

Bases de données (CS443)

#5, Modèle relationnel: SQL

Arthur Baudet

Grenoble INP - Esisar

2023–2024



Transparents F. Duchateau, pour univ Lyon1, CC by SA.

<https://perso.liris.cnrs.fr/fabien.duchateau/BDW1/>

Dialectes

Suivant le SGBD des différences de syntaxe mineures peuvent apparaître.

On pourra se reporter à la page ci-dessus pour l'écriture de sous-requêtes.

Rappel du jeu de données

ÉLÈVE (idE, *nomE*, *moyenneLycee*, *effectifLycee*)
CANDIDATURE (*#idE*, *#nomU*, département, *décision*)
UNIVERSITÉ (*nomU*, *ville*, *effectif*)

idE	nomE	moyenneLycee	effectifLycee
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

Dans ces transparents, les tables sont abrégées en *E*, *C* et *U*

Jeu de données francisé inspiré du cours [Databases de Stanford](#)

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures

Ce n'est pas un parcours exhaustif du langage, mais uniquement quelques concepts. On regardera en particulier :

- la correspondance avec les/des opérations de l'algèbre relationnelle
- les fonctions support (comptage, traitement de chaînes) supplémentaires.

Une documentation sera fournie avec les TPs.

Démo - sqllitebrowser

The screenshot displays the SQLite Browser application window. The menu bar includes 'Fichier', 'Édition', 'Vue', 'Outils', and 'Aide'. The toolbar contains icons for file operations and execution. The main window has tabs for 'Nouvelle Base de Données', 'Ouvrir une Base de Données', 'Ouvrir un Projet', and 'Attacher une Base de Données'. Below the tabs are buttons for 'Structure de la Base de Données', 'Parcourir les données', 'Éditer les Pragma', and 'Exécuter le SQL'. The 'Exécuter le SQL' button is active, and the 'Éditer le contenu d'une Cellule' panel is open on the right.

The SQL editor on the left contains the following query:

```
1 SELECT Nom,Prenom FROM utilisateur
2
```

The result set is displayed in a table with two columns: 'Nom' and 'Prenom'.

	Nom	Prenom
1	SPOCK	Eillian
2	DENNIS	Jean-Michel
3	CALLEB	Oliver
4	JENKINS	Stefan

The status bar at the bottom indicates: 'L'exécution s'est terminée sans erreur. Résultat : 4 enregistrements ramenés en 3ms. À la ligne 1 : SELECT Nom,Prenom FROM utilisateur'.

The 'Éditer le contenu d'une Cellule' panel on the right shows the 'Mode' set to 'Texte'. The content area contains the text '1 SPOCK'. Below this, it states 'Type actuel des données de la cellule : Texte / Numérique' and '5 caractères'. The 'Journal SQL' panel shows the executed SQL query and its result: '1 -- EXECUTER TOUT DANS 'test'', '2 --', '3 -- À la ligne 1:', '4 SELECT Nom,Prenom FROM utilisateur', '5 -- Résultat : 4 enregistrements ramenés en 3', and '6'.

Différences entre la théorie et SQL

- ▶ Possibilité de doublons
- ▶ Possibilité d'ordonner le résultat des requêtes
- ▶ Notion de valeur non définie
- ▶ Absence de certains opérateurs ensemblistes



Construire les tables

Les commandes (requêtes de modification) pour construire les tables seront vues en TP. Voici les mots clés :

- CREATE TABLE
- CONSTRAINT ... PRIMARY KEY, NOT NULL, ...
- INSERT INTO, DELETE, ...

Un exemple pour la route :

```
CREATE TABLE auteur (  
  num_a INTEGER,  
  nom VARCHAR(30),  
  CONSTRAINT cle_auteur PRIMARY KEY (num_a)  
);
```


Éléments de langage

① Basics

- Projection (select... from)

- Sélection (where)

- Chaînes

- Tri, limites

- Agrégation (group by), et calculs

② Jointures

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures

Syntaxe

Syntaxe minimale d'une requête SQL :

```
SELECT att1, att2, ...  
FROM nom_table ;
```

- ▶ SELECT et FROM sont des clauses SQL
- ▶ Projection : récupération des valeurs contenues dans la table *nom_table*, en ne gardant que les attributs *att₁, att₂, ...*
- ▶ On peut remplacer *att₁, att₂, ...* par * pour utiliser tous les attributs des tables listées dans la clause FROM

