

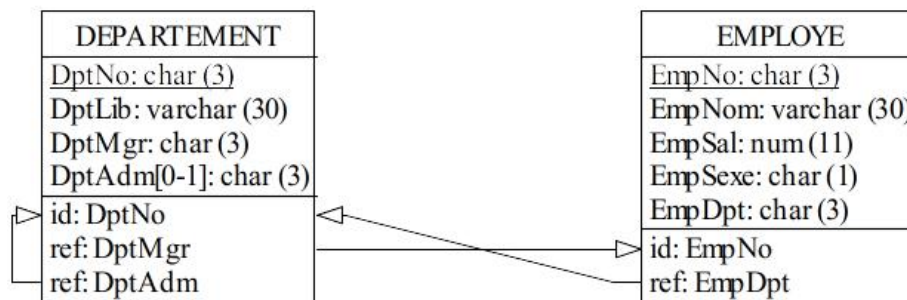
DON2 – Laboratoires DON2

TD7 – Agrégation

Exercice 1

Algèbre relationnelle

Voici un schéma conceptuel de base de données relative à la gestion des employés et départements d'une entreprise.



Les données de ces deux tables sont dans les fichiers `employe.csv` et `departement.csv`.

Donnez les données correspondantes aux expressions d'algèbre relationnelle suivantes, **attention**, certaines expressions sont erronées.

1. $\gamma_{\{empdpt; COUNT(*)\}}(employe)$

COUNT(*)
5
6
2

L'opérateur γ retourne une *relation*, donc seules les valeurs distinctes y apparaissent.

Note : L'expression

$$\gamma_{\{empdpt; empdpt, COUNT(*)\}}(employe)$$

aurait quant à elle donné

empdpt	COUNT(*)
D21	5
C01	6
A00	2
E11	6
D11	6
E21	5

2. $\gamma_{\{dptno; COUNT(dptadm)\}}(departement)$

COUNT(dptadm)
1
0

Les étapes du raisonnement :

- (a) On regroupe les lignes de la table departement selon l'identifiant primaire dptno, donc il y a un groupe par département.
- (b) Ensuite, on compte pour chaque groupe le nombre de valeurs (non nécessairement distinctes) pour dptadm.
Il faut se rappeler que les fonctions agrégatives (telles que COUNT) ne comptent pas les valeurs NULL.
Comme il n'y a qu'une seule ligne par groupe, le résultat du comptage est donc 0 ou 1 selon que l'attribut dptadm y est NULL ou pas.
- (c) Enfin, on ne garde que les lignes distinctes.

3. $\gamma_{\{dptno:dptlib, COUNT(dptmgr)\}}(departement)$

Cette expression est erronée. « Il est impossible de demander le détail d'un groupe. » De manière générale, on se conformera à la règle suivante : dans la *liste d'expressions affichées* ne peuvent figurer que

- (a) des attributs apparaissant dans la *liste d'attributs pour le groupement*,
- (b) des fonctions agrégatives d'attributs quelconques.

Ici dptlib apparaît directement dans la liste d'expressions affichées mais pas dans la liste d'attributs pour le groupement, ce qui n'est pas autorisé. Par contre, dptmgr apparaît dans la liste d'expressions affichées mais pas dans la liste d'attributs pour le groupement, mais ce n'est pas un problème car l'attribut apparaît comme argument de la fonction agrégative COUNT.

Note concernant la traduction en SQL : Jusqu'à SQL-92, cette règle faisait partie du standard SQL. Depuis SQL :1999, des attributs non mentionnés dans la liste d'attributs pour le groupement sont acceptés dans la liste d'expressions affichées *si ils dépendent fonctionnellement* d'attributs qui sont mentionnés dans la liste d'attributs pour le groupement. Par exemple, ici dptlib dépend fonctionnellement de dptno puisque dptno est un identifiant primaire, donc la requête SQL correspondant à l'expression ci-dessus serait acceptée dans les standards SQL :1999 et suivants. Cependant, dans Oracle Apex la règle énoncée plus haut s'applique strictement et une requête du type `SELECT dptlib, COUNT(dptmgr) FROM departement GROUP BY dptno` ne sera pas acceptée.

4. $\gamma_{\{empdpt, empsexe:empdpt, empsexe, MAX(empsal)\}}(\sigma_{\{empno \leq 100\}}(employe))$

empdpt	empsexe	MAX(empsal)
C01	M	110300
C01	F	110200
E11	M	100800
E11	F	100200
D21	M	140200
E21	M	104000

5. $\gamma_{\{COUNT(DISTINCT dptmgr)\}}(departement)$

COUNT(DISTINCT dptmgr)
8

L'agrégation ne donne qu'un seul groupe correspondant à toute la table puisqu'il n'y a pas d'attribut sélectionné pour le regroupement (pas d'attribut avant le ';').

