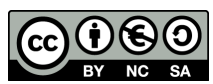
BDW - Sous-requêtes en SQL

Fabien Duchateau

fabien.duchateau [at] univ-lyon1.fr

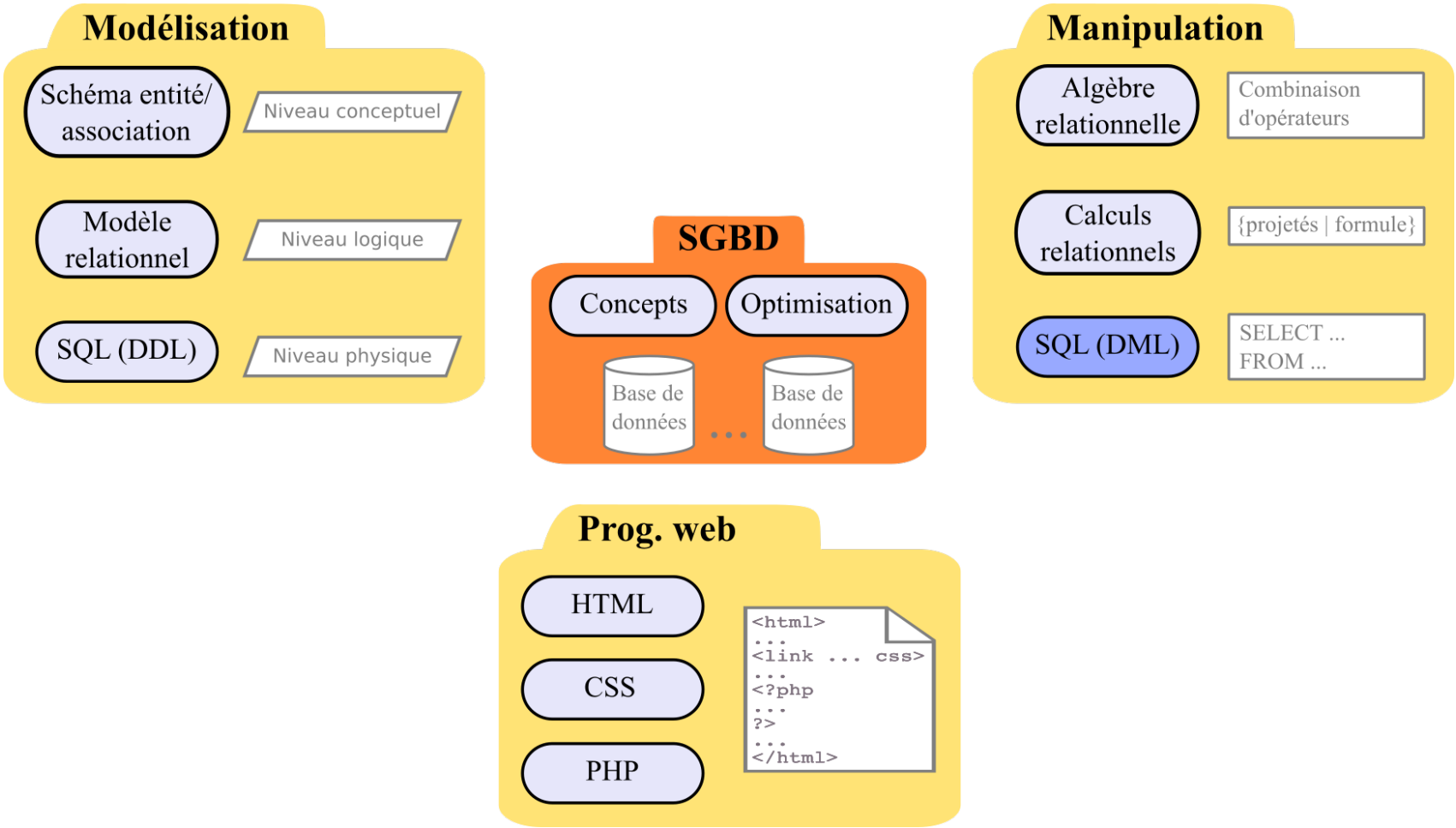
Université Claude Bernard Lyon 1

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)

Rappels

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

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

- Différents types de jointures (utilisées dans la clause FROM), par example NATURAL JOIN ou INNER JOIN ... ON

Motivation

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

Pour trouver les universités qui ont un effectif supérieur à celui de *UCB*, une solution consiste à **récupérer l'effectif de *UCB*** et à le comparer avec l'effectif des autres universités

Sous-requêtes

Une sous-requête est une requête incluse dans une autre requête (principale), i.e., la requête principale utilise le résultat de la sous-requête

- Augmentation de la puissance d'expression du langage
- Les sous-requêtes sont utilisables dans les clauses :
  - SELECT (à condition que pour chaque ligne sélectionnée par la requête principale, on ne sélectionne qu'une ligne dans la sous-requête)
  - FROM (à condition de renommer le résultat)
  - WHERE et HAVING

