

# Introduction aux bases de données

L2A

Semestre 4

Mehdi Benzine

# Objectifs du cours

- Comprendre les objectifs, les langages et les architectures des bases de données
- Apprendre à concevoir, manipuler et interroger des bases de données relationnelles

Volume horaire:

3h cours + 1.5 h TD + 1.5h TP par semaine

# Bibliographie

1. Bases de données Georges Gardarin S8/63323-25 S12/1318-1323
2. Création de bases de données Nicolas Larousse S8/72256-...59 Collection Syntex
3. Introduction aux bases de données relationnelles Mata Toledo S4/20035-20037 , S4/20510
4. Programmation SQL Mata Toledo S4/20513
5. Introduction aux bases de données Chris J. Date S8/59384-87 S8/62600-62602
6. SQL pour Oracle Christian Soutou S8/77467
7. Bases de données relationnelles concepts, mise en œuvre et exercices Claude Chrismont S8/74683



# Introduction

- Les organisations gèrent des volumes de données très grands
  - Giga, Tera, Peta –octets
  - Numériques, Textuelles, Multimédia (images, vidéos,...)
- Il faut pouvoir facilement
  - Archiver les données sur mémoires secondaires permanente
  - Retrouver les données pertinentes à un traitement
  - Mettre à jour les données variant dans le temps
- Les données sont structurées et identifiées
  - Données élémentaires ex: nom, note en BD, date de naissance ...
  - Données composées ex: CV, résultats de l'année...
  - Identifiant humain ex: numéro d'étudiant, numéro d'immatriculation de véhicule
- Qu'est-ce qu'une base de données ?
  - Collection de données structurées
  - Interrogeable et modifiable par des langages de haut niveau (proches du langage naturel)
  - Partagée par plusieurs applications/utilisateurs

# Historique

Années 60:

- Fichiers séquentiels

- Accès séquentiel aux données puis sur clé.

Années 70:

- Bases de données hiérarchiques puis réseaux

- Données stockées dans des fichiers et reliées par des pointeurs.

- Interrogation par navigation

Années 80:

- Bases de données relationnelles

- Relation entre ensembles de données

- Interrogation par un langage proche du langage naturel

- Modèle le plus utilisé jusqu'à aujourd'hui.

Années 90 et 2000:

- Bases de données objets, XML ...



# Gestion des données (1)

La gestion de données par l'utilisation de fichiers présente de nombreux inconvénients:

1. Absence de standardisation  
plusieurs formats de stockage, plusieurs langages
2. Redondance des données (duplication)  
Problèmes de mise à jour, incohérence des données
3. Interrogation par langage de programmation  
Difficulté de maintenance  
Coût élevé

# Gestion des données (2)

- 4. Pannes (arrêt brutal, panne de disque ...)  
pas de solution standardisée
- 5. Partage de données  
pas de solution standardisée
- 6. Confidentialité  
Pas de solution standardisée

Pas de solution standardisée => une solution doit être conçue et implémentée pour chaque application.



# L'approche "bases de données"

## Modélisation des données

- Éliminer la **redondance** de données
- **Centraliser** et **organiser** correctement les données
- Plusieurs niveaux de modélisation
- Outils de conception

## Logiciel de **S**ystème de **G**estion de **B**ases de **D**onnées (SGBD)

- Factorisation des modules de contrôle des applications: interrogation, pannes, confidentialité, partage ...
- Administration facilitée des données



# SGBD relationnels (SGBDR)

Logiciels commerciaux/payants:

- Oracle (plus de 60% du marché mondial des SGBDR)
- Microsoft SQL server
- IBM DB2
- Sybase Anywhere
- ...

Logiciels libres/gratuits:

