Département Informatique et Statistique, Faculté de SEG, Université Lumière Lyon 2
M1 Informatique – Année 2010-2011

**Langages de requêtes – TD n° 3**

J. Darmont (*http://eric.univ-lyon2.fr/~jdarmont/*), 05/01/11

## Base de données exemple

Considérons la base de données dont le schéma et l'extension sont donnés ci-dessous.

EMP (EMPNO, ENAME, JOB, MGR#, HIREDATE, SAL, COMM, DEPTNO#)
DEPT (DEPTNO, DNAME, LOC)

Clés primaires
Clés étrangères#

**EMP**

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|
| 7369 | SMITH | CLERK | 7902 | 17/12/80 | 800.00 | NULL | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 20/02/81 | 1600.00 | 300.00 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 22/02/81 | 1250.00 | 500.00 | 30 |
| 7566 | JONES | MANAGER | 7839 | 02/04/81 | 2975.00 | NULL | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 28/09/81 | 1250.00 | 1400.00 | 30 |
| 7698 | BLAKE | MANAGER | 7839 | 01/05/81 | 2850.00 | NULL | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09/06/81 | 2450.00 | NULL | 10 |
| 7788 | SCOTT | ANALYST | 7566 | 09/11/87 | 3000.00 | NULL | 20 |
| 7839 | KING | PRESIDENT | NULL | 17/11/81 | 5000.00 | NULL | 10 |
| 7844 | TURNER | SALESMAN | 7698 | 08/09/81 | 1500.00 | 0.00 | 30 |
| 7876 | ADAMS | CLERK | 7788 | 23/09/87 | 1100.00 | NULL | 20 |
| 7900 | JAMES | CLERK | 7698 | 03/12/81 | 950.00 | NULL | 30 |
| 7902 | FORD | ANALYST | 7566 | 03/12/81 | 3000.00 | NULL | 20 |
| 7934 | MILLER | CLERK | 7782 | 23/01/82 | 1300.00 | NULL | 10 |

**DEPT**

| DEPTNO | DNAME | LOC |
|---|---|---|
| 10 | ACCOUNTING | NEW-YORK |
| 20 | RESEARCH | DALLAS |
| 30 | SALES | CHICAGO |
| 40 | OPERATIONS | BOSTON |

## Création de la base de données, contraintes d'intégrité

1) Créer la table DEPT. Pour cela, il est possible d'utiliser l'interface graphique de *SQL Developer* (onglet *Connections* / bouton droit sur l'élément *Tables* / *New Table*). Ne pas oublier de définir le numéro de département DEPTNO comme clé primaire. Intégrer également la *contrainte de domaine* suivante : le nom d'un département (DNAME) ne peut être que ACCOUNTING, RESEARCH, SALES ou OPERATIONS.
2) Remplir la table DEPT. Il est également possible d'utiliser l'interface graphique de *SQL Developer* pour cela (onglet *Connections* / développer l'élément *Tables* / double-cliquer sur DEPT / onglet *Data* dans la partie droite de l'écran / icône ).
3) Recopier la table EMP en passant la commande suivante :
`CREATE TABLE EMP AS SELECT * FROM SCOTT.EMP;`
(création de la table EMP par copie de la table EMP de l'utilisateur SCOTT).
4) Valider les mises à jour précédentes (`COMMIT` ou F11 sous *SQL Developer*). **À partir de la question suivante, utiliser uniquement des requêtes SQL, que ce soit pour mettre à jour ou consulter les tables.**



5) Insérer le n-uplet (7369, 'Bidon', NULL, NULL, NULL, NULL, NULL, NULL) dans la table EMP. Ça fonctionne. Est-ce normal ?
6) Annuler l'insertion précédente (`ROLLBACK` ou F12 sous *SQL Developer*).
7) Ajouter les contraintes d'intégrité nécessaires à la table EMP (clé primaire et clé**s** étrangère**s**) à l'aide de la commande `ALTER TABLE EMP ADD CONSTRAINT`…
8) Dans EMP, ajouter les nouveaux employés :
<7369, 'WILSON', 'MANAGER', 7839, '17/11/91', 3500.00, 600.00, 10> ;
<7657, 'WILSON', 'MANAGER', 7839, '17/11/91', 3500.00, 600.00, 50> ;
<7657, 'WILSON', 'MANAGER', 7000, '17/11/91', 3500.00, 600.00, 10> ;
<7657, 'WILSON', 'MANAGER', 7839, '17/11/91', 3500.00, 600.00, 10>.
Remarques ?
9) Valider l'insertion précédente.

## Mise à jour de la base de données

1) Changer la localisation (LOC) du département SALES de CHICAGO à PITTSBURGH.
2) Dans EMP, augmenter de 10 % le salaire (SAL) des vendeurs dont la commission (COMM) est supérieure à 50 % du salaire.
3) Dans EMP, attribuer aux employés en poste avant le 01/01/1982 (HIREDATE) et ayant une commission non spécifiée (NULL) une commission égale à la moyenne des commissions.
4) Annuler les trois mises à jour précédentes.
5) Dans DEPT, supprimer le département n° 20 (DEPTNO). Remarque ?