Équivalences

- ▶ En SQL :
SELECT att_1, att_2, \dots
FROM nom_table ;
- ▶ En algèbre relationnelle :
 $\pi_{att_1, att_2, \dots}(nom_table)$
- ▶ En calcul relationnel tuple :
 $\{t.att_1, t.att_2, \dots \mid nom_table(t)\}$

Exemple de projection

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	Irene	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 l'effectif des universités

Exemple de projection

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	Irene	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 l'effectif des universités

```
22 SELECT nomU, effectif
23 FROM Université;
```

Exemple de projection

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	Irene	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 l'effectif des universités

22 **SELECT** nomU, effectif
 23 **FROM** Université;

nomU	effectif
INSA	36000
UCB	15000
UJF	10000
UJM	21000

Exemple de projection (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	Gisele	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 universités

Exemple de projection (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	Gisele	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 universités

```
25 | SELECT nomU, ville, effectif
26 | FROM Université;
```

ou

```
28 | SELECT * FROM Université;
```

Exemple de projection (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	Gisele	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 universités

25 **SELECT** nomU, ville, effectif
 26 **FROM** Université;

ou

28 **SELECT** * **FROM** Université;

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

Suppression des doublons

Mot clé **DISTINCT** pour supprimer les n-uplets en doublon :

```
SELECT DISTINCT  $att_1, att_2, \dots$   
FROM  $nom\_table$  ;
```

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

En rouge, un exemple de n-uplet en doublon dans la table (les trois valeurs des 2 n-uplets étant identiques)

Suppression des doublons (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

Les villes où se trouvent des universités

Suppression des doublons (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

Les villes où se trouvent des universités

30 | `SELECT ville FROM Université;`

ville
Lyon
Lyon
Grenoble
Saint-Étienne

32 | `SELECT DISTINCT ville FROM Université;`

ville
Lyon
Grenoble
Saint-Étienne

Renommage et alias

Mot-clé AS pour renommer un attribut :

```
SELECT  $att_1$  AS  $att'_1$ ,  $att_2$  AS  $att'_2$ , ...  
FROM  $nom\_table$  ;
```

- L'attribut att_1 sera renommé en att'_1 , etc.

Alias de table comme moyen de désambiguïsation ou pour utiliser deux fois la même table :

```
SELECT DISTINCT  $t.att_1$ ,  $att_2$ , ...  
FROM  $nom\_table$  [AS]  $t$  ;
```

- L'alias de la table est t , accès à l'attribut att_1 par $t.att_1$

Renommage et alias (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	Gisele	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 la ville des universités, avec renommage

```
34 | SELECT nomU AS nom-univ, ville AS ville-univ FROM
    | Université;
```

Renommage et alias (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	Gisele	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 la ville des universités, avec renommage

34 **SELECT** nomU **AS** nom-univ, ville **AS** ville-univ **FROM**
Université;

nom-univ	ville-univ
INSA	Lyon
UCB	Lyon
UJF	Grenoble
UJM	Saint-Étienne

Renommage et alias (3)

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 des universités avec alias de table

36 | `SELECT u.nomU, u.ville FROM Université AS u;`

Renommage et alias (3)

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 des universités avec alias de table

36 | `SELECT u.nomU, u.ville FROM Université AS u;`

nomU	ville
INSA	Lyon
UCB	Lyon
UJF	Grenoble
UJM	Saint-Etienne

En résumé

- ▶ Clause SELECT \equiv projection (d'attributs)
- ▶ Clause FROM \equiv liste des tables utilisées
- ▶ Renommage d'attribut (AS) et alias de table
- ▶ Syntaxe d'une requête SQL (crochets = option) :

```
SELECT att1 [, att2 [ AS att'2 ], ...]  
FROM nom_table1 [, nom_table2 [ alias ] ...];
```

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures

Syntaxe

Clause WHERE pour la sélection :

```
SELECT att1, att2, ...  
FROM nom_table  
WHERE condition ;
```

- ▶ La clause WHERE permet de sélectionner les lignes en filtrant celles qui ne remplissent pas la condition (i.e., éliminées du résultat)

Conditions

Condition du **WHERE** : une combinaison d'expressions connectées par **AND** (\wedge) ou **OR** (\vee)

Une expression effectue une opération entre un attribut et une constante ou entre deux attributs

Expressions simples :

- ▶ Opérateurs de comparaison (=, !=, <, <=, >, >=)
- ▶ Différents types de données utilisés pour les constantes :
 - ▶ nombres : 1, 36000, 1.5
 - ▶ chaînes de caractères : 'UCBL', 'Grenoble'
 - ▶ dates : '1986-04-26' (le formatage des dates peut varier selon le SGBD)

