

Introduction aux SGBD

Texte issu du cours de Claudia Roncacio (2006)

Définitions :

Base de données : C'est une collection de données inter-reliées, une entité cohérente, logique et véhiculant une certaine sémantique.

Système de gestion de BD : Ensemble de programmes qui permettent à des utilisateurs de créer et maintenir une BD.

Activités supportées :

- Définition d'une BD (spécification des types de données à stocker – LDD)
- Construction d'une BD (stockage des données – LMD)
- Manipulation des données (consultation, mise à jour, suppression – LDD)

Le SGBD sépare :

- La description des données dans le catalogue :
 - Schéma de la BD (peu évolutif)
 - Langage de définition des données (LDD)
- Données :
 - Extension
 - Langage de manipulation des données (LMD)

Modèle de données : C'est un ensemble de concepts permettant de décrire la structure d'une BD. La plupart des modèles incluent des opérations de manipulation des données.

Ex : Relationnel, Objet, Relationnel-Objet, semi-structuré, réseau, hiérarchique.

Plan du cours :

- Fonctions SGBD
- Modèle Relationnel
 - Algèbre
 - SQL
- Début Projet : Appli
 - JDBC
 - Transaction
- Modélisation
 - Dépendance fonctionnelle
 - EP
 - Passage Relationnel

Fonctions d'une SGBD

- Partage de données : Le SGBD doit permettre l'accès multi-utilisateur concurrents aux données.
 - Mécanismes de contrôle de concurrence
 - Notion de vue : permet de définir pour chaque « catégorie » d'utilisateurs la portion de la BD qu'il voit et dans la forme qui l'intéresse
- Gérer les autorisations d'accès
 - BD multi-utilisateurs : droits par utilisateurs (lecture, mise à jour, admin)
- Représentation de relations complexes entre les données
 - Dépend du modèle de données plus ou moins riches. Les plus expressifs sont les modèles à objets complexes
- Vérifier les contraintes d'intégrité (CI)
 - Contraintes:
 - Statiques : sur un état des données (predicats)
 - Dynamiques : sur plusieurs états
- Contrôler la redondance de l'information
 - Avec la redondance on a des problèmes de coût en mémoire, de temps de réponse et de cohérence.
 - Le SGBD doit offrir des moyens d'avoir une gestion unifiée des données.
- Assurer la sécurité et la reprise après panne
 - Les SGBD fournissent des mécanismes pour garantir que les modifications se font de manière correcte et durable.
 - Notion de transactions
 - Mécanismes de journalisation/reprise après panne. Utilisation de sauvegardes.
- Interfaces multiples
 - Différents type d'accès : graphique, langage procédural et langage déclaratif.

Architecture ANSI/SPARC :

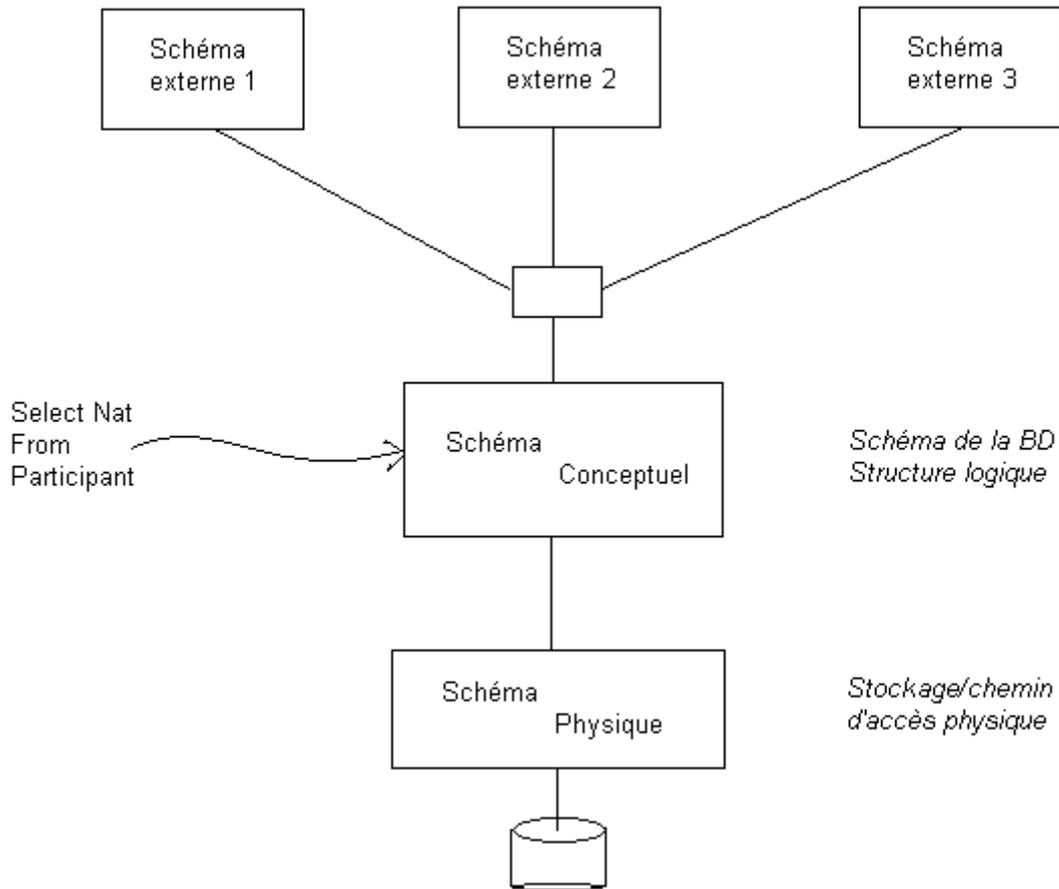
Relations:

Participant(ID : int, Nom : string, Prénom : string, Nat : string)

ID	Nom	Prénom	Nat
007	MOHR	Roger	Française
002	MOSSIERE	Jacques	Française
003	RONCANCIO	Claudia	Colombienne
001	PATERNAULT	Louis	Française

Inscriptions(ID : int, Tarif : float, Status : string)

ID	Tarif	Status
007	3000	Impayé
002	3000	Payé
001	1000	Payé
003	1500	Payé



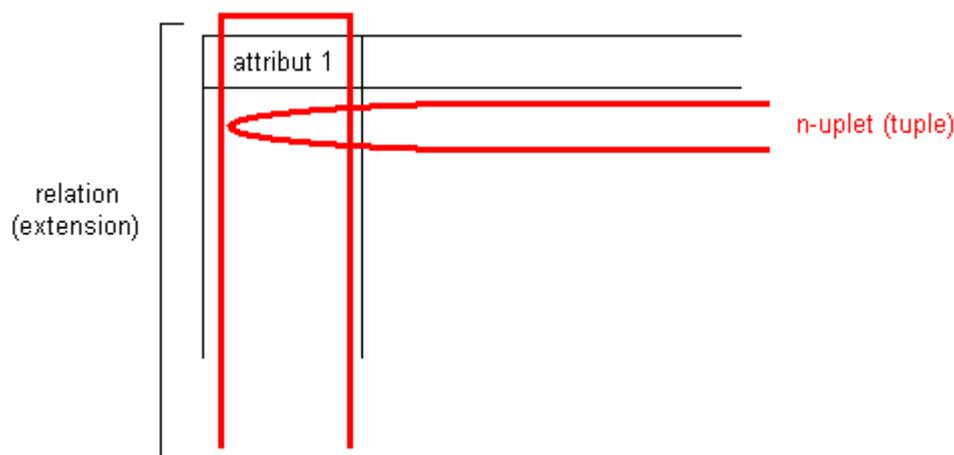
Create View :
 Select Nom, Prenom
 from Inscriptions, Participants
 where Inscriptions.ID = Participants .ID and Status = « impayé »

Indépendance programmes/données :
 Indépendance physique et logique. Permettre la modification d'un schéma à un certain niveau en limitant l'impact sur les autres niveaux.

Modèle Relationnel (Codd, 1970)

- Indépendance physique
- Bases solides pour traiter les problèmes de cohérence
- Langage déclaratif pour LMD, algèbre, standardisation (SQL).
- Modèle extensible : relations imbriquées, objet.

Relation R1 (attribut1 : type, ...) Schéma de la relation



Modèle fondé sur la théorie des relations (donc des ensembles)

Domaine : Ensemble de valeurs caractérisées par un nom.

Intention : par exemple int ou string.

Extension : par exemple CouleursVin { Rouge, Blanc, Rosé }

Relation : Sous-ensemble du produit cartésien d'une liste de domaines. Une relation à un nom.

Exemple : Localisation (Cru : string, Region : string, Prix : int)

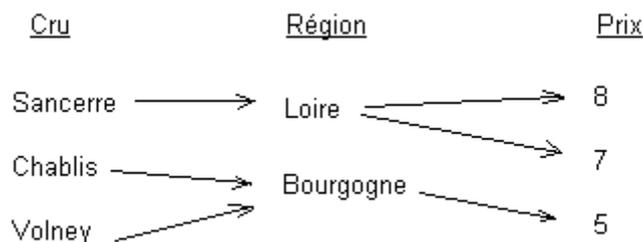


Schéma de relation (description de la relation) :

Ensemble d'attributs : Chaque attribut a un nom et un domaine. La relation a un nom. Les noms des attributs d'une même relation sont différents.

