

### Création d'une base de données

Soit le schéma relationnel suivant :

SERVICE (NUM\_SERVICE, LIB\_SERVICE, HIERARCHIE#)  
EMPLOYE (NUM\_EMPLOYE, NOM, PRENOM, SERVICE#)  
NATURE\_CONGE (NUM\_NATURE\_CONGE, LIB\_NATURE\_CONGE)  
DROIT\_CONGE (EMPLOYE#, NATURE\_CONGE#, NB\_JOURS)  
CONGE\_PRIS (NUM\_CONGE, EMPLOYE#, DATE\_DEBUT\_CONGE, DATE\_FIN\_CONGE, NATURE\_CONGE#)

L'extension des tables est disponible en fin d'énoncé.

1. Créer la structure des quatre tables : SERVICE, EMPLOYE, NATURE\_CONGE, DROIT\_CONGE et CONGE\_PRIS. Ne pas oublier de spécifier les contraintes d'intégrité (clés primaires et étrangères).
2. Modifier la structure de la table CONGE\_PRIS en ajoutant la contrainte de domaine spécifiant que DATE\_FIN\_CONGE est strictement supérieure à DATE\_DEBUT\_CONGE. (La date de début correspond au jour où l'employé ne travaille pas, et la date de fin de congé correspond à la date de reprise du travail).
3. Remplir la table NATURE\_CONGE en utilisant l'interface graphique de *SQL Developer*.
4. Remplir les tables SERVICE, EMPLOYE, DROIT\_CONGE et CONGE\_PRIS à partir des tables SERVICE2, EMPLOYE, DROIT\_CONGE et CONGE\_PRIS de l'utilisateur DARMONT, respectivement.

### Mise à jour de la base de données

1. Modifier le nombre de jours de congés annuels pour l'employé LHERMITTE : il a droit à 10 jours de congés annuels.
2. Modifier le nombre de jours de congés du type 3 : pour les employés qui en avaient moins de 4, il faut leur en rajouter 10.
3. Supprimer le congé pris par l'employé 8 qui finissait le 13/02/05.
4. Supprimer le congé pris par l'employé CLAVIER qui commençait le 15/02/05.

\* D'après C. Favre, 2004.

### Requêtes sur la base de données

1. Nom et prénom de tous les employés par ordre alphabétique du nom.
2. Nom et prénom des employés dont le nom commence par un C.
3. Nom et prénom des employés dont le nom finit par un T.
4. Nombre total d'employés.
5. Nombre total de services.
6. Nom et prénom des employés travaillant dans l'atelier montage.
7. Service, nom et prénom des employés travaillant dans le service numéro 1100 ou dans un des services dépendant de 1100 (2 méthodes).
8. Identifiant des services dans lesquels travaillent strictement plus d'une personne.
9. Libellé des services où aucun employé n'a été affecté.
10. Libellé des services où au moins un employé travaille (2 méthodes : 2<sup>ème</sup> méthode en utilisant la requête précédente).
11. Nombre total de jours de congés déjà pris par employé.
12. Nombre total de jours de congés auxquels chaque employé a droit.
13. Nom et prénom des employés ayant un nombre de jours de congé supérieur à un nombre saisi au clavier (paramètre).
14. Nombre d'employés par service.
15. Nom et prénom des employés qui prennent au moins un jour de congé entre le 18/12/05 et le 26/12/05 (c'est le cas si le jour de début ou le jour de fin des congés est compris dans cette période).
16. Nom et prénom des employés qui prennent au moins un jour de congé entre deux dates saisies au clavier.
17. Détecter les nom et prénom des employés non référencés dans la table DROIT\_CONGE.
18. Employés ayant droit à des congés pour motif de récupération.
19. Nom et prénom des employés ayant posé des congés (éliminer les doublons).
20. Structure hiérarchique avec les niveaux des services (NUM\_SERVICE, LIB\_SERVICE), sachant que le service qui dirige l'ensemble de l'entreprise est la direction générale.
21. Structure hiérarchique avec les niveaux des employés (NOM, PRENOM et libellé du service) en fonction de leur service d'appartenance.
22. Nombre d'employés pour chaque niveau de service.

