

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



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

Restriction
 $R \sigma \text{ cond}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Projection
 $R \pi \{A, C\}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Jointure naturelle
 $R \bowtie S$

A	B
a1	b1
a2	b1
a3	b3
a4	b4

 \bowtie

B	C
b1	c1
b2	c2
b3	c3
b3	c4

 $=$

A	B	C
a1	b1	c1
a2	b1	c1
a3	b3	c3
a3	b3	c4

Extension
(augmentation)
 $R \alpha C:f$

A	B
a1	b1
a2	b1
a3	b3
a4	b4

 $\alpha C:f =$

A	B	C
a1	b1	f(a1,b1)
a2	b1	f(a2,b1)
a3	b3	f(a3,b3)
a4	b4	f(a4,b4)

Renommage
 $R \rho A:C$

A
...

 $\rho A:C =$

C
...

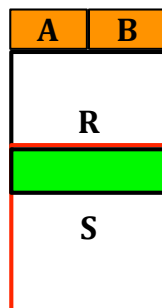
Note : Le symbole de projection π est souvent omis, à l'instar de la multiplication dans les polynômes.

LE LANGAGE SQL

ALGÈBRE RELATIONNELLE - OPÉRATEURS ENSEMBLISTES DU SELECT

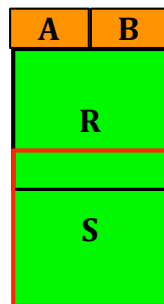
Intersection

$$R \cap S$$



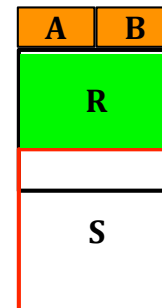
Union

$$R \cup S$$



Différence

$$R - S$$



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

LE LANGAGE SQL

SELECT (SYNTAXE SIMPLIFIÉE)

requête ::=

[*Contexte*]

SELECT *opMode* { * | *projection-extension* }

FROM *listeDeJointures*

[*Restriction*]

[**Groupement**]

[*OpComplémentaire*]

[**Ordonnement**]

[**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).

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. Ordonnancement
8. Divers

Nous suivrons 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*) | *autresDénnotations*

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 *dénotationTable*, en pratique un nom de table ou une requête entre parenthèses.
- Le mot **AS** permet de (re)nommer la *dénotationTable* 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, SIMPLIFICATIONS!

- 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 ?

ÉVALUATION – EXEMPLE DE DONNÉES (INNER JOIN)

Étudiant

<u>matricule</u>	nom	adresse
15113150	Paul	>Δ ^ϕ σ▷ ^ϕ b
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

Activité

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

TypeÉvaluation

<u>code</u>	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

Résultat

<u>matricule</u>	<u>TE</u>	<u>activité</u>	<u>trimestre</u>	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

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

**jointure-gauche
(LEFT JOIN)**

A	B
a1	b1
a2	b1
a3	b3
a4	b4

 LEFT JOIN

B	C
b1	c1
b2	c2
b3	c3
b5	c4

 =

A	B	C
a1	b1	c1
a2	b1	c1
a3	b3	c3
a4	b4	NULL

**jointure-complète
(FULL JOIN)**

A	B
a1	b1
a2	b1
a3	b3
a4	b4

 FULL JOIN

B	C
b1	c1
b2	c2
b3	c3
b5	c4

 =

A	B	C
a1	b1	c1
a2	b1	c1
NULL	b2	c2
a3	b3	c3
a4	b4	NULL
NULL	b5	c4

**jointure-droite
(RIGHT JOIN)**

A	B
a1	b1
a2	b1
a3	b3
a4	b4

 RIGHT JOIN

B	C
b1	c1
b2	c2
b3	c3
b5	c4

 =

A	B	C
a1	b1	c1
a2	b1	c1
NULL	b2	c2
a3	b3	c3
NULL	b5	c4

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

<u>matricule</u>	nom	adresse
15113150	Paul	>Δ ^ϕ σ▷ ^ϕ b
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

Activité

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

TypeÉvaluation

<u>code</u>	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

Résultat

<u>matricule</u>	<u>TE</u>	<u>activité</u>	<u>trimestre</u>	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

AU TABLEAU

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

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.

