

BDW - Opérateurs ensemblistes 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

Modélisation

Schéma entité/
association

Niveau conceptuel

Modèle
relationnel

Niveau logique

SQL (DDL)

Niveau physique

SGBD

Concepts

Optimisation

Base de
données

...

Base de
données

Manipulation

Algèbre
relationnelle

Combinaison
d'opérateurs

Calculs
relationnels

{projetés | formule}

SQL (DML)

SELECT ...
FROM ...

Prog. web

HTML

CSS

PHP

```
<html>
...
<link ... css>
...
<?php
?>
...
</html>
```

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

Opérateurs ensemblistes

Comment utiliser les opérateurs ensemblistes en SQL ?

- ▶ Permettent de combiner les résultats de plusieurs `SELECT`
- ▶ Pas de doublons (`DISTINCT` implicite)
- ▶ Les `SELECT` doivent contenir le même nombre d'attributs
- ▶ Les noms des attributs du résultat sont ceux du premier `SELECT`
 - ▶ C'est l'ordre des attributs qui compte
- ▶ Un opérateur ensembliste peut inclure un `ORDER BY` et un `LIMIT` de manière globale
 - ▶ dans ce cas, les deux `SELECT` sont mis entre parenthèses

Opérateurs ensemblistes (2)

Dans la norme SQL :

- ▶ \cup : UNION
- ▶ \cap : INTERSECT
- ▶ $-$: EXCEPT ou MINUS
- ▶ \div : non implémentée (cf cours *regroupements*)

Sous MariaDB, INTERSECT et EXCEPT implémentés depuis la version 10.3 (UNION disponible depuis longtemps)

<https://mariadb.com/kb/en/mariadb/data-manipulation/>
<http://sqlpro.developpez.com/cours/divrelationnelle/>

Plan

Union

Intersection

Différence

Opérateur union

```
(SELECT att1, ..., attn FROM ...)  
UNION  
(SELECT att1, ..., attn FROM ...)  
[ ORDER BY atti [, attj, ...] ]  
[ LIMIT n ] ;
```

- ▶ Deux requêtes SELECT reliées par un opérateur UNION
- ▶ Les attributs *att*_{*i*} et *att*_{*j*} sont inclus dans *att*₁, ..., *att*_{*n*}

<https://mariadb.com/kb/en/mariadb/union/>

Exemple d'union

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 nom et identifiant des élèves qui ont plus de 18 de moyenne ou qui sont accepté.e.s en informatique

```

19 SELECT idE, nomE FROM Éleve
20 WHERE moyenneLycée > 18
21 UNION
22 SELECT e.idE, nomE FROM Élève e, Candidature c
23 WHERE e.idE = c.idE AND décision = 'O' AND
    département = 'informatique';

```

idE	nomE
123	Ana
678	Farid
456	Damien
876	Irene
654	Ana
345	Chloe
898	Hector

Un moment de réflexion

Pourquoi ne peut-on pas faire une jointure entre les tables Candidature et Élève suivie d'une sélection avec les 3 conditions ?

```
SELECT e.idE, nomE  
FROM Élève e NATURAL JOIN Candidature c  
WHERE décision = 'O' AND département = 'informatique' AND  
moyenneLycee > 18;
```


Un moment de réflexion

Pourquoi ne peut-on pas faire une jointure entre les tables Candidature et Élève suivie d'une sélection avec les 3 conditions ?

```
SELECT e.idE, nomE  
FROM Élève e NATURAL JOIN Candidature c  
WHERE décision = 'O' AND département = 'informatique' AND  
moyenneLycee > 18;
```

⇒ La jointure fait "disparaître" des tuples : l'élève (456, Damien, 19.5, 1000) n'a pas candidaté et sera supprimé lors de la jointure malgré une moyenne de 19.5

Plan

Union

Intersection

Différence

Opérateur intersection

```
(SELECT att1, ..., attn FROM ...)  
INTERSECT  
(SELECT att1, ..., attn FROM ...);
```

- ▶ Deux requêtes `SELECT` reliées par un opérateur `INTERSECT`
- ▶ Opérateur implémenté à partir de MariaDB 10.3

Exemple d'intersection

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 nom et identifiant des élèves qui ont candidaté à la fois en *bioinformatique* et en *informatique*

```

35 | SELECT e.idE, nomE
36 | FROM Élève e NATURAL JOIN Candidature
37 | WHERE département = 'bioinformatique'
38 | INTERSECT
39 | SELECT e.idE, nomE
40 | FROM Élève e NATURAL JOIN Candidature
41 | WHERE département = 'informatique';

