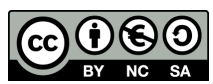
BDW - Jointures SQL

Fabien Duchateau

fabien.duchateau [at] univ-lyon1.fr

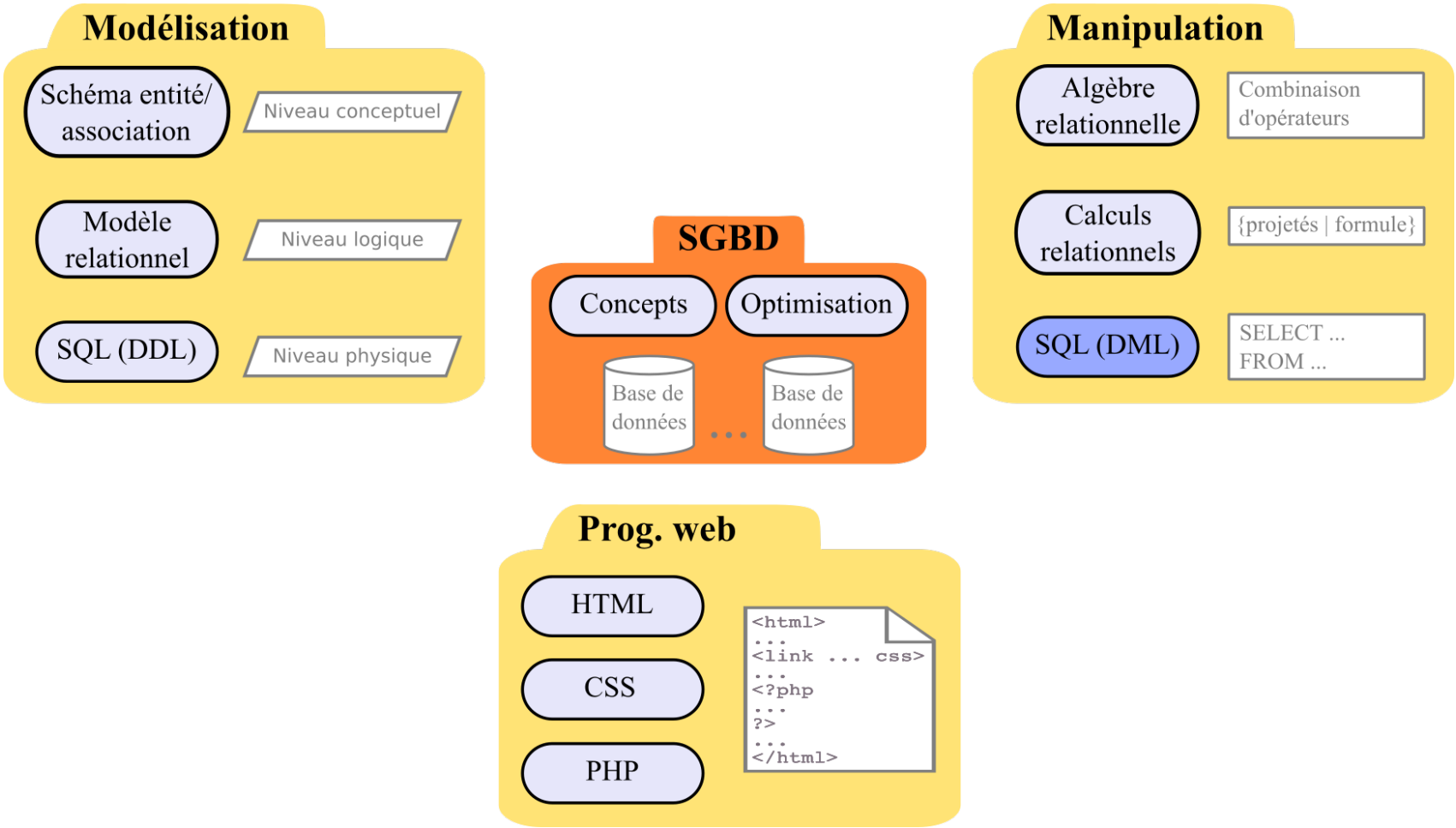
Université Claude Bernard Lyon 1

2023 - 2024



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

Positionnement dans BDW

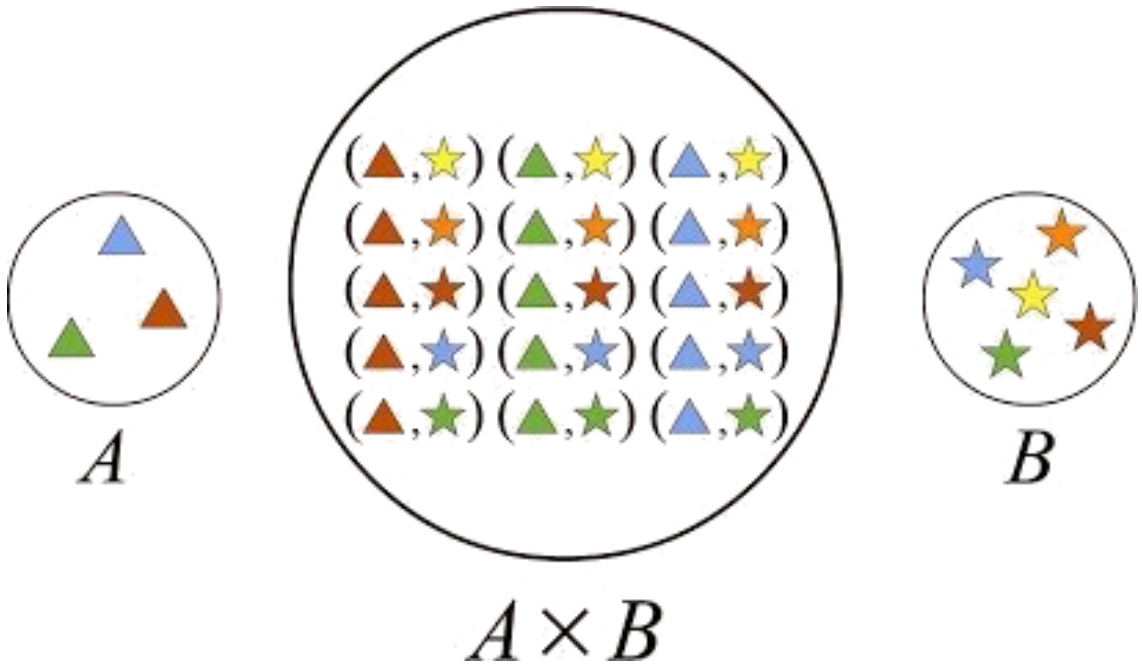


Ces diapositives utilisent le genre féminin (e.g., chercheuse, développeuses) plutôt que l'écriture inclusive (moins accessible, moins concise, et pas totalement inclusive)

Requêtes sur plusieurs tables

Interrogation de plusieurs tables en utilisant la clause FROM

- Produit cartésien



Types de jointure

Si l'on met une ou plusieurs conditions entre attributs de différentes tables, on ne parle plus de produit cartésien mais de **jointure**

- Jointure interne
- Jointure externe
- Semi-jointure
- Auto-jointure

Conditions de jointure

Les conditions de jointures spécifient comment les tables sont jointes (quels attributs et quelle comparaison)

- Équi-jointure = la condition de jointure est une égalité
- Theta-jointure = la condition de jointure n'est pas (forcément) une égalité ($>$, ..., $\leq$, IN, LIKE, ...)
- Naturelle = la condition de jointure applique une égalité entre les attributs identiques (même label) des deux tables

Plan

Produit cartésien

Jointure naturelle

Jointure interne

Jointure externe

Semi-jointure

Auto-jointure

Syntaxe

SELECT att_1, att_2, ...
FROM nom_table_1, nom_table_2, ... **;**

- Plusieurs tables séparées par des virgules = produit cartésien entre ces différentes tables

- Si la requête utilise un attribut *att* présent dans plusieurs tables, on doit l'écrire *nom_table.att* ou utiliser un alias de table *alias.att*

Exemple de produit cartésien

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 paires de nom d'université et de nom d'élève

```
19 SELECT nomU,nomE
20 FROM Université, Élève;
```

nomU	nomE
INSA	Ana
UCB	Ana
UJF	Ana
UJM	Ana
INSA	Bob
...	...