---

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

Sous-requêtes

Une sous-requête est une requête incluse dans une autre requête (principale), i.e., la requête principale utilise le résultat de la sous-requête

- Augmentation de la puissance d'expression du langage
- Les sous-requêtes sont utilisables dans les clauses :
  - SELECT (à condition que pour chaque ligne sélectionnée par la requête principale, on ne sélectionne qu'une ligne dans la sous-requête)
  - FROM (à condition de renommer le résultat)
  - WHERE et HAVING

Problème : le résultat des requêtes est variable en termes de tuples et d'attributs

---

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

Plan

Sous-requête scalaire

- Retourne une seule valeur (i.e., une ligne et un attribut)

Exemple de requête scalaire :

SELECT nomE
FROM Élève
WHERE idE = 123 ;

nomE
Ana

- S'utilise (généralement) là où une requête attend un nom d'attribut ou une constante
- Ci après, illustration de sous-requêtes scalaires dans trois clauses (WHERE, FROM, SELECT)

---

http://mariadb.com/kb/en/mariadb/subqueries-scalar-subqueries/

Exemple de sous-requête scalaire dans la clause WHERE

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

Table Élève

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

Table Université

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

Table Candidature

Les universités qui sont dans la même ville que *UCB*

```
18 SELECT *
19 FROM Université
20 WHERE ville = (SELECT ville
21     FROM Université
22     WHERE nomU = 'UCB');
```

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000

Exemple de sous-requête scalaire dans la clause FROM

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

Table Élève

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

Table Université

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

Table Candidature

Les universités qui sont dans la même ville que *UCB*

```
34 SELECT *
35 FROM Université u, (SELECT ville
36       FROM Université
37       WHERE nomU = 'UCB') villeUCB
38 WHERE u.ville = villeUCB.ville;
```

nomU	ville	effectif	ville
INSA	Lyon	36000	Lyon
UCB	Lyon	15000	Lyon

http://mariadb.com/kb/en/mariadb/subqueries-in-a-from-clause/

Exemple de sous-requête scalaire dans la clause SELECT

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

Pour chaque université de Lyon, la différence d'effectif avec celui de l'université *UJM*

```
42 SELECT nomU, effectif - (SELECT effectif
43        FROM Université
44        WHERE nomU = 'UJM') AS diff
45 FROM Université
46 WHERE ville = 'Lyon';
```

Exemple de sous-requête scalaire dans la clause SELECT

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

Pour chaque université de Lyon, la différence d'effectif avec celui de l'université *UJM*

```
42 SELECT nomU, effectif - (SELECT effectif
43        FROM Université
44        WHERE nomU = 'UJM') AS diff
45 FROM Université
46 WHERE ville = 'Lyon';
```

nomU	diff
INSA	15000
UCB	-6000

Plan

Sous-requête multi-lignes

- Retourne plusieurs tuples / lignes
- Nécessite des opérateurs (par exemple pour comparer une valeur avec toutes celles retournées par la sous-requête)

Exemple de requête multi-lignes :

```
SELECT nomU
FROM Université
WHERE ville = 'Lyon';
```

nomU
INSA
UCB

Opérateurs pour sous-requêtes multi-lignes - IN, ANY

- *att* □ **ANY** (*sous_requete*)
  avec □ parmi $\{=, <, >, <=, >=, !=\}$
  - vrai s'il existe un *b* parmi les lignes renvoyées par *sous_requete* tel que *att* □ *b* soit vrai

- *att* [**NOT**] **IN** (*sous_requete*)
  - vrai si la valeur de *att* apparaît dans le résultat de *sous_requete* (ou n'apparait pas dans le cas du **NOT IN**)
  - **IN** est équivalent à $=$ **ANY**
  - **NOT IN** est équivalent à $!=$ **ANY**

---

https://mariadb.com/kb/en/mariadb/subqueries-and-any/

Exemple de sous-requêtes multi-lignes - IN

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 à *Lyon*

```
62 SELECT c.idE, nomE
63 FROM Candidature c NATURAL JOIN Élève e
64 WHERE nomU IN (SELECT nomU
65       FROM Université
66       WHERE ville = 'Lyon');
```

idE	nomE
123	Ana
234	Bob
678	Farid
765	Joana
876	Irene
898	Hector

Opérateurs pour sous-requêtes multi-lignes - ALL, EXISTS

- *att* □ **ALL** (*sous_requete*)
  avec □ parmi {=, <, >, <=, >=, !=}
  - vrai si pour toutes les lignes *b* renvoyées par *sous_requete*, *att* □ *b* est vrai

