Introduction aux Bases de Données

Support de cours

Compléments de cours:
http://dbweb.enst.fr/teaching/INF225.html
Autres supports de cours:

- *Database Management Systems*, Ramakrishnan & Gehrke
  - (voir aussi http://pages.cs.wisc.edu/~dbbook/)
- *Bases de Données*, G. Gardarin
- *Database systems: the complete book*, Molina, Ullman, Widom

Venez en amphi! Questions!

Plan de la 1ère partie du cours

- Bases de données et SGBD
- Modèle (de données) relationnel
  - Algèbre relationnelle

Base de Données et SGBD

- Une collection de données cohérentes entre elles, généralement de taille importante.
- Modélise une *entreprise* du monde réel
  - Entités (ex., films, acteurs)
  - Associations (ex., qui joue dans quel film)
- Un *Système de Gestion de Bases de Données (SGBD)* est un logiciel destiné au stockage et à la manipulation de bases de données.

Tables (ou relations)

Réalisateurs

Identifiant	Prénom	Nom
45601	George	Lucas
. . .		

RéalisateurFilm:

Identifiant_realisateur	Numéro_film
45601	10230
. . .	

Films:

Numéro	Titre	Année
10230	Star Wars	1977
. . .		

Role du SGBD

1. Création/stockage de données (vol. importants)
2. Interrogation/Mises à jour
3. Changement/évolution de la structure
4. Concurrence des accès (plusieurs utilisateurs)
5. Reprise sur panne
6. Intégrité des données et sécurité

…

Approches possibles…

- Fichiers
- Feuilles de calcul (ex., Excel)
- SGBD

1) Création/stockage de données?

- Fichiers
- Feuilles de calcul (ex., Excel)
- SGBD

2) Interrogation/Mises à jour?

- Fichiers
- Feuilles de calcul (ex., Excel)
- SGBD

3) Changement/évolution de la structure?

- Fichiers
- Feuilles de calcul (ex., Excel)
- SGBD

4) Concurrence?

Plusieurs utilisateurs accèdent / modifient les mêmes données simultanément

- Que peut-il arriver?
- Comment les SE (OS) gèrent ce problème?
- Insuffisant pour les BD; pourquoi ?

5) Reprise sur panne?

- Transférer 100 d'un compte vers un autre:

```
X = Read(compte #1);
X.somme = X.somme - 100;
Write(compte #1, X);

Y = Read(compte #2);
Y.somme = Y.somme + 100;
Write(compte #2, Y);
```

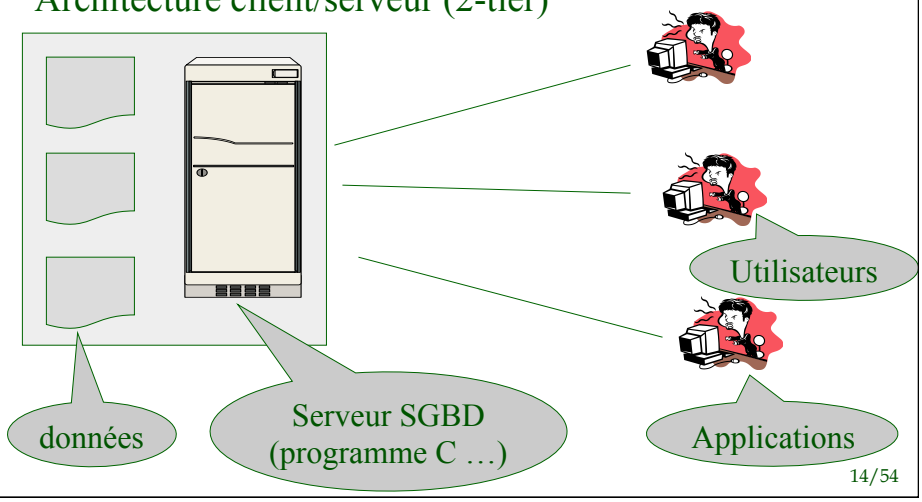
Panne !

6) Intégrité des données et sécurité?

