

# Base de données

---

## Chapitre 5 : L'algèbre relationnelle

Stéphane Malandain – sbd – 2024

# Introduction

---

# introduction

---

## Objectifs

- Base théorique pour l'expression des requêtes
- Le langage SQL est inspiré de l'algèbre relationnelle

# introduction

---

## Algèbre relationnelle

Langage fournissant un ensemble formel d'opérateurs permettant d'extraire des informations des relations

# introduction

---

Les opérateurs sont unaires ou binaires. Ils prennent une ou deux relations et retournent une nouvelle relation. Le résultat **ne comporte pas de doublons**.

- Opérateurs unaires :
  - Sélection
  - Produit
- Opérateurs binaires:
  - Produit cartésien
  - Jointures
  - Différence
  - Division
  - Union
  - Intersection

# introduction

---

Pour ce chapitre, nous allons utiliser un schéma de base de données simplifié représentant les emprunts de livres d'une bibliothèque. Un emprunt correspond à une personne et un livre.

Schéma relationnel :

```
Personne(id_personne, prenom, nom, email, date_naissance, localite)
Livre(id_livre, titre, id_theme
      id_theme  $\subseteq$  Theme.id_theme
Theme(id_theme, intitule)
Emprunt(id_personne, id_livre, date_emprunt, date_retour, date_echeance)
      id_personne  $\subseteq$  Personne.id_personne
      id_livre  $\subseteq$  Livre.id_livre
      date_retour OPTIONNEL
```

## La sélection ( $\sigma$ )

---

La **sélection** ( $\sigma$ ) est une opération unaire permettant d'appliquer un filtre sur une relation. (correspond à la clause SQL : WHERE)

Syntaxe :  $\sigma_{\text{condition}}(\text{Relation})$

## La sélection ( $\sigma$ )

Exemple : Liste des personnes habitant "Lausanne"

Requête :  $\sigma_{\text{localite=Lausanne}}$  (*Personne*)

**Personne**

<u>id_personne</u>	prenom	nom	email	date_naissance	localite
23	Aurélia	Racloz	a.racloz@gmail.cn	01.05.1999	Bulle
18	Stéphane	Ischi	st.ischi@tartempion.net	03.10.1980	Lausanne
14	Danielle	Zuffret	zuffret@commerce.cm	02.03.1971	Penthaz
16	Naïm	Daiz	naim.daiz@gmail.cn	05.05.1993	Lausanne
29	Jérôme	Doutaz	jerome@doutaz.ch	01.12.2001	Genève

$\sigma_{\text{localite=Lausanne}}$  (*Personne*)

<u>id_personne</u>	prenom	nom	email	date_naissance	localite
18	Stéphane	Ischi	st.ischi@tartempion.net	03.10.1980	Lausanne
16	Naïm	Daiz	naim.daiz@gmail.cn	05.05.1993	Lausanne



## La sélection ( $\sigma$ )

---

Exercice : Liste des emprunts en cours qui ont du retard

Requête :  $\sigma$  ...

## La projection ( $\pi$ )

---

La **projection** ( $\pi$ ) d'une relation permet de conserver un sous ensemble de ses attributs. Les **doublons sont supprimés** du résultat (correspond à la clause SQL : SELECT DISTINCT)

Syntaxe :  $\pi_{\{\text{attribut1, attribut2, ...}\}} (Relation)$

## La projection ( $\pi$ )

Exemple : Liste de toutes les personnes en ne gardant que leur prénom, nom et email

Requête :  $\pi_{\{\text{nom}, \text{prenom}, \text{email}\}}(\text{Personne})$

$\pi_{\{\text{nom}, \text{prenom}, \text{email}\}}(\text{Personne})$

prenom	nom	email
Aurélia	Racloz	a.racloz@gnail.cn
Stéphane	Ischi	st.ischi@tartempion.net
Danielle	Zuffret	zuffret@commerce.cm
Naïm	Daiz	naim.daiz@gnail.cn
Jérôme	Doutaz	jerome@doutaz.ch

