



Bases de données : (encore) un peu plus loin dans SQL

Karën Fort (repris d'Alice Millour)

karen.fort@sorbonne-universite.fr



Sources d'inspiration

- ▶ il s'agit du cours d'Alice Millour (2020)
- ▶ <https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1966846-fonctions-dagregation>
- ▶ <https://openclassrooms.com/fr/courses/4449026-initiez-vous-a-lalgebre-relationnelle-avec-le-langage-sql/4558491-ameliorez-vos-agregations-grace-a-having>
- ▶ <https://sql.sh/cours/alias>

Retours sur le TD

Agréger les résultats

Filtrer les résultats après une agrégation


Les alias

Pour finir

Erreurs les plus courantes : typage

SQL est sensible au type des données

Erreur

Requête SQL : 

```
SELECT * FROM `Etudiants` WHERE genre=M LIMIT 0, 25
```

MySQL a répondu : 

```
#1054 - Champ 'M' inconnu dans where clause
```

Erreurs les plus courantes : DISTINCT

DISTINCT porte sur **tous** les éléments à afficher

```
SELECT DISTINCT auteur, titre FROM Livres ORDER By auteur
```

Profilage [Éditer en ligne] [Éditer] [Explic

Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: Trier

+ Options

	auteur ▲ 1	titre
<input type="checkbox"/> Éditer Copier Supprimer	Anonyme	La loi c'est pas moi
<input type="checkbox"/> Éditer Copier Supprimer	Baleine	Autant en emporte l'event
<input type="checkbox"/> Éditer Copier Supprimer	Carmin	L'informatique pour les 1ere S curieux
<input type="checkbox"/> Éditer Copier Supprimer	Corneille	Le Cid
<input type="checkbox"/> Éditer Copier Supprimer	Godoc	SQL les fondamentaux du langage
<input type="checkbox"/> Éditer Copier Supprimer	Larsson	La fille qui revait d'un bidon d'essence et d'une ...
<input type="checkbox"/> Éditer Copier Supprimer	Larsson	La reine dans le palais des courants d'air
<input type="checkbox"/> Éditer Copier Supprimer	Larsson	Les hommes qui n'aimaient pas les femmes

Erreurs les plus courantes : JOIN

la jointure doit être faite **avant** les restrictions

Erreur

Requête SQL : ?

```
SELECT prenom FROM Emprunts WHERE date_ret IS NULL INNER JOIN Etudiants ON Emprunts.num_et=Etudiants.num_et LIMIT 0, 25
```

MySQL a répondu : ?

```
#1064 - Erreur de syntaxe près de 'INNER JOIN Etudiants ON Emprunts.num_et=Etudiants.num_et LIMIT 0, 25' à la ligne 1
```

Exercice 1

1. `SELECT filiere FROM Programme` (ou, sans doublon, `SELECT DISTINCT filiere FROM Programme`)
2. `SELECT auteur FROM Livres` (ou trié, sans doublon : `SELECT DISTINCT auteur FROM Livres ORDER BY auteur`)
3. `SELECT titre FROM Livres WHERE nb_pages=150`
4. `SELECT nom, prenom FROM Etudiants WHERE prenom="Marie"`
5. `SELECT titre FROM Emprunts WHERE date_ret IS NULL`
6. `SELECT num_et FROM Emprunts WHERE date_ret IS NULL`

NATURAL JOIN

une écriture simplifiée d'un **INNER JOIN**, dans le cas où les colonnes ont :

- ▶ le même nom
- ▶ le même type

Exemple

7 `SELECT prenom FROM Emprunts NATURAL JOIN Etudiants WHERE date_ret IS NULL`

vs

`SELECT prenom FROM Emprunts INNER JOIN Etudiants ON Emprunts.num_et=Etudiants.num_et WHERE date_ret IS NULL`