- Fichiers
- Feuilles de calcul (ex., Excel)
- SGBD

SGBD

"Architecture client/serveur (2-tier)"



Role du SGBD

1. Création/stockage de données (vol. importants)
2. Interrogation/Mises à jour
3. Changement/évolution de la structure
4. Concurrence des accès (plusieurs utilisateurs)
5. Reprise sur panne
6. Intégrité des données et sécurité

…

DML

DDL

Transactions ACID

Grant, Revoke, roles

Avantages

En plus des 6 points précédents:

- Indépendance données/applications (suite)
- Temps de développement d'application réduit
- Administration des données uniforme
- Efficacité!

Modèle de données

- Un *modèle de données* est un ensemble de concepts sur les données.
- Un *schéma* est une description d'un ensemble de données, s'appuyant sur un modèle de données.
- Le *modèle relationnel* est le plus répandu.
  - Concepts de base: *relation*, table avec tuples et des colonnes.
  - Chaque relation a un *schéma*, qui décrit ses colonnes.

1ère example: IMDB

Réalisateurs

Identifiant	Prénom	Nom
45601	George	Lucas
. . .		

RéalisateurFilm:

Identifiant_realisateur	Numéro_film
45601	34351
. . .	

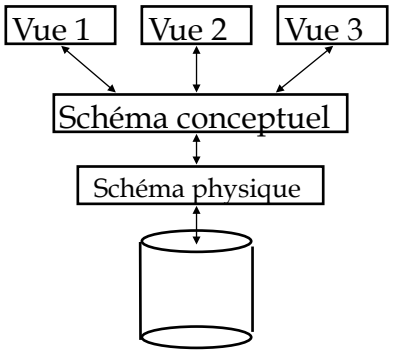
Films:

Numéro	Titre	Année
34351	Star Wars	1977
. . .		

Indépendance données / applications

Les 3 niveaux d'abstraction:

- Plusieurs *vues*, un seul *schéma conceptuel (logique)* et *schéma physique*.
  - Les vues décrivent comment l'utilisateur voit les données.
  - Le schéma conceptuel définit la structure logique des données.
  - Le schéma physique décrit les fichiers et les index utilisés.



2ème exemple: base de données d'une agence bancaire

- Schéma conceptuel (relations):
  - *Clients(idl: integer, Nom: string, Prénom: string, Profession: string)*
  - *Comptes(idc:string, type : string, solde:string, idl: integer)*
  - *Opérations(idc:string, dte:date, type:string, montant : integer)*
- Schéma physique :
  - Les relations sont stockés dans des fichiers non séquentiel, index, etc
- Schéma externe (Vues):
  - *Cumul_mensuel(mois: string, idc:string, Nom: String, Prénom: string, total_crédits:integer, total_débits: integer)*

Indépendance d/a: avantages

- Les applications sont isolées des changements de structure et du mode de stockage des données.
- *Indépendance logique des données*: Protection des changements de structure des données au niveau *logique*.
- *Indépendance physique des données*: Protection des changement de structure au niveau physique.

☛ *Un des plus importants bénéfices de l'utilisation des SGBD*

Contrôle de concurrence

- L'exécution concurrente de programmes est essentielle pour un SGBD.
  - Les accès disque sont fréquents et relativement *lents*,
  - L'exécution partielle des actions de différents programmes peut aboutir à des incohérences:
- Les SGBD assurent que la concurrence soit réalisée sans problème: chaque utilisateur a l 'impression d'être seul à travailler sur le système.

Exécution d'un programme au-dessus d'une BD

- Concept clé : *transaction*, une séquence atomique d'actions sur une BD (lectures/écritures).
- Chaque transaction est sensée laisser la BD dans un *état cohérent* après l'avoir prise dans un état cohérent.
  - Les utilisateurs peuvent spécifier des *contraintes d'intégrité* simples sur les données.
    - Le SGBD n'a pas conscience de la sémantique des traitement effectués.
  - Le fait qu'une transaction préserve la cohérence de la BD est au bout du compte de la responsabilité de l'utilisateur!

Sûreté des traitements

- Les SGBD assurent la cohérence des données même en cas de crash.
- Idée: garder un *journal ou log* (historique) de toutes les actions élémentaires de m-à-j et de validation réalisées par le SGBD :
  - Avant qu'un changement ne soit réalisé, l'action est tracée dans un log file.
  - Après un crash, l'effet des transactions non abouties est annulé à l'aide du fichier log.

Les bases de données font «les affaires» de beaucoup de monde ...

