BDW - Regroupements SQL

Fabien Duchateau

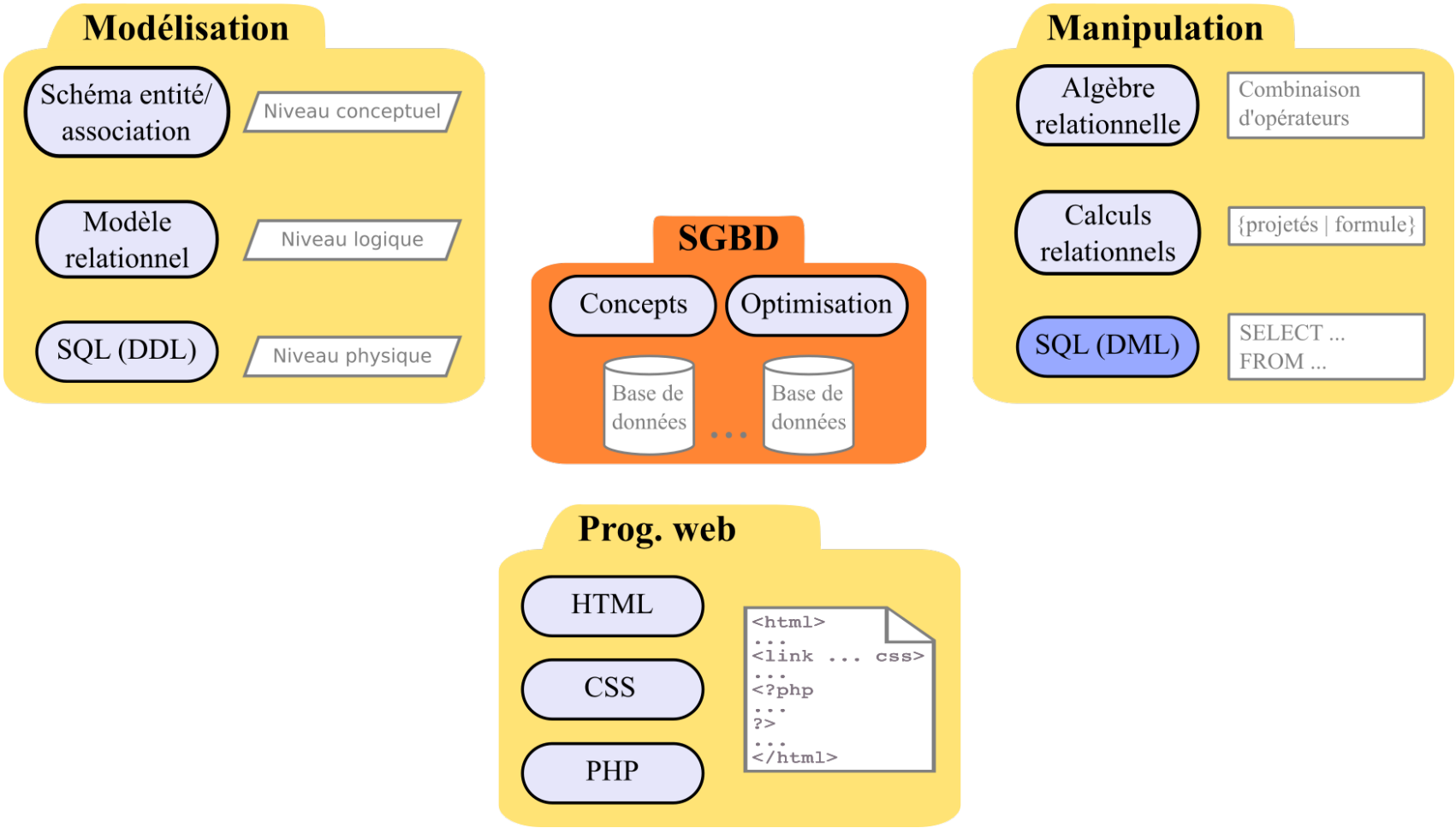
fabien.duchateau [at] univ-lyon1.fr

Université Claude Bernard Lyon 1

2023 - 2024

https://perso.liris.cnrs.fr/fabien.duchateau/BDW/

Positionnement dans BDW



Ces diapositives utilisent le genre féminin (e.g., chercheuse, développeuses) plutôt que l'écriture inclusive (moins accessible, moins concise, et pas totalement inclusive)

Rappel

Syntaxe d'une requête SQL (crochets = option) :

SELECT att_1 [, att_2 [**AS** att'_2], ...]
FROM nom_table_1 [, nom_table_2 [**AS** alias]]
[**WHERE** *condition*]
[**ORDER BY** att_i [, att_j, ...]]
[**LIMIT** n] **;**

Comment calculer ...