```

idE	nomE
345	Chloe

Un moment de réflexion

Pourquoi ne peut-on pas faire une jointure entre les tables Candidature et Élève suivie d'une sélection sur les 2 conditions ?

```
SELECT e.idE, nomE FROM Élève e NATURAL JOIN  
Candidature  
WHERE département = 'bioinformatique' AND département =  
'informatique' ;
```

Un moment de réflexion

Pourquoi ne peut-on pas faire une jointure entre les tables Candidature et Élève suivie d'une sélection sur les 2 conditions ?

```
SELECT e.idE, nomE FROM Élève e NATURAL JOIN  
Candidature  
WHERE département = 'bioinformatique' AND département =  
'informatique' ;
```

⇒ **Aucun tuple dans le résultat, car la valeur de *département* pour un même tuple ne peut pas prendre en même temps la valeur 'informatique' et la valeur 'bioinformatique'**

Un moment de réflexion

Pourquoi ne peut-on pas faire une jointure entre les tables Candidature et Élève suivie d'une sélection sur les 2 conditions ?

```
SELECT e.idE, nomE FROM Élève e NATURAL JOIN  
Candidature  
WHERE département = 'bioinformatique' AND département =  
'informatique';
```

⇒ **Aucun tuple dans le résultat, car la valeur de *département* pour un même tuple ne peut pas prendre en même temps la valeur 'informatique' et la valeur 'bioinformatique'**

Par quoi peut-on remplacer une requête avec intersection ?

Un moment de réflexion

Pourquoi ne peut-on pas faire une jointure entre les tables Candidature et Élève suivie d'une sélection sur les 2 conditions ?

```
SELECT e.idE, nomE FROM Élève e NATURAL JOIN  
Candidature  
WHERE département = 'bioinformatique' AND département =  
'informatique';
```

⇒ **Aucun tuple dans le résultat, car la valeur de *département* pour un même tuple ne peut pas prendre en même temps la valeur 'informatique' et la valeur 'bioinformatique'**

Par quoi peut-on remplacer une requête avec intersection ?

⇒ **Par une sous-requête avec *IN* ou *EXISTS***

Plan

Union

Intersection

Différence

Opérateur différence

```
(SELECT att1, ..., attn FROM ...)  
EXCEPT  
(SELECT att1, ..., attn FROM ...);
```

- ▶ Deux requêtes `SELECT` reliées par un opérateur `EXCEPT`
- ▶ Opérateur implémenté à partir de MariaDB 10.3

Exemple de différence

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

L'identifiant des élèves qui ont candidaté en *informatique* mais pas en *bioinformatique*

```

52 SELECT idE FROM Candidature
53 WHERE département = 'informatique'
54 EXCEPT
55 SELECT idE FROM Candidature
56 WHERE département = 'bioinformatique';

```

idE	nomE
123	Ana
543	Hector
876	Ana
898	Irene

Un moment de réflexion

Est-ce possible de faire la somme des valeurs de deux attributs (e.g., des attributs *effectif* et *effectifLycee*) ?

Un moment de réflexion

Est-ce possible de faire la somme des valeurs de deux attributs (e.g., des attributs *effectif* et *effectifLycee*) ?

⇒ **Oui, en créant une sous-requête avec une union entre les attributs à sommer, puis la requête principale fait la somme sur la colonne résultante de la sous-requête**

Un moment de réflexion

Est-ce possible de faire la somme des valeurs de deux attributs (e.g., des attributs *effectif* et *effectifLycee*) ?

⇒ **Oui, en créant une sous-requête avec une union entre les attributs à sommer, puis la requête principale fait la somme sur la colonne résultante de la sous-requête**

Par quoi peut-on remplacer une requête avec différence ?

Un moment de réflexion

Est-ce possible de faire la somme des valeurs de deux attributs (e.g., des attributs *effectif* et *effectifLycee*) ?

⇒ **Oui, en créant une sous-requête avec une union entre les attributs à sommer, puis la requête principale fait la somme sur la colonne résultante de la sous-requête**

Par quoi peut-on remplacer une requête avec différence ?

⇒ **Par une sous-requête avec NOT IN ou NOT EXISTS**

En résumé

- ▶ Opérateurs UNION, INTERSECT et EXCEPT/MINUS dans la norme SQL et dans MariaDB, mais pas la division
- ▶ Division avec des regroupements ou avec des sous-requêtes
NOT EXISTS

```
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)