

IFT3030

Base de données

Chapitre 9

Concepts avancés

☒ Introduction
☒ Architecture
☒ Modèles de données
☒ Modèle relationnel
☒ Algèbre relationnelle
☒ SQL
☒ Conception
☒ Fonctions avancées
☒ **Concepts avancés**
☒ Modèle des objets
☒ BD à objets

Plan du cours

- Introduction
- Architecture
- Modèles de données
- Modèle relationnel
- Algèbre relationnelle
- SQL
- Conception
- Fonctions avancées
- **Concepts avancés**
- Modèle des objets
- BD à objets

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

Introduction

- En plus des concepts de base du model relationnel, les SGBD relationnels implantent des concepts avancés
- Concepts présentés
 - Vues
 - Optimisation de requêtes

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

La notion de vue

- Généralités
 - Les vues sont pour le relationnel ce que sont les niveaux externes pour les modèles légués
 - Garantissent l'indépendance logique entre les applications et la base de données
 - Renforcent la sécurité
- Exemple
 - Vente (Numv, Nump, Numf, Date, Qte)
 - Produit (Nump, Des, Type, Prix)
 - Fournisseur (Numf, Nom, Region, Tel)
 - les produits vendus par le fournisseur FLUX
 - les ventes du produit numéro 63755 par région
 - les fournisseurs de l'île de Montréal

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

La notion de vue

■ Définition

- Table virtuelle dont le schéma et le contenu sont dérivés de la base de données par une requête

■ Syntaxe SQL

```
CREATE VIEW <une_vue> [ <liste_alias> ]
AS <une_requête>
[WITH CHECK OPTION]
```

■ Exemple

- les produits vendus par le fournisseur FLUX
- les ventes par région
- les fournisseurs de l'île de Montréal

5

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

La notion de vue

■ Syntaxe SQL (suite)

```
DROP VIEW <une_vue>
```

6

La notion de vue

- Mise à jour des données à travers une vue
 - Problèmes : traduire toute mise à jour, insertion ou suppression portant sur une vue en une mise à jour des tables de base
 - Exemples
 - Modifier un produit du fournisseur FLUX
 - Rajouter un fournisseur à la liste des fournisseurs de l'île de Montréal
 - Solution
 - En pratique, mise à jour réservé aux vues monotable avec obligation de contenir la clé

7

La notion de vue

- Mise à jour des données à travers une vue
 - Règle pour la mise à jour avec SQL
 1. La vue est définie comme une sélection (pas d'union par exemple)
 2. Pas de DISTINCT dans le SELECT
 3. Tout élément du SELECT est une référence à un attribut de la table sous-jacente
 4. La clause FROM contient une référence à une seule table
 5. Cette table peut être soit une table de base soit une vue qui peut être mise à jour
 6. Pas de clause WHERE qui contient une sous-requête dont la clause FROM fait référence à la table en question
 7. Pas de GROUP BY
 8. Pas de HAVING

8

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

La notion de vue

Cas particulier: Vue concrète

- Une vue calculée à partir des tables de base et stockée physiquement sur disque
- Elle est mise à jour dès qu'une transaction modifie les tables sous-jacentes par différentiel quand c'est possible
- Particulièrement intéressante pour mémoriser les agrégats

9

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

La notion de vue

Cas particulier: Vue concrète

- Syntaxe SQL
- ```
CREATE SNAPSHOT <une_vue_concrete>
AS <une_requête>
[REFRESH [FAST | COMPLETE | FORCE] [START WITH
date] [NEXT date]]
```
- Exemple
    - ventes de produits par jour

10

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

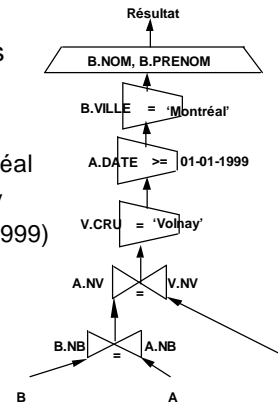
# Optimisation de requêtes

## Objectif

- Maintenir des performances acceptables lors de l'exécution des requêtes

## Exemple :

- Requête : Nom et prénoms des buveurs Montréalais de Volnay depuis 1999
- Contenu de la base :
  - 10 millions de buveurs dont 1 m à Montréal
  - 10 millions d'abus dont 10000 de Volnay et 1m depuis 1999 (2000 de Volnay depuis 1999)
  - 1000 vins
- Coût d'exécution :  $10m * 10m + \dots$

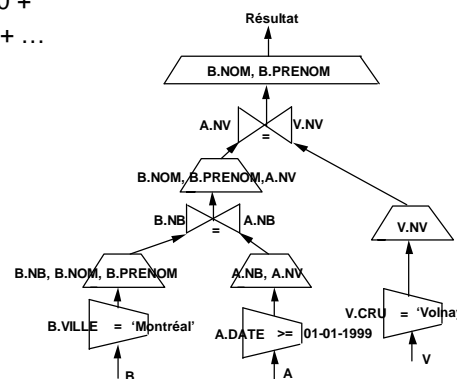


- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

# Optimisation de requêtes

## Autre solution

- Coût d'exécution :  $10m + 10m + 10000 + 1m * 1m + 2000 + \dots$



## Optimisation de requêtes

### ■ Problème :

- suivant l'ordre des opérateurs algébriques dans un arbre, le coût d'exécution est différent

### ■ Pourquoi?

- le coût des opérateurs varie en fonction du volume des données traitées. Plus le nombre de n-uplets des relations traitées est petit, plus les coûts cpu et d'E/S sont minimisés
- certains opérateurs diminuent le volume des données  
e.g., restriction et projection