Total de 48 tuples (12×4)

Plan

Produit cartésien

Jointure naturelle

Jointure interne

Jointure externe

Semi-jointure

Auto-jointure

Syntaxe

SELECT att_1, att_2, ...
FROM nom_table_1 **NATURAL JOIN** nom_table_2
[**WHERE** *autres_conditions*] **;**

- Soient att_{c_1}, ..., att_{c_k} les attributs communs des tables nom_table_1 et nom_table_2
- Les instances de nom_table_1 et nom_table_2 qui possèdent des valeurs égales sur tous leurs attributs communs att_{c_1}, ..., att_{c_k} sont "assemblées" en un tuple qui est ajouté dans le résultat

Syntaxe de la jointure naturelle tronquée

SELECT att_1, att_2, ...
FROM nom_table_1 **NATURAL JOIN** nom_table_2
USING $(att_{c_x}, \ldots, att_{c_y})$
[**WHERE** *autres_conditions*] ;

- Soient att_{c_1}, ..., att_{c_k} les attributs communs des tables nom_table_1 et nom_table_2, et $\{att_{c_x}, \ldots, att_{c_y}\} \in \{att_{c_1}, \ldots, att_{c_k}\}$
- Le mot-clé USING permet de faire une jointure naturelle sur un sous-ensemble des attributs communs $\{att_{c_x}, \ldots, att_{c_y}\}$ de nom_table_1 et nom_table_2

Exemple de jointure naturelle

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 élèves avec plus de 19 de moyenne qui ont candidaté

```
30 SELECT DISTINCT e.idE, nomE,
       moyenneLycée, nomU
31 FROM Élève e NATURAL JOIN
     Candidature c
32 WHERE moyenneLycée > 19;
```

idE	nomE	moyenneLycee	nomU
123	Ana	19.5	INSA
123	Ana	19.5	UCB
123	Ana	19.5	UJM
876	Irene	19.5	UCB
876	Irene	19.5	UJF

Plan

Produit cartésien

Jointure naturelle

Jointure interne

Jointure externe

Semi-jointure

Auto-jointure

Syntaxe

La **jointure interne** est fréquemment utilisée : seuls les tuples qui respectent la condition de jointure sont conservés

SELECT att_1, att_2, ...
FROM nom_table_1 **INNER JOIN** nom_table_2
ON $nom\_table_1.att_x \ \Theta \ nom\_table_2.att_x$
[**WHERE** *autres_conditions*] **;**

- Soit Θ un opérateur parmi $=$, $\neq$, $<$, $>$, $\leq$, $\geq$, LIKE, ...
- La condition de jointure $nom\_table_1.att_x \ \Theta \ nom\_table_2.att_x$ s'exprime avec le mot-clé ON
- Les autres conditions s'expriment dans le WHERE et sont appliquées après la condition de jointure

Exemple de jointure interne

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 élèves qui ont candidaté dans une université grenobloise

```
34 SELECT e.idE, nomE
35 FROM Élève e INNER JOIN Candidature c ON e.idE
       = c.idE
36       INNER JOIN Université u ON c.nomU = u.nomU
37 WHERE u.ville = 'Grenoble';
```

idE	nomE
345	Chloe
543	Chloe
876	Irene
876	Irene

Syntaxe obsolète de jointure interne

SELECT att_1, att_2, ...
FROM nom_table_1, nom_table_2, ...
WHERE $nom\_table_1.att_x \ \Theta \ nom\_table_2.att_x$
[**AND** *autres_conditions*] ;

- Jointure interne $\equiv$ sélection sur le produit cartésien
- La condition de jointure $nom\_table_1.att_x \ \Theta \ nom\_table_2.att_x$ s'exprime ici dans le WHERE

Syntaxe obsolète, à éviter !

Exemple de jointure interne obsolète

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 élèves qui ont candidaté dans une université grenobloise

```
39 SELECT e.idE, nomE
40 FROM Élève e, Candidature c, Université u
41 WHERE e.idE = c.idE AND c.nomU = u.nomU AND
       u.ville ='Grenoble';
```

idE	nomE
345	Chloe
543	Chloe
876	Irene
876	Irene

Exemple de jointure interne obsolète

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 élèves qui ont candidaté dans une université grenobloise

```
39 SELECT e.idE, nomE
40 FROM Élève e, Candidature c, Université u
41 WHERE e.idE = c.idE AND c.nomU = u.nomU AND
       u.ville = 'Grenoble';
```