Arité d'une relation : Nombre d'attributs de la relation.

N-uplet (tuple) :

- Ligne de données d'une relation
- Un élément du produit cartésien des domaines des attributs de la relation.

Dans la relation, les n-uplets ne sont pas ordonnés.

Contraintes d'intégrité :

- Structurelle
- Valeurs (statiques ou dynamiques)

Contraintes de domaine :

Exemple : Relation Vins

Cru : string not null

Couleur : string in { Rouge, Rosé, Blanc }

Clés :

Clé primaire : ensemble minimal d'attributs dont la connaissance des valeurs permet d'identifier un n-uplet de la relation de façon unique.

Employé(**NSS : int**, Nom : string, Prénom : string, Sal : int)
Dept (**Ndept : int**, Chef : int, Budget : int)

Contrainte d'identité : les attributs de la clé primaire doivent avoir une valeur non nulle.

Contrainte de référence (ou CI référentielle) : CI portant sur une relation R_i consistant à imposer que la valeur d'un groupe d'attributs apparaisse comme valeur d'une clé (unicité) dans une autre relation.

Clé étrangère : groupe d'attributs qui apparaît comme clé dans une autre relation.

Algèbre Relationnel

Opérations :

– Ensemblistes

– Union : $R1 \cup R2 = R3$

Opération sur deux relations $R1$ et $R2$ de même schéma qui consiste à construire une relation $R3$ de même schéma qui contient l'union des n-uplets de $R1$ et $R2$.

– Différence : $R1 - R2 = R3$

Opération sur deux relations $R1$ et $R2$ de même schéma qui consiste à construire une relation $R3$ de même schéma qui contient les n-uplets qui sont dans $R1$ et pas dans $R2$.

– Produit cartésien : $R1 \times R2 = R3$

Opération sur deux relations $R1$ et $R2$ qui construit une relation $R3$ ayant pour schéma la concaténation des schémas de $R1$ et $R2$ et pour n-uplets toutes les combinaisons des n-uplets de $R1$ et $R2$.

– Spécifiques

– Projection : $\Pi_{\Delta_1, \Delta_2 \dots \Delta_k}(R1)$

Opération sur une relation $R1$ qui consiste à composer une nouvelle relation en enlevant de $R1$ les attributs non mentionnés, en opérandes, ceci au niveau schéma et données. Les n-uplets en double sont supprimés.

– Sélection ou restriction : $\sigma_{\text{Qualité = bonne}}(\text{Vignoble})$

Opération sur une relation $R1$ produisant une relation $R2$ de même schéma mais qui comporte seulement les n-uplets de $R1$ qui vérifient la condition de la sélection. Attention : la condition porte sur un n-uplet à la fois.

– Jointure : $R1 \bowtie R2$

$(R1.\Delta \text{ op } R2.\Delta) (\wedge \text{ ou } \vee)$

Opération consistant à rapprocher selon une condition les n-uplets de deux relations $R1$ et $R2$ afin de former $R3$ qui contient tous les n-uplets obtenus par concaténation d'un n-uplet de $R1$ et de $R2$ et qui vérifient la condition.

– Intersection: $R1 \cap R2 = R3 = (R1 \cup R2) - (R1 - R2) - (R2 - R1)$

Opération sur deux relations $R1$ et $R2$ de même schéma qui consiste à construire une relation $R3$ de même schéma qui contient l'intersection des n-uplets de $R1$ et $R2$.

– Jointure naturelle : $R1 \bowtie R2$

(pas de condition explicite)

Opération consistant à rapprocher les n-uplets de $R1$ et $R2$ afin de former $R3$ dont les attributs sont l'union des attributs de $R1$ et $R2$ et dont les n-uplets sont obtenus en composant un n-uplet de $R1$ avec un de $R2$ ayant les mêmes valeurs pour les attributs de même nom.

– Semi-jointure :

$R1(X)$ et $R2(Y)$ avec X, Y ensemble d'attributs.

$\Pi_X(R1 \bowtie_c R2)$: avec semi-jointure $R1 \bowtie_c R2$ (On conserve seulement les attributs de $R1$)

Opération portant sur deux relations $R1$ et $R2$ donnant en résultat les n-uplets de $R1$ qui participent à la jointure.