- Les utilisateurs finaux et les éditeurs de SGBD
  - DB2 (IBM) – 35%, SQL Server (MS) - 19%, Oracle 33%, Sybase
  - MySQL, Postgres, …
- Les développeurs d'applications BD

- *Administrateurs de BD (DBA)*
  - conception logique / physique de schéma
  - gère la sécurité et les droits
  - la disponibilité des données, reprise sur panne
  - paramétrage *(tuning)* en fonction des besoins & statistiques

Doit comprendre le fonctionnement des SGBD!

Résumé

- Les SGBD sont utilisés pour maintenir et interroger un volume de données important.
- Quelques bénéfices : reprise sur panne, accès concurrent, développement rapide d'applications, sécurité des données.
- Les niveaux d'abstraction permettent l'indépendance données/ applications.
- Backend pour applications BD traditionnelles, applications Web (ex., Amazon, Ebay).
- La R&D en BD est très active (nouvelles technos, nouvelles applications, nouveaux modèles, architectures, etc)
  - BDs probabilistes, stream data (flux de données), « data management » au-delà des SGBD relationnels: social & collaborative data management, le « cloud » , Web-scale data management

INF225

Leçons:
- Modèle relationnel
- SQL
- Vues
- Design des BD: modélisation & normalisation
- Données semi-structurées (XML)

TPs:
- SQL
- PL/SQL, mises à jours et vues
- Applications BD & Web

INF345 BD Avancées (S1P2/P4 ?)
- transactions, indexage, optimisation de requêtes, BDs reparties, MapReduce, recherche sur le Web, …
- projet TinyBase

Plan de la 1ère partie du cours

- Bases de données et SGBD
- Modèle (de données) relationnel
  - Algèbre relationnelle

Concepts descriptifs

T. Codd (IBM San Jose) en 1970

- Domaine de valeurs (ensemble de valeurs)
  - Exemple : Entier, Réel, …, Franc, Salaire = {5 000..100 000}, Point = {(X:Réel,Y:Réel)}
- Le produit cartésien D1x D2x ... x Dn est l'ensemble des tuples (n-uplets) : <V1,V2,…, Vn> tel que Vi ∈ Di
  - Exemple : D1(clients) = {Dupond, Dupont, Tryphon}, D2(solde) = {100, 10000}

Dupond	100
Dupont	100
Tryphon	100
Dupond	10000
Dupont	10000
Tryphon	10000

Relation

- Sous-ensemble du produit cartésien d'une liste de domaines
- Chaque relation possède un nom
- Vision tabulaire
  - Une relation est une table à deux dimensions
  - Une ligne est un tuple
  - Un nom est associé à chaque colonne
- Attribut :
  - nom donné à une colonne d'une relation
  - prend ses valeurs dans un domaine

Exemple de relation

VINS	CRU	MILL	REGION	COULEUR
	CHENAS	1983	BEAUJOLAIS	ROUGE
	TOKAY	1980	ALSACE	BLANC
	TAVEL	1986	RHONE	ROSE
	CHABLIS	1986	BOURGOGNE	BLANC
	ST-EMILION	1987	BORDELAIS	ROUGE

Clé

- Groupe d'attributs minimum d'une relation qui détermine chaque tuple de façon unique
  - Exemples : {Cru,Millésime} dans la table VINS, NSS pour des PERSONNES
- Clé primaire : Clé choisie par le concepteur de la base de données
- Clé étrangère : Groupe d'attributs d 'une relation R_1 devant apparaître comme clé primaire dans une autre relation R_2
  - Les clés étrangères définissent les contraintes d'intégrité référentielles

Schéma

- Schéma d'une relation : Nom de la relation, liste des attributs avec leurs domaines, (+Clé primaire)
- Exemple :

  VINS (NV : Entier, Cru : Texte, Mill : Entier, Degré : Réel, Région : Texte)

  Par convention, la clé primaire est soulignée
- Intention et extension
  - Un schéma de relation définit l'intention de la relation
  - Une instance de table représente une extension de la relation
- Schéma d'une BD relationnelle
  - C'est l'ensemble des schémas des relations composantes

Exemple de schéma

BUVEURS (NB, Nom, Prénom, Type)
VINS (NV, Cru, Mill, Degré)
ABUS (NB, NV, Date, Quantité)

clés étrangères

ABUS.NV Référence VINS.NV
ABUS.NB Référence BUVEURS.NB

Résumé des concepts descriptifs

- Relation ou Table
- Attribut ou Colonne
- Domaine ou Type
- Clé
- Clé primaire
- Clé étrangère