Exemple de sélection

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

Nom et effectif des universités avec un effectif inférieur à 20000

Exemple de sélection

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	Irene	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

Nom et effectif des universités avec un effectif inférieur à 20000

```

40 SELECT nomU, effectif
41 FROM Université
42 WHERE effectif < 20000;
```


Exemple de sélection

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

Nom et effectif des universités avec un effectif inférieur à 20000

```

40 SELECT nomU, effectif
41 FROM Université
42 WHERE effectif < 20000;

```

nomU	effectif
UCB	15000
UJF	10000

Exemple de sélection (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

Informations sur les universités de Lyon avec un effectif inférieur à 25000

Exemple de sélection (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

Informations sur les universités de Lyon avec un effectif inférieur à 25000

```

44 SELECT *
45 FROM Université
46 WHERE ville = 'Lyon'
47 AND effectif < 20000;
```

Exemple de sélection (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

Informations sur les universités de Lyon avec un effectif inférieur à 25000

```

44 SELECT *
45 FROM Université
46 WHERE ville = 'Lyon'
47 AND effectif < 20000;
```

nomU	ville	effectif
UCB	Lyon	15000

Autres opérateurs

- ▶ Opérateur **IN** (val_1, val_2, \dots) :
 - ▶ spécifie un ensemble de valeur possibles

- ▶ Opérateur **BETWEEN** val_1 **AND** val_2 :
 - ▶ spécifie un intervalle de valeurs (bornes val_1 et val_2 incluses)
 - ▶ attention à ne pas confondre le **AND** du **BETWEEN** avec celui qui correspond au \wedge

Exemple de sélection (3)

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	Gisele	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 universités de Lyon ou Grenoble

```

49 | SELECT *
50 | FROM Université
51 | WHERE ville IN ('Lyon', 'Grenoble');

53 | SELECT *
54 | FROM Université
55 | WHERE ville = 'Lyon' OR ville = 'Grenoble';

```

Exemple de sélection (3)

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	Gisele	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 universités de Lyon ou Grenoble

```

49 | SELECT *
50 | FROM Université
51 | WHERE ville IN ('Lyon', 'Grenoble');

```

```

53 | SELECT *
54 | FROM Université
55 | WHERE ville = 'Lyon' OR ville = 'Grenoble';

```

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000

Exemple de sélection (4)

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	Gisele	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 avec un effectif entre 12000 et 25000

```

57 SELECT * FROM Université
58 WHERE effectif BETWEEN 12000 AND
    25000;

60 SELECT * FROM Université
61 WHERE effectif >= 12000 AND
    effectif <= 25000;
```


Exemple de sélection (4)

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	Gisele	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 avec un effectif entre 12000 et 25000

```

57 SELECT * FROM Université
58 WHERE effectif BETWEEN 12000 AND
    25000;

60 SELECT * FROM Université
61 WHERE effectif >= 12000 AND
    effectif <= 25000;
```

nomU	ville	effectif
UCB	Lyon	15000
UJM	Saint-Étienne	21000

Valeurs non définies

En pratique, il est possible d'avoir des valeurs non définies :

- ▶ Elles sont représentées par le mot clé `NULL`
- ▶ On peut tester si une valeur n'est pas définie grâce à la condition **IS NULL** (ou son contraire **IS NOT NULL**)



Exemple de sélection (5)

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	NULL
UJM	Saint-Etienne	21000

Les informations sur les universités avec un effectif non défini

```
65 | SELECT * FROM Université
66 | WHERE effectif IS NULL;
```

Exemple de sélection (5)

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	NULL
UJM	Saint-Etienne	21000

Les informations sur les universités avec un effectif non défini

```
65 | SELECT * FROM Université
66 | WHERE effectif IS NULL;
```

nomU	ville	effectif
UJF	Grenoble	NULL

En résumé

- ▶ Clause WHERE \equiv sélection (d'instances)
- ▶ Condition = combinaison (AND, OR) d'expressions (=, !=, <, <=, >, >=)
- ▶ Opérateurs spécifiques (IN, BETWEEN), valeur non définie (NULL)
- ▶ Syntaxe d'une requête SQL (crochets = option) :

```
SELECT att1 [, att2 [ AS att'2 ], ...]  
FROM nom_table1 [, nom_table2 [ alias ], ...]  
[ WHERE condition ] ;
```

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures

SELECT ... FROM ... WHERE attribut **LIKE** pattern

le pattern est une forme de *regex*.