ÉVALUATION – EXEMPLE DE DONNÉES (RESTRICTION)

Étudiant

<u>matricule</u>	nom	adresse
15113150	Paul	>Δ ^ϕ σ▷ ^{ϕb}
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

Activité

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

TypeÉvaluation

<u>code</u>	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

Résultat

<u>matricule</u>	<u>TE</u>	<u>activité</u>	<u>trimestre</u>	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

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 ≥ 90 et la cote 'B' si la note ≥ 70 et < 90 .
4. Lister les notes pour tous les étudiants. Afficher '0' si l'étudiant n'a pas de note pour l'activité.

ÉVALUATION – EXEMPLE DE DONNÉES (PROJECTION ET EXTENSION)

Étudiant

<u>matricule</u>	nom	adresse
15113150	Paul	>Δ ^ϕ σ▷ ^ϕ b
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

Activité

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

TypeÉvaluation

<u>code</u>	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

Résultat

<u>matricule</u>	<u>TE</u>	<u>activité</u>	<u>trimestre</u>	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

AU TABLEAU

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

<u>matricule</u>	nom	adresse
15113150	Paul	>Δ ^ϕ σ▷ ^ϕ b
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

Activité

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

TypeÉvaluation

<u>code</u>	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

Résultat

<u>matricule</u>	<u>TE</u>	<u>activité</u>	<u>trimestre</u>	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

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.

ÉVALUATION – EXEMPLE DE DONNÉES (WITH)

Étudiant

<u>matricule</u>	nom	adresse
15113150	Paul	>Δ ^ϕ σ▷ ^ϕ b
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

Activité

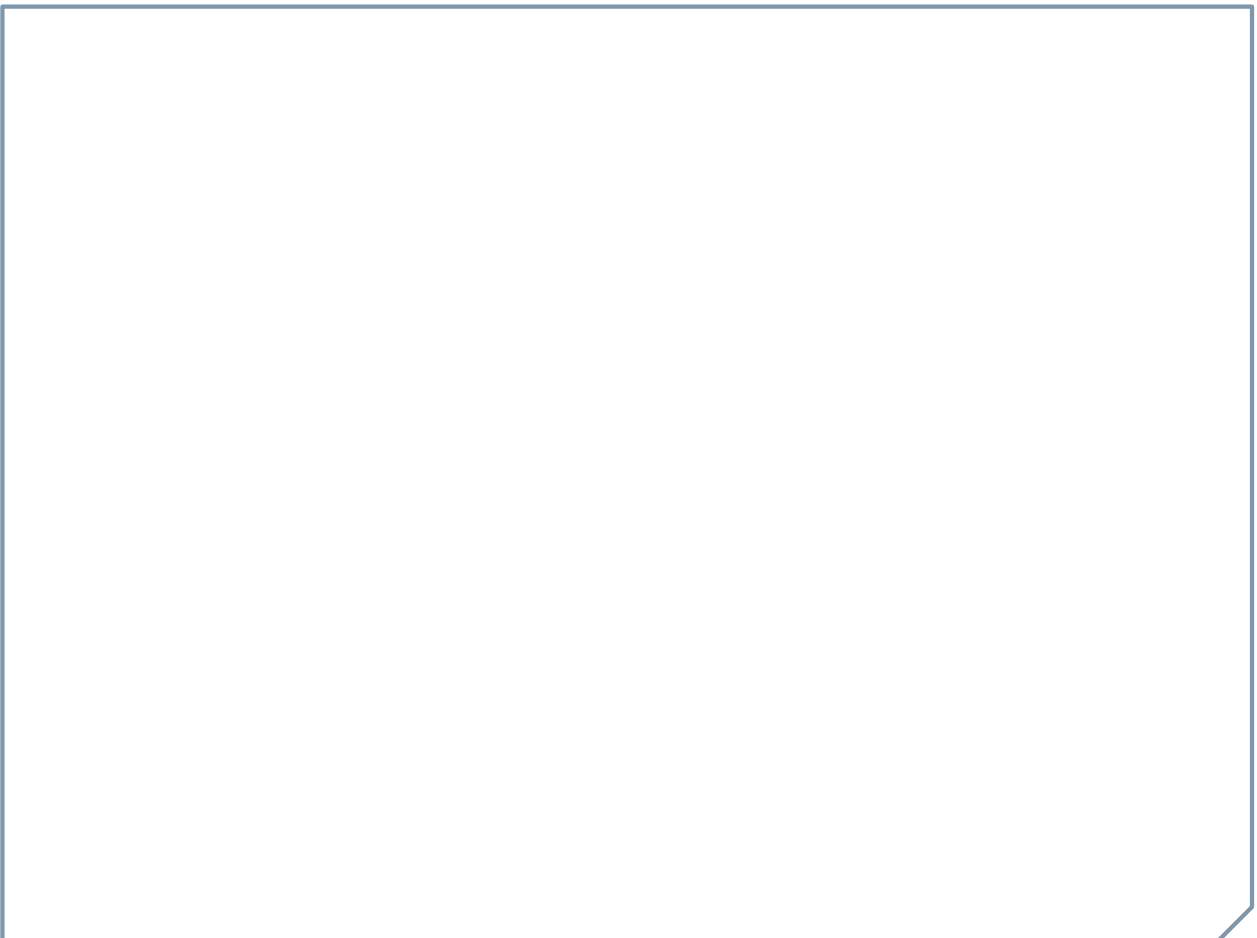
<u>sigle</u>	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