Plan de la 1ère partie du cours

- Bases de données et SGBD
- Modèle (de données) relationnel
  - Algèbre relationnelle

Concepts de manipulation

- DML (langage de manipulation des données)
  - expressions (requêtes) pour
    extraire, modifier, effacer, insérer des tuples
- manipulation ensembliste

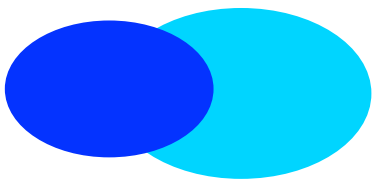
- Algèbre relationnelle *procédurale*
- Calcul relationnel *déclaratif*

SQL: standard pour les SGBD relationnelle

Expressions SQL (calcul) -> Plan d'exécution (algèbre)

Concepts de manipulation: l'algèbre

- Un ensemble d'opérations formelles (algèbre relationnelle), permettant d'exprimer les requêtes sous forme algébrique
  - Ces opérations sont utiles pour l'optimisation des requêtes
- Opérations algébriques :
  - Unaires : O(Relation) → Relation
  - Binaires : Relation X Relation → Relation
- Opérations ensemblistes
  - UNION notée ∪
  - INTERSECTION notée ∩
  - DIFFERENCE notée –
  - …



Algèbre relationnelle

- Cinq opérateurs de base :
  - Projection π
  - Restriction σ
  - Produit cartésien X
  - Deux opérateurs ensemblistes :
    - Union ∪
    - Intersection ∩
- + opérateur de renommage ρ
- + opérateurs composés :
  - Jointure ⋈
  - Division ÷
  - Différence -

Projection

- Elimination des attributs non désirés et suppression des tuples en double
- Opérateur unaire, $\pi_{A1,A2,...Ap}$ (R)

VINS	Cru	Mill	Région	Qualité
	VOLNAY	1983	BOURGOGNE	A
	VOLNAY	1979	BOURGOGNE	B
	CHENAS	1983	BEAUJOLAIS	A
	JULIENAS	1986	BEAUJOLAIS	C

$\pi_{cru,Région}$

π(VINS)	Cru	Région
	VOLNAY	BOURGOGNE
	CHENAS	BEAUJOLAIS
	JULIENAS	BEAUJOLAIS

Restriction (ou selection)

- Obtention des tuples de R satisfaisant un critère Q
- Opérateur unaire, notée $\sigma_Q(R)$
- Q est le critère de qualification de la forme :

  A_i θ Valeur

  θ ∈ { =, <, >=, >, <=, ≠}
- Il est possible de réaliser des "ou" (union) et des "et" (intersection) de critères simples

Exemple de restriction

VINS	Cru	Mill	Région	Qualité
	VOLNAY	1983	BOURGOGNE	A
	VOLNAY	1979	BOURGOGNE	B
	CHENAS	1983	BEAUJOLAIS	A
	JULIENAS	1986	BEAUJOLAIS	C

$\sigma_{MILL>=1986}$

VINS	Cru	Mill	Région	Qualité
	JULIENAS	1986	BEAUJOLAIS	C

$\sigma_{CRU="VOLNAY"}$

$\sigma_{CRU="CHENAS"}$

Jointure ⋈

- Composition des deux relations sur un domaine commun
- Relation binaire équivalente à une restriction sur un produit cartésien
  R1 $\bowtie_{\theta}$ R2 équivalent à σ_{θ} **(R1 X R2)**
- Critère de jointure
  - Attributs de même nom égaux : Attribut = Attribut
    - Jointure naturelle
  - Comparaison d'attributs : Attribut1 θ Attribut2
    - Thêta-jointure

Exemple de jointure

VINS	Cru	Mill	Qualité
	VOLNAY	1983	A
	VOLNAY	1979	B
	CHABLIS	1983	A
	JULIENAS	1986	C

⋈

LOCALISATION	Cru	Région	QualMoy
	VOLNAY	Bourgogne	A
	CHABLIS	Bourgogne	A
	CHABLIS	Californie	B

↓

VINSREG	Cru	Mill	Qualité	Région	QualMoy
	VOLNAY	1983	A	Bourgogne	A
	VOLNAY	1979	B	Bourgogne	A
	CHABLIS	1983	A	Bourgogne	A
	CHABLIS	1983	A	Californie	B

Exemples de requêtes

Effacer les vins de qualité C.

