



SQL

Langage algébrique

**André Miralles**

# Plan

- » **Présentation générale de SQL**
- » **Présentation des 6 opérateurs**
- » **Quelques sites utiles**



# Présentation générale de SQL

**Structured  
Query Language**

# Présentation générale de **SQL**

## » Langage SQL

### > Langage déclaratif

- + Consiste à décrire les **propriétés du point d'arrivée** (le résultat) en **fonction de celles du point de départ** (les données de la base, dans notre cas)
  - Utilisation de **formules logiques** qui indiquent comment l'existence d'un fait  $f_1$  au départ implique l'existence d'un fait  $f_2$  à l'arrivée

### > Langage algébrique basé sur l'algèbre relationnel

- + Consiste en un **ensemble d'opérations** qui permettent de **manipuler des relations**, considérées comme des **ensembles de nuplets (attributs)**



# Présentation générale de **SQL**

## » Algèbre relationnel est composé de **6 opérateurs**

- > **Projection** (  $\pi$  )
- > **Sélection** (  $\sigma$  )
- > **Renommage** (  $\rho$  )
- > **Produit cartésien** (  $\times$  )
- > **Union** (  $\cup$  )
- > **Différence** (  $-$  )

## » Ces opérateurs ont **deux propriétés** majeures

- > **Clôture** : un **opérateur** s'applique à des **relations** et le résultat est une **relation**
- > **Composition** : un **opérateur** peut prendre en entrée le **résultat d'un autre opérateur** pour définir des **requêtes algébriques complexes**

## » Opérateurs additionnels

- > Opérateur **Jointure** (  $\bowtie$  )
  - + Composition d'opérateurs
- > Opérateur **Égalité** (  $:=$  )

# Présentation générale de **SQL**

- » **SQL : langage composé de plusieurs langages**
  - > **Data Definition Language (DDL)**
    - + Langage **création** de **bases de données**, de **tables**, de **contraintes**, etc.
  - > **Data Manipulation Language (DML)**
    - + Langage de **manipulation** de **données**
  - > **Data Control Language (DCL)**
    - + Langage de **gestion des droits**
  - > **Transaction Control Language (TCL)**
    - + Langage de **contrôle de l'exécution des transactions**
  - > **Procedural Language / Structured Query Language (PL/SQL)**
    - + Langage de **programmation du SGBD Oracle Database**



# Présentation des 6 opérateurs

**Structured  
Query Language**

# Opérateur de *Projection*

- » Opérateur notée  $\pi$
- » La relation  $\pi_{A_1, A_2, \dots, A_k}(R)$  **conserve seulement les attributs**  $A_1, A_2, \dots, A_k$  de la relation  $R$
- » Exemple  $\pi_{Fournisseur, Ville}(Fournisseurs)$ 
  - > Fournisseur et Ville de la relation Fournisseurs

> **SELECT** Fournisseur, Ville **FROM** Fournisseurs

Fournisseur	Adresse fournisseur	Ville
VIDEO SA	13 rue du cherche-midi	PARIS
HITEK LTD	25 Bond Street	LONDON



Fournisseur	Ville
VIDEO SA	PARIS
HITEK LTD	LONDON

- > **SELECT \* FROM** Fournisseurs
  - + Conserve tous les attributs



# Opérateur de *Sélection*

- » Opérateur notée  $\sigma$
- » La relation  $\sigma_F(R)$  **extraît les nuplets** (tuples) de la relation  $R$  qui **satisfont la condition  $F$**
- » Exemple  $\sigma_{Ville="PARIS"}(Fournisseurs)$ 
  - > SELECT \* FROM Fournisseurs WHERE Ville = 'PARIS'

Fournisseur	Adresse fournisseur	Ville
VIDEO SA	13 rue du cherche-midi	PARIS
HITEK LTD	25 Bond Street	LONDON



Fournisseur	Adresse fournisseur	Ville
VIDEO SA	13 rue du cherche-midi	PARIS

# Opérateur de *Renommage*

- » Opérateur notée  $\rho$
- » L'opérateur  $\rho_{A \rightarrow C, B \rightarrow D}(R)$  **renomme** les **attributs  $A$  et  $B$**  de la relation  $R$  **en  $C$  et  $D$**  respectivement
- » Exemple  $\rho_{Ville \rightarrow Capitale}(Fournisseurs)$

> SELECT Ville **AS** Capitale FROM Fournisseurs

Fournisseur	Adresse fournisseur	Ville
VIDEO SA	13 rue du cherche-midi	PARIS
HITEK LTD	25 Bond Street	LONDON