8 `SELECT titre, date_emp FROM Livres NATURAL JOIN Emprunts WHERE nb_pages=150`

Dans ce cours

- ▶ aller plus loin dans le `SELECT` : compter les éléments, trouver le minimum ou calculer la moyenne d'une colonne, etc.
- ▶ grouper les lignes qui partagent une même valeur pour faire des statistiques plus fines sur les données

Retours sur le TD

Agréger les résultats

Compter en SQL

GROUP BY

Filtrer les résultats après une agrégation

Les alias

Pour finir

Retours sur le TD

Agréger les résultats

Compter en SQL

GROUP BY

Filtrer les résultats après une agrégation

Les alias

Pour finir

L'agrégation

- ▶ Requête sans agrégation : renvoie une **liste** = liste des lignes sélectionnées par la requête
- ▶ Requête avec agrégation : renvoie une **valeur** = résultat d'une fonction d'agrégation appliquée à une colonne

Fonction d'agrégation

permet (principalement) de calculer des statistiques sur les données :

- ▶ **COUNT**(colonne) = compter les éléments **not NULL**
- ▶ **MIN**(colonne), **MAX**(colonne) = trouver les éléments minimum et maximum d'une colonne
- ▶ **SUM**(colonne) = sommer les éléments d'une colonne
- ▶ **AVG**(colonne) = calculer la valeur moyenne des éléments d'une colonne

Attention : il ne doit pas y avoir d'espace entre la commande SQL et la colonne entre parenthèse

(rappel) requête sans agrégation

Dans la BD Enquete

```
SELECT filiere FROM Diplomes
```

✓ Affichage des lignes 0 - 24 (total de 43, traitement en 0.0007 seconde(s).)

```
select filiere from Diplomes
```

Profilage [Éditer en ligne]

1 > >> | Tout afficher | Nombre de lignes : 25 | Filtrer le:

+ Options

	filiere
<input type="checkbox"/> Éditer Copier Supprimer	Informatique
<input type="checkbox"/> Éditer Copier Supprimer	Informatique
<input type="checkbox"/> Éditer Copier Supprimer	Informatique
<input type="checkbox"/> Éditer Copier Supprimer	Informatique
<input type="checkbox"/> Éditer Copier Supprimer	Sciences pour l'ingenieur
<input type="checkbox"/> Éditer Copier Supprimer	Sciences pour l'ingenieur
<input type="checkbox"/> Éditer Copier Supprimer	Sciences pour l'ingenieur
<input type="checkbox"/> Éditer Copier Supprimer	Sciences pour l'ingenieur
<input type="checkbox"/> Éditer Copier Supprimer	Sciences de l'education
<input type="checkbox"/> Éditer Copier Supprimer	Sciences de l'education
<input type="checkbox"/> Éditer Copier Supprimer	Sciences de l'education

Requête avec agrégation : COUNT(colonne)

Dans la BD Enquête

SELECT COUNT(filiere) FROM Diplomes

43 filières

A screenshot of a database query interface. At the top, a green status bar indicates 'Affichage des lignes 0 - 0 (total de 1,1)'. Below it, the SQL query is displayed: `select count(filiere) from Diplomes`. Underneath the query, there is a control bar with a checkbox for 'Tout afficher' and the text 'Nombre de lignes :'. Below this, there is a '+ Options' link and a table header `count(filiere)`. The table contains a single row with the value 43.

<code>count(filiere)</code>
43

(avec DISTINCT) 4 filières différentes

A screenshot of a database query interface. At the top, a green status bar indicates 'Affichage des lignes 0 - 0 (total de 1,1)'. Below it, the SQL query is displayed: `select count(DISTINCT filiere) from Diplomes`. Underneath the query, there is a control bar with a checkbox for 'Tout afficher' and the text 'Nombre de lignes :'. Below this, there is a '+ Options' link and a table header `count(DISTINCT filiere)`. The table contains a single row with the value 4.

