

# Bases de Données 1

## #5 - Algèbre relationnelle

Matthieu Nicolas  
Polytech S5 - II

Slides réalisées à partir de celles de Claude Godart et Malika Smaïl

# Langage de manipulation de données

- Jusqu'à maintenant, on a vu **comment représenter nos données** grâce à un **langage de description de données** (LDD)...
  - Dans notre cas, il s'agit du **schéma relationnel**
- ... il est grand temps de voir **comment** on peut ensuite **les manipuler**
- Se fait à l'aide d'un **langage de manipulation de données** (LMD)
  - Dans notre cas, il s'agit de l'**algèbre relationnelle**

# Algèbre relationnelle

- Ensemble d'**opérateurs sur des relations** **produisant** en résultat **de nouvelles relations**
- Objectif : **élaborer de nouvelles informations** à partir des **relations de départ** et d'une **composition séquentielle d'opérateurs**

# Types d'opérateurs

- Opérateurs **unaires**
  - *Projection, Restriction*
- Opérateurs **binaires sur schémas différents**
  - *Produit cartésien, Jointure, Division*
- Opérateurs **binaires sur mêmes schémas**
  - *Union, Intersection, Différence*

# Plan

- Opérateurs unaires
- Opérateurs binaires sur schémas différents
- Opérateurs binaires sur mêmes schémas
- Requêtes algébriques
- Algèbre relationnelle et SQL

# Opérateurs unaires

Base de Données 1  
#5 - Algèbre relationnelle

# Projection - 1

- Sélectionne les attributs à récupérer d'une relation

**Vins**(num, cru, millésime)

num	cru	millesime
100	Chablis	1974
110	Mecurey	1978
120	Mâcon	1977
200	Sancerre	1977

opérateur    relation    attributs

↓                      ↓                      ↓

**PROJECT(Vins, cru, millésime)**

**Res**(cru, millésime)

cru	millesime
Chablis	1974
Mecurey	1978
Mâcon	1977
Sancerre	1977

# Projection - 2

- La projection est un opérateur permettant de générer une nouvelle relation  $R'(A')$  composée d'un sous-ensemble des attributs de la relation initiale  $R(A)$
- Les tuples de cette nouvelle relation sont obtenus en **éliminant des tuples de  $R$  les attributs de  $A$  n'appartenant pas à  $A'$  et en supprimant les tuples en double**



# Projection - 3

- **Signature** : relation x liste d'attributs  $\rightarrow$  relation
- **Notation** :  $R' = \text{PROJECT}(R, A')$ 
  - avec  $R(A)$  une relation
  - $A'$  un attribut ou ensemble d'attributs de  $A$

# Restriction - 1

- Sélectionne les tuples à récupérer d'une relation à l'aide d'une condition booléenne

**Vins**(num, cru, millésime)

num	cru	millésime
100	Chablis	1974
110	Mecurey	1978
120	Mâcon	1977
200	Sancerre	1977



**RESTRICT(Vins, millésime = 1977)**

num	cru	millésime
120	Mâcon	1977
200	Sancerre	1977

# Restriction - 2

- La restriction (ou sélection) est un opérateur permettant de générer une nouvelle relation  $R'(A)$  composée d'un sous-ensemble des tuples de la relation initiale  $R(A)$
- Les tuples de cette nouvelle relation sont obtenus en ne conservant de  $R$  que les tuples pour lesquels la condition booléenne est vraie

# Restriction - 3

- **Signature** : relation x condition booléenne  $\rightarrow$  relation
- **Notation** :  $\text{Res} = \text{RESTRICT}(\text{R}, \text{C})$ 
  - avec R une relation
  - C une condition booléenne, simple ou composée

# Condition booléenne

- La condition booléenne peut être une expression logique simple qui compare la valeur d'un **attribut** avec une **constante** de son domaine à l'aide d'un **opérateur** ( $=$ ,  $<$ ,  $>$ ,  $\neq$ ,  $\geq$ ,  $\leq$ )
  - $cru = \text{"Chablis"}$
  - $millesime > 1977$
- ou composée avec des connecteurs **ET** ( $\wedge$ ) et **OU** ( $\vee$ )
  - $(cru = \text{"Chablis"} \vee cru = \text{"Mâcon"}) \wedge millesime < 1988$

