TP n°1 p1/7

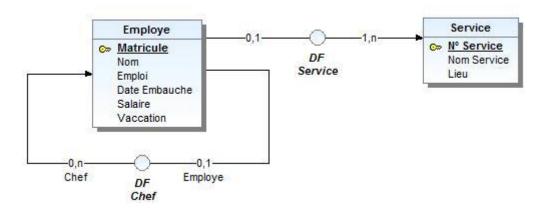
TP n°1

# PRÉSENTATION

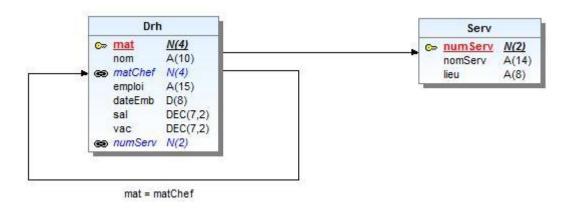
Il s'agit d'aider une DRH à développer une application pour la gestion du personnel.

Les informations nécessaires à cette activité de la DRH ont été structurées dans les modèles de données conceptuels (MCD) et logiques relationnels (MLDR) donnés ci-après.

#### MCD



#### MLDR



 $TP n^{\circ}1$  p2/7

# **■** CONTENU DES TABLES

## Table SERV

NUMSER	NUMSERV NOMSERV		LIEU		
	_				
	1	INFORMATIQUE	BAT	688	
	2	PRODUCTIQUE	BAT	680	
	3	ELECTRONIQUE	BAT	688	
	4	ADMINISTRATION	BAT	607	

# Table DRH

MAT	NOM	MATCHEF	EMPLOI	DATEEMB	SAL	VAC	NUMSERV
5147	BASILE	5344	FORMATEUR	23/01/85	1300	190	1
5277	GENEVIEVE	5476	FORMATEUR	03/05/82	1400	300	3
5319	RAYMOND	5476	FORMATEUR	03/05/85	1250	850	3
5344	LUCIEN	5617	CHEF	23/07/82	2000		1
5432	ALIX	5476	FORMATEUR	08/02/81	800	100	3
5476	GUILLAUME	5617	CHEF	17/03/84	1850		3
5560	PAULIN	5617	CHEF	09/11/86	1550		4
5566	TATIANA	5344	FORMATEUR	09/12/80	1850	30	1
5617	YVETTE		PRESIDENT	01/04/80	3800		4
5622	NINA	5476	SECRETAIRE	28/11/81	900		3
5654	RENE	5344	FORMATEUR	02/09/81	700	190	1
5708	MARCEL	5344	FORMATEUR	22/05/87	600	550	1
5700	ROSELINE	5344	SECRETAIRE	20/11/83	1800		1
5712	MARIUS	5566	ADMINISTRATIF	17/09/82	800		4

TP n°1 p3/7

#### QUESTION 1

Constituer un fichier *question1.sql* contenant un script SQL qui permet de créer la base de données de travail contenant les données sur le personnel et les services. Ce fichier est à constituer à partir des quatre fichiers listés ci-dessous.

- 1. TP1\_DRH\_SERV\_DROPCREATE.SQL qui contient les ordres de destruction et de création des deux tables DRH et SERV avec les définitions de clé primaires.
- **2.** TP1\_DRH\_SERV\_INSERT.SQL qui contient les ordres d'insertion de données dans les deux tables DRH et SERV.
- **3.** TP1\_DRH\_SERV\_FK.SQL qui contient les ordres de définitions des clés étrangères indispensables aux deux tables DRH et SERV.
- **4.** TP1\_DRH\_SERV\_SELECT.SQL qui permet de visualiser la structure et le contenu des deux tables ainsi que les contraintes d'intégrité définies explicitement sur les deux tables.

#### **QUESTION 2**

Constituer un fichier *question2.sql* contenant le script PL/SQL d'implémentation des deux fonctions suivantes utiles pour la suite.