6. $\gamma_{\{dptno, COUNT(*)\}}((departement) \bowtie_{dptno=empdpt} (employe))$

Cet expression est erronée. « Il est impossible de demander le détail d'un groupe. »
Puisqu'il n'y pas d'attribut pour le regroupement, le seul groupe constitué est la relation obtenue suite à la jointure. On ne peut donc pas demander à ce groupe un seul dptno qui le caractériserait.

Voir aussi l'explication à l'exercice 3.

7. $\gamma_{\{dptmgr:dptmgr, COUNT(*)\}}((\sigma_{\{dptadm='A00'\}}(departement)) \bowtie_{dptno=empdpt} (employe))$

dptmgr	COUNT(*)
030	6

La sélection des dptadm = 'A00' donne les départements B01 et C01 mais seul C01 a des employés. La jointure ne retient donc que ce département dirigé par 030 avec 6 employés.

8. $\gamma_{\{dptmgr:dptmgr, COUNT(*)\}}((departement) \bowtie_{dptno=empdpt} (\sigma_{\{empsexe='F'\}}(employe)))$

dptmgr	COUNT(*)
030	1
340	1
060	2
070	1
100	1

9. $\gamma_{\{dptmgr:dptmgr, COUNT(*)\}}((departement) \bowtie_{dptno=empdpt} (\pi_{\{empsexe\}}(employe)))$

Cet expression est erronée. L'exécution de cet expression commence par la projection sur empsexe de la table employe. Cette projection retourne une relation avec un seul attribut empsexe. Il est donc impossible de faire la jointure sur cette relation puisque cette jointure demande l'existence de l'attribut empdpt dans le résultat de la projection.

10. $\sigma_{\{COUNT(*)>3\}}(\gamma_{\{dptmgr:dptmgr, COUNT(*)\}}((\sigma_{\{dptadm='A00'\}}(departement)) \bowtie_{dptno=empdpt} (employe)))$

dptmgr	COUNT(*)
030	6

Exercice 2 SQL

Sur le schéma de la question précédente répondez aux questions suivantes, par une requête SQL.

1. Donnez la liste des employés (empno, empnom, empsexe, dptlib, dptmgr) en mettant en premier lieu les femmes puis les hommes et en les triant en majeur sur le libellé du département et en mineur sur leur nom.

```
SELECT empno, empnom, empsexe, dptlib, dptmgr
FROM employe
JOIN departement ON dptno = empDpt
ORDER BY empsexe, dptlib, empnom ;
```

2. Donnez la masse salariale par sexe de l'entreprise.

```
SELECT empsexe, SUM(empsal)
FROM employe
GROUP BY empsexe ;
```

3. Donnez la liste des masses salariales par département ordonnée sur le libellé des départements.

```
SELECT dptno, dptlib, SUM(empsal)
FROM employe
JOIN departement ON dptno = empDpt
GROUP BY dptno, dptlib
ORDER BY dptlib ;
```

4. Affichez le nombre de femmes employées par département.

```
SELECT dptno, dptlib, COUNT(empsexe)
FROM employe
WHERE empsexe = ' F '
JOIN departement ON dptno = empDpt
GROUP BY dptno, dptlib ;
```

5. Affichez les libellés des départements dont la moyenne des salaires dépasse 90 000 unités.

```
SELECT dptlib, AVG(empsal)
FROM employe
JOIN departement ON dptno = empDpt
GROUP BY dptno, dptlib
HAVING AVG(empsal) > 90000 ;
```

6. Affichez les départements ayant plus de 3 employés masculins, ordonnés par ordre décroissant de nombre d'employés.

```
SELECT dptno, dptlib, empsexe, COUNT(*)
FROM employe
JOIN departement ON dptno = empDpt
WHERE empsexe = ' M '
GROUP BY dptno, dptlib, empsexe
HAVING COUNT(*) > 3
ORDER BY COUNT(*) DESC ;
```

7. Affichez le nombre de départements administrés par un autre département.

```
SELECT COUNT(*)
FROM departement
WHERE dptadm IS NOT null ;
ou
```

```
SELECT COUNT(dptadm)  
FROM departement ;
```

8. Affichez le nombre de départements administrant un autre département.

```
SELECT COUNT(DISTINCT dptadm)  
FROM departement ;
```

9. Affichez la liste ordonnée des noms des managers qui dirigent plus de 3 employés.

```
SELECT mgr.empnom, COUNT(*)  
FROM employe  
JOIN departement ON dptno = employe.empDpt  
JOIN employe mgr ON mgr.empno = dptmgr  
GROUP BY mgr.empno, mgr.empnom  
HAVING COUNT(*) > 3  
ORDER BY mgr.empnom ;
```

10. Affichez le plus grand, le plus petit et la moyenne des salaires par département sans prendre en compte les managers.

Attention ici, nous utilisons un select imbriqué.

```
SELECT MAX(empsal), MIN(empsal), AVG(empsal)  
FROM employe  
JOIN departement ON empdpt = dptno  
WHERE empno NOT IN (SELECT dptmgr  
FROM departement) ;
```