.

```
ALTER TABLE Emp ADD CONSTRAINT Ck_SalCom CHECK ((Salaire +  
Commission) BETWEEN 800 AND 8000) ;  
INSERT INTO Emp (NumEmp, Nom, Salaire, Commission, NumDept) VALUES  
(1200, 'DUPONT', 600, 100, 30) ; Insertion impossible.
```

12. Modifier la table EMP de manière à ce que le numéro de département NUMDEPT définisse une clé étrangère correspondant à NUMDEPT de DEPT. Tester la contrainte.

```
ALTER TABLE Emp ADD CONSTRAINT FK_EMP_DEPT FOREIGN KEY (NumDept)  
REFERENCES Dept(NumDept) ;  
INSERT INTO Emp (NumEmp, Nom, Salaire, Commission, NumDept) VALUES  
(1200, 'DUPONT', 1500, 100, 50) ; Insertion impossible.
```

13. Modifier la table EMP de manière à ce que le numéro d'employé soit clé primaire. Tester la contrainte.

```
ALTER TABLE Emp ADD CONSTRAINT PK_Emp PRIMARY KEY(NumEmp) ;  
INSERT INTO Emp (NumEmp, Nom, Salaire, Commission, NumDept) VALUES  
(7369, 'DUPONT', 1500, 100, 20) ; Insertion impossible NumEmp Unique.
```

14. Tenter de définir la contrainte que COMM ne puisse pas avoir de valeur nulle.

```
ALTER TABLE Emp MODIFY Commission NOT NULL ;  
Impossible car présence de valeurs nulles dans la colonne Commission.
```

15. Tenter de supprimer la contrainte de clé primaire sur DEPT (Utiliser la clause DROP? Avec et sans la clause CASCADE).

```
ALTER TABLE Dept DROP CONSTRAINT PK_DEPT ; Impossible.  
ALTER TABLE Dept DROP CONSTRAINT PK_DEPT CASCADE ; Ok
```

16. Y'a-t-il un moyen pour exprimer la contrainte que les départements ont tous des employés ?
Non car le check ne peut s'appliquer que sur un attribut voir plusieurs attributs de la même table.

17. Pour récapituler, visualisez la liste des contraintes en affichant la vue du dictionnaire de données USER_CONSTRAINTS.

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME FROM USER_CONSTRAINTS ;
```

IV – Interrogation :

1. Donner la liste de tous les départements
`SELECT * FROM Dept ;`
2. Donner la liste des départements sans employés
`SELECT NumDept, Nom FROM Dept WHERE NumDept NOT IN (SELECT DISTINCT NumDept FROM Emp) ;`
3. Qui est le responsable de l'entreprise (il s'agit de l'employé sans responsable)
`SELECT * FROM Emp WHERE Responsable IS NULL ;`
4. Donner la liste des employés du plus grand salaire au plus petit
`SELECT * FROM Emp ORDER BY Salaire DESC ;`
5. Quel est le département qui emploie le plus d'employés ?
`CREATE VIEW NbEmp (NumDept, Total) AS (SELECT NumDept, COUNT(*) FROM Emp GROUP BY NumDept) ;`
`SELECT Nom FROM Dept WHERE NumDept IN (SELECT NumDept FROM NbEmp WHERE Total = (SELECT MAX(Total) FROM NbEmp)) ;`
6. Donner la masse salariale par département (total des salaires par département)
`SELECT NumDept, SUM(Salaire) AS TotalSalaires FROM Emp GROUP BY NumDept ;`
7. Quel est le total des commissions par département ?
`SELECT NumDept, SUM(Commission) AS TotalCommissions FROM Emp GROUP BY NumDept ;`

V – Les vues :

Une vue (appelée souvent table virtuelle) est structurée en lignes et en colonnes comme une relation mais n'occupe pas de place disque. Une vue peut être interrogée exactement comme une table, par contre il existe des restrictions sur les mises à jour autorisées.

1. Créer une table PROJET(NUMP, NOMP) en définissant NUMP comme clé primaire, puis insérer 5 projets numérotés entre 101 et 105.
2. Ajouter une colonne NUMP à la table EMP et une contrainte de clé étrangère sur cette colonne pour référencer la table PROJET. Affecter chaque employé à un projet.
3. Création de vue à partir d'une relation :
 - (a) Créer une vue EMP10 qui permet de donner pour le département numéro 10, les numéros des employés, leurs noms et leurs postes.
 - (b) Donner le contenu de la vue.
4. Impact d'une modification d'une relation sur une vue et vice versa :
 - (a) En utilisant la relation EMP, changer le poste de l'employée HELENE en ANALYSTE. Vérifier le contenu de EMP10.
 - (b) En utilisant la relation EMP, insérer un nouvel employé (7999, 'BROWN', 'SECRETAIRE', 7902, to_date('25-JAN-80','dd-mon-yy'), 800,NULL, 10, 102). Vérifier le contenu de EMP10.
 - (c) En utilisant la vue EMP10, donner les noms et les postes des employés du département numéro 10.
 - (d) En utilisant la vue EMP10, remettre le poste de l'employée HELENE à SECRETAIRE.

- (e) Donner le contenu de la vue EMP10. Que peut-on déduire ?
5. Création des vues par jointure sur les relations :
- (a) Créer une vue PROJSTAFF contenant les noms des employés, les noms et les numéros des projets sur lesquels ils sont affectés. Afficher le contenu de la vue.
 - (b) Créer une vue PROJECTS contenant les noms des projets, les noms et les numéros des employés, ainsi que la ville de leurs départements. Afficher le contenu de la vue.
 - (c) Que peut-on dire sur l'insertion du tuple ('DURAND','SATELLITE', 179) dans PROJSTAFF.
6. Interrogation des vues :
- (a) Donner les noms des employés qui sont affectés au projet numéro 101.
 - (b) Donner les noms des projets et les noms des employés travaillant à Paris.
7. Interrogation par jointure de relation et de vue
- (a) Donner les postes et les projets des employés travaillant à Paris. (Utiliser la vue PROJECTS et la relation EMP)
8. Création de vue à partir d'une autre vue :
- (a) En utilisant la vue PROJECTS, créer une autre vue PARIS donnant les noms des projets et les noms des employés travaillant à Paris. Lister ensuite le contenu de cette vue.
9. Création de vue avec utilisation de champs calculés :
- (a) Créer une vue PAY contenant les noms des employés, leurs salaires mensuels et leurs salaires annuels ainsi que leur numéro de département.
 - (b) Lister ensuite ceux du département 30.
 - (c) Peut-on modifier cette vue ?
10. Création de vue avec utilisation des agrégats :
- (a) Créer une vue DEPT_SAL contenant pour chaque département, le minimum, le maximum, la moyenne et le cumul des salaires.

VI – Les index :

Créer un index sur la table EMP sur la colonne nom afin d'améliorer les performances de recherche sur le nom d'employé. Tester le temps d'exécution d'une recherche sur le nom d'employé avant et après la mise en place de l'index, avec les commandes **SET TIMING ON** et **SET TIMING OFF**.

```
CREATE INDEX NDXNomEmp ON EMP (Nom ASC) ;
```