### Vues

1. Créer une vue point\_conges regroupant nom, prénom, identifiant et libellé du service, identifiant et libellé de la nature du congé, nombre de jours de congé auxquels il a droit et le nombre de congé qu'il a pris.
2. Lister le contenu de la vue point\_conges, remarque ?

3. A partir de la vue point\_conges, répondre aux questions suivantes :
  - a. nombre total de jours de congés accordés à l'ensemble des employés présents dans la vue par l'entreprise ;
  - b. nombre total de jours de congés donnés à l'ensemble des employés présents dans la vue par service ;
  - c. nombre de jours total de congés qui restent à prendre pour chaque employé présent dans la vue ;
  - d. moyenne du nombre de jours de congé posés par employé présent dans la vue ;
  - e. remarque ?

### Vues et catalogue du système

1. Créer la vue VEMP (NUM\_EMPLOYE, NOM, PRENOM) à partir de la table EMPLOYE et vérifier son contenu.
2. A travers la vue VEMP, modifier le nom de l'employé 11 et consulter le contenu de la vue VEMP et de la table EMPLOYE.
3. Créer la vue VAFFECT (NUM\_EMPLOYE, NOM, PRENOM, LIB\_SERVICE) qui regroupe les nom et prénom des employés avec le libellé de leur service d'affectation.
4. A travers la vue VAFFECT, modifier le libellé du service de l'employé 6 et consulter le contenu de la vue VAFFECT et de la table EMPLOYE, que se passe-t-il ?
5. Lister toutes les contraintes définies sur les tables de la base de données étudiée dans ce TD à l'aide de la vue système USER\_CONSTRAINTS. Afficher pour chaque contrainte son nom, la table à laquelle elle s'applique et son type.
6. Créer la vue MES\_CONTRAINTES à partir du résultat de la requête précédente, vérifier son contenu.

### Privilèges

1. Donner à vos voisins le droit de consulter votre table EMPLOYE et consulter la-leur.
2. Modifier le nom de l'employé 2 dans la table EMPLOYE de vos voisins, que se passe-t-il ?
3. Donner à vos voisins le droit de mise à jour dans la table EMPLOYE, modifier le nom de l'employé 2 dans leur table.

### Extension des tables

#### SERVICE2

NUM_SERVICE	LIB_SERVICE	HIERARCHIE
1000	Direction Générale	
1100	Ressources Humaines	1000
1101	Service Payes	1100
1102	Service Recrutement	1100
1103	Service Formation	1100
1200	Comptabilité	1000
1201	Service Facturation	1200
1202	Service Fournisseur	1200
1300	Marketing	1000
1400	Production	1000
1401	Atelier Montage	1400
1402	Atelier Finition	1400

#### NATURE\_CONGE

NUM_NATURE_CONGE	LIB_NATURE_CONGE
1	Congé Annuel
2	RTT Entreprise
3	RTT Salarié
4	Récupération

#### DROIT\_CONGE

EMPLOYE	NATURE_CONGE	NB_JOURS
4	1	20
2	1	3
11	4	4
3	1	20
6	2	10
6	3	3
8	3	5
1	2	4
11	1	20
7	1	15
9	1	17
7	2	11
8	2	1
7	4	1
6	1	15
7	3	6
9	3	4
11	3	5
4	4	3
11	2	12
12	4	2

#### EMPLOYE

NUM_EMPLOYE	NOM	PRENOM	SERVICE
1	JUGNOT	Gérard	1000
2	CLAVIER	Christian	1103
3	HAZEL	Marie-Anne	1100
4	LHERMITTE	Thierry	1202
5	BALASKO	Josiane	1200
6	MOYNOT	Bruno	1103
7	CHEVIT	Maurice	1401
8	BLANC	Michel	1300
9	LAVANANT	Dominique	1400
10	GIRAUD	Roland	1402
11	SUCH	Michel	1201
12	LAPORTE	Guy	1401