- [**NOT**] **EXISTS** (*sous_requete*)
  - vrai si *sous_requete* retourne au moins un tuple résultat (ou ne retourne pas de tuple dans le cas du **NOT EXISTS**)
  - s'utilise principalement avec sous-requêtes corrélées

---

https://mariadb.com/kb/en/mariadb/subqueries-and-all/
https://mariadb.com/kb/en/mariadb/subqueries-and-exists/

Exemple de sous-requêtes multi-lignes - ALL

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

Table Élève

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

Table Université

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

Table Candidature

La ou les universités qui ont le plus grand effectif

```
70 SELECT *
71 FROM Université
72 WHERE effectif >= ALL (SELECT effectif
73       FROM Université);
```

nomU	ville	effectif
INSA	Lyon	36000

Plan

Sous-requête multi-colonnes

- Retourne plusieurs attributs / colonnes
- Nécessite de former un n-uplet noté **(**att_1**,** att_2**, ...)** qui sera comparé avec le résultat de la sous-requête
- Le n-uplet doit avoir autant d'attributs que le résultat de la sous-requête

Exemple de requête multi-colonnes :

SELECT nomU, ville
FROM Université
WHERE ville = 'Grenoble' ;

nomU	**ville**
UJF	Grenoble

https://mariadb.com/kb/en/mariadb/subqueries-row-subqueries/

Exemple de sous-requête multi-colonnes

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

Table Élève

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

Table Université

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

Table Candidature

Les universités avec même ville et même effectif que *UCB*

```
77 SELECT *
78 FROM Université
79 WHERE (ville, effectif) =
80       (SELECT ville, effectif
81       FROM Université
82       WHERE nomU = 'UCB');
```

nomU	ville	effectif
UCB	Lyon	15000

Sous-requête multi-lignes et multi-colonnes

- Dans ce cas, il faut utiliser à la fois un opérateur (multi-lignes) et former un n-uplet (multi-colonnes)

Exemple : les élèves qui ont candidaté dans la même université et le même département que l'élève *123*

```
SELECT idE, nomU, département
FROM Candidature
WHERE idE != 123 AND
(nomU, département) = ANY (SELECT nomU, département
                           FROM Candidature
                           WHERE idE = 123);
```

idE	nomU	département
898	INSA	informatique
876	UCB	informatique
898	UCB	informatique
345	UJM	électronique

Plan

Sous-requête corrélée

- Une sous-requête corrélée est une sous-requête qui possède une référence (alias) vers une table qui se trouve dans la requête principale
- S'utilise principalement avec [**NOT**] **EXISTS**

Exemple de corrélation :

```
SELECT *
FROM Élève aliasE
WHERE EXISTS (SELECT *
              FROM Candidature c
              WHERE c.idE = aliasE.idE);
```

Exemple de sous-requête corrélée

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

Table Élève

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

Table Université

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

Table Candidature

Les universités qui ont reçu des candidatures

```
86 SELECT nomU
87 FROM Université u
88 WHERE EXISTS (SELECT *
89     FROM Candidature c
90     WHERE c.nomU = u.nomU);
```

nomU
INSA
UCB
UJF
UJM

Plan

Sous-requête imbriquée

- Possible d'imbriquer les sous-requêtes avec plus d'un niveau de profondeur

Exemple de sous-requêtes imbriquées

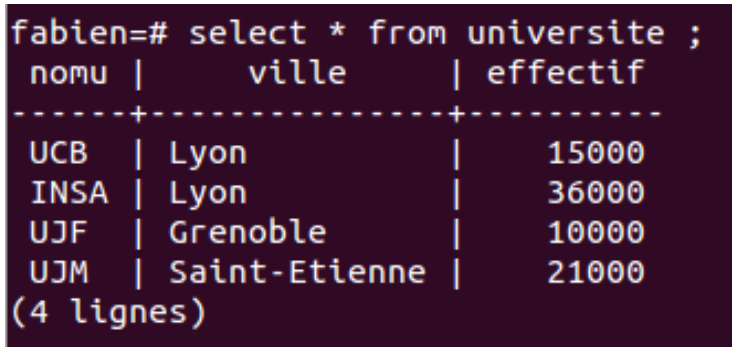
Les universités qui sont dans la même ville qu'une université qui a une plus grande capacité que 'UCB'

```
100 SELECT *
101 FROM Université
102 WHERE ville IN (
103     SELECT DISTINCT ville
104     FROM Université
105     WHERE effectif > (
106         SELECT effectif
107         FROM Université
108         WHERE nomU = 'UCB'));
```

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

En résumé

- Les sous-requêtes permettent d'augmenter la puissance de calcul du langage
- Bien réfléchir au résultat retourné par une sous-requête (scalaire, multi-lignes, multi-colonnes)
- Utiliser la méthode appropriée (opérateur, n-uplet) pour intégrer la sous-requête



Démo avec PostgreSQL (scripts SQL en ligne)