OBSOLÈTE

idE	nomE
345	Chloe
543	Chloe
876	Irene
876	Irene

Plan

Produit cartésien

Jointure naturelle

Jointure interne

Jointure externe

Semi-jointure

Auto-jointure

Syntaxe

SELECT att_1, att_2, ...
FROM nom_table_1 < **LEFT** | **RIGHT** | **FULL** >
[**OUTER**] **JOIN** nom_table_2
ON $nom\_table_1.att_x = nom\_table_2.att_x$
[**WHERE** *autres_conditions*] **;**

- Non exprimable en Algèbre Relationnelle
- Une requête avec jointure OUTER JOIN retourne les tuples qui remplissent la condition de la jointure, mais aussi certains tuples qui ne la satisfont pas
- Ces tuples qui ne satisfont pas la condition de jointure dépendent du mot-clé LEFT, RIGHT ou FULL

Syntaxe (2)

Sélection des tuples de la jointure externe :

- LEFT (ou RIGHT) : les tuples de la table de gauche (ou de droite) sans correspondance dans l'autre table sont inclus dans le résultat avec une valeur NULL pour les attributs de l'autre table

- FULL : toutes les lignes de chacune des tables sont retournées. Les lignes sans correspondance ont leurs attributs complétés par des valeurs NULL

Exemple de jointure externe à gauche

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 élèves avec une moyenne supérieure à 19 et les éventuelles universités où ils/elles ont candidaté

```
45 SELECT DISTINCT e.idE, nomE, nomU
46 FROM Élève e LEFT OUTER JOIN
       Candidature c
47   ON c.idE = e.idE
48 WHERE moyenneLycee > 19;
```

idE	nomE	nomU
123	Ana	INSA
123	Ana	UCB
123	Ana	UJM
456	Damien	NULL
654	Ana	NULL
876	Irène	UCB
876	Irène	UJF

Exemple de jointure externe complète

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

Requête identique, mais avec un FULL OUTER JOIN

```
50 SELECT e.idE, nomE, nomU
51 FROM Élève e FULL OUTER JOIN
       Candidature c
52   ON c.idE = e.idE
53 WHERE moyenneLycee > 19;
```

idE	nomE	nomU
123	Ana	UJM
123	Ana	INSA
123	Ana	UCB
456	Damien	NULL
654	Ana	NULL
876	Irène	UJF
876	Irène	UCB

Exemple de jointure externe complète (2)

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

Pourquoi la requête FULL OUTER JOIN donne le même résultat que celle avec LEFT OUTER JOIN ?

Plan

Produit cartésien

Jointure naturelle

Jointure interne

Jointure externe

Semi-jointure

Auto-jointure

Syntaxe

SELECT *nom_table*$_2$.*
FROM *nom_table*$_1$ **NATURAL JOIN** *nom_table*$_2$ **;**

- Une semi-jointure est une jointure qui ne garde dans le résultat que les attributs d'une seule table (ici les attributs de *nom_table*$_2$)

- Construite avec *nom_table*.* dans la clause SELECT (tous types de jointures acceptés, i.e., interne, externe)

Exemple de semi-jointure

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 informations sur les élèves qui ont candidaté dans une université grenobloise

```
61 SELECT e.*
62 FROM Élève e NATURAL JOIN
       Candidature c
63   NATURAL JOIN Université
64 WHERE ville = 'Grenoble';
```

idE	nomE	moyenneLycee	effectifLycee
345	Chloe	17.5	500
543	Chloe	17	2000
876	Irene	19.5	400
876	Irene	19.5	400

Plan

Produit cartésien

Jointure naturelle

Jointure interne

Jointure externe

Semi-jointure

Auto-jointure

Syntaxe

SELECT t1.att_1, t2.att_1, ...
FROM nom_table_1 t1 **NATURAL JOIN** nom_table_1 t2 **;**

- Auto-jointure = jointure d'une table avec elle même (tous types de jointures acceptés, i.e., interne, externe), en utilisant des alias de table (ici *t1* et *t2*)

- Exemples fréquents d'auto-jointure : *personne/parents*, *employée/supérieure hiérarchique*, *pièce/composant*