## Exemple de composition d'une projection et d'une selection

Exemple : Liste des emails des personnes habitant à Lausanne

Requête :  $\pi_{\{email\}}(\sigma_{localite=Lausanne}(Personne))$

$\pi_{\{email\}}(\sigma_{localite=Lausanne}(Personne))$

email

st.ischi@tartempion.net

naim.daiz@gmail.cn

## Exemple de composition d'une projection et d'une sélection

Requête :  $\pi_{\{email\}}(\sigma_{localite=Lausanne}(Personne))$

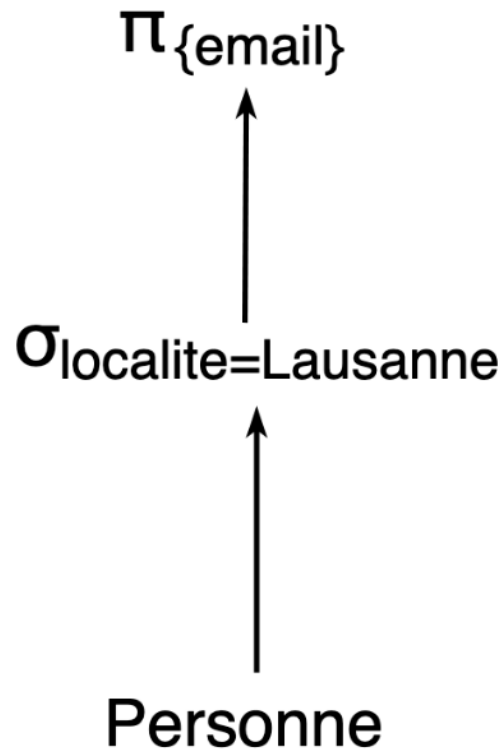


Fig.1 : Arbre d'expression algébrique

## Produit cartésien ( $\times$ )

---

Le produit cartésien produit une nouvelle relation composée de toutes les combinaisons possibles des tuples des deux relations.

**Exemple** : Produit cartésien des relations Livre et Theme

**Requête** : *Livre*  $\times$  *Theme*

## Produit cartésien (×)

Livre

id_livre	titre	id_theme
2-X-4	Les Acqueducs	1
5-X-1	Python pour les nuls	2
7-X-3	Base de données avancée	2

Theme

id_theme	intitule
1	Architecture
2	Informatique
3	Mathématique

*Livre × Theme*

id_livre	titre	Livre.id_theme	Theme.id_theme	intitule
2-X-4	Les Acqueducs	1	1	Architecture
5-X-1	Python pour les nuls	2	1	Architecture
7-X-3	Base de données avancée	2	1	Architecture
2-X-4	Les Acqueducs	1	2	Informatique
5-X-1	Python pour les nuls	2	2	Informatique
7-X-3	Base de données avancée	2	2	Informatique
2-X-4	Les Acqueducs	1	3	Mathématique
5-X-1	Python pour les nuls	2	3	Mathématique
7-X-3	Base de données avancée	2	3	Mathématique

## Produit cartésien ( $\times$ )

---

Le produit cartésien n'a pas d'intérêt en tant que tel. Par contre, il fournit une opération intermédiaire pour joindre les enregistrements de deux relations.

**Exemple** : Lister les livres (nom du livre et nom du thème) avec leur thème associé

**Requête** :

$$\pi_{\{\text{titre, intitule}\}}(\sigma_{\text{Theme.id\_theme} = \text{Livre.id\_theme}}(\text{Livre} \times \text{Theme}))$$



## La jointure (⋈)

---

### Jointure

La **jointure** permet de produire une relation dont les lignes sont composées en joignant les lignes des deux relations en entrée qui possèdent une correspondance commune (même valeurs d'attributs).

**Syntaxe :**  $R1 \bowtie_{condition} R2$

**Exemple :** Lister les livres (nom du livre et nom du thème) avec leur thème associé

**Requête :**

$\pi_{\{titre, intitule\}} (Theme \bowtie_{Theme.id\_theme = Livre.id\_theme} Livre)$

## La jointure (⋈)

*Livre* ⋈ *Livre.id\_theme = Theme.id\_theme* *Theme*

id_livre	titre	Livre.id_theme	Theme.id_theme	intitule
2-X-4	Les Acqueducs	1	1	Architecture
5-X-1	Python pour les nuls	2	2	Informatique
7-X-3	Base de données avancée	2	2	Informatique

$\pi \{titre, intitule\}(Livre \bowtie_{Livre.id\_theme = Theme.id\_theme} Theme)$

titre	intitule
Les Acqueducs	Architecture
Python pour les nuls	Informatique
Base de données avancée	Informatique

## La jointure ( $\bowtie$ )

### Requête :

$$\pi_{\{\text{titre, intitule}\}} (\text{Theme} \bowtie_{\text{Theme.id\_theme} = \text{Livre.id\_theme}} \text{Livre})$$