#### CONGE\_PRIS

NUM_CONGE	EMPLOYE	DATE_DEBUT_CONGE	DATE_FIN_CONGE	NATURE_CONGE
1	11	01/01/05	05/01/05	1
2	4	10/01/05	16/01/05	4
3	2	15/02/05	18/02/05	1
4	8	12/02/05	13/02/05	2
5	1	07/05/05	11/05/05	2
6	6	12/07/05	15/07/05	3
7	6	03/10/05	05/10/05	1
8	11	09/07/05	15/07/05	1
9	8	09/08/05	14/08/05	3
10	9	10/04/05	14/04/05	1
11	9	15/09/05	17/09/05	1
12	1	15/07/05	17/07/05	2
13	12	22/12/05	24/12/05	4
14	11	24/12/05	29/12/05	1
15	4	15/12/05	19/12/05	1
16	6	22/11/05	25/11/05	3
17	12	04/12/05	06/12/05	4

## Correction

-- Création de la base de données

```
create table SERVICE(NUM_SERVICE number(4), LIB_SERVICE varchar(50), HIERARCHIE
number(4), constraint service_cle_pri primary key(NUM_SERVICE), constraint
hierarchie_cle_etr foreign key(HIERARCHIE) references SERVICE(NUM_SERVICE));
```

```
create table EMPLOYE(NUM_EMPLOYE number(2), NOM varchar(30), PRENOM varchar(30),
SERVICE number(4), constraint employe_cle_pri primary key(NUM_EMPLOYE),
constraint employe_cle_etr foreign key(SERVICE) references
SERVICE(NUM_SERVICE));
```

```
create table NATURE_CONGE(NUM_NATURE_CONGE number(2), LIB_NATURE_CONGE
varchar(30), constraint natureconge_cle_pri primary key(NUM_NATURE_CONGE));
```

```
create table DROIT_CONGE(EMPLOYE number(2), NATURE_CONGE number(2), NB_JOURS
number(2), constraint droitconge_cle_pri primary key(EMPLOYE, NATURE_CONGE));
```

```
create table CONGE_PRIS(NUM_CONGE number(2), EMPLOYE number(2), DATE_DEBUT_CONGE
date, DATE_FIN_CONGE date, NATURE_CONGE number(2), constraint congepris_cle_pri
primary key(NUM_CONGE), constraint congepris_employe_cle_etr foreign
key(EMPLOYE) references EMPLOYE(NUM_EMPLOYE), constraint
congepris_natconge_cle_tr foreign key(NATURE_CONGE) references
NATURE_CONGE(NUM_NATURE_CONGE));
```

```
alter table CONGE_PRIS add constraint date_conge_valide
CHECK(DATE_DEBUT_CONGE<DATE_FIN_CONGE);
```

```
insert into NATURE_CONGE values(&num_conge, '&lib_type_conge');
insert into SERVICE select * from DARMONT.SERVICE2;
insert into EMPLOYE select * from DARMONT.EMPLOYE;
insert into DROIT_CONGE select * from DARMONT.DROIT_CONGE;
insert into CONGE_PRIS select * from DARMONT.CONGE_PRIS;
```

-- Mise à jour de la base de données

```
update DROIT_CONGE set NB_JOURS = 10 where EMPLOYE = (select NUM_EMPLOYE from
EMPLOYE where NOM='LHERMITTE') and NATURE_CONGE = (select NUM_NATURE_CONGE from
NATURE_CONGE where LIB_NATURE_CONGE = 'Congé Annuel');
```

```
update DROIT_CONGE set NB_JOURS = (NB_JOURS + 10) where NATURE_CONGE = 3 and
NB_JOURS < 4;
```

```
delete from CONGE_PRIS where EMPLOYE = 8 and DATE_FIN_CONGE = '13/02/05';
```

```
delete from CONGE_PRIS where EMPLOYE = (select NUM_EMPLOYE from EMPLOYE where
NOM = 'CLAVIER') and DATE_DEBUT_CONGE = '15/02/05';
```