Exemple d'auto-jointure

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

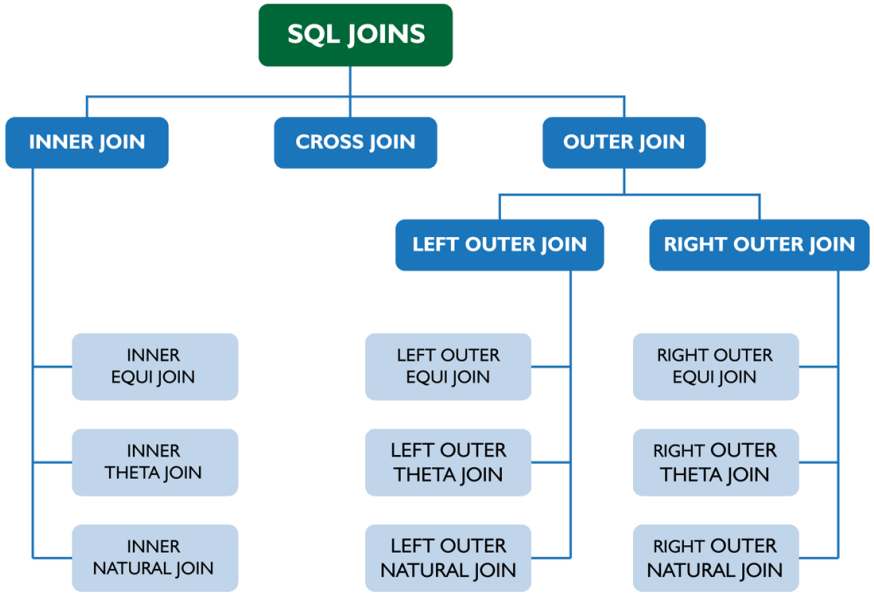
Les paires d'élèves qui ont candidaté dans la même université et le même département

```
68 SELECT DISTINCT c1.idE, c2.idE
69 FROM Candidature c1 INNER JOIN Candidature
       c2 ON c1.nomU = c2.nomU
70   AND c1.département = c2.département
71   AND c1.idE < c2.idE;
```

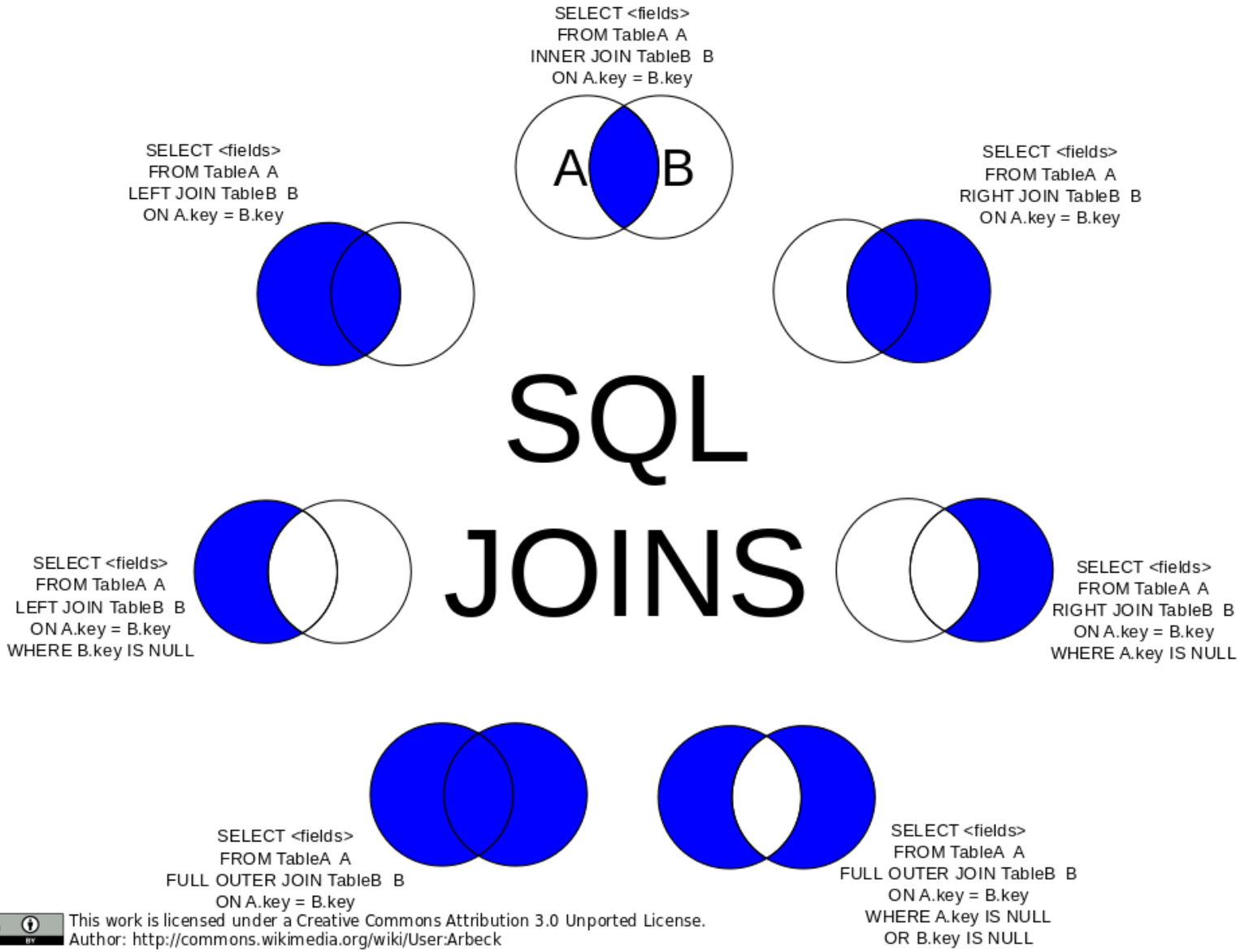
c1.idE	c2.idE
123	898
123	876
123	345
678	765
876	898