- La moyenne des élèves qui ont candidaté en bio ?
- Le nombre de candidatures pour chaque université ?
- Les universités qui ont plus de trois candidatures ?

Plan

Clause GROUP BY

Fonctions d'agrégation

Clause HAVING

Division

Syntaxe

SELECT att_1 [, att_2 [**AS** att'_2], ...]
FROM nom_table_1 [, nom_table_2 [**AS** alias]]
[**WHERE** *condition*]
[**GROUP BY** att_k [, att_l, ...]]
[**ORDER BY** att_i [, att_j, ...]]
[**LIMIT** n] ;

Le GROUP BY, exécuté après le WHERE, indique de procéder à une répartition du résultat en groupes de n-uplets :

- Deux n-uplets sont dans un même groupe s'ils ont les mêmes valeurs pour chaque attribut de regroupement att_k, att_l, ...

https://mariadb.com/kb/en/mariadb/select/#group-by

Exemple de regroupement

```
18 SELECT nomE
19 FROM Élève
20 WHERE idE < 700;
```

nomE
Ana
Bob
Chloe
Damien
Chloe
Eleonore
Ana
Farid

```
22 SELECT nomE
23 FROM Élève
24 WHERE idE < 700
25 GROUP BY nomE;
```

nomE
Ana
Chloe
Damien
Bob
Eleonore
Farid

Conséquences du regroupement

- La requête ne renvoie **qu'un seul n-uplet par groupe**
- Le SELECT et le ORDER BY ne peuvent utiliser que des attributs présents dans le GROUP BY
  - dans un groupe, un attribut de regroupement a la même valeur pour tous les n-uplets, donc utilisable directement
  - les autres attributs ont des valeurs diverses, donc non utilisables directement (quelle valeur utiliser ?)

On peut quand même utiliser les attributs qui ne font pas partie des attributs de regroupement via des fonctions !

Plus exactement, le comportement est indéfini (erreur, résultat incorrect)

Plan

Caractéristiques des fonctions d'agrégation

Fonctions agissant sur un ensemble de valeurs atomiques :

- Compter un nombre d'élèves
- Calculer la moyenne des effectifs

Comment les utiliser ?

- Si aucun GROUP BY n'est spécifié dans la requête, la fonction s'applique à toutes les lignes
- Si un GROUP BY est spécifié dans la requête, la fonction s'applique pour chaque groupe

---

https://mariadb.com/kb/en/mariadb/functions-and-modifiers-for-use-with-group-by/

Caractéristiques des fonctions d'agrégation (2)

- Où utiliser ces fonctions ?
  - Dans le SELECT et dans le ORDER BY
  - **Pas** dans le WHERE (qui a lieu **avant** regroupement)

- L'intérêt de ces fonctions est de permettre l'utilisation des attributs qui ne sont pas des attributs de regroupements

- Le mot clé DISTINCT peut être placé dans la fonction d'agrégation (avant l'attribut) pour ne prendre en compte que les valeurs distinctes

Une dizaine de fonctions, dont COUNT, MIN/MAX, AVG, SUM

Fonction COUNT

COUNT(att) : le nombre de valeurs de l'attribut *att*

COUNT(**DISTINCT** att) : le nombre de valeurs distinctes de l'attribut *att*

- Les valeurs NULL ne sont pas comptées
- * peut remplacer *att*, cela compte alors le nombre de n-uplets

Exemple de COUNT sans regroupement

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Le nombre d'élèves

```
28 SELECT COUNT(*)
29 FROM Élève;
```

ou

```
36 SELECT COUNT(nomE)
37 FROM Élève;
```

count
12

Exemple de COUNT avec DISTINCT

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Le nombre de noms d'élèves distincts

```
40 SELECT COUNT(DISTINCT nomE)
41 FROM Élève;
```

count
10

Déroulement pas à pas d'une requête avec regroupement

Le nombre de candidatures par université, avec tri par nom :