# Opérateurs binaires sur mêmes schémas

Base de Données 1  
#5 - Algèbre relationnelle

# Union - 1

- Regroupe les tuples de 2 relations identiques

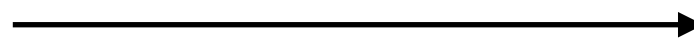
**Vins-1**(num, cru, millésime)

num	cru	millésime
100	Chablis	1974
110	Mecurey	1978

**Vins-2**(num, cru, millésime)

num	cru	millésime
120	Mâcon	1977
200	Sancerre	1977

**UNION(Vins-1, Vins-2)**



**Res**(num, cru, millésime)

num	cru	millésime
100	Chablis	1974
110	Mecurey	1978
120	Mâcon	1977
200	Sancerre	1977

# Union - 2

- **Signature** : relation x relation  $\rightarrow$  relation
- **Notation** :  $\text{Res} = \text{UNION}(\text{R1}, \text{R2})$ 
  - avec  $\text{R1}(\text{A})$  et  $\text{R2}(\text{A})$  des relations avec un schéma identique



# Intersection - 1

- Répertorie les tuples appartenant simultanément aux 2 relations initiales

**Vins-1**(num, cru, millésime)

num	cru	millésime
100	Chablis	1974
110	Mecurey	1978

**Vins-2**(num, cru, millésime) **INTERSECT(Vins-1, Vins-2)**

num	cru	millésime
110	Mecurey	1978
200	Sancerre	1977

**Res**(num, cru, millésime)

num	cru	millésime
110	Mecurey	1978

# Intersection - 2

- **Signature** : relation x relation  $\rightarrow$  relation
- **Notation** : Res = INTERSECT(R1, R2)
  - avec R1(A) et R2(A) des relations avec un schéma identique

# Différence - 1

- Liste les tuples apparaissant dans la première relation mais pas dans la seconde

**Vins-1**(num, cru, millésime)

num	cru	millesime
100	Chablis	1974
110	Mecurey	1978
120	Mâcon	1977

**Vins-2**(num, cru, millésime)

num	cru	millesime
100	Chablis	1974
200	Sancerre	1977

**MINUS(Vins-1, Vins-2)**



**Res**(num, cru, millésime)

num	cru	millesime
110	Mecurey	1978
120	Mâcon	1977

# Différence - 2

- La différence est un opérateur permettant de générer une nouvelle relation  $R'(A)$  composée d'un sous-ensemble des tuples de la relation initiale  $R_1(A)$
- Les tuples de cette nouvelle relation sont obtenus en retirant des tuples de  $R_1$  les tuples apparaissant aussi dans la  $R_2$

# Différence - 3

- **Signature** : relation  $\times$  relation  $\rightarrow$  relation
- **Notation** :  $Res = MINUS(R1, R2)$ 
  - avec  $R1(A)$  et  $R2(A)$  des relations avec un schéma identique

# Opérateurs binaires sur schémas différents

Base de Données 1  
#5 - Algèbre relationnelle

# Produit cartésien - 1

- Effectue toutes les combinaisons possibles entre les tuples des relations initiales

**Vins**(num, cru, millésime)

num	cru	millesime
100	Chablis	1974
200	Sancerre	1979

**Viticulteurs**(nom, ville, region)

nom	ville	region
Nicolas	Pouilly	Bourgogne
Martin	Bordeaux	Bordelais

↓  
**PRODUCT(Vins, Viticulteurs)**

num	cru	millesime	nom	ville	region
100	Chablis	1974	Nicolas	Pouilly	Bourgogne
100	Chablis	1974	Martin	Bordeaux	Bordelais
200	Sancerre	1979	Nicolas	Pouilly	Bourgogne
200	Sancerre	1979	Martin	Bordeaux	Bordelais

# Produit cartésien - 2

- Le produit cartésien est un opérateur permettant de générer une nouvelle relation  $R'(A1 + A2)$  à partir des relations initiales  $R1(A1)$  et  $R2(A2)$
- Les tuples de cette nouvelle relation sont obtenus en combinant chaque tuple de  $R1$  avec chaque tuple de  $R2$