<code>count(DISTINCT filiere)</code>
4

Requête avec agrégation : autres exemples

Dans la BD Enquête

(à tester vous-même)

- ▶ `SELECT MIN(nb_pages) FROM Livres`
- ▶ `SELECT MAX(nb_pages) FROM Livres`
- ▶ `SELECT SUM(nb_pages) FROM Livres`
- ▶ `SELECT AVG(nb_pages) FROM Livres`

Retours sur le TD

Agréger les résultats

Compter en SQL

GROUP BY

Filtrer les résultats après une agrégation

Les alias

Pour finir

La clause GROUP BY

GROUP BY colonne

groupe les résultats d'une requête qui partagent la même valeur dans la colonne spécifiée

`SELECT filiere FROM Diplomes GROUP BY filiere`

A screenshot of a database query execution interface. At the top, a green status bar indicates "Affichage des lignes 0 - 3 (total de 4), traitement en 0.0008 sec". Below this, the SQL query is displayed: `select filiere from Diplomes group by filiere`. A checkbox labeled "Profilage" is visible. Below the query, there are controls for "Tout afficher" (checked), "Nombre de lignes : 25", and "Filtrer les li". A section titled "+ Options" shows a table with a header "filiere" and four rows of results, each with edit, copy, and delete icons.

	filiere
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Informatique
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Sciences de l'education
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Sciences du langage
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Sciences pour l'ingenieur

Faire des statistiques sur une table

combiner fonctions d'agrégation et `GROUP BY`

Compter les étudiants

- ▶ nombre de diplômés (43)

```
SELECT COUNT(num_et) FROM Diplomes
```

- ▶ nombre de diplômés *par filière* :

```
SELECT COUNT(num_et) FROM Diplomes GROUP BY filière
```

+ Options

count(num_et)
4
17
13
9

la fonction d'agrégation s'applique à chaque **groupe de résultats**

Faire des statistiques sur une table (2)

combiner fonctions d'agrégation et `GROUP BY`

Compter les coureurs (base : BDD_L3_ARMENTA)

- ▶ nombre de participations (16)

```
SELECT COUNT(NumeroCoureur) FROM Participer
```

- ▶ nombre de participations *par coureur* :

```
SELECT COUNT(NumeroCoureur) FROM Participer GROUP BY NumeroCoureur
```

<code>count(NumeroCoureur)</code>
1
4
1
1
3
2
1
3

Rendre les statistiques lisibles

Champs à sélectionner

```
SELECT filiere, COUNT(num_et)
FROM Diplomes
GROUP BY filiere
```

filiere	count(num_et)
Informatique	4
Sciences de l'education	17
Sciences du langage	13
Sciences pour l'ingenieur	9

dans une clause **SELECT** comprenant un **GROUP BY**, on ne peut avoir que deux types d'éléments :

- ▶ un élément présent dans la clause **GROUP BY** (ou qui en dépend fonctionnellement) : ici **filiere**
- ▶ une fonction d'agrégation : ici **COUNT(num_et)**

Rendre les statistiques lisibles

Champs à sélectionner

```
SELECT Coureur.NomCoureur, COUNT(Participer.NumeroCoureur)
FROM Participer
INNER JOIN Coureur ON Participer.NumeroCoureur = Coureur.NumeroCoureur
GROUP BY Coureur.NumeroCoureur
```

NomCoureur	count(Coureur.NumeroCoureur)
Sébastien Reichenbach	1
Peter Sagan	4
Romain Bardet	1
Stefan Küng	1
Egan Bernal	3
Kasper Asgreen	2
Jonathan Castroviejo	1
Geraint Thomas	3

Erreur classique

Champ mal sélectionné

Erreur

Requête SQL :

```
select Coureur.NomCoureur, Participer.TempsRéalise, count(Coureur.NumeroCoureur) from Participer  
join Coureur on Participer.NumeroCoureur=Coureur.NumeroCoureur  
group by Coureur.numeroCoureur LIMIT 0, 25
```