Ex : `select distinct titre from livre where titre like 'H%' from utilisateur :`

titre
Harry Potter à l'école des sorciers
Harry Potter et la chambre des secrets

Traitement des chaînes 2/2

- On peut comparer des chaînes, ...
- On peut concaténer des chaînes, utiliser upper

```
select upper(nom || ' ' || prenom) as nom_prenom from utilisateur;
```

nom_prenom
CALLEB OLIVER
JENKINS STEFAN
SPOCK ELLIAN
DENNIS JEAN-MICHEL

noter l'utilisation de **as** pour nommer le résultat et encore : BETWEEN (nombres, chaînes).

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures

Syntaxe du tri

Clause ORDER BY pour trier le résultat d'une requête :

```
SELECT att1, att2, ...  
FROM nom_table  
WHERE condition  
ORDER BY atti, attj, ... ;
```

- ▶ Le résultat de la requête est trié selon l'ordre naturel croissant de l'attribut *att*_{*i*}
- ▶ En cas d'égalité entre deux lignes au niveau de l'attribut *att*_{*i*}, on utilise l'attribut suivant *att*_{*j*}, etc.
- ▶ Le nom d'un attribut peut être suivi par ASC ou DESC pour indiquer un ordre croissant (défaut) ou décroissant

Exemple de tri

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	Irene	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 universités, triées par effectif croissant

```
71 SELECT * FROM Université
72 ORDER BY effectif ASC;
```

Exemple de tri

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	Irene	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 universités, triées par effectif croissant

```
71 SELECT * FROM Université
72 ORDER BY effectif ASC;
```

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

Exemple de tri (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	Gisele	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 universités de plus de 12000 étudiant-e-s, avec un tri par ville décroissante puis par effectif croissant

```

77 SELECT * FROM Université
78 WHERE effectif > 12000
79 ORDER BY ville DESC,
    effectif ASC;

```

Exemple de tri (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	Gisele	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 universités de plus de 12000 étudiant-e-s, avec un tri par ville décroissante puis par effectif croissant

```

77 SELECT * FROM Université
78 WHERE effectif > 12000
79 ORDER BY ville DESC,
    effectif ASC;

```

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

Syntaxe de la limitation

Clause LIMIT pour conserver un nombre restreint de résultats :

```
SELECT att1, att2, ...  
FROM nom_table  
WHERE condition  
ORDER BY atti, attj, ...  
LIMIT n ;
```

- ▶ Les n premières instances (après le tri) sont conservés dans le résultat
- ▶ Pas un top-K, qui récupère toutes les instances avec les n meilleures valeurs

Dans la norme SQL, la syntaxe est FETCH FIRST n ROWS ONLY

Exemple de limitation

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

Deux universités qui ont les effectifs les plus élevés

nomU	effectif
INSA	36000
UJM	21000

Exemple de limitation

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

Deux universités qui ont les effectifs les plus élevés

```

81 SELECT nomU, effectif FROM
    Université
82 ORDER BY effectif DESC LIMIT 2;
```

nomU	effectif
INSA	36000
UJM	21000

En résumé

- ▶ Clause ORDER BY \equiv tri du résultat
- ▶ Clause LIMIT \equiv troncature du résultat aux premières instances
- ▶ Syntaxe d'une requête SQL (crochets = option) :

```
SELECT att1 [, att2 [ AS att'2 ], ...]  
FROM nom_table1 [, nom_table2 [AS alias ]]  
[ WHERE condition ]  
[ ORDER BY atti [, attj, ...] ]  
[ LIMIT n ] ;
```

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures

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

Table CANDIDATURE

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

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



nomU	count
UJM	6
UCB	6
UJF	4
INSA	3

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

⇒

nomU	count
UJM	6
UCB	6
UJF	4
INSA	3

⇒

nomU	count
INSA	3
UCB	6
UJF	4
UJM	6

Having

Idem que where, mais sur les regroupements (voir TP).