SELECT nomU, **COUNT**(*) **FROM** *C*
GROUP BY nomU **ORDER BY** nomU ;

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

$\Rightarrow$

nomU	count
UJM	6
UCB	6
UJF	4
INSA	3

$\Rightarrow$

nomU	count
INSA	3
UCB	6
UJF	4
UJM	6

Exemple de regroupement sur plusieurs attributs

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Nombre de candidatures par université puis par département pour les universités *INSA* et *UCB*

```
60 SELECT nomU, département, COUNT(*) AS nbC
61 FROM Candidature c
62 WHERE nomU IN ('INSA', 'UCB')
63 GROUP BY nomU, département;
```

nomU	département	nbC
INSA	biologie	1
INSA	informatique	2
UCB	électronique	1
UCB	histoire	2
UCB	informatique	3

Piège à éviter

Quelle requête correspond au "*nombre de candidatures pour chaque élève*" ?

```
66 SELECT nomE, COUNT(*)
67 FROM Élève e INNER JOIN
       Candidature c
68 ON e.idE = c.idE
69 GROUP BY nomE;
```

⇓

nomE	count
Joana	3
Ana	4
Chloe	5
Irene	3
Hector	2
Bob	1
Farid	1

```
72 SELECT e.idE, nomE, COUNT(*)
73 FROM Élève e INNER JOIN
       Candidature c
74 ON e.idE = c.idE
75 GROUP BY e.idE, nomE;
```

⇓

idE	nomE	count
765	Joana	3
123	Ana	4
345	Chloe	4
876	Irene	3
543	Chloe	1
898	Hector	2
234	Bob	1
678	Farid	1

Fonctions MIN et MAX

MIN(att) : la valeur minimale de l'attribut *att*

MAX(att) : la valeur maximale de l'attribut *att*

- Le mot clé DISTINCT n'a pas d'effet avec ces fonctions

Le chat, Geluck

Exemple de MIN/MAX sans regroupement

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

La moyenne la plus élevée

```
78 SELECT MAX(moyenneLycée) AS moyenneMax
79 FROM Élève;
```

moyenneMax
19.5

Exemple de MIN/MAX avec sous-requête

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table Élève

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table Université

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table Candidature

Les universités où ont candidaté les élèves avec la moyenne la plus faible

```
82 SELECT e.idE, nomE, nomU
83 FROM Candidature c INNER JOIN Élève e ON c.
       idE = e.idE
84 WHERE moyenneLycée =
85    (SELECT MIN(moyenneLycée) FROM Élève);
```

idE	nomE	nomU
765	Joana	UCB
765	Joana	UJM
765	Joana	UJM

Un moment de réflexion

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Que modifier pour obtenir tous les élèves avec la moyenne la plus faible et les éventuelles universités où ils/elles ont candidaté ?

⇒ **Remplacer la jointure interne par une jointure externe (à droite)**

idE	nomE	nomU
567	Eleonore	
765	Joana	UCB
765	Joana	UJM
765	Joana	UJM

Fonctions AVG et SUM

AVG(att) : la moyenne des valeurs de l'attribut *att*

SUM(att) : la somme des valeurs de l'attribut *att*

Le chat, Geluck

Exemple de AVG avec regroupement

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table Élève

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table Université

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table Candidature

La moyenne des candidat.e.s par université (avec arrondi)

```
93 SELECT nomU, round(AVG(moyenneLycée))
94 FROM Élève NATURAL JOIN Candidature
95 GROUP BY nomU;
```

nomU	round
UJM	17
UCB	18
UJF	18
INSA	19

Un moment de réflexion

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Que modifier pour obtenir **la ou les** universités dont la moyenne de ses candidat.e.s est la plus faible ?

⇒ **Créer une sous-requête à partir de la requête précédente, et récupérer la moyenne minimale dans la requête principale**

Autres fonctions SQL

- **STD**(att) ou **STDDEV_POP**(att) : l'écart-type de *att*
- **VAR_POP**(att) ou **VARIANCE**(att) : la variance de *att*
- **GROUP_CONCAT**(att) : une concaténation des valeurs de *att* dans une chaîne de caractères
- ...