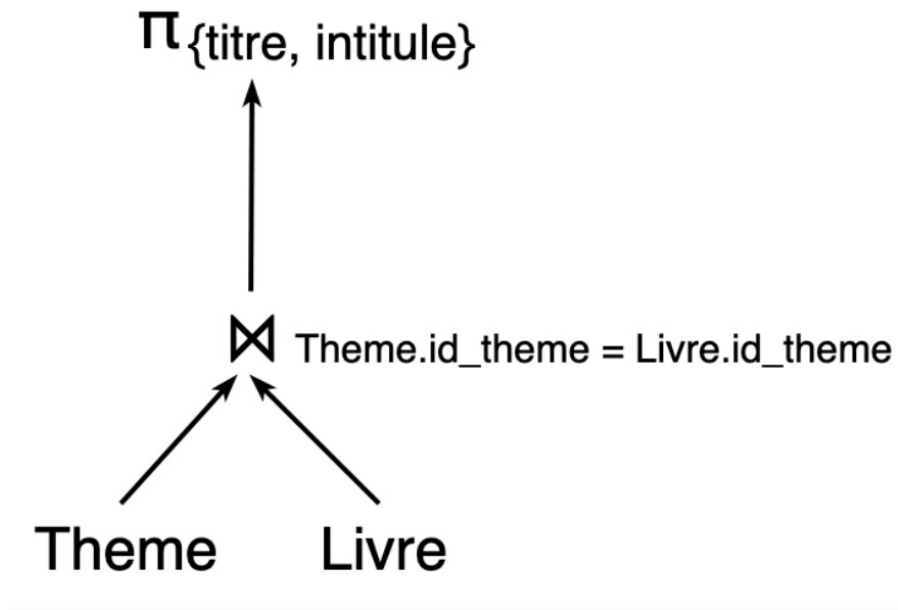


Figure 2 : Arbre d'expression algébrique

## Les jointures externes ( $\bowtie$ $\bowtie$ $\bowtie$ )

### Requête :

$$\pi_{\{\text{titre, intitule}\}} (\text{Theme} \bowtie_{\text{Theme.id\_theme} = \text{Livre.id\_theme}} \text{Livre})$$

Une jointure externe permet d'ajouter les lignes d'une relation qui n'a pas de correspondance dans l'autre table

Il existe trois sortes de jointures externes :

- La jointure gauche, notée  $R1 \bowtie R2$
- La jointure droite, notée  $R1 \bowtie R2$
- La jointure pleine, notée  $R1 \bowtie R2$

## Les jointures externes (⋈⋈⋈)

**Exemple :** Lister les livres (nom du livre et nom du thème) avec leur thème associé en faisant ressortir les thèmes qui n'ont pas de livres associés.

**Requête :**

$$\pi_{\{titre, intitule\}}(Livre \bowtie_{Livre.id\_theme = Theme.id\_theme} Theme)$$

$$\pi_{\{titre, intitule\}}(Livre \bowtie_{Livre.id\_theme = Theme.id\_theme} Theme)$$

titre	intitule
Les Acqueducs	Architecture
Python pour les nuls	Informatique
Base de données avancée	Informatique
NULL	Mathématique

## Exemple de composition d'une requête

---

**Exemple :** Lister tous les emprunts (nom, prénom, titre du livre et intitulé) des personnes habitant Lausanne qui ont emprunté des livres sur le thème "Informatique"

## Exemple de composition d'une requête

### Requête avec renommage :

$$\begin{aligned} T &\leftarrow \textit{Theme} \\ L &\leftarrow \textit{Livre} \\ E &\leftarrow \textit{Emprunt} \\ P &\leftarrow \textit{Personne} \\ R &\leftarrow (T \bowtie_{T.id\_Theme=L.id\_Theme} L \bowtie_{L.id\_Livre=E.id\_livre} E \bowtie_{E.id\_personne=P.id\_personne} P) \\ \pi_{\{nom, prenom, titre, intitule\}} &(\sigma_{intitule=Informatique \wedge localite=Lausanne}(R)) \end{aligned}$$