## Interrogation de la base de données

Exprimer en SQL les requêtes suivantes.

1) Nom (ENAME), salaire, commission, salaire+commission de tous les vendeurs (SALESMAN).
2) Nom des vendeurs par ordre décroissant du ratio commission/salaire.
3) Nom des vendeurs dont la commission est inférieure à 25% de leur salaire.
4) Nombre d'employés du département n° 10.
5) Nombre d'employés ayant une commission.
6) Nombre de fonctions (JOB) différentes.
7) Salaire moyen par fonction (sans tenir compte des commissions).
8) Total des salaires du département SALES.
9) Nom des employés avec le nom de leur département.
10) Nom, fonction et salaire de l'employé ayant le salaire le plus élevé.
11) Nom des employés gagnant plus que JONES.
12) Nom des employés occupant la même fonction que JONES.
13) Nom des employés ayant même manager (MGR) que CLARK.
14) Nom et fonction des employés ayant même fonction et même manager que TURNER.

15) Nom des employés embauchés avant tous les employés du département n° 10.

16) Liste des employés en indiquant pour chacun son nom et celui de son manager.

17) Nom des employés ne travaillant pas dans le même département que leur manager.

*Requêtes hiérarchiques*

18) Structure hiérarchique des employés (NOM, JOB, EMPNO, MGR), sachant que l'employé au sommet de la hiérarchie n'a pas de manager.

19) Liste des employés dépendant de JONES en indiquant leur niveau (LEVEL) dans cette hiérarchie.

20) Salaire moyen pour chaque niveau d'employé.

21) Liste des employés qui travaillent pour JONES, sauf SCOTT.

22) Liste des employés qui travaillent pour JONES, sauf SCOTT et ceux qui travaillent pour SCOTT.

23) Liste des employés dans l'ordre hiérarchique, présentée sous forme indentée (niveau 0 de la hiérarchie aligné à gauche, niveau 1 décalé vers la droite, niveau 2 plus décalé, etc.).

24) Idem, mais en indiquant en plus le niveau hiérarchique à gauche.

*Requêtes complémentaires*

25) Nom, salaire, commission, salaire+commission de tous les employés.

26) Même requête que la requête n° 16, en faisant figurer tous les employés (y compris ceux qui n'ont pas de manager, donc).

27) Indiquer pour chaque employé sa catégorie de salaire, telle qu'elle est référencée dans la table SALGRADE de l'utilisateur SCOTT.

28) Mettre à jour le salaire de tous les employés en suivant la règle suivante :

$$nouveau\_salaire = (6 - niveau\_hiérarchique) \times 50 \times ancienneté\_en\_mois\ .$$



**CORRECTION**

**Création de la base de données, contraintes d'intégrité**

1)
```
CREATE TABLE DEPT( DEPTNO NUMBER(2),
                   DNAME CHAR(20),
                   LOC CHAR(20),
                   CONSTRAINT DEPT_CLEP PRIMARY KEY (DEPTNO),
                   CONSTRAINT DEPT_DOM1 CHECK (DNAME IN
                    ('ACCOUNTING','RESEARCH','SALES','OPERATIONS')));
```

2)
```
INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW-YORK');
INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');
INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');
INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');
```

3)
```
CREATE TABLE EMP AS SELECT * FROM SCOTT.EMP;
```

4)
```
COMMIT;
```

5)
```
INSERT INTO EMP VALUES (7369, 'Bidon', NULL, NULL, NULL, NULL, NULL, NULL);
```

6)
```
ROLLBACK;
```

7)
```
ALTER TABLE EMP ADD CONSTRAINT EMP_CLEP PRIMARY KEY(EMPNO);
ALTER TABLE EMP ADD CONSTRAINT EMP_CLET_DEPTNO FOREIGN KEY(DEPTNO)
                                                 REFERENCES DEPT(DEPTNO);
ALTER TABLE EMP ADD CONSTRAINT EMP_CLET_MGR FOREIGN KEY(MGR)
                                           REFERENCES EMP(EMPNO);
```

8)
```
INSERT INTO EMP VALUES
(7657, 'WILSON', 'MANAGER', 7839, '17/11/91', 3500.00, 600.00, 10);
```

9)
```
COMMIT;
```

**Mise à jour de la base de données**

1)
```
UPDATE DEPT SET LOC='PITTSBURGH' WHERE DNAME='SALES';
```

2)
```
UPDATE EMP SET SAL=SAL*1.1 WHERE COMM>0.5*SAL;
```

3)
```
UPDATE EMP
SET COMM=(SELECT AVG(COMM) FROM EMP)
WHERE HIREDATE<'01/01/82'
AND COMM IS NULL;
```

4)
```
ROLLBACK;
```

5)
```
DELETE FROM DEPT WHERE DEPTNO=20;
```

**Interrogation de la base de données**

1)
```
SELECT ENAME, SAL, COMM, SAL+COMM FROM EMP WHERE JOB='SALESMAN';
```

2)
```
SELECT ENAME FROM EMP ORDER BY COMM/SAL DESC;
```

3)
```
SELECT ENAME FROM EMP WHERE COMM<.25*SAL;
```

4)
```
SELECT COUNT(EMPNO) FROM EMP WHERE DEPTNO=10;
```

5)
```
SELECT COUNT(EMPNO) FROM EMP WHERE COMM IS NOT NULL;
```

6)
```
SELECT COUNT(DISTINCT JOB) FROM EMP;
```

7)
```
SELECT JOB, AVG(SAL) FROM EMP GROUP BY JOB;
```

8)
```
SELECT SUM(SAL) FROM EMP, DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO
                                AND DNAME='SALES';
```