TypeÉvaluation

<u>code</u>	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

Résultat

<u>matricule</u>	<u>TE</u>	<u>activité</u>	<u>trimestre</u>	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



SYNTHÈSE

LE LANGAGE SQL SELECT

(EXEMPLES
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 :

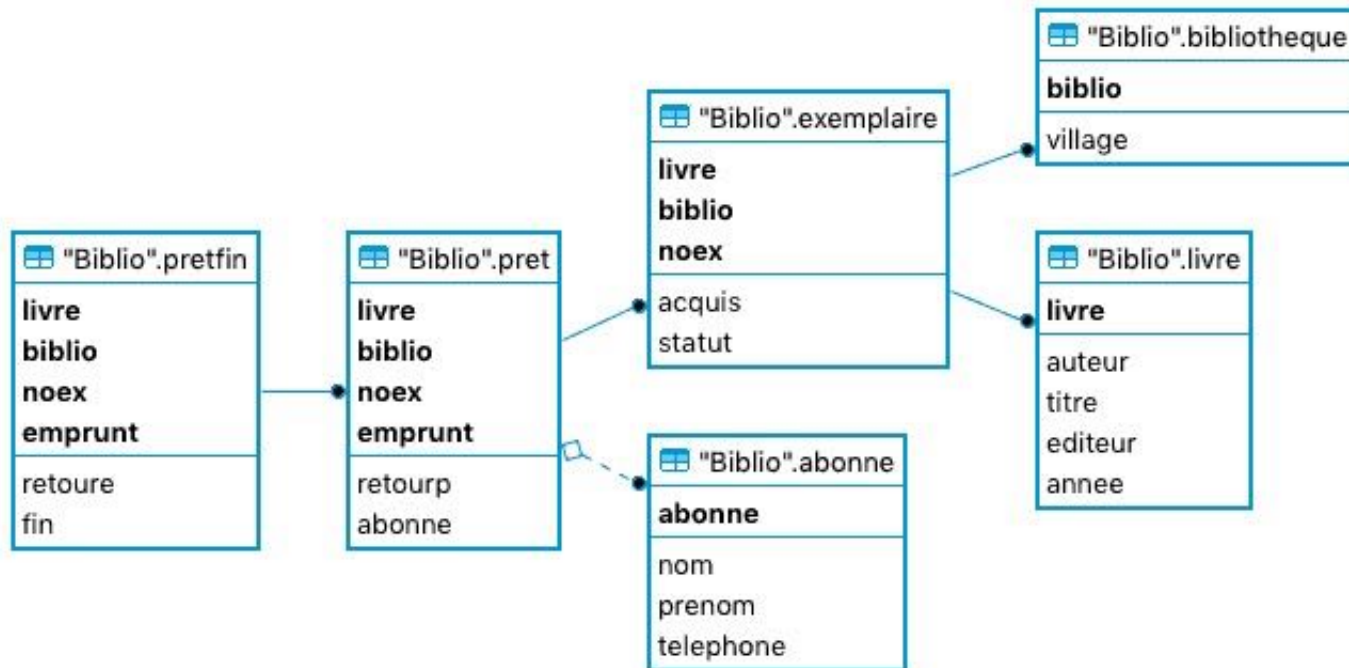
```
SELECT C.nom, T.tel  
FROM  
    C RIGHT JOIN T ON C.id = T.id  
UNION  
SELECT C.nom, T.tel  
FROM  
    C LEFT JOIN T ON C.id = T.id
```

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^e ed.), chapitre 7
- Elmasri et Navathe (6^e 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.
2nd 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
- Le site de PostgreSQL (en français)
 - <http://docs.postgresqlfr.org>