- 1. Ecrire une fonction **NbEmploye** qui prend en entrée un numéro du service et qui renvoie le nombre (il peut être égal à 0) d'employés dans ce service.
- **2.** Ecrire une fonction **NbFormateur** qui prend en entrée un numéro du service et qui renvoie le nombre (il peut être égal à 0) de formateur dans ce service.

Utilisations possibles de ces fonctions :

TP n°1 p4/7

# **QUESTION 3**

Ecrire dans un fichier question3.sql un script SQL et PL/SQL qui permet de :

- 1. Ajouter une colonne NBPERS et NBFORM à la table SERV.
- 2. Ecrire un bloc PL/SQL qui permet de mettre à jour la table **SERV** automatiquement à l'aide d'un curseur. S'agissant d'une mise à jour par UPDATE via un curseur, on utilisera, lors de la déclaration du curseur, la clause FOR UPDATE de l'instruction SELECT et, lors de la mise à jour, la clause WHERE CURRENT OF de l'instruction UPDATE.

#### Résultat attendu:

Select * From Serv;										
NUMSERV	NOMSERV	LIEU	J	NBPERS	NBFORM					
2 3	INFORMATIQUE PRODUCTIQUE ELECTRONIQUE ADMINISTRATION	BAT BAT BAT BAT	680 688	6 0 5 3	4 0 3 0					

TP n°1 p5/7

#### **OUESTION 4**

On désire mettre à jour automatiquement les colonnes NBPERS et NBFORM de la table **SERV** chaque fois que la table **DRH** est modifiée.

Pour cela, on utilisera un TRIGGER sur la table **DRH** qui, lors d'un UPDATE sur la colonne *emploi*, d'un DELETE ou d'un INSERT sur cette table, mettra à jour les colonnes NBPERS ou NBFORM impactées par ces modifications.

On suppose qu'une personne une fois présente dans la table ne change pas de service. Conseil :

On utilisera un TRIGGER avec notamment les clauses <u>AFTER</u> et <u>FOR EACH ROW</u> et il n'y aura pas de gestion des erreurs car celle-ci est effectuée par les contraintes d'intégrité mises en place à la question 1.

On demande donc d'écrire dans un fichier *question4.sql* le script PL/SQL qui crée ce TRIGGER.

Pour vérifier si le TRIGGER fonctionne, on pourra effectuer les opérations suivantes :

```
Update Drh Set emploi = 'FORMATEUR', vac = 300
        Where mat = 5712;
Delete From Drh Where mat = 5708;
Insert Into Drh Values
      (5800, 'MAURICE', 5617, 'CHEF', SYSDATE, 12000, 0, 2);
Insert Into Drh Values
       (5900, 'SERGE', 5476, 'FORMATEUR', SYSDATE, 5000, 1000, 3);
select * from serv;
select * from drh;
                                                                    LIEU
                                                                                                        NBPERS
       NUMSERV NOMSERV
                                                                                                                                      NBFORM
                        1 INFORMATIQUE BAT 688
                        2 PRODUCTIQUE BAT 680
                                                                                                                     1
                        3 ELECTRONIQUE BAT 688
                        4 ADMINISTRATION BAT 607
     MAT NOM MATCHEF EMPLOI
                                                                                           DATEEMB
                                                                                                                                SAL VAC NUMSERV

        5147 BASILE
        5344 FORMATEUR
        23/01/85
        1300
        190

        5277 GENEVIEVE
        5476 FORMATEUR
        03/05/82
        1400
        300

        5319 RAYMOND
        5476 FORMATEUR
        03/05/85
        1250
        850

        5344 LUCIEN
        5617 CHEF
        23/07/82
        2000
        5432 ALIX
        5476 FORMATEUR
        08/02/81
        800
        100

        5476 GUILLAUME
        5617 CHEF
        17/03/84
        1850
        5560 PAULIN
        5617 CHEF
        09/11/86
        1550

        5566 TATIANA
        5344 FORMATEUR
        09/12/80
        1850
        30

        5617 YVETTE
        PRESIDENT
        01/04/80
        3800

        5622 NINA
        5476 SECRETAIRE
        28/11/81
        900

        5654 RENE
        5344 FORMATEUR
        02/09/81
        700
        190

        5700 ROSELINE
        5344 SECRETAIRE
        20/11/83
        1800

        5712 MARIUS
        5566 FORMATEUR
        17/09/82
        800
        300

        5800 MAURICE
        5617 CHEF
        08/03/05
        12000
        0

        5900 SERGE
        5476 FORMATEUR
        08/03/05
        5000
        1000
    </t
 _____
                                       ______
                                                                                                                                                                                         3
                                                                                                                                                                                         3
                                                                                                                                                                                        1
                                                                                                                                                                                         3
                                                                                                                                                                                         3
                                                                                                                                                                                         4
                                                                                                                                                                                         1
                                                                                                                                                                                         3
                                                                                           17/09/82 800
08/03/05 12000
08/03/05 5000
                                                                                                                                                                                         1
                                                                                                                                                                                         4
                                                                                                                                                                                         2
```