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures

Syntaxe

```
SELECT att1, att2, ...  
FROM nom_table1, nom_table2, ... ;
```

- ▶ 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	Gisele	17	800
876	Irene	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 att1, att2, ...  
FROM nom_table1 NATURAL JOIN nom_table2  
[ WHERE autres_conditions ] ;
```

- ▶ Soient $att_{c_1}, \dots, att_{c_k}$ 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 $att_{c_1}, \dots, att_{c_k}$ sont "assemblées" en un tuple qui est ajouté dans le résultat

Syntaxe de la jointure naturelle tronquée

```
SELECT  $att_1, att_2, \dots$   
FROM  $nom\_table_1$  NATURAL JOIN  $nom\_table_2$   
USING ( $att_{c_x}, \dots, att_{c_y}$ )  
[ WHERE  $autres\_conditions$  ] ;
```

- ▶ Soient $att_{c_1}, \dots, att_{c_k}$ les attributs communs des tables nom_table_1 et nom_table_2 , et $\{att_{c_x}, \dots, att_{c_y}\} \in \{att_{c_1}, \dots, att_{c_k}\}$
- ▶ Le mot-clé **USING** permet de faire une jointure naturelle sur un sous-ensemble des attributs communs $\{att_{c_x}, \dots, att_{c_y}\}$ 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	Gisele	17	800
876	Irene	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	moyenneLycée	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 att1, att2, ...  
FROM nom_table1 INNER JOIN nom_table2  
ON nom_table1.attx  $\Theta$  nom_table2.attx  
[ WHERE autres_conditions ] ;
```

- ▶ Soit Θ un opérateur parmi $=$, \neq , $<$, $>$, \leq , \geq , LIKE, ...
- ▶ La condition de jointure *nom_table*₁.*att*_x Θ *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	Irene	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 att1, att2, ...  
FROM nom_table1, nom_table2, ...  
WHERE nom_table1.attx  $\Theta$  nom_table2.attx  
[ AND autres_conditions ] ;
```

- ▶ Jointure interne \equiv sélection sur le produit cartésien
- ▶ La condition de jointure *nom_table*₁.*att*_x Θ *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	Irene	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	Irene	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

Plan

Produit cartésien

Jointure naturelle

Jointure interne

Jointure externe

Semi-jointure

Auto-jointure

Syntaxe

```
SELECT att1, att2, ...  
FROM nom_table1 < LEFT | RIGHT | FULL >  
[ OUTER ] JOIN nom_table2  
ON nom_table1.attx = nom_table2.attx  
[ 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	Gisele	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 moyenneLycée > 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	Gisele	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 moyenneLycée > 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	Irene	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_table2.*  
FROM nom_table1 NATURAL JOIN nom_table2;
```

- ▶ 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	Gisele	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	moyenneLycée	effectifLycée
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.att1, t2.att1, ...  
FROM nom_table1 t1 NATURAL JOIN nom_table1 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	Gisele	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

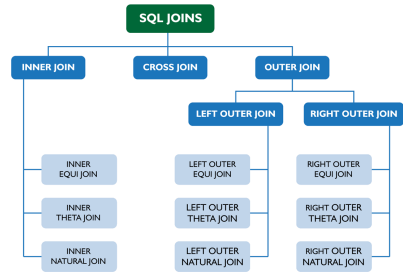
```

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;
```

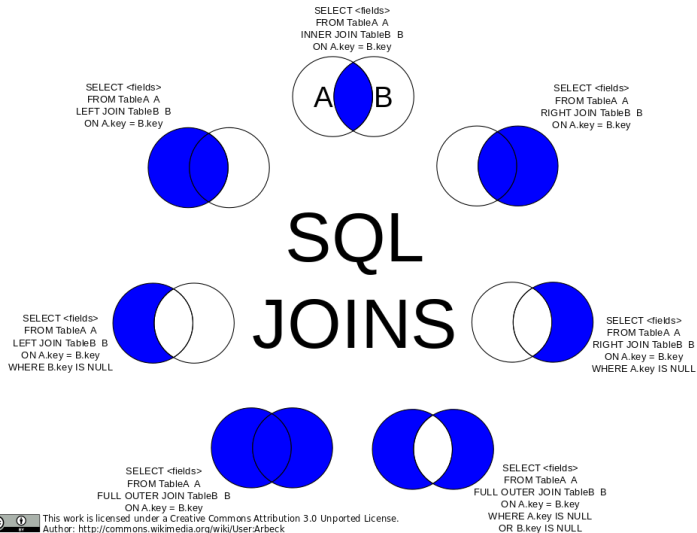
c1.idE	c2.idE
123	898
123	876
123	345
678	765
876	898

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



This work is licensed under a Creative Commons Attribution 3.0 Unported License.
Author: <http://commons.wikimedia.org/wiki/User:Arbeck>

Exemple de deux requêtes à résultat équivalent ?

Éléments de langage

① Basics

Projection (select... from)

Sélection (where)

Chaînes

Tri, limites

Agrégation (group by), et calculs

② Jointures