9)
```
SELECT ENAME, DNAME FROM EMP, DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

10)
```
SELECT ENAME, JOB, SAL FROM EMP WHERE SAL = (SELECT MAX(SAL) FROM EMP);
```

11)
```
SELECT ENAME FROM EMP WHERE SAL>(SELECT SAL FROM EMP WHERE ENAME='JONES');
```

12)
```
SELECT ENAME FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='JONES')
                        AND ENAME<>'JONES';
```

13)
```
SELECT ENAME FROM EMP WHERE MGR=(SELECT MGR FROM EMP WHERE ENAME='CLARK')
                        AND ENAME<>'CLARK';
```

14)
```
SELECT ENAME FROM EMP WHERE (JOB, MGR) IN
  (SELECT JOB, MGR FROM EMP WHERE ENAME='TURNER')
  AND ENAME<>'TURNER';
```

15)
```
SELECT ENAME FROM EMP WHERE HIREDATE < ALL
  (SELECT HIREDATE FROM EMP WHERE DEPTNO=10);
```

16)
```
SELECT SUBALTERNE.ENAME, SUPERIEUR.ENAME FROM EMP SUBALTERNE, EMP SUPERIEUR
WHERE SUBALTERNE.MGR=SUPERIEUR.EMPNO;
```

17)
```
SELECT SUB.ENAME FROM EMP SUB, EMP SUP WHERE SUB.MGR=SUP.EMPNO
                                       AND SUB.DEPTNO<>SUP.DEPTNO;
```

18)
```
SELECT LEVEL, EMPNO, ENAME, JOB, MGR FROM EMP
CONNECT BY MGR = PRIOR EMPNO
START WITH MGR IS NULL
ORDER BY LEVEL;
```

19)
```
SELECT LEVEL, ENAME FROM EMP
WHERE ENAME <> 'JONES'
CONNECT BY MGR = PRIOR EMPNO
START WITH ENAME = 'JONES'
ORDER BY LEVEL;
```



20)
```
SELECT LEVEL, AVG(SAL) FROM EMP
CONNECT BY MGR = PRIOR EMPNO
START WITH MGR IS NULL
GROUP BY LEVEL;
```

21)
```
SELECT ENAME FROM EMP
WHERE ENAME <> 'JONES' AND ENAME <> 'SCOTT'
CONNECT BY MGR = PRIOR EMPNO
START WITH ENAME = 'JONES';
```

22)
```
SELECT ENAME FROM EMP
WHERE ENAME <> 'JONES'
CONNECT BY MGR = PRIOR EMPNO AND ENAME <> 'SCOTT'
START WITH ENAME = 'JONES';
```

23)
```
SELECT LPAD(' ', 3*LEVEL, '-') || ENAME FROM EMP
CONNECT BY MGR = PRIOR EMPNO
START WITH MGR IS NULL
ORDER BY LEVEL;
```

24)
```
SELECT LEVEL || LPAD(' ', 3*LEVEL, '-') || ENAME FROM EMP
CONNECT BY MGR = PRIOR EMPNO
START WITH MGR IS NULL
ORDER BY LEVEL;
```

25)
```
SELECT ENAME, SAL, COMM, SAL+NVL(COMM, 0) FROM EMP;
```

26)
```
SELECT SUBALTERNE.ENAME, SUPERIEUR.ENAME
FROM EMP SUBALTERNE, EMP SUPERIEUR
WHERE SUBALTERNE.MGR=SUPERIEUR.EMPNO
UNION
SELECT ENAME, NULL
FROM EMP
WHERE MGR IS NULL;
--
SELECT SUB.ENAME SUBALTERNE,
       (SELECT SUP.ENAME FROM EMP SUP
       WHERE SUB.MGR = SUP.EMPNO) SUPERIEUR
FROM EMP SUB;
```

27)
```
SELECT ENAME, GRADE FROM EMP, SCOTT.SALGRADE
WHERE SAL BETWEEN LOSAL AND HISAL;
```

28)
```
UPDATE EMP E1
SET SAL = (6 - (SELECT LEVEL FROM EMP E2
      WHERE E1.EMPNO = E2.EMPNO
      CONNECT BY MGR = PRIOR EMPNO
      START WITH MGR IS NULL)) * 50 * MONTHS_BETWEEN(SYSDATE, HIREDATE);
```