MySQL a répondu :

```
#1055 - Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column  
'BDD_L3_ARMENTA.Participer.TempsRéalise' which is not functionally dependent on columns in GROUP BY clause; this  
is incompatible with sql_mode=only_full_group_by
```

le champ Participer.TempsRéalise :

- ▶ ne dépend pas fonctionnellement de Coureur.NumeroCoureur
- ▶ n'est pas une fonction d'agrégation

GROUP BY sur plusieurs champs

- ▶ nombre de diplômés par filière et *par niveau* :

```
SELECT filiere, niveau, COUNT(num_et)
FROM Diplomes
GROUP BY filiere, niveau
```

filiere	niveau	count(num_et)
Informatique	L2	2
Informatique	L3	2
Sciences de l'education	L1	9
Sciences de l'education	L2	3
Sciences de l'education	L3	5
Sciences du langage	L1	8
Sciences du langage	L2	1
Sciences du langage	L3	4
Sciences pour l'ingenieur	L2	5
Sciences pour l'ingenieur	L3	4

GROUP BY, exemples

(à tester vous-même)

- ▶ durée moyenne des cours par filière :

```
SELECT filiere, AVG(nb_heures)
FROM Programme
GROUP BY filiere
```

- ▶ première date d'obtention de diplôme par filière et par niveau :

```
SELECT filiere, niveau, MIN(date_d)
FROM Diplomes
GROUP BY filiere, niveau
```

Retours sur le TD

Agréger les résultats

Filtrer les résultats après une agrégation

Les alias

Pour finir

Filtrer les résultats : HAVING

HAVING condition

restreint les résultats à ceux qui respectent la **condition**

- ▶ la clause '**WHERE** condition' restreint avant l'agrégation
- ▶ la clause '**HAVING** condition' restreint après l'agrégation

Filtrer les résultats : HAVING

Si on cherche les livres qui ont été empruntés au moins 3 fois depuis le 1^{er} janvier 2012

- ▶ restriction sur la date d'emprunt avec **WHERE**
- ▶ restriction sur le nombre d'emprunts avec **HAVING**

```
SELECT titre, COUNT(date_emp)
FROM Emprunts
WHERE date_emp > "2012-01-01" # restriction sur la date d'emprunt
GROUP BY titre
HAVING COUNT(date_emp) >= 3 # restriction sur le nombre d'emprunts
```

Retours sur le TD

Agréger les résultats

Filtrer les résultats après une agrégation

Les alias

Pour finir

Clarifier les requêtes grâce aux alias

La clause AS

(optionnel) pour changer le nom de colonnes et de tables

- ▶ formater un résultat (alias de colonne) :

```
SELECT num_et AS "Numéro d'étudiant" FROM Etudiants
```

```
select num_et as "Numéro d'étudiant" from Etudiants
```

1 > >> | Tout afficher | Nombre de lignes :

+ Options
← T →

	Numéro d'étudiant	
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer		1
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer		2
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer		3
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer		4
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer		5

Clarifier les requêtes grâce aux alias

La clause AS

(optionnel) pour changer le nom de colonnes et de tables

- ▶ raccourcir les requêtes (alias de table) :

```
SELECT i.num_et, i.annee, i.filiere FROM Inscriptions AS i
```

- ▶ nommer des résultats de fonctions d'agrégation :

```
SELECT num_et, COUNT(annee) AS nb_insc  
FROM Inscriptions  
GROUP BY num_et  
HAVING nb_insc = 1
```


Retours sur le TD

Agréger les résultats

Filtrer les résultats après une agrégation

Les alias

Pour finir

CQFR : Ce Qu'il Faut Retenir

TD



- ▶ maîtrise des fonctions d'agrégation, de **GROUP BY**, de **AS**
- ▶ champs à sélectionner quand on combine fonctions d'agrégation et **GROUP BY**

TD à rendre à la fin du TD, noté

Voir support papier (à rendre également)