- Division : R (A1, A2 ... Ak ... An) et S(A1, A2 ... Ak)
 $R / S = T(A_{k+1}, \dots, A_n)$
 Opération consistant à construire le quotient de relation R par une relation S dont les n-uplets sont ceux qui, concaténés à tout n-uplet de S donnent un n-uplet de R.
 $Vins / bons = \Pi_{cru}(vins) - \Pi_{cru} [(\Pi_{cru}(vins) \times Bons) - Vins]$

- Jointure externe : R1 |>.<| R2
 Opération qui à partir de la jointure de R1 et R2 génère une relation R3 enrichit des n-uplets de R1 et R2 qui ne participent pas à la jointure avec les valeurs *null* pour les attributs de l'autre relation. Il existe une jointure gauche ou droite pour garder les n-uplets de R1 ou R2 respectivement.

Exemples Algèbre Relationnelle - C.L. Roncancio

extraits de Gardarin Valduriez...

Vins1(NV, Cru, Mill, Degré)

NV	Cru	Mill.	Degré
110	Chablis	1979	12
120	Sancerre	1980	12.5
130	Tokay	1980	12.5

Vins2(NV, Cru, Mill, Degré)

NV	Cru	Mill.	Degré
140	Chenas	1981	12.7
130	Tokay	1980	12.5

Vins1 U Vins2

NV	Cru	Mill.	Degré
110	Chablis	1979	12
120	Sancerre	1980	12.5
130	Tokay	1980	12.5
140	Chenas	1981	12.7

Vins1 - Vins2

NV	Cru	Mill.	Degré
110	Chablis	1979	12
120	Sancerre	1980	12.5

Crus(Cru, Region, Pays)

Cru	Region	Pays
Chenas	Beaujolais	France
Volnay	Bourgogne	France
Chanturgues	Auvergne	France

Année(Mill, Qualité)

Mill	Qualité
1979	Bonne
1980	Passable

Crus x Année = Vignoble

Cru	Region	Pays	Mill	Qualité
Chenas	Beaujolais	France	1979	Bonne
Chenas	Beaujolais	France	1980	Passable
Volnay	Bourgogne	France	1979	Bonne
Volnay	Bourgogne	France	1980	Passable
Chanturgues	Auvergne	France	1979	Bonne
Chanturgues	Auvergne	France	1980	Passable

Vignoble : Qualité = "Bonne"

Cru	Region	Pays	Mill	Qualité
Chenas	Beaujolais	France	1979	Bonne
Volnay	Bourgogne	France	1979	Bonne
Chanturgues	Auvergne	France	1979	Bonne

Vignoble [Mill, Qualité]

Mill	Qualité
1979	Bonne
1980	Passable

Vins(Cru, Mill, Qualité)

Cru	Mill	Qualité
Chablis	1977	Bonne
Chablis	1978	Mediocre
Volnay	1977	Bonne
Volnay	1978	Bonne
Saumur	1978	Bonne

Localisation(Cru, Region, QualitéMoy)

Cru	Region	QualitéMoy
Chablis	Bourgogne	Moyenne
Chablis	Californie	Bonne
Chenas	Beaujolais	Bonne

Bons(Mill, Qualité)

Mill	Qualité
1977	Bonne
1978	Bonne

Vins / Bons

Cru
Volnay

Vins * Localisation (jointure naturelle)

Cru	Mill	Qualité	Region	QualitéMoy
Chablis	1977	Bonne	Bourgogne	Moyenne
Chablis	1977	Bonne	Californie	Bonne
Chablis	1978	Mediocre	Bourgogne	Moyenne
Chablis	1978	Mediocre	Californie	Bonne

Structured Query Language (SQL)

Introduction

Développé par ANSI et standardisé par ISO/IEC en 1986. SQL contient le LDD (schéma) et le LMD (manipulation des données : insertion, suppression, mise à jour, interrogation).

Principaux mots-clefs (SFW) :

- *select* <liste des attributs à retenir>
- *from* <nom des relations arguments>
- *where*<conditions de sélection jointure>

Projection :

Commandes(Num, CNom, PNom, Qté)

Π_{CNom} (Commandes) :

```
Select Num, Cnom, Pnom, Qté
from Commandes
      ou
Select *
from Commandes
```

Sélection :

Exemple :

```
Select PNom
from Commandes
where CNom = « Jean » and Qté>200
```

Conditions de selection de base :

- attribut op valeur
 - attribut op attribut
- avec op : = != > <

Autres comparaison :

- Between 100 and 200
- is null / is not null
- Attribut in (v1,v2... vk)

Jointure :

Fourniture(PNum, Fnom, Prix)

$\Pi_{\text{PNom, FNom, Prix}}$ (Fourniture |><| Π_{CNom} ($\sigma_{\text{CNom=Jean}}$ (Commandes))) :

```
Select Commande.PNom, FNom, Prix
from Commandes, Fourniture
where CNom = « Jean » and Fourniture.PNom = Commande.PNom
```

Attention : Si on ne met pas de condition de jointure, on obtient un produit cartésien et pas une jointure naturelle.

