

A spiral-bound notebook with a light beige, textured cover. The metal spiral binding is visible on the left side. The text is centered on the page in a dark brown, serif font.

Le langage de requêtes SQL  
et  
Algèbre Relationnelle

# SQL: le langage de requêtes

---

- ✓ Simple Query Language
- ✓ Principal langage de requêtes des BD relationnelles
- ✓ Basé sur l'algèbre relationnelle

# Principales opérations

---

- ✓ Projection ( $\Pi$ ): restriction sur les colonnes
- ✓ Sélection ( $\sigma$ ): restriction sur les lignes
- ✓ Produit cartésien ( $\times$ ) : association de toutes les lignes d'une table à toutes les lignes de l'autre
- ✓ Jointure ( $\Join$ ): association de certaines lignes d'une table avec certaines lignes d'une autre (reconstitution de l'information)
- ✓ Renommage ( $\rho$ ): changement de nom des colonnes

# Afficher le contenu d'une table

---

SQL

```
select *  
from adherent;
```

AR

```
adherent
```

# Projection

---

SQL

```
select nom, prenom  
from adherent;
```

AR

$\Pi_{\text{nom, prenom}}(\text{adherent})$

# Sélection

---

SQL

```
select *  
from adherent  
where nom='Dupont';
```

AR

$\sigma_{\text{nom}='Dupont'}(\text{adherent})$

# Jointure

---

SQL

```
select *  
from adherent  
natural join  
participe
```

AR

```
adherent ∞ participe
```

# Jointure

---

SQL

```
select *  
from adherent  
natural join  
participe
```

AR

```
adherent ∞ participe
```

# Produit cartésien

---

SQL

```
select nom, intitule  
from adherent,  
participe;
```

AR

```
adherent x participe
```

**Attention!**

- ✓ AR: base théorique
- ✓ SQL: Implantation de l'AR  
=> il y a quelques différences

# Différences SQL/AR

---

## ✓ En cas de projection

- En SQL: il y a risque de doublons (plusieurs lignes strictement identiques)
- En AR: les doublons n'existent pas (manipulation d'ensembles)

## ✓ Le nom des colonnes

- En SQL : `table.colonne`
- En AR : seul le nom de colonne compte

# Syntaxe d'une requête SQL

---

**Select** <liste\_col>

**From** <liste\_tables>

**Where** <condition>

- ✓ <liste\_col> va effectuer une projection
- ✓ <liste\_tables> va effectuer un produit cartésien ou des jointures
- ✓ <condition> va effectuer une sélection
- ✓ la clause **Where** est facultative
- ✓ les commandes SQL se terminent par ;

# Exemple

---

- SQL

```
Select nom, prenom, intitule
```

```
From adherent natural join participe
```

```
Where Adresse='Orléans';
```

- Algèbre Relationnelle

$$\Pi_{\text{nom, prenom, intitule}} \left( \sigma_{\text{adresse='Orléans'}} (\text{adherent} \bowtie \text{participe}) \right)$$

# Conditions SQL

---

`<condition> ::= <exp_bool> |  
( <condition> ) |  
<condition><connect><condition> |  
NOT <condition>`

`<connect> ::= AND | OR`

`<exp_bool> ::= <comparaison>`

✓ Il existe de nombreuses expressions booléennes possibles, pour l'instant nous nous attacherons aux comparaisons

# Comparaisons SQL

---

`<comparaison> ::= <expr> <op> <expr>`

<code>&lt;op&gt; ::=</code>	<code>=</code>	<i>égal</i>
	<code>&lt;&gt;</code>   <code>!=</code>	<i>différent</i>
	<code>&lt;</code>   <code>&lt;=</code>   <code>&gt;</code>   <code>&gt;=</code>	<i>ordre</i>
	<code>IS NULL</code>	<i>absence de valeur</i>

`<expr> ::= <constante> |`

`<nom_colonne> |`

`<nom_table>.<nom_colonne> |`

`<expr> <opération> <expr>`

# Comparaisons de chaînes

---

- ✓ l'opérateur de comparaison approximative  
**LIKE**
- ✓ `<chaîne> LIKE ' <masque> '`  
où `<masque>` est une suite de caractères  
contenant éventuellement les symboles `%` et `-`
- ✓ `%` remplace n'importe quelle suite de car
- ✓ `-` remplace exactement un car
- ✓ `' abc% '` désigne tous les mots commençant  
par `' abc '`

# Expressions

---

- ✓ `<opération>` sont standards pour l'arithmétique
  - pour les chaînes `||` désigne la concaténation
  - `<date>+<entier>` ajoute un nombre de jours à une date
  - `<date>-<entier>` enlève un nombre de jours à une date
  - `<date>-<date>` donne le nombre de jours entre les deux dates