## Exemple de composition d'une requête

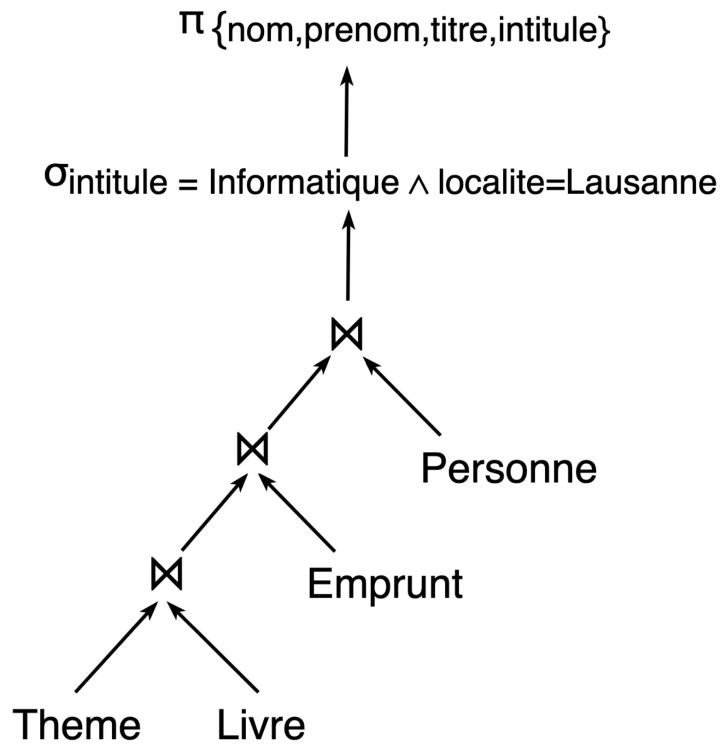


Figure 3 : Arbre non optimisé

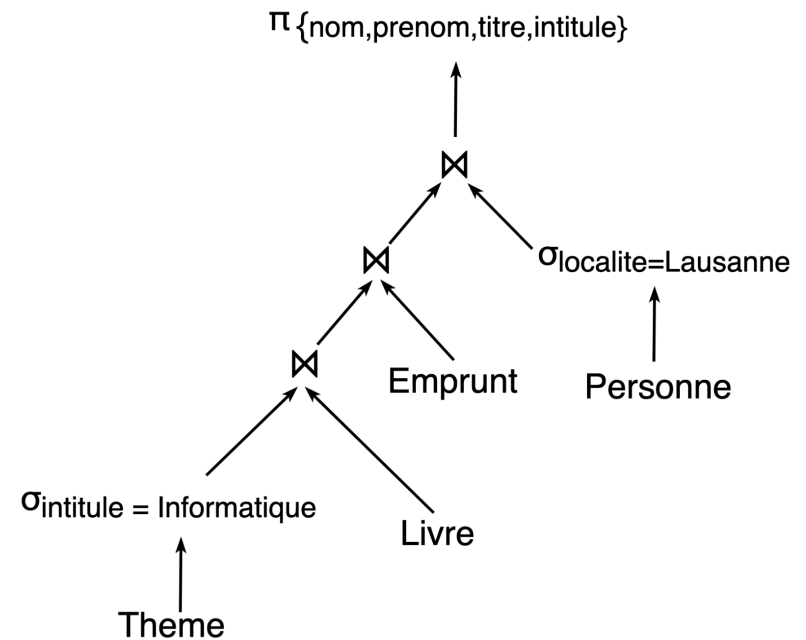


Figure 4 : Arbre optimisé



## Opérations ensemblistes

---

Les **opérations ensemblistes** sont des opérations binaires qui s'effectuent sur des relations possédant les mêmes attributs.

- L'union, notée  $\cup$ 
  - **Exemple:** Le nom des clients et des employés
  - **Requête:**  $\pi_{\{nom\}}(Client) \cup \pi_{\{nom\}}(Employe)$
- L'intersection, notée  $\cap$ 
  - **Exemple:** Le nom des clients qui sont également employés
  - **Requête:**  $\pi_{\{nom\}}(Client) \cap \pi_{\{nom\}}(Employe)$
- La différence, notée  $-$ 
  - Exprime généralement une négation
  - **Exemple:** Les personnes (nom, prénom) qui n'ont jamais emprunté de livre
  - **Requête :**  
 $\pi_{\{nom, prenom\}}(Personne) - \pi_{\{nom, prenom\}}(Emprunt \bowtie Personne)$

## La division (/)

Le résultat de la **division** de deux relations notée  $R1/R2$  est composée des tuples de  $R1$  qui sont associés à **tous** les éléments de  $R2$ . Le schéma de la relation  $R2$  **doit obligatoirement être composée** d'un sous-ensemble d'attributs du schéma de la relation  $R1$ .

**Exemple:** Les personnes (nom, prénom) qui ont emprunté tous les livres

**Requête:**

$$R \leftarrow \text{Personne} \bowtie_{\text{Personne.id\_personne}=\text{Emprunt.id\_personne}} \text{Emprunt}$$
$$\pi\{\text{nom}, \text{prenom}, \text{id\_livre}\}(R) / \pi\{\text{id\_livre}\}(\text{Livre})$$

## La division (/)

$$\pi_{\{nom, prenom, id\_livre\}}(Personne \bowtie_{Personne.id\_personne = Emprunt.id\_personne} Emprunt)$$

nom	prenom	id_livre
Aurélia	Racloz	2-X-4
Stéphane	Ischi	2-X-4
Stéphane	Ischi	5-X-1
Naïm	Daiz	7-X-3
Jérôme	Doutaz	5-X-1
Aurélia	Racloz	7-X-3
Jérôme	Doutaz	2-X-4
Naïm	Daiz	2-X-4
Naïm	Daiz	5-X-1
Jérôme	Doutaz	7-X-3

$$\pi_{\{id\_livre\}}(Livre)$$

id_livre
2-X-4
5-X-1
7-X-3

$$\pi_{\{nom, prenom, id\_livre\}}(Personne \bowtie Emprunt) / \pi_{\{id\_livre\}}(Livre)$$

nom	prenom
Naïm	Daiz
Jérôme	Doutaz