Classification des jointures

- La condition $=$ naturelle, équi-jointure, θ-jointure
- Les n-uplets conservés dans le résultat $=$ jointure interne, jointures externes (et produit cartésien)
- Les attributs conservés dans le résultat $=$ semi-jointure

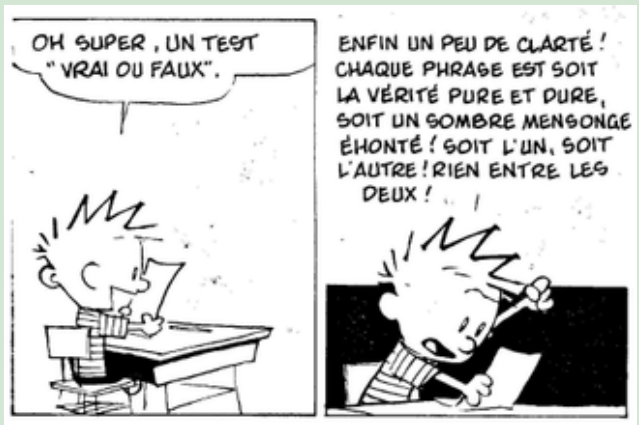


En résumé



Un moment de réflexion

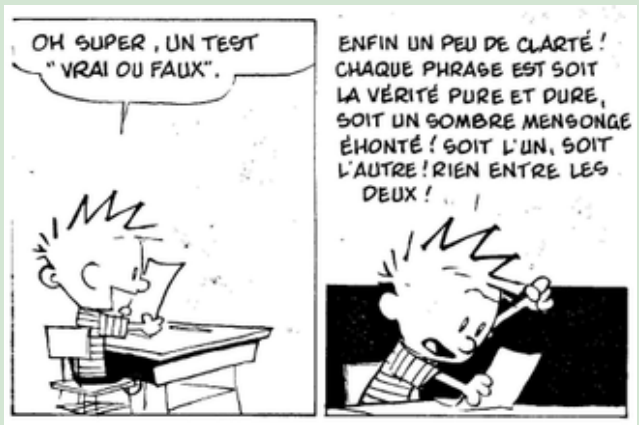
Qu'est ce que l'hypothèse du monde clos ? Quel opérateur permet de "contrer" cette hypothèse ?



Un moment de réflexion

Qu'est ce que l'hypothèse du monde clos ? Quel opérateur permet de "contrer" cette hypothèse ?

Hypothèse du monde clos : l'absence d'information (valeur nulle) implique de filtrer le n-uplet (e.g., si on requête les universités de Lyon, une université qui n'a pas de ville n'est pas retournée).
La jointure externe et/ou une condition "**OR** *att* **IS NULL**" permettent de contrer cette hypothèse en retournant les n-uplets qui ont une valeur nulle ou aucune correspondance de jointure



Démo phpMyAdmin

Démo avec phpMyAdmin (scripts SQL en ligne)