



Module M4

Base de données

Chapitre 4

SQL

Calcul et Regroupement

Auteurs : Véronique Laine / Laura Monceaux



Ce chapitre présente les différentes actions que l'on peut réaliser sur un ensemble de tuples, notamment les calculs.

4.1 Calcul sur un ensemble de valeurs de plusieurs tuples

Il existe dans le langage SQL un certain nombre de fonctions de calculs permettant de réaliser des calculs sur un ensemble de tuples :

- ***SUM(att)*** calcule la somme des valeurs de l'attribut ***att*** pour l'ensemble des tuples sélectionnés
- ***AVG(att)*** calcule la moyenne des valeurs de l'attribut ***att*** pour l'ensemble des tuples sélectionnés
- ***MIN(att)*** retourne la valeur minimale de l'attribut ***att*** pour l'ensemble des tuples sélectionnés
- ***MAX(att)*** retourne la valeur maximale de l'attribut ***att*** pour l'ensemble des tuples sélectionnés
- ***COUNT(att)*** compte le nombre de valeurs de l'attribut ***att*** pour l'ensemble des tuples sélectionnés
- ***COUNT(*)*** compte le nombre de tuples pour l'ensemble des tuples sélectionnés
- ***COUNT(distinct att)*** compte le nombre de valeurs différentes de l'attribut ***att*** pour l'ensemble des tuples sélectionnés
- etc.

Les fonctions définies ci dessus sont les fonctions les plus courantes mais il en existe bien d'autres.

Exemple : si l'on désire la moyenne de la note de cc pour le module M1 :

```
SELECT AVG(notecc)
FROM Notes
WHERE codemat='M1';
```

La moyenne des notes de cc se fera sur l'ensemble des tuples sélectionnés dans la table Notes, c'est à dire correspondant aux étudiants inscrits dans le module M1.

Exemple : si l'on désire connaître le nombre de modules de notre base – on calcule le nombre de valeurs différentes pour l'attribut codemat dans la table Notes :

```
SELECT COUNT(distinct codemat)
FROM Notes;
```

On peut également faire des calculs utilisant les opérateurs arithmétiques, dans ce cas, le calcul se fait pour chaque tuple.

Exemple : si l'on désire la moyenne du module M1 :

```
SELECT AVG((notecc+noteex)/2)
FROM Notes
WHERE codemat='M1';
```

Pour chaque tuple sélectionné, on va calculer la moyenne de l'étudiant (notecc+notetp)/2 puis

on fera ensuite la moyenne de toutes les moyennes des étudiants.

Avec des requêtes simples, les fonctions de calcul ne peuvent être utilisées que dans la clause SELECT, puisque seule clause dans laquelle on peut faire des calculs sur un ensemble de tuples. En effet, nous rappelons que dans la clause WHERE les conditions portent sur UN et UN seul tuple.

Pourtant, si l'on veut calculer la moyenne de tous les modules de notre base, on aimerait regrouper tous les tuples concernant un même module et créer des sous groupes sur lesquels on calculerait la moyenne. Pour se faire nous allons introduire la notion de regroupement.

4.1 Regroupement

Si l'on veut faire des actions sur des sous ensembles (groupes) de tuples, il faut partitionner la table en groupes.

- La clause GROUP BY permet de définir un groupe (sous ensemble)
 - on regroupe les tuples selon un certain nombre d'attributs, représentatifs du groupe que l'on veut former
 - **attention** faire un groupe a un impact sur la clause SELECT : après avoir fait un regroupement, la requête ne peut retourner que des attributs spécifiés dans le GROUP BY soit des calculs sur chacun des groupes formés
 - ainsi tout attribut de la clause SELECT devra être présent dans la clause GROUP BY
- La clause HAVING permet de réaliser des conditions sur les groupes formés par le GROUP BY.

La syntaxe de la commande SELECT avec un regroupement sera ainsi la suivante:

```
SELECT ....
FROM ....
[WHERE prédicats]
GROUP BY liste_attributs
[HAVING condition_groupe]
[ORDER BY] ...;
```

Exemple : si l'on désire la moyenne du CC de chaque module, on va d'abord regrouper les tuples concernant la même matière (codemat), puis faire le calcul sur chacun de ces groupes

NumEtu	codemat	noteCC	noteEX
123456	M1	12	14
234567	M1	17	14
457832	M1	14	7
987655	M1	13	8
234567	M2	10	8
457832	M2	11	12
345667	M3	5	7
135790	M3	17	14
123456	M4	4	10
345667	M4	12	12

Groupe	codemat	Moyenne
Groupe codemat= 'M1'	M1	14
Groupe codemat= 'M2'	M2	10,5
Groupe codemat= 'M3'	M3	11
Groupe codemat= 'M4'	M4	8

```
SELECT codemat, AVG(notecc) MoyenneCC  
FROM Notes  
GROUP BY codemat;
```

Ainsi pour cette requête, on obtient le résultat suivant :

<u>codemat</u>	<u>MoyenneCC</u>
M1	14
M2	10,5
M3	11
M4	8

Exemple : si l'on désire connaître les modules qui ont une moyenne de CC supérieure à 10. On reprend la requête précédente, et une fois les groupes réalisés, on ne conserve que les modules qui ont une moyenne de CC supérieure ou égale à 10 dans la clause HAVING (clause permettant de faire des sélections sur les groupes).

```
SELECT codemat, AVG(notecc) MoyenneCC  
FROM Notes  
GROUP BY codemat  
HAVING AVG(notecc) >= 10;
```

En conclusion, les fonctions de calcul en SQL sont utilisables :

- soit sur l'ensemble de la table et donc présentes dans la clause SELECT,
- soit sur des groupes de tuples que l'on aura au préalable formé par la clause GROUP BY et donc les fonctions de calculs peuvent être utilisées soit dans la clause SELECT soit dans la clause HAVING et le calcul se fait pour chacun des groupes.