-- Interrogation de la base de données

```
select NOM, PRENOM from EMPLOYE order by NOM;
```

```
select NOM, PRENOM from EMPLOYE where NOM like 'C%';
```

```
select NOM, PRENOM from EMPLOYE where NOM like '%T';
```

```
select count(NUM_EMPLOYE) from EMPLOYE;
```

```
select count(NUM_SERVICE) from SERVICE;
```

```
select NOM, PRENOM from EMPLOYE, SERVICE where SERVICE.NUM_SERVICE =
EMPLOYE.SERVICE AND SERVICE = (select NUM_SERVICE from SERVICE where LIB_SERVICE
= 'Atelier Montage');
```

```
select SERVICE, NOM, PRENOM from EMPLOYE where SERVICE = 1100 or SERVICE in
(select NUM_SERVICE from SERVICE where HIERARCHIE = 1100);
-- ou
select SERVICE, NOM, PRENOM from EMPLOYE where SERVICE = 1100
union
select SERVICE, NOM, PRENOM from EMPLOYE where SERVICE in (select NUM_SERVICE
from SERVICE where HIERARCHIE = 1100);
```

```
select NUM_SERVICE
from SERVICE, EMPLOYE
where SERVICE.NUM_SERVICE = EMPLOYE.SERVICE
group by NUM_SERVICE
having count(NUM_EMPLOYE) > 1;
```

```
select LIB_SERVICE from SERVICE where not exists (select * from EMPLOYE where
SERVICE = NUM_SERVICE);
```

```
select LIB_SERVICE from SERVICE where exists (select * from EMPLOYE where
EMPLOYE.SERVICE = SERVICE.NUM_SERVICE);
```

```
-- ou
select LIB_SERVICE from SERVICE
minus
select LIB_SERVICE from SERVICE where not exists (select * from EMPLOYE where
SERVICE = NUM_SERVICE);
```

```
select EMPLOYE, sum(DATE_FIN_CONGE-DATE_DEBUT_CONGE) from CONGE_PRIS group by
EMPLOYE;
```

```
select EMPLOYE, sum(NB_JOURS) from DROIT_CONGE group by EMPLOYE;
```

```
select NOM, PRENOM from EMPLOYE where (select sum(NB_JOURS) from DROIT_CONGE
where EMPLOYE.NUM_EMPLOYE = DROIT_CONGE.EMPLOYE) > &nombre_jours;
```

```
select SERVICE, count(NUM_EMPLOYE) from EMPLOYE group by SERVICE;
```

```
select NOM, PRENOM from EMPLOYE, CONGE_PRIS where CONGE_PRIS.EMPLOYE =
EMPLOYE.NUM_EMPLOYE AND DATE_DEBUT_CONGE > '18/12/05' AND DATE_DEBUT_CONGE <
'26/12/05'
union
select NOM, PRENOM from EMPLOYE, CONGE_PRIS where CONGE_PRIS.EMPLOYE =
EMPLOYE.NUM_EMPLOYE AND DATE_FIN_CONGE > '18/12/05' AND DATE_FIN_CONGE <
'26/12/05';
```

```
select NOM, PRENOM from EMPLOYE, CONGE_PRIS where CONGE_PRIS.EMPLOYE =
EMPLOYE.NUM_EMPLOYE AND DATE_DEBUT_CONGE > '&date_debut' AND DATE_DEBUT_CONGE <
'&date_fin'
union
select NOM, PRENOM from EMPLOYE, CONGE_PRIS where CONGE_PRIS.EMPLOYE =
EMPLOYE.NUM_EMPLOYE AND DATE_FIN_CONGE > '&date_debut' AND DATE_FIN_CONGE <
'&date_fin';
```

```
select NOM, PRENOM from EMPLOYE where not exists (select * from DROIT_CONGE
where DROIT_CONGE.EMPLOYE = EMPLOYE.NUM_EMPLOYE);
```

```
select distinct NOM, PRENOM from EMPLOYE, DROIT_CONGE where EMPLOYE.NUM_EMPLOYE
= DROIT_CONGE.EMPLOYE and NATURE_CONGE = (select NUM_NATURE_CONGE from
NATURE_CONGE where LIB_NATURE_CONGE = 'Récupération');
```

```
select distinct NOM, PRENOM from EMPLOYE, CONGE_PRIS where EMPLOYE.NUM_EMPLOYE =
CONGE_PRIS.EMPLOYE;
```

```
select level, NUM_SERVICE, LIB_SERVICE from SERVICE connect by prior NUM_SERVICE
= HIERARCHIE start with LIB_SERVICE = 'Direction Générale';
```

```
select level, NOM, LIB_SERVICE from EMPLOYE, SERVICE where EMPLOYE.SERVICE =
SERVICE.NUM_SERVICE connect by prior NUM_SERVICE = HIERARCHIE start with
LIB_SERVICE = 'Direction Générale';
```

```
select LIB_SERVICE, count(NUM_EMPLOYE) from EMPLOYE, SERVICE where
EMPLOYE.SERVICE = SERVICE.NUM_SERVICE group by LIB_SERVICE connect by prior
NUM_SERVICE = HIERARCHIE start with LIB_SERVICE='Direction Générale';
```