---

https://mariadb.com/kb/en/mariadb/group_concat/

En résumé

Fonctions liées au regroupement (mais également utilisables dans une requête sans GROUP BY) :

- Comptage avec COUNT
- Bornes avec MIN et MAX
- Statistiques avec SUM, AVG, etc.

```
fabien=# select * from universite ;
 nomu |     ville      | effectif
------+----------------+----------
 UCB  | Lyon           |    15000
 INSA | Lyon           |    36000
 UJF  | Grenoble       |    10000
 UJM  | Saint-Etienne  |    21000
(4 lignes)
```

Démo avec PostgreSQL (scripts SQL en ligne)

Plan

Syntaxe

SELECT att_1 [, att_2 [**AS** att'_2], ...]
FROM nom_table_1 [, nom_table_2 [**AS** alias]]
[**WHERE** *condition*]
[**GROUP BY** att_k [, att_l, ...]]
[**HAVING** *condition_groupe*]
[**ORDER BY** att_i [, att_j, ...]]
[**LIMIT** n] ;

La clause HAVING sélectionne les groupes qui satisfont la condition *condition_groupe*

- Une clause GROUP BY est requise
- Exécutée entre le GROUP BY et le ORDER BY

Condition du HAVING

- Le WHERE ne porte que sur les n-uplets individuels, **avant regroupement**

- La condition du HAVING porte sur les groupes et pas sur les n-uplets individuels :
  - utilisation directe des attributs de regroupement
  - utilisation des autres attributs à travers les fonctions liées au regroupement
  - évaluation booléenne (comme la condition du WHERE)

Déroulement pas à pas d'une requête avec HAVING

Les départements d'université avec au moins 2 candidatures :

SELECT nomU, département, **COUNT**(*) **AS** nbC **FROM** *C*
GROUP BY nomU, département **HAVING COUNT**(*) ≥ 2 **;**

nomU	département
INSA	informatique
UCB	électronique
UCB	informatique
UJM	électronique
INSA	biologie
UJF	bioinformatique
UJM	bioinformatique
UJM	électronique
UJM	informatique
UJF	informatique
UCB	histoire
UCB	histoire
UJM	histoire
UJM	psychologie
UCB	informatique
UJF	biologie
UJF	biologie marine
INSA	informatique
UCB	informatique

ement	décision
atique	O
nique	N
atique	O
atique	O

nomU	département	nbC
INSA	informatique	2
UCB	électronique	1
UCB	informatique	3
UJM	électronique	2
INSA	biologie	1
UJF	bioinformatique	1
UJM	bioinformatique	1
UJM	informatique	1
UJF	informatique	1
UCB	histoire	2
UJM	histoire	1
UJM	psychologie	1
UJF	biologie	1
UJF	biologie marine	1

nomU	département	nbC
INSA	informatique	2
UCB	histoire	2
UCB	informatique	3
UJM	électronique	2

Table CANDIDATURE

Exemple de HAVING

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Nombre de candidatures par décision, avec un minimum de 5 candidatures

```
202 SELECT décision, COUNT(*) AS nb
203 FROM Candidature c
204 GROUP BY décision
205 HAVING COUNT(*) > 5;
```

décision	nb
O	11
N	8

Un moment de réflexion

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Que modifier dans la requête précédente pour prendre en compte de nouvelles valeurs de décision (e.g., 'LA' pour *liste d'attente*) ?

⇒ **Rien ! Le regroupement récupère toutes les valeurs de décision automatiquement**

décision	nb
O	11
N	8
LA	...
...	...

Exemple de HAVING avec sous-requête

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Élèves ayant candidaté dans une université avec au moins 5 disciplines

```
117 SELECT e.idE, nomE, COUNT(*) AS nbC
118 FROM Élève e NATURAL JOIN Candidature c
119 WHERE nomU IN(SELECT nomU FROM Candidature
        GROUP BY nomU
120    HAVING COUNT(DISTINCT département) >= 5)
121 GROUP BY e.idE, nomE;
```

idE	nomE	nbC
123	Ana	1
765	Joana	2
345	Chloe	3

Plan

Retour sur la division

- Pas d'opérateur de division en SQL
- Solutions en utilisant un regroupement, une double négation, des différences, etc.
- Attention : la division avec regroupement ne fonctionne pas toujours !

