

# Bases de données

## SQL LMD – partie 2 Sélections simples

BD105

v311a

2020-09-22

Département d'informatique  
Faculté des sciences



Christina.Khnaisser@USherbrooke.ca  
<http://info.USherbrooke.ca/ckhnaisser>  
Luc.Lavoie@USherbrooke.ca  
<http://info.USherbrooke.ca/llavoie>

## PLAN

- Historique
- SELECT (version simplifiée)
  - Jointure (et renommage)
  - Restriction
  - Projection, extension et renommage
  - Union, intersection et différence
  - Contexte
- Synthèse
- Exercices
- Références



2020-09-22

BD105-SQL/LMD-02 : Sélections simples (v311a) — Christian Kinnaisser et Luc Laviole  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

## LE LANGAGE SQL

- L'histoire du SELECT
- Correspondance opérateurs relationnels
- Jointure (et renommage)
- Restriction
- Projection et extension
- Union, intersection et différence
- Contexte

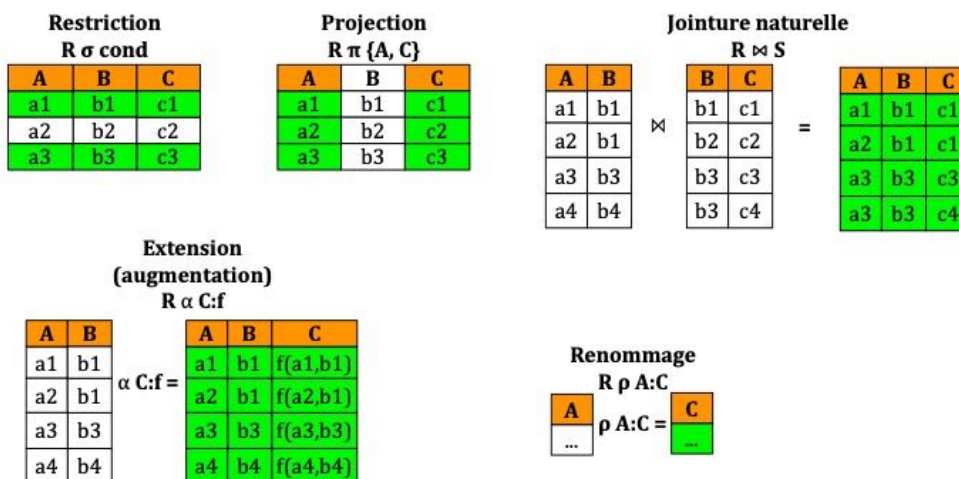
## LE LANGAGE SQL

### L'HISTOIRE DU SELECT

- La séquence d'opérations *jointure-restriction-projection-extension* contribue en pratique à une proportion dominante des requêtes.
- En réunissant cette séquence dans une même instruction, les concepteurs du langage SQL voulaient
  - donner une allure procédurale au langage, jugeant qu'elle faciliterait l'adhésion des programmeurs;
  - offrir un raccourci notationnel aux programmeurs;
  - faciliter l'optimisation conjointe des opérations.
- Les opérateurs ensemblistes seront intégrés par la suite, de façon un peu artificielle.
- Au fil du temps, plusieurs opérateurs (ou pseudo-opérateurs) seront ajoutés – certains d'entre eux seront couverts dans le module suivant.

## LE LANGAGE SQL

### ALGÈBRE RELATIONNELLE - OPÉRATEURS DE BASE DU SELECT



Note : Le symbole de projection  $\pi$  est souvent omis, à l'instar de la multiplication dans les polynômes.

2020-09-22

BDI05-SQL-LMD-02 : Sélections simples (v311a) — Christina Khnaisser et Luc Lavoie  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

5

Opérateurs relationnels propres : 3  
Opérateurs ensemblistes : 4  
Opérateur structurel : 1 (renommage)

IGE 487

Le produit est-il vraiment nécessaire ?

Et le renommage ?

Quel est l'ensemble de base minimal (nécessaire et suffisant) ?

Cet ensemble est-il unique ?

Voir

DATE, C. J. ; DARWEN, H.

*Databases, types, and the relational model: The third manifesto.*

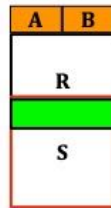
3rd ed., Addison-Wesley Inc., 2008.

ISBN 0-321-39942-0

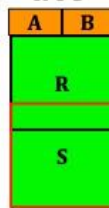
## LE LANGAGE SQL

### ALGÈBRE RELATIONNELLE - OPÉRATEURS ENSEMBLISTES DU SELECT

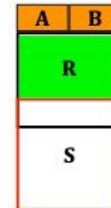
**Intersection**  
 $R \cap S$



**Union**  
 $R \cup S$



**Différence**  
 $R - S$



Note : Le symbole de projection  $\pi$  est souvent omis, à l'instar de la multiplication dans les polynômes.

Opérateurs relationnels propres : 3  
Opérateurs ensemblistes : 4  
Opérateur structurel : 1 (renommage)

IGE 487

Le produit est-il vraiment nécessaire ?

Et le renommage ?

Quel est l'ensemble de base minimal (nécessaire et suffisant) ?

Cet ensemble est-il unique ?

Voir

DATE, C. J. ; DARWEN, H.