# Produit cartésien - 3

- **Signature** : relation x relation  $\rightarrow$  relation
- **Notation** :  $\text{Res} = \text{PRODUCT}(\text{R1}, \text{R2})$ 
  - avec  $\text{R1}(\text{A1})$  et  $\text{R2}(\text{A2})$  des relations

# Jointure - 1

- Fusionne les tuples issus des 2 relations initiales pour lesquelles la condition de jointure est vraie

**Vins**(num, cru, millésime)

num	cru	millesime
150	Riesling	1984
110	Mecurey	1978
120	Mâcon	1977

**Viticulteurs**(nom, numVin, region)

nom	numVin	region
Nicolas	150	Alsace
Martin	120	Bourgogne

**JOIN(Vins, Viticulteurs,  
Vins.num = Viticulteurs.numVin)**

num	cru	millesime	nom	region
150	Riesling	1984	Nicolas	Alsace
120	Mâcon	1977	Martin	Bourgogne

# Jointure - 2

- La jointure est un opérateur permettant de générer une nouvelle relation  $R'(A1 + A2)$  regroupant les informations provenant de différentes relations à partir de leur(s) attribut(s) commun(s)
- Les tuples de cette nouvelle relation sont obtenus en fusionnant les couples de tuples de  $R1$  et de  $R2$  pour lesquels la condition de jointure est vraie

# Jointure - 3

- **Signature** : relation x relation x condition booléenne  $\rightarrow$  relation
- **Notation** :  $\text{Res} = \text{JOIN}(\text{R1}, \text{R2}, \text{C})$ 
  - avec R1 et R2 des relations
  - C porte sur les attributs de R1 et de R2, appelés **attributs de jointure**
- Une **jointure** est **équivalente** à un **produit cartésien** suivi d'une opération de **restriction**
  - $\text{JOIN}(\text{R1}, \text{R2}, \text{C}) = \text{RESTRICT}(\text{PRODUCT}(\text{R1}, \text{R2}), \text{C})$

# Jointures externes - 1

- Il existe d'autres types de jointures
- Les jointures externes permettent, **en plus du résultat de la jointure, d'inclure les tuples d'une relation** (ou des 2) n'ayant aucune correspondance dans l'autre relation

# Jointures externes - 2

- **Jointure externe à gauche** : jointure de R1 et de R2 + le reste des tuples de R1
  - LEFT-JOIN(R1, R2, C)
- **Jointure externe à droite** : jointure de R1 et de R2 + le reste des tuples de R2
  - RIGHT-JOIN(R1, R2, C)
- **Jointure externe pleine** : jointure de R1 et de R2 + le reste des tuples de R1
  - LEFT-JOIN(R1, R2, C)

# Jointures externes - 3

**Vins**(num, cru, millésime)

num	cru	millésime
150	Riesling	1984
110	Mecurey	1978
120	Mâcon	1977

**Viticulteurs**(nom, numVin, region)

nom	numVin	region
Nicolas	150	Alsace
Martin	120	Bourgogne

↓  
**LEFT-JOIN(Vins, Viticulteurs,  
Vins.num = Viticulteurs.numVin)**

num	cru	millésime	nom	region
150	Riesling	1984	Nicolas	Alsace
110	Mecurey	1978	NULL	NULL
120	Mâcon	1977	Martin	Bourgogne

# Division - 1

- Récupère les tuples de la première table qui sont en relation avec tous les tuples de la seconde table

**Produit**(numProd, libelle, pu)

numProd	libelle	pu
p1	Parasite	14.99
p2	Mario	54.99

**Stock**(numProd, numDep, qte)

numProd	numDep	qte
p1	d1	1000
p1	d2	1200
p2	d1	2000

**DIV(  
PROJECT(Stock, numProd, numDep),  
PROJECT(Produit, numProd))**



numDep
d1

**Res**(numDep)



# Division - 2

- La division est un opérateur permettant de générer une nouvelle relation  $R'(A1 - A2)$  à partir des relations initiales  $R1(A1)$  et  $R2(A2)$
- Les tuples de cette nouvelle relation sont obtenus en conservant uniquement les **tuples de R1 qui possèdent une liaison avec tous les tuples de R2**