-- Vues

```
create view point_conges as select NOM, PRENOM, NUM_SERVICE, LIB_SERVICE,
NUM_NATURE_CONGE, LIB_NATURE_CONGE, DATE_FIN_CONGE - DATE_DEBUT_CONGE as
NB_JOURS_POSES, NB_JOURS from SERVICE, EMPLOYE, NATURE_CONGE, CONGE_PRIS,
DROIT_CONGE where SERVICE.NUM_SERVICE = EMPLOYE.SERVICE and EMPLOYE.NUM_EMPLOYE
= CONGE_PRIS.EMPLOYE and EMPLOYE.NUM_EMPLOYE = DROIT_CONGE.EMPLOYE and
NATURE_CONGE.NUM_NATURE_CONGE = DROIT_CONGE.NATURE_CONGE and
NATURE_CONGE.NUM_NATURE_CONGE = CONGE_PRIS.NATURE_CONGE;
```

```
select * from point_conges;
```

-- On ne retrouve dans la vue que les employés qui étaient à la fois présents  
-- dans la table CONGE\_PRIS et DROIT\_CONGE. Ainsi, par exemple, ceux qui n'ont  
-- pas posé de congés mais qui y avaient droit ne figurent pas dans la vue.

```
select sum(NB_JOURS_POSES) from point_conges;
```

```
select LIB_SERVICE, sum(NB_JOURS_POSES) from point_conges group by LIB_SERVICE;
```

```
select LIB_TYPE_CONGE, sum(NB_JOURS_DROIT_CONGE) from point_conges group by
LIB_TYPE_CONGE;
```

```
select NOM, (sum(NB_JOURS)-sum(NB_JOURS_POSES)) from point_conges group by NOM;
```

```
select avg(sum(NB_JOURS_POSES)) from point_conges group by NOM;
```

-- Vues et catalogue du système

```
create view VEMP as select NUM_EMPLOYE, NOM, PRENOM from EMPLOYE;
select * from VEMP;
```

```
update VEMP set NOM = 'Autre_Nom' where NUM_EMPLOYE = 11;
select * from VEMP;
select * from EMPLOYE;
```

```
create view VAFFECT as select NUM_EMPLOYE, NOM, PRENOM, LIB_SERVICE from
EMPLOYE, SERVICE where EMPLOYE.SERVICE = SERVICE.NUM_SERVICE;
update VAFFECT set LIB_SERVICE = 'Autre_Lib_Service' where NUM_EMPLOYE = 6;
select * from VAFFECT;
select * from EMPLOYE;
```

```
select table_name, constraint_name, constraint_type from user_constraints where
table_name in ('EMPLOYE', 'SERVICE', 'TYPE_CONGE', 'DROIT_CONGE', 'CONGE_PRIS');
```

```
create view MES_CONTRAIINTES as select table_name, constraint_name,
constraint_type from user_constraints where table_name in ('EMPLOYE', 'SERVICE',
'TYPE_CONGE', 'DROIT_CONGE', 'CONGE_PRIS');
select * from MES_CONTRAIINTES;
```

-- Privilèges

```
grant select on EMPLOYE to login_voisin ;
select * from login_voisin.EMPLOYE;
```

```
update login_voisin.EMPLOYE set NOM = 'Autre_nom' where NUM_EMPLOYE = 2;
```

```
grant update on EMPLOYE to login_voisin;
```

```
update login_voisin.EMPLOYE set NOM = 'Autre_nom' where NUM_EMPLOYE = 2;
```