*Databases, types, and the relational model: The third manifesto.*

3rd ed., Addison-Wesley Inc., 2008.

ISBN 0-321-39942-0

## LE LANGAGE SQL

### SELECT (SYNTAXE SIMPLIFIÉE)

```

requête ::=
    [ Contexte ]
    SELECT opMode { * | projection-extension }
    FROM listeDeJointures
    [ Restriction ]
    [ Groupement ]
    [ OpComplémentaire ]
    [ Ordonnancement ]
    [ Divers ]
opMode ::=
    [ DISTINCT | ALL ]
  
```

Les catégories en vert  
sont traitées au module  
suivant

- Le mode indique l'expression retourne une relation (**DISTINCT**) ou une collection (**ALL**).
- L'étoile indique que tous les attributs de l'expression sont retenus (sans projection).

```

[ WITH [ RECURSIVE ] requête_with [, ...] ]
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
    * | expression [ [ AS ] nom_d_affichage ] [, ...]
    [ FROM éléments_from [, ...] ]
    [ WHERE condition ]
    [ GROUP BY expression [, ...] ]
    [ HAVING condition [, ...] ]
    [ WINDOW nom_window AS ( définition_window ) [, ...] ]
    [ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
    [ ORDER BY expression [ ASC | DESC | USING opérateur ] [ NULLS { FIRST |
LAST } ] [, ...] ]
    [ LIMIT { nombre | ALL } ]
    [ OFFSET début ] [ ROW | ROWS ] ]
    [ FETCH { FIRST | NEXT } [ total ] { ROW | ROWS } ONLY ]
    [ FOR { UPDATE | SHARE } [ OF nom_table [, ...] ] [ NOWAIT ] [...]]
  
```

avec *éléments\_from* qui peut être :

```

[ ONLY ] nom_table [ * ] [ [ AS ] alias [ ( alias_colonne [, ...] ) ] ]
( select ) [ AS ] alias [ ( alias_colonne [, ...] ) ]
nom_requête_with [ [ AS ] alias [ ( alias_colonne [, ...] ) ] ]
nom_fonction ( [ argument [, ...] ] ) [ AS ] alias [ ( alias_colonne [, ...] |
définition_colonne [, ...] ) ]
nom_fonction ( [ argument [, ...] ] ) AS ( définition_colonne [, ...] )
éléments_from [ NATURAL ] type_jointure éléments_from [ ON condition_jointure |
USING ( colonne_jointure [, ...] ) ]

```

et *requête\_with* est :

```

nom_requête_with ( ( nom_colonne [, ...] ) ) AS ( select | valeurs | insert | update |
delete )

```

```

TABLE [ ONLY ] nom_table [ * ]

```



## LE LANGAGE SQL

### SELECT (SÉQUENCE D'EXÉCUTION)

Les clauses de l'énoncé SELECT sont exécutées dans l'ordre suivant:

1. Contexte
2. FROM ...
3. Restriction
4. Groupement
5. SELECT ...
6. OpComplémentaire
7. Ordonnement
8. Divers

*Nous suivons cet ordre de présentation, sauf pour le contexte qui sera présenté en dernier.*

## LE LANGAGE SQL

### SELECT (SYNTAXE SIMPLIFIÉE) - LISTE DE JOINTURES

*listeDeJointures* ::=  
*denotationTable* [ [ **AS** ] *alias* ] [ *jointure* ... ]  
*denotationTable* ::=  
*nomTable* | ( *requête* ) | *autresDénottations*  
*jointure* ::=  
*jointure\_naturelle* | *produit* | *jointure\_qualifiée* |  
*jointure\_externe*

Les catégories en gris  
ne sont pas traitées dans  
le présent cours.

- Une *listeDeJointures* permet de faire plusieurs jointures à la suite.
- Chaque terme est représenté par une *dénottationTable*, en pratique un nom de table ou une requête entre parenthèses.
- Le mot **AS** permet de (re)nommer la *dénottationTable* en même temps.
- Rappel : une requête n'est pas autre chose qu'une expression **SELECT**.

**LE LANGAGE SQL**  
**SELECT (SYNTAXE SIMPLIFIÉE) – JOINTURE NATURELLE**

*jointure\_naturelle ::=*

**NATURAL** [ **INNER** ] **JOIN** *denotationTable* [ [ **AS** ] *alias* ]

- C'est la jointure de la théorie relationnelle.
- Le mot **INNER** est superfétatoire (sic).
- Le mot **AS** permet de (re)nommer la *dénotationTable* en même temps.

## LE LANGAGE SQL

### SELECT (SYNTAXE SIMPLIFIÉE) - PRODUIT

*produit* ::=  
*opProduit dénotationTable* [ [ **AS** ] *alias* ]

*opProduit* ::=  
**CROSS JOIN** | ,

- C'est le produit de la théorie relationnelle.
- Le mot **AS** permet de (re)nommer la *dénotationTable* en même temps.

## LE LANGAGE SQL

### SELECT (SYNTAXE SIMPLIFIÉE) – JOINTURE QUALIFIÉE

*jointure\_qualifiée* ::=

[ **INNER** ] **JOIN** *denotationTable* [ [ **AS** ] *alias* ] *qualificationJ*

*qualificationJ* ::=

**ON** *conditionJ*

| **USING** ( *listeNomCol* )

- La *qualificationJ* permet de renommer les attributs de jointures en même temps.
- Le mot **INNER** est superfétatoire (sic).
- Le mot **AS** permet de (re)nommer la *dénotationTable* en même temps.