Variables N-Uplets :

Désigne un n-uplet à la fois lorsqu'on l'utilise dans le where ou dans le select.

$\Pi_{\text{PNom, FNom, Prix}}$ (Fourniture |><| Π_{CNom} ($\sigma_{\text{CNom=Jean}}$ (Commandes))) :

```
Select Commande.PNom, FNom, Prix
from Commandes C, Fourniture F
where CNom = « Jean » and F.PNom = C.PNom
```

Expression ensemblistes :

Union :

```
Select Pnom
from Commande
where Cnom = Jean
Union
Select Pnum
from Commande
where Qté > 7000
```

Différence :

```
Select Pnom
from Commande
where Cnom = Jean
Except
Select Pnum
from Commande
where Qté > 7000
```

Intersection : *Intersect* (s'utilise de la même manière)

Imbrication de SFW :

```
Select F.PNom, Fnom, Prix
from Fourniture F, Commandes C
where F.PNom = C.PNom and Cnom = 'Jean'
```

```
Select Pnom, Fnom, Prix
from Fourniture
where Pnom in (Select Pnom
from Commandes
where Cnom = 'Jean')
```

Imbrication SFW dans une condition de sélection seulement à droite de l'opérateur.

ANY / ALL / EXISTS :

On cherche les fournisseurs de pains au chocolat à un coût inférieur au coût maximum des croissants :

```
Select Fnom
from Fourniture
where Pnom = 'Pain au chocolat' and Prix < Any (Select prix
from Fourniture
where Pnom = 'Croissant')
```

ANY : La condition Any f op ANY (SFW) est vraie si et seulement si la comparaison F op V est vraie pour **une** valeur V du résultat du SFW imbriqué.

ALL : La condition f op ALL (SFW) est vraie si et seulement si la comparaison f op V est vraie pour **toutes** les valeurs V du résultat du SFW imbriqué.

On cherche les fournisseurs qui proposent au moins un produit :

```
Select FNom
from Fournisseur
where EXISTS ( Select *
from Fourniture
where Fnom = F.FNom)
```

EXISTS : La condition EXISTS (SFW) est vraie si et seulement si le résultat du SFW n'est pas vide.

Fonctions de calcul :

Fonctions pour faire des calculs sur les valeurs des attributs ou pour compter

- Count()
- Min / MAX / AVG ...

```
Select count( * ), FNom  
from Fournisseur  
where ville = 'Grenoble'
```

Ces fonctions ne peuvent être utilisées qu'après select car il n'y a qu'un résultat pour count et plusieurs résultats pour FNom. On ne peut demander un attribut et une fonction de calcul en même temps.

On souhaite connaître le nombre de produits ainsi que le produit le moins cher.

```
Select count(distinct Pnom) as Total, min(Prix) as MinPrix  
from Fourniture
```

Réponse :

Total	MinPrix
300	0.35

Le mot-clef distinct sert à éliminer les doublons.

Agregation : Group by

```
Select count( * ), Ville, Status  
from Fournisseur  
group by Ville
```

Réponse:

Count(*)	Ville
13	Grenoble
8	Lyon
10	Paris

On peut sélectionner le nom de Ville car la requête est « group by » Ville mais pas Status. Pour avoir le status :

```
Select count( * ), Ville , Status  
from Fournisseur  
group by Ville, Status
```

Réponse:

Count(*)	Ville	Status
10	Grenoble	Actif
3	Grenoble	Inactif
8	Lyon	Actif
10	Paris	Inactif

- Opération de regroupement selon un ou plusieurs attributs.
- Les fonctions de calcul indiquées dans le select sont appliquées à chaque partition.
- Une réponse par partition.

Select count(*), Ville , Status
from Fournisseur
where Status = 'Actif'
group by Ville, Status

Réponse:

Count(*)	Ville
10	Grenoble
8	Lyon

Having :

Permet de donner une condition pour éliminer des groupes. Les conditions sont évaluées par groupe formé. Si la condition est vraie, le groupe est considéré dans la réponse.

Select count(*), Ville , Status
from Fournisseur
where Status = 'Actif'
group by Ville, Status
having count(*) > 9

Réponse:

Count(*)	Ville
10	Grenoble