Capitale
PARIS
LONDON

**Très utile lorsque le nom de l'attribut d'une relation est le même que celui d'une autre relation**



# Opérateur de *Produit Cartésien*

- » Opérateur notée  $\times$
- » Le résultat du **produit cartésien**  $R \times S$  est une **relation associant à chaque nuplet de  $R$  un nuplet de  $S$**
- » Exemple *Fournisseurs*  $\times$  *Villes*

> SELECT \* FROM Fournisseurs **CROSS JOIN** Villes

Fournisseur
VIDEO SA
HITEK LTD

$\times$

Ville
PARIS
LONDON



Fournisseur	Ville
VIDEO SA	PARIS
VIDEO SA	LONDON
HITEK LTD	PARIS
HITEK LTD	LONDON

> Remarque : « Fournisseurs **CROSS JOIN** Villes » est une relation incluse dans la clause FROM

# Opérateur d'*Union*

- » Opérateur notée  $\cup$
- » Le résultat de l'**union**  $R \cup S$  est une **relation contenant l'union minimale** (suppression doublons) **des nuplets de  $R$  et de  $S$** 
  - > Les **schémas** des 2 relations **doivent être les mêmes**
    - + Souvent nécessité de renommage

## » Exemple $Villes \cup Capitales$

- > `SELECT * FROM Villes`
  - + **UNION**
- > `SELECT Capitale AS Ville FROM Capitales`

Ville
PARIS
LONDRES
LYON
BARCELONE



Capitale
PARIS
LONDRES
BERLIN



Ville
PARIS
LONDRES
BERLIN
LYON
BARCELONE



# Opérateur de *Différence*

- » Opérateur notée  $-$
- » Le résultat de la **différence**  $R - S$  est une **relation contenant les nuplets de  $R$  et qui ne sont pas dans  $S$** 
  - > Les **schémas** des 2 relations **doivent être les mêmes**
    - + Souvent nécessité de renommage

**Attention opérateur non permutable**

$$R - S \neq S - R$$

# Opérateur de *Différence*

## » Exemple *Villes – Capitales*

> SELECT \* FROM Villes

+ EXCEPT

> SELECT Capitale AS Ville FROM Capitales

Ville
PARIS
LONDRES
LYON
BARCELONE

–

Capitale
PARIS
LONDRES
BERLIN



Ville
LYON
BARCELONE

## » Exemple *Capitales – Villes*

> SELECT Capitale AS Ville FROM Capitales

+ EXCEPT

> SELECT \* FROM Villes

Capitale
PARIS
LONDRES
BERLIN

–

Ville
PARIS
LONDRES
LYON
BARCELONE



Ville
BERLIN



# Présentation des opérateurs additionnels

**Structured  
Query Language**

# Opérateur de *Jointure*

- » Opérateur notée  $\bowtie$
- » Le résultat de la **jointure**  $R \bowtie_F S$  est le **sous-ensemble du produit cartésien**  $R \times S$  qui satisfait la **condition**  $F$

$$> R \bowtie_F S \equiv \sigma_F (R \times S)$$

- » Exemple  $Fournisseurs \bowtie_{Ville="PARIS"} Villes$

> SELECT \* FROM Fournisseurs, Villes **WHERE** Ville='PARIS'

> SELECT \* FROM Fournisseurs **JOIN** Villes **ON** Ville='PARIS'



# Opérateur de *Jointure*

» Exemple *Fournisseurs* ⋈<sub>Ville="PARIS"</sub> *Villes*

Fournisseur
VIDEO SA
HITEK LTD

X

Ville
PARIS
LONDON



Fournisseur	Ville
VIDEO SA	PARIS
VIDEO SA	LONDON
HITEK LTD	PARIS
HITEK LTD	LONDON

Fournisseur
VIDEO SA
HITEK LTD

⋈<sub>Ville="PARIS"</sub>

Ville
PARIS
LONDON



Fournisseur	Ville
VIDEO SA	PARIS
HITEK LTD	PARIS

# Opérateur d'Égalité

- » Opérateur notée  $\coloneqq$
- » Le **résultat** d'une expression algébrique est **affecté à une nouvelle relation**
- » Utilisation la plus courante
  - > **Affectation** d'une expression algébrique
  - > Corollaire  $\rightarrow$  **Simplification** expression algébrique
    - +  $P \coloneqq R \times S$                       Exemple :  $ProdCart \coloneqq Fournisseurs \times Villes$
    - +  $\sigma_F(P)$                                        $\sigma_F(ProdCart)$
- » Autre utilisation importante
  - > **Renommage** des relations
    - + ... **FROM** *Personne* **AS**  $P1$ 
      - »  $P1 \coloneqq Personne$                        $P2 \coloneqq Personne$
      - »  $ProduitCartésien \coloneqq P1 \times P2$





# Quelques sites utiles

# Quelques sites utiles

## » SQL

> <http://sql.bdpedia.fr/alg.html>