Requête type pour la division : les universités qui proposent **toutes** les disciplines de la vie (bio...)

Autre façon de comprendre : il n'existe aucune discipline de la vie que l'université ne propose pas

---

http://sqlpro.developpez.com/cours/divrelationnelle/

Illustration d'une division

Les universités avec toutes les disciplines de la vie (bio...)

Dividende

nomU	département
INSA	informatique
UCB	électronique
UJM	électronique
INSA	biologie
UJF	bioinformatique
UJF	bioinformatique
UJM	bioinformatique
UJM	informatique
UJF	informatique
UCB	histoire
UJM	histoire
UJM	psychologie
UJF	biologie
UJF	biologie
UJF	biologie marine
UJF	biologie marine
UCB	informatique

Diviseur

département
bioinformatique
biologie
biologie marine

Résultat

nomU
UJF

Exemple de division avec double négation

Intuition : on cherche une université pour laquelle il n'existe pas une discipline en bio que cette université ne propose pas

Les universités avec toutes les disciplines de la vie (bio...)

```
SELECT DISTINCT nomU
FROM U u
WHERE NOT EXISTS (SELECT * FROM C c1
        WHERE c1.département LIKE 'bio%'
        AND NOT EXISTS (SELECT *
                FROM C c2
                WHERE u.nomU = c2.nomU
                AND c1.département = c2.département));
```

nomU
UJF

Exemple de division avec regroupement

Intuition : on cherche une université qui a un nombre de disciplines en bio égal au nombre total de disciplines en bio

Les universités avec toutes les disciplines de la vie (bio...)

```
SELECT DISTINCT nomU
FROM C WHERE département IN
        (SELECT département FROM C
        WHERE département LIKE 'bio%')
GROUP BY nomU
HAVING COUNT(DISTINCT département) =
        (SELECT COUNT(DISTINCT département)
        FROM C WHERE département LIKE 'bio%') ;
```

nomU
UJF

Division (non) exacte

Division exacte ou entière :

- Pas de reste, toutes les valeurs de la table *dividende* sont dans la table *diviseur*
- Exemple : *les universités avec toutes les disciplines de la vie (bio...) et uniquement celles de la vie*

Division non exacte :

- Avec reste, la table *dividende* peut contenir des valeurs supplémentaires par rapport à la table *diviseur*
- Exemple : *les universités avec toutes les disciplines de la vie (bio...), et éventuellement d'autres disciplines*

Illustration d'une division avec reste

Les universités avec toutes les disciplines de la vie (bio...)

Division exacte ou avec reste ?

Dividende

nomU	département
INSA	informatique
UCB	électronique
UJM	électronique
INSA	biologie
UJF	bioinformatique
UJF	bioinformatique
UJM	bioinformatique
UJM	informatique
UJF	informatique
UJF	informatique
UCB	histoire
UJM	histoire
UJM	psychologie
UJF	biologie
UJF	biologie

Diviseur

département
bioinformatique
biologie
biologie marine

Résultat

nomU
UJF

Reste

département
informatique

Exemple de division avec reste

Les élèves qui ont candidaté dans toutes les universités de Lyon (et d'autres)

```
174 SELECT DISTINCT c1.idE
175 FROM Candidature c1
176 WHERE NOT EXISTS(
177     SELECT *
178     FROM Université u
179     WHERE u.ville = 'Lyon' AND NOT EXISTS(
180           SELECT *
181           FROM Candidature c2
182           WHERE c2.idE = c1.idE AND c2.nomU = u.nomU
183 ));
```