13

## Optimisation de requêtes

### ■ Processus

1. Transformer la requête dans un format interne
2. Trouver une forme canonique
3. Décider de la stratégie d'exécution de bas niveau
4. Générer les plans de requêtes et choisir le meilleur

14

## Optimisation de requêtes

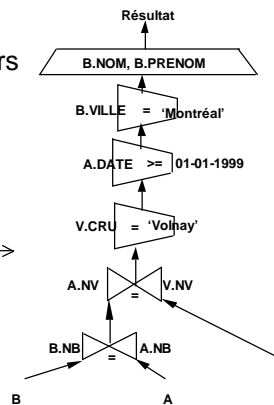
### ■ Transformer la requête dans un format interne

- Arbre de requête en algèbre relationnelle

#### ■ Exemple :

- Requête : Noms et prénoms des buveurs  
Montréalais de Volnay depuis 1999

```
SELECT NOM, PRENOM
FROM BUVEURS B, ABUS A, VINS V
WHERE B.NB = A.NB
AND A.NV = V.NV
AND CRU = 'Volnay'
AND DATE >= 01-01-1999
AND VILLE = 'Montréal'
```



15

## Optimisation de requêtes

### ■ Trouver une forme canonique

- Définitions

- Soit Q un ensemble de requêtes et R une relation d'équivalence entre ces requêtes (deux requêtes sont équivalentes si elles produisent le même résultats), un sous ensembles C de Q est un ensemble de formes canoniques de Q pour R si et seulement si toute requête q de Q est équivalente à exactement une requête c de C

- c est la forme canonique de q

16



- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

# Optimisation de requêtes

## ■ Trouver une forme canonique

### – Règles de transformation

1. Une suite restrictions sur une relation donnée peut être transformée en une restriction conjonctive sur cette relation
2. Dans une suite de projections sur une relation donnée, on peut garder uniquement la dernière
3. Une restriction d'une projection peut être transformée en une projection sur une restriction
4. Distributivité :  $f(A \circ B) \equiv f(A) \circ f(B)$  ou  $f$  est une opération sur une table (restriction, projection) et  $\circ$  une opération binaire (union, intersection, différence, jointure)

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

# Optimisation de requêtes

## ■ Trouver une forme canonique

### – Règles de transformation

4. Distributivité :  $f(A \circ B) \equiv f(A) \circ f(B)$

|             | Union | Intersection | Différence | Jointure                                               |
|-------------|-------|--------------|------------|--------------------------------------------------------|
| Restriction | Oui   | Oui          | Oui        | Si 2 cond., une pour chaque table                      |
| Projection  | Oui   | Oui          | Non        | Si les attributs de jointure inclus dans la projection |

## Optimisation de requêtes

### ■ Trouver une forme canonique

#### – Règles de transformation

5. Commutativité :  $A \circ B \equiv B \circ A$

union, intersection, jointure (pas différence, division)

6. Associativité :  $(A \circ B) \circ C \equiv A \circ (B \circ C)$

union, intersection, jointure (pas différence, division)

7. Idempotence :  $A \circ A \equiv A$

union, intersection, jointure (pas différence, division)

8. Simplification d'expressions arithmétiques

Exemple :  $a * b + a * c \equiv a * (b + c)$

9. Transformation d'expressions logiques

Exemple :  $a > b \text{ AND } b > 3 \equiv a > b \text{ AND } b > 3 \text{ AND } a > 3$

## Optimisation de requêtes

### ■ Trouver une forme canonique

#### – Règles de transformation

10. Transformations sémantiques

– Dépendent de la sémantique du domaine

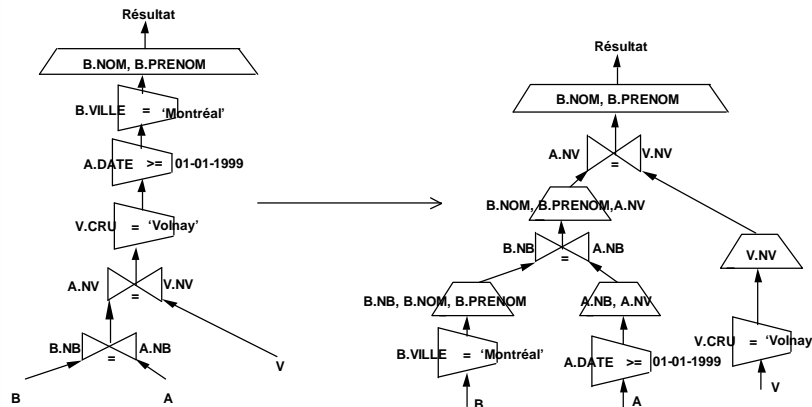
– Reliés aux contraintes d'intégrité

– Exemple : suppression de jointure dans certains cas

$\text{Project}(\text{Join}(\text{Abus}, \text{Vins}), \text{NV}) \equiv \text{Project}(\text{Abus}, \text{NV})$

# Optimisation de requêtes

## Trouver une forme canonique



# Optimisation de requêtes

## Décider de la stratégie d'exécution de bas niveau

- Tient compte des facteurs suivants
  - Les indexes
  - La distribution des valeurs de données mémorisées
  - L'emplacement des physique des données mémorisées
  - Etc.

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ **Concepts avancés**
- ☒ Modèle des objets
- ☒ BD à objets

## Optimisation de requêtes

### ■ Générer les plans de requêtes et choisir le meilleur

- Chaque opérateur algébrique est implémenté par plusieurs procédures
- Un plan d'exécution est une combinaison de procédures pour une requête donnée
- Il existe plusieurs plans possibles pour une requête
- Le coût de chaque plan est la somme des coûts des procédures qu'il contient
- Le coût de chaque procédure est estimé en fonction des résultats possibles de la procédure précédente
- Tous les plans ne sont pas générés. Des techniques de réduction de l'espace de recherche sont utilisées