 $TP n^{\circ}1$  p6/7

#### **OUESTION 5**

Dans le même ordre d'idée que les questions 2, 3 et 4 :

• Ecrire une fonction **MasseSalariale** qui prend en entrée un numéro du service et qui renvoie la masse salariale de ce service. Rendre 0 si le service ne compte aucun salarié.

- Ajouter une colonne BUDGET à la table SERV qui représente la masse salariale de chaque service.
- Mettre à jour dynamiquement cette colonne en tenant compte des salaires et des vacations des employés actuels de chaque service. Cette colonne vaut 0 si le service ne compte aucun salarié.

```
        Select * From Serv;

        NUMSERV NOMSERV
        LIEU
        NBPERS
        NBFORM
        BUDGET

        1 INFORMATIQUE
        BAT 688
        5
        3
        8060

        2 PRODUCTIQUE
        BAT 680
        1
        0
        12000

        3 ELECTRONIQUE
        BAT 688
        6
        4
        13450

        4 ADMINISTRATION
        BAT 607
        3
        1
        6450
```

• Mettre à jour automatiquement cette colonne à chaque mouvement de salarié (nouvelle embauche d'un salarié dans un service, débauche d'un salarié, promotion d'un salarié, etc.).

```
Delete From Drh Where mat = 5900;
Delete From Drh Where mat = 5800;
Insert Into Drh
 Values (5708, 'MARCEL', 5344, 'FORMATEUR', SYSDATE, 600, 550, 1);
Update Drh Set emploi = ' ADMINISTRATIF', vac = NULL
 Where mat = 5712;
Select * from serv;
  NUMSERV NOMSERV
               LIEU NBPERS NBFORM BUDGET
 _____ ___
       1 INFORMATIQUE BAT 688
                                6
                                              9210
                                     0
                                0
       2 PRODUCTIQUE BAT 680
                                              0
       3 ELECTRONIQUE BAT 688
                                             7450
                                5
                                       3
                                            6150
       4 ADMINISTRATION BAT 607 3 0
```

 $TP n^{\circ}1$  p7/7

### **QUESTION 6**

Ecrire des triggers qui rendent possible le respect des règles de gestion suivantes :

Seuls les formateurs ont droit à une vacation.

```
Update Drh Set emploi = 'FORMATEUR'
Where mat = 5712;
Update Drh Set emploi = ' ADMINISTRATIF'
Where mat = 5708;
ORA-20001: Un formateur a droit à une vacation !
ORA-20002: Un non formateur n'a pas droit à une vacation !
```

Un salarié ne doit pas être son propre chef.

```
Update Drh Set matchef = 5617
Where mat = 5617;
ORA-20004: Un salarié ne doit pas être son propre chef. !
```

• La date d'embauche d'un salarié est une information stable, i.e. non modifiable.

```
Update Drh Set dateEmb = SYSDATE
    Where mat = 5712;
ORA-20005: La date d'embauche d'un salarié est non modifiable.
```

• Le salaire d'un salarié ne doit jamais diminuer.

```
Update Drh Set sal = 3000
    Where mat = 5617;
ORA-20003: Le salaire d'un employé ne doit jamais diminuer !
Update Drh Set sal = 500, vac = 700
    Where mat = 5708;
-- OK
```