# Division- 3

- **Signature** : relation x relation  $\rightarrow$  relation
- **Notation** :  $\text{Res} = \text{DIV}(\text{R1}, \text{R2})$ 
  - avec  $\text{R1}(\text{A1})$  et  $\text{R2}(\text{A2})$  des relations

# Requêtes algébriques

Base de Données 1  
#5 - Algèbre relationnelle

# Requêtes algébriques

- **Combiner l'utilisation** des opérateurs nous permet de composer des **requêtes algébriques**
- Ces **requêtes** nous **permettent d'extraire les données** existantes de notre BD ou **d'en créer** de nouvelles

# Exemple requête algébrique

## - 1

- Étant donné le schéma relationnel suivant
  - **Client**(numCli, nom, prenom, ddn, rue, CP, ville)
  - **Produit**(numProd, libelle, pu, numFour)
  - **Commande**(numCom, numCli, numProd, date, qte)
- Donner les produits commandés en quantité supérieure à 100 lors d'une commande et dont le prix dépasse 1000€
  - On affichera les numéros de produit, leur libellé et leur prix unitaire ainsi que la date de la commande.

# Exemple requête algébrique

## - 2

- $R1 = \text{RESTRICT}(\text{Produit}, \text{pu} > 1000)$
- $R2 = \text{RESTRICT}(\text{Commande}, \text{qte} > 100)$
- $R3 = \text{JOIN}(\text{Produit}, \text{Commande}, \text{Produit.numProd} = \text{Commande.numProd})$
- **Res** =  $\text{PROJECT}(R3, \text{numProd}, \text{libelle}, \text{pu}, \text{date})$

# Algèbre relationnelle et SQL

Base de Données 1  
#5 - Algèbre relationnelle

# Ensemble minimal d'opérateurs

- Les requêtes SQL dans les SGBD relationnels sont transformées en expressions algébriques
- Cinq opérateurs sont nécessaires (ensemble minimal)
  - **union, différence, projection, produit cartésien, sélection**
- Les autres opérateurs s'expriment en fonction des précédents



# Quelques propriétés des opérateurs

- $\text{RESTRICT}(R, C1 \wedge C2) = \text{RESTRICT}(\text{RESTRICT}(R, C1), C2)$
- $\text{RESTRICT}(\text{RESTRICT}(R, C1), C2) = \text{RESTRICT}(\text{RESTRICT}(R, C2), C1)$
- $\text{PROJECT}(\text{PROJECT}(R, \text{liste1}), \text{liste2}) = \text{PROJECT}(R, \text{liste2})$
- $\text{JOIN}(R1, R2, C) = \text{JOIN}(R2, R1, C)$
- $\text{JOIN}(\text{JOIN}(R1, R2, C1), R3, C2) = \text{JOIN}(R1, \text{JOIN}(R2, R3, C2), C1)$ 
  - où  $R, R1, R2, R3$  sont des **relations**,  $C, C1, C2$  des **conditions** et  $\text{liste1}$  et  $\text{liste2}$  des **listes d'attributs**
- Ces **propriétés** sont **utilisées** pour l'**optimisation** de requêtes

# Bilan

- L'algèbre relationnelle est **non-opérationnel**
- On **exprime** dans les requêtes **ce qu'on veut obtenir**, et **non comment l'obtenir**
  - Il s'agit d'un **langage déclaratif**
- Offre donc la **liberté au SGBD** relationnel **pour traiter la requête** de la façon qu'il veut
- Lui **permet d'optimiser les requêtes** en coulisses

# Résumé

- L'algèbre relationnelle est le **Langage de Manipulation de Données du modèle relationnel**
- Met à disposition **8 opérateurs** pour **manipuler** et **interroger** des **relations**
- **Utiliser séquentiellement** ces opérateurs nous permet de **créer nos requêtes**
- **L'aspect déclaratif** du langage **permet** aux SGBD relationnels **d'optimiser** les requêtes