Vins := Vins - $\sigma_{qual=C}$ (Vins)

Quelles sont les régions qui ont produit du vin de qualité A en 1983?

Res := π_{region} [$\sigma_{mill=1983,\ qual=A}$ (Localisation I><I Vins)]

Division

- Ce n'est pas un opérateur de base, mais utile pour exprimer des requêtes du type:
  Trouver les personnes qui ont réservé tous les bateaux.
- Si *A* a 2 attributs, *x* et *y*; et *B* a un seul attribut, *y*:
  - *A/B* = $\{\langle x\rangle \mid \exists \langle x, y\rangle \in A \ \ \forall \langle y\rangle \in B\}$
  - ***A/B* contient tous les tuples x (personne) t.q. pour chaque tuple y (bateau) dans *B*, il existe un tuple *xy* dans A.**.
- En général, *x* et *y* sont des listes d'attributs; *y* est la liste d'attributs de B, et $x \cup y$ est la liste d'attributs de A.

Exemples de division A/B

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3

Exprimer A/B avec les opérateurs de base

- *L'idée*: Pour *A/B*, calculer tous les tuples *x* qui ne sont pas `disqualifiés' par un tuple *y* de *B*.
  - *x* est *disqualifié* si avec le tuple *y* de *B*, ont obtient une paire *xy* qui n'est pas dans A.

Valeurs *x* disqualifiées: $\pi_x((\pi_x(A) \times B) - A)$

A/B: $\pi_x(A)$ − valeurs x disqualifiées

Optimisation de requêtes

Res := π_{region} [$\sigma_{mill=1983,\ qual=A}$ (Localisation I><I Vins)]

vs

Res' := π_{region} [Localisation I><I $\sigma_{mill=1983,\ qual=A}$ (Vins)]

- Plans d'exécution (arbres algébriques)
  - Heuristiques: projeter les attributs inutiles, « pousser » les restrictions, ordonner l'exécution des jointures, etc.

Algèbre

- Notion de relation
  - Schéma : Nom de relation + ensemble d'attributs
  - Extension : ensemble de tuples (n-uplets)
- Cinq opérations de base
  - Π, Projection
  - σ, Restriction
  - X, Produit cartésien
  - ∪, Union
  - -, Différence
- Autres opérations déduites
  - I><I, Jointure
  - ∩, Intersection
  - ÷, Division
  - ρ, Renommage

Calcul relationnel
ou calcul à variable n-uplet

- Langage formel basé sur la logique des prédicats du premier ordre
- Requêtes en calcul relationnel {t | ϕ(t) }
  - t désigne une variable n-uplet et ϕ(t) une formule bien formée
- Formules, composées à partir de :
  - termes atomiques : variables t, noms de relations et constantes
  - opérateurs ∈, =, <, >, …
  - connecteurs AND (∧) , OR (v) et négation
  - Quantificateurs ∃ et ∀
- SQL
  - Version commerciale du calcul relationnel

Exemple de requête en calcul

- Schéma :

  Viticulteurs (NVT, Nom, Prénom, Ville, Région)
  VINS (NV, Cru, Millésime, Degré, NVT, Prix)
  BUVEURS (NB, Nom, Prénom, Ville)
  ABUS (NV, NB,Date,Qté)

Quels sont les viticulteurs qui ont produit au moins un vin de 1983 (nom et cru) ?

{t:cru, nom | ∃ v ∃ w (v ∈ vins ^ w ∈ viticulteurs ^
t(cru) = v(cru) ^ t(nom)=w(nom) ^
w(nvt) = v(nvt) ^ v(mill) =1983 }

Résumé du modèle relationnel

- Un ensemble de concepts bien compris et bien formalisé
- Un modèle unique, normalisé et de plus en plus riche
- Optimisation!
- Un formalisme qui s'étend plutôt bien
  - algèbre d'objets
  - algèbre pour le semi-structuré (XML)

A lire sur le Web

SQL for Web Nerds, de Philip Greenspun,

http://philip.greenspun.com/sql/introduction