**LE LANGAGE SQL****JOINTURES, SIMPLIFIONS!**

- Les jointures qualifiées sont les jointures de prédilection
  - **USING** ( $a_1, \dots, a_n$ )  
lorsque l'attribut est le même dans les deux tables.
  - **ON**  $a_1=b_1$  **AND** ... **AND**  $a_n=b_n$   
lorsque les noms d'attributs ne sont pas les mêmes.

**Note**

- La variante ON est beaucoup plus riche qu'indiqué, mais il existe de nombreuses bonnes raisons pour ne pas utiliser cette richesse. L'exposé de ces raisons dépasse toutefois la portée du cours. En conséquence, nous nous limiterons à la seule forme recommandable.

## ÉVALUATION – EXEMPLE DE REQUÊTES (INNER JOIN)

1. Quels sont les types d'évaluation ?
2. Quels sont les types d'évaluation ayant des notes ?
3. Quels sont les étudiants.es inscrits.es ?
4. Quels sont les étudiants.es non inscrits.es ?

2020-09-22

BD105-SQL/LMD-02 : Sélections simples (V311a) — Christian Khmaliset et Luc Lavote  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

14

```
SELECT * FROM TypeEvaluation;
```

```
SELECT * FROM TypeEvaluation JOIN Resultat ON (TE =code);
```

```
SELECT * FROM Etudiant JOIN Resultat USING(matricule);
```

```
-- V2 intersect
```

```
SELECT matricule FROM Etudiant
```

```
INTERSECT
```

```
SELECT matricule FROM Resultat;
```

```
SELECT * FROM Etudiant LEFT JOIN Resultat USING(matricule);
```

```
-- V2 différence
```

```
SELECT matricule FROM Etudiant
```

```
EXCEPT
```

```
SELECT matricule FROM Resultat;
```

Voir le code :

Exemples/Evaluation/Evaluation\_req.sql

## ÉVALUATION – EXEMPLE DE DONNÉES (INNER JOIN)

### Étudiant

matricule	nom	adresse
15113150	Paul	>Δ <sup>ς</sup> σ <sup>ς</sup> ϑ <sup>ς</sup>
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

### Activité

sigle	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

### TypeÉvaluation

code	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

### Résultat

matricule	TE	activité	trimestre	note
15113150	TP	IFT187	20133	80
15112354	FI	IFT187	20123	78
15113150	TP	IFT159	20133	75
15112354	FI	GMQ103	20123	85
15110132	IN	IMN117	20123	90
15110132	IN	IFT187	20133	45
15112354	FI	IFT159	20123	52

On remarque l'omission de la dénotation des types des attributs.

C'est fréquemment le cas dans les représentations graphiques, afin de les alléger.

Cela ne porte pas à conséquence dans la mesure où une définition textuelle les accompagne, ce qui est le cas ici.

Nous verrons bientôt d'autres représentations graphiques plus complètes.

Note : >Δ<sup>ς</sup>σ<sup>ς</sup>ϑ<sup>ς</sup> se prononce (approximativement) Puvirnitug



**AU TABLEAU**

### COMMENT LES JOINTURES DE SQL GÈRENT-ELLES LES ATTRIBUTS DU RÉSULTAT ?

La jointure naturelle (celle de la théorie relationnelle et le NATURAL JOIN de SQL) opère sur les attributs de même nom et élimine (automatiquement) la redondance des attributs de leur résultat, par exemple :

```
select *  
from Resultat natural join Etudiant
```

aura pour entête, puisque matricule est le seul attribut commun :  
matricule, TE, activité, trimestre, note, nom, adresse

En conséquence, la référence à l'attribut matricule dans la clause SELECT ne pose aucun problème, par exemple

```
select matricule, nom, TE, activité, trimestre, note  
from Resultat natural join Etudiant
```

## COMMENT LES JOINTURES DE SQL GÈRENT-ELLES LES ATTRIBUTS DU RÉSULTAT ?

Par contre, si on utilise la jointure conditionnelle (JOIN ... ON), ça ne sera pas le cas, ainsi :

```
select *  
from Resultat join Etudiant  
on Resultat.matricule = Etudiant.matricule
```

aura pour entête :

```
Resultat.matricule, TE, activité, trimestre, note,  
Etudiant.matricule, nom, adresse
```

En conséquence, la référence à l'attribut matricule dans la clause SELECT devra préciser lequel des deux est souhaité (même s'ils sont rigoureusement égaux!), par exemple :

```
select  
Etudiant.matricule, nom, TE, activité, trimestre, note  
from Resultat join Etudiant  
on Resultat.matricule = Etudiant.matricule
```

## COMMENT LES JOINTURES DE SQL GÈRENT-ELLES LES ATTRIBUTS DU RÉSULTAT ?

Qu'en est-il alors de la jointure restreinte (JOIN ... USING) ?

Son comportement est analogue à celui de la jointure naturelle pour les attributs communs retenus par la clause USING et analogue à la jointure conditionnelle pour les autres attributs communs.

:-)

Comme quoi une tentative de simplification du renommage peut aussi entraîner parfois une augmentation de la complexité.

**LE LANGAGE SQL****SELECT (SYNTAXE SIMPLIFIÉE) – JOINTURE EXTERNE**

*jointure\_externe ::=*

*jExterne JOIN denotationTable [ [ AS ] alias ] qualificationJ*

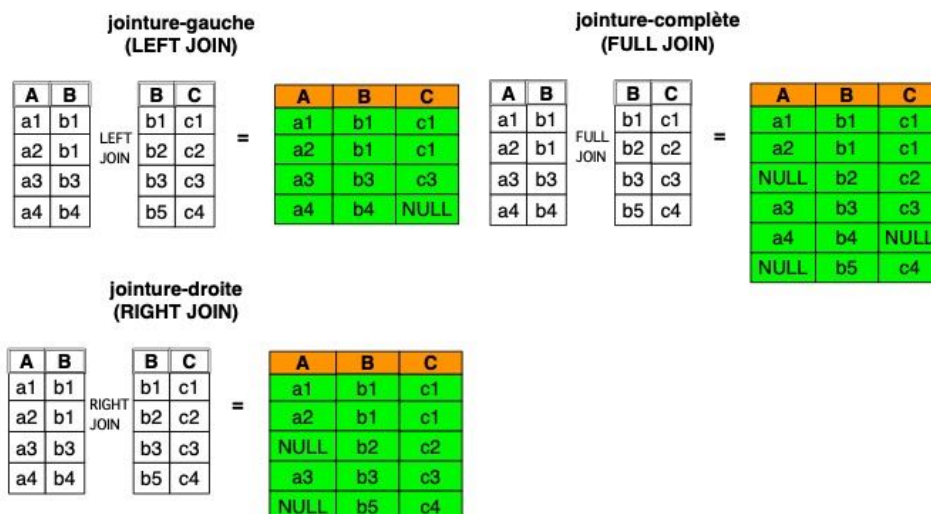
*jExterne ::=*

**LEFT [OUTER] | RIGHT [OUTER] | FULL [OUTER]**

- Une jointure externe permet de ne pas « perdre » les données des tuples sans correspondant en affectant des NULL aux attributs de valeur inconnue.
- Le mot **OUTER** est superfétatoire (sic).
- Le mot **AS** permet de (re)nommer la *dénotationTable* en même temps.

## LE LANGAGE SQL

### ILLUSTRATION JOINTURE EXTERNE



```

CREATE TABLE R (A CHAR(2) PRIMARY KEY, B CHAR(2));
CREATE TABLE S (B CHAR(2) PRIMARY KEY, C CHAR(2));
INSERT INTO R VALUES
  ('a1', 'b1'),
  ('a2', 'b1'),
  ('a3', 'b3'),
  ('a4', 'b4');
INSERT INTO S VALUES
  ('b1', 'c1'),
  ('b2', 'c2'),
  ('b3', 'c3'),
  ('b5', 'c4');
SELECT * FROM R LEFT JOIN S USING(B);
SELECT * FROM R RIGHT JOIN S USING(B);
SELECT * FROM R FULL JOIN S USING(B);

```

## LE LANGAGE SQL

### JOINTURES EXTERNES SIMPLIFIÉES

```
SELECT C.nom, T.tel  
FROM  
  C RIGHT JOIN T ON C.id = T.id
```

*est équivalent à :*

```
SELECT C.nom, T.tel  
FROM  
  T LEFT JOIN C ON C.id = T.id
```

```
SELECT C.nom, T.tel  
FROM  
  C FULL JOIN T ON C.id = T.id
```

*est équivalent à :*

```
SELECT C.nom, T.tel  
FROM  
  C RIGHT JOIN T ON C.id = T.id  
UNION  
SELECT C.nom, T.tel  
FROM  
  C LEFT JOIN T ON C.id = T.id
```

## ÉVALUATION – EXEMPLE DE REQUÊTES (OUTER JOIN)

1. Quels sont les étudiants qui n'ont aucune évaluation à ce jour ?



## ÉVALUATION – EXEMPLE DE DONNÉES (OUTER JOIN)

### Étudiant

matricule	nom	adresse
15113150	Paul	>Δ <sup>ς</sup> σ <sup>ς</sup> ϑ <sup>ς</sup> b
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

### Activité

sigle	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

### TypeÉvaluation

code	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

### Résultat

matricule	TE	activité	trimestre	note
15113150	TP	IFT187	20133	80
15112354	FI	IFT187	20123	78
15113150	TP	IFT159	20133	75
15112354	FI	GMQ103	20123	85
15110132	IN	IMN117	20123	90
15110132	IN	IFT187	20133	45
15112354	FI	IFT159	20123	52

On remarque l'omission de la dénotation des types des attributs.

C'est fréquemment le cas dans les représentations graphiques, afin de les alléger.

Cela ne porte pas à conséquence dans la mesure où une définition textuelle les accompagne, ce qui est le cas ici.

Nous verrons bientôt d'autres représentations graphiques plus complètes.

Note : >Δ<sup>ς</sup>σ<sup>ς</sup>ϑ<sup>ς</sup>b se prononce (approximativement) Puvirnitug

**AU TABLEAU**

- À venir... quand nous traiterons des attributs annulables

2020-09-22

BD105-SQL/LMD-02 : Sélections simples (v311a) — Christian Rinaisier et Luc Laviole  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

25

-- Solution stricte

WITH

NbA AS

(

SELECT matricule, COUNT(\*) AS nbAct

FROM Resultat

GROUP BY matricule

)

SELECT matricule, nom, adresse, COALESCE (nbAct,0) AS nbAct

FROM Etudiant LEFT JOIN NbA USING(matricule)

ORDER BY matricule ;

-- solution avec induction, mais est-elle bonne ? Pourquoi ?

SELECT matricule, nom, adresse, COUNT(\*) AS nbAct

FROM Etudiant LEFT JOIN Resultat USING(matricule)

GROUP BY matricule

ORDER BY matricule ;

```
-- solution avec induction, mais est-elle bonne ? Pourquoi ?  
SELECT matricule, nom, adresse, COUNT(activite) AS nbAct  
FROM Etudiant LEFT JOIN Resultat USING(matricule)  
GROUP BY matricule  
ORDER BY matricule ;
```

## LE LANGAGE SQL SELECT (SYNTAXE SIMPLIFIÉE) - RESTRICTION

*Restriction ::=*

**WHERE** *condition*

- C'est la restriction de la théorie relationnelle.
- La *condition* suit la syntaxe présentée antérieurement. Les attributs y sont limités aux seuls attributs obtenus par la jointure de la clause **FROM** qui précède.

### ÉVALUATION – EXEMPLE DE REQUÊTES (RESTRICTION)

1. Quels sont les étudiants inscrits en IFT 187?
2. Quels sont les étudiants inscrits à une activité d'informatique à l'automne 2013?
3. Quels sont les étudiants en situation d'échec au final à l'automne 2012?
4. Produire le relevé de notes d'Éliane.

2020-09-22

BD105-SQL/LMD-02 : Sélections simples (v311a) — Christian Rinaisser et Luc Lavoté  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

27

Voir

Exemples/Evaluation/Evaluation\_req.sql

Exercices

Lister les résultats des TP

Lister les résultats des TP où la note est < 50

Lister les résultats des TP où la note entre 70 et 100

Liste les activités d'informatique

## ÉVALUATION – EXEMPLE DE DONNÉES (RESTRICTION)

### Étudiant

matricule	nom	adresse
15113150	Paul	>Δ <sup>ς</sup> σ <sup>ς</sup> ᵇ
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

### Activité

sigle	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

### TypeÉvaluation

code	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

### Résultat

matricule	TE	activité	trimestre	note
15113150	TP	IFT187	20133	80
15112354	FI	IFT187	20123	78
15113150	TP	IFT159	20133	75
15112354	FI	GMQ103	20123	85
15110132	IN	IMN117	20123	90
15110132	IN	IFT187	20133	45
15112354	FI	IFT159	20123	52

On remarque l'omission de la dénotation des types des attributs.

C'est fréquemment le cas dans les représentations graphiques, afin de les alléger.

Cela ne porte pas à conséquence dans la mesure où une définition textuelle les accompagne, ce qui est le cas ici.

Nous verrons bientôt d'autres représentations graphiques plus complètes.

Note : >Δ<sup>ς</sup>σ<sup>ς</sup>ᵇ se prononce (approximativement) Puvirnitug

**AU TABLEAU**

**LE LANGAGE SQL****SELECT (SYNTAXE SIMPLIFIÉE) – PROJECTION ET EXTENSION**

*projection-extension* ::=

{ *expression* [ [ **AS** ] *nomCol* ] ... , }

- Tout retenir (par **SELECT \***) est peu flexible.
- La clause *projection-extension* permet de ne retenir que les expressions désirées et de les nommer.
- Une *expression* suit la syntaxe qui a été présentée antérieurement. Les attributs y sont limités aux seuls attributs obtenus par la jointure de la clause **FROM** qui suit.
- Lorsqu'une expression se réduit à un identificateur, il s'agit en fait simplement d'une projection de la théorie relationnelle.



### ÉVALUATION – EXEMPLE DE REQUÊTES (PROJECTION ET EXTENSION)

1. Lister le sigle et le titre des activités d'informatique.
2. Lister le sigle des activités ayant au moins un étudiant inscrit.
3. Étendre la table résultat pour ajouter la cote 'A' si la note est  $\geq 90$  et la cote 'B' si la note  $\geq 70$  et  $< 90$ .
4. Lister les notes pour tous les étudiants. Afficher '0' si l'étudiant n'a pas de note pour l'activité.

2020-09-22

BD105-SQL/LMD-02 : Sélections simples (v311a) — Christian Khmaliset et Luc Lavote  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

31

```
SELECT sigle, titre FROM Activite WHERE SUBSTR(sigle, 1, 3) IN ('IFT', 'IGE');
```

```
SELECT activite FROM Resultat;  
SELECT DISTINCT activite FROM Resultat;
```

```
SELECT *,  
    CASE  
        WHEN note  $\geq$  90 THEN 'A'  
        WHEN note BETWEEN 70 AND 89 THEN 'B'  
        ELSE 'C'  
    END AS cote  
FROM Resultat;
```

```
SELECT matricule, te, activite, trimestre, COALESCE(note, 0)  
FROM Etudiant LEFT JOIN Resultat USING(matricule);
```

Voir

Exemples/Evaluation/Evaluation\_req.sql

## ÉVALUATION – EXEMPLE DE DONNÉES (PROJECTION ET EXTENSION)

### Étudiant

matricule	nom	adresse
15113150	Paul	>Δ <sup>ς</sup> σ <sup>ς</sup> ᵇ
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

### Activité

sigle	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

### TypeÉvaluation

code	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

### Résultat

matricule	TE	activité	trimestre	note
15113150	TP	IFT187	20133	80
15112354	FI	IFT187	20123	78
15113150	TP	IFT159	20133	75
15112354	FI	GMQ103	20123	85
15110132	IN	IMN117	20123	90
15110132	IN	IFT187	20133	45
15112354	FI	IFT159	20123	52

On remarque l'omission de la dénotation des types des attributs.

C'est fréquemment le cas dans les représentations graphiques, afin de les alléger.

Cela ne porte pas à conséquence dans la mesure où une définition textuelle les accompagne, ce qui est le cas ici.

Nous verrons bientôt d'autres représentations graphiques plus complètes.

Note : >Δ<sup>ς</sup>σ<sup>ς</sup>ᵇ se prononce (approximativement) Puvirnitug

## AU TAQBLEAU

2020-09-22

BD105-SQT/LMD-02 : Sélections simples (431A) — Christiana Rinaisser et Luc Laviole  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

**LE LANGAGE SQL****SELECT (SYNTAXE SIMPLIFIÉE) - OpCOMPLÉMENTAIRE**

*OpComplémentaire ::=*

- INTERSECT opMode requête*
- | *UNION opMode requête*
- | *EXCEPT opMode requête*
- | *jointure [ opComplémentaire ]*

- L'expression d'opération ensembliste est donc lourde, mais fonctionnelle.
- Attention à *opMode*!
- En pratique, la deuxième forme de jointure est inutile (surtout quand nous aurons le *Contexte*).

### ÉVALUATION – EXEMPLE DE REQUÊTES (U, I, D)

1. Lister les étudiants qui ont des notes.
2. Lister les étudiants qui n'ont pas eu de note.
3. ...

## ÉVALUATION – EXEMPLE DE DONNÉES (U, I, D)

### Étudiant

matricule	nom	adresse
15113150	Paul	>Δ <sup>ς</sup> σ <sup>ς</sup> ϑ <sup>ς</sup>
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

### Activité

sigle	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

### TypeÉvaluation

code	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

### Résultat

matricule	TE	activité	trimestre	note
15113150	TP	IFT187	20133	80
15112354	FI	IFT187	20123	78
15113150	TP	IFT159	20133	75
15112354	FI	GMQ103	20123	85
15110132	IN	IMN117	20123	90
15110132	IN	IFT187	20133	45
15112354	FI	IFT159	20123	52

On remarque l'omission de la dénotation des types des attributs.

C'est fréquemment le cas dans les représentations graphiques, afin de les alléger.

Cela ne porte pas à conséquence dans la mesure où une définition textuelle les accompagne, ce qui est le cas ici.

Nous verrons bientôt d'autres représentations graphiques plus complètes.

Note : >Δ<sup>ς</sup>σ<sup>ς</sup>ϑ<sup>ς</sup> se prononce (approximativement) Puvirnitug

2020-09-22

BD105-SQT/LMD-02 : Sélections simples (4311a) — Christiana Rinaisser et Luc Laviole  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

37

## LE LANGAGE SQL SELECT (SYNTAXE SIMPLIFIÉE) - CONTEXTE

*Contexte* ::=

**WITH** [ **RECURSIVE** ] { *requêteCont* ... , }

*requêteCont* ::=

*nomReqCont* [ ( { *nomCol* ... , } ) ] **AS** ( *opDefTable* )

*opDefTable* ::=

*requête* | *valeurs* | *insertion* | *miseAJour* | *retrait*

- Ceci est tout simplement la notation mathématique usuelle permettant de définir des sous-expressions (*opDefTable*) en leur donnant un nom (*nomReqCont*) utilisable dans de l'expression qui suit afin d'en faciliter l'écriture... et la lecture!
- Nous l'utiliserons abondamment.



**GÉRER LA COMPLEXITÉ DES REQUÊTES****EMBOITEMENT VS WITH**

- Au sein d'une requête, le plus souvent, moyennant quelques contorsions syntaxiques, il est possible d'utiliser une expression SELECT à la place d'une référence à une table.
- Assez rapidement, la lisibilité en souffre.
- L'énoncé WITH devient alors l'instrument privilégié de décomposition syntaxique (diviser pour régner).
- Le seul moment où il est nécessaire de recourir à l'emboitement : lorsque le SELECT fait référence à une variable liée définie dans le contexte englobant.

## ÉVALUATION – EXEMPLE DE REQUÊTES (WITH)

1. Quels étudiants ne sont inscrits à aucune activité en 2013 ?
2. Lister la note des examens finaux des activités d'informatique du trimestre 20123 pour tous les étudiants. Afficher '0' si l'étudiant n'a pas de note pour l'activité.

2020-09-22

BD105-SQL/LMD-02 : Sélections simples (311a) — Christian Rinaisier et Luc Lavot  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

40

### WITH

```
Inscrit2013 AS -- (Résultat  $\sigma$  (20131≤trimestre≤20133))  $\pi$  {matricule}
(
  SELECT matricule
  FROM Resultat
  WHERE trimestre BETWEEN '20131' AND '20133'
),
NonInscrit2013 AS -- Étudiant – (Étudiant  $\bowtie$  Inscrits2013)  $\pi$  {matricule}
(
  SELECT *
  FROM Etudiant
  EXCEPT
  SELECT *
  FROM Etudiant JOIN Inscrit2013 USING (matricule)
)
SELECT *
FROM NonInscrit2013
;
```

-- Très créatif :-)

```
WITH ActiviteInfo AS
(
  SELECT sigle as activite
  FROM Activite
  WHERE SUBSTR(sigle, 1, 3) IN ('IFT', 'IGE', 'IMN')
),
EtudiantInfo AS
(
  SELECT matricule, activite, '20123' AS trimestre, 'FI' AS te
  FROM Etudiant CROSS JOIN ActiviteInfo
),
etudiantNote AS
(
  SELECT matricule,
         activite,
         trimestre,
         te,
         coalesce(note, 0) AS note
  FROM Resultat RIGHT JOIN EtudiantInfo
  USING(matricule, activite, trimestre, te)
)
SELECT * FROM etudiantNote
WHERE te = 'FI' AND trimestre = '20123';
```

## ÉVALUATION – EXEMPLE DE DONNÉES (WITH)

### Étudiant

matricule	nom	adresse
15113150	Paul	>Δ <sup>ς</sup> σ <sup>ς</sup> ϑ <sup>ς</sup>
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

### Activité

sigle	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

### TypeÉvaluation

code	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

### Résultat

matricule	TE	activité	trimestre	note
15113150	TP	IFT187	20133	80
15112354	FI	IFT187	20123	78
15113150	TP	IFT159	20133	75
15112354	FI	GMQ103	20123	85
15110132	IN	IMN117	20123	90
15110132	IN	IFT187	20133	45
15112354	FI	IFT159	20123	52

On remarque l'omission de la dénotation des types des attributs.

C'est fréquemment le cas dans les représentations graphiques, afin de les alléger.

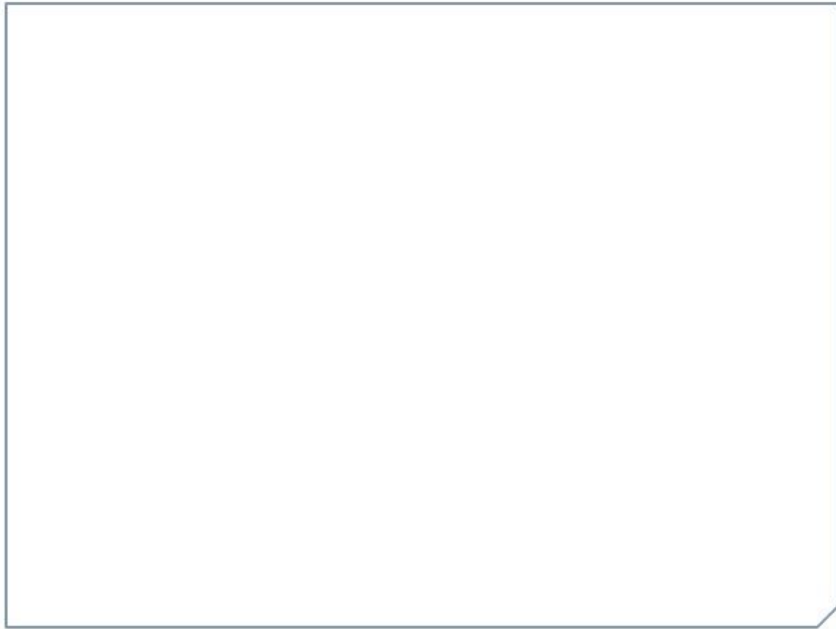
Cela ne porte pas à conséquence dans la mesure où une définition textuelle les accompagne, ce qui est le cas ici.

Nous verrons bientôt d'autres représentations graphiques plus complètes.

Note : >Δ<sup>ς</sup>σ<sup>ς</sup>ϑ<sup>ς</sup> se prononce (approximativement) Puvirnitug



## SYNTHÈSE



2020-09-22

BD105-SQT/LMD-02 : Sélections simples (4311a) — Christian Rinaisier et Luc Laviole  
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

## LE LANGAGE SQL SELECT (EXEMPLES SIMPLES)

Norme ISO

Dialecte

PostgreSQL

### Soit les relations (tables)

- A (x, y, z)
- B (v, w)
- C (v, w, x)
- D (x, y, z)

- Renommage :  $A \rho \{x:q\}$ 
  - SELECT x AS q, y, z FROM A
- Restriction :  $A \sigma \text{ cond}$ 
  - SELECT \* FROM A WHERE cond
- Projection :  $A \pi \{y, z\}$ 
  - SELECT DISTINCT y, z FROM A
- Produit :  $A \times B$ 
  - SELECT \* FROM A, B
- Jointure :  $A \bowtie C$ 
  - SELECT \* FROM A NATURAL JOIN C
- Union :  $A \cup D$ 
  - SELECT \* FROM A UNION DISTINCT  
SELECT \* FROM D
- Intersection :  $A \cap D$ 
  - SELECT \* FROM A INTERSECT DISTINCT  
SELECT \* FROM D
- Différence :  $A - D$ 
  - SELECT \* FROM A EXCEPT DISTINCT  
SELECT \* FROM D

## LE LANGAGE SQL SELECT (EXEMPLES SIMPLES)

### Dialecte Oracle

### Soit les relations (tables)

- A (x, y, z)
- B (v, w)
- C (v, w, x)
- D (x, y, z)

- Renommage :  $A \rho \{x:q\}$ 
  - SELECT x AS q, y, z FROM A
- Restriction :  $A \sigma \text{ cond}$ 
  - SELECT \* FROM A WHERE cond
- Projection :  $A \pi \{y, z\}$ 
  - SELECT DISTINCT y, z FROM A
- Produit :  $A \times B$ 
  - SELECT \* FROM A, B
- Jointure :  $A \bowtie C$ 
  - SELECT \* FROM A NATURAL JOIN C
- Union :  $A \cup D$ 
  - SELECT \* FROM A UNION DISTINCT  
SELECT \* FROM D
- Intersection :  $A \cap D$ 
  - SELECT \* FROM A INTERSECT DISTINCT  
SELECT \* FROM D
- Différence :  $A - D$ 
  - SELECT \* FROM A MINUS DISTINCT  
SELECT \* FROM D



## LE LANGAGE SQL

### JOINTURES EXTERNES SIMPLIFIÉES

```
SELECT C.nom, T.tel  
FROM  
  C RIGHT JOIN T ON C.id = T.id
```

*est équivalent à :*

```
SELECT C.nom, T.tel  
FROM  
  T LEFT JOIN C ON C.id = T.id
```

```
SELECT C.nom, T.tel  
FROM  
  C FULL JOIN T ON C.id = T.id
```

*est équivalent à :*

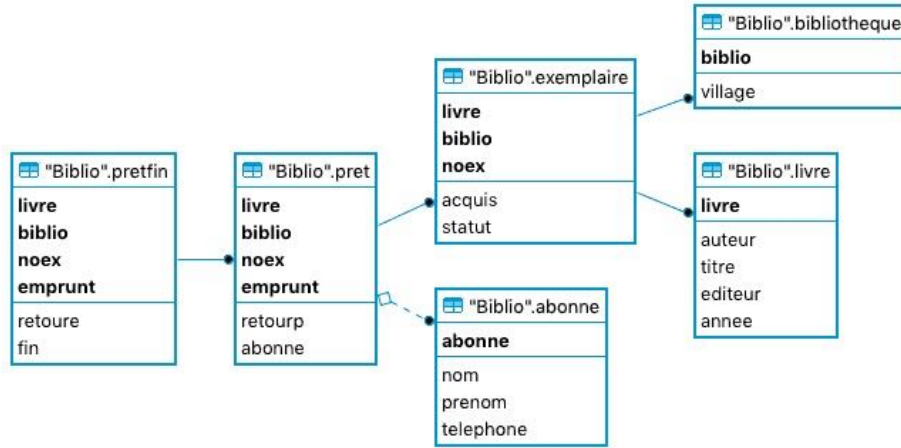
```
SELECT C.nom, T.tel  
FROM  
  C RIGHT JOIN T ON C.id = T.id  
UNION  
SELECT C.nom, T.tel  
FROM  
  C LEFT JOIN T ON C.id = T.id
```

## EXERCICES

- Bibliovik
- ... et les autres

## EXERCICE

### BIBLIOVIK – DIAGRAMME RELATIONNEL



**EXERCICES****BIBLIOVIK - REQUÊTES**

- Lister les abonnés.
- Lister les livres de l'auteur « Michel Tremblay », afficher le titre et l'année seulement.
- Lister les livres acquis en 2019 (le livre doit apparaître une seule fois).
- Quels sont les livres apparus en 2020 qui sont commandés ?
- Quels sont les livres prêtés à l'abonné A123456 ?
- Quels sont les emprunts en retard ?
- Quels sont les livres qui n'existent pas dans la bibliothèque ?

### ...ET LES AUTRES

- Le répertoire public de cours
  - <smb://dinf-argus.dinf.fsci.usherbrooke.ca/public/Cours/IFT187/Exemples>
- propose plusieurs autres exemples, dont
  - Evaluation
  - Pharmacie
  - Gaspard

## RÉFÉRENCES

- Elmasri et Navathe (4<sup>e</sup> ed.), chapitre 7
- Elmasri et Navathe (6<sup>e</sup> ed.), chapitre 4
- [Loney2008]  
Loney, Kevin ;  
*Oracle Database 11g: The Complete Reference.*  
Oracle Press/McGraw-Hill/Osborne, 2008.  
ISBN 978-0071598750.
- [Date2012]  
Date, Chris J. ;  
*SQL and Relational Theory : How to Write Accurate SQL Code.*  
2<sup>nd</sup> edition, O'Reilly, 2012.  
ISBN 978-1-449-31640-2.
- Le site d'Oracle (en anglais)
  - [http://docs.oracle.com/cd/E11882\\_01/index.htm](http://docs.oracle.com/cd/E11882_01/index.htm)
- Le site de PostgreSQL (en français)
  - <http://docs.postgresqlfr.org>