idE	count
898	2
123	3

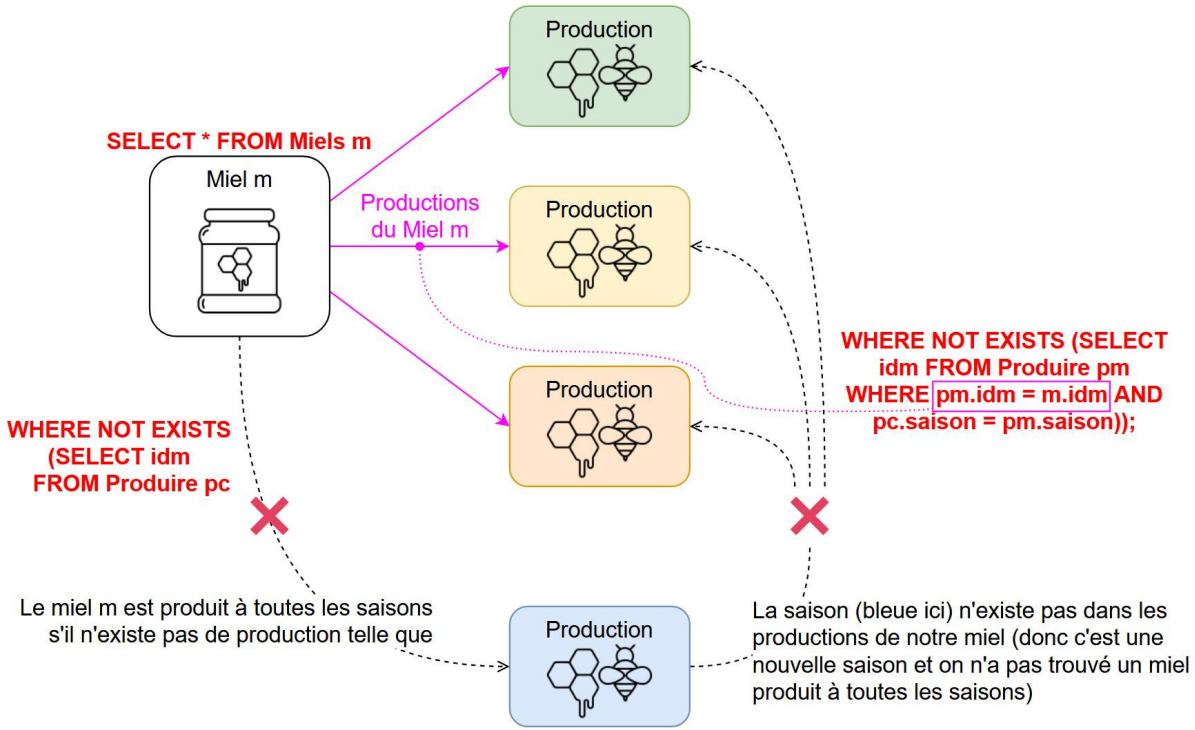
Exemple de division sans reste (exacte)

Les élèves qui ont candidaté dans toutes les universités de Lyon, et **uniquement** celles de Lyon

```
186 SELECT DISTINCT c1.idE
187 FROM Candidature c1
188 WHERE NOT EXISTS(
189    SELECT *
190    FROM Université u
191    WHERE u.ville = 'Lyon' AND NOT EXISTS(
192          SELECT *
193          FROM Candidature c2
194          WHERE c2.idE = c1.idE AND c2.nomU = u.nomU))
195 GROUP BY c1.idE
196 HAVING COUNT(*) = (SELECT COUNT(DISTINCT nomU)
197          FROM Université
198          WHERE ville = 'Lyon');
```

idE
898

Schéma explicatif de la division (Julien, 2021)




```sql
SELECT *
FROM Miels m
WHERE NOT EXISTS -- Il n'existe pas de production ayant une saison qui n'apparait pas dans les productions de m
    (SELECT idm
    FROM Produire p_contrexemple
    WHERE NOT EXISTS -- Il n'existe pas de production de m qui a la même saison que notre contre exemple (donc s'il n'existe pas
                     -- de production de m avec la même saison ça veut dire qu'on a trouvé une production avec une nouvelle saison,
                     -- non existante pour le miel m)
        (SELECT idm                                    -- `p_miel.idm = m.idm` fait une jointure entre le miel m qu'on considère
         FROM Produire p_miel                          -- et les productions, ici on a donc juste les productions de m qui ont la
         WHERE p_miel.idm = m.idm                      -- même saison que la production potentiellement "contre exemple" (et donc
           AND p_contrexemple.saison = p_miel.saison)); -- s'il n'y en a pas c'est bien un contre exemple)
```

En résumé

- Clause GROUP BY pour créer des groupes
- Fonctions liées au regroupement
- Clause HAVING pour filtrer les groupes
- Division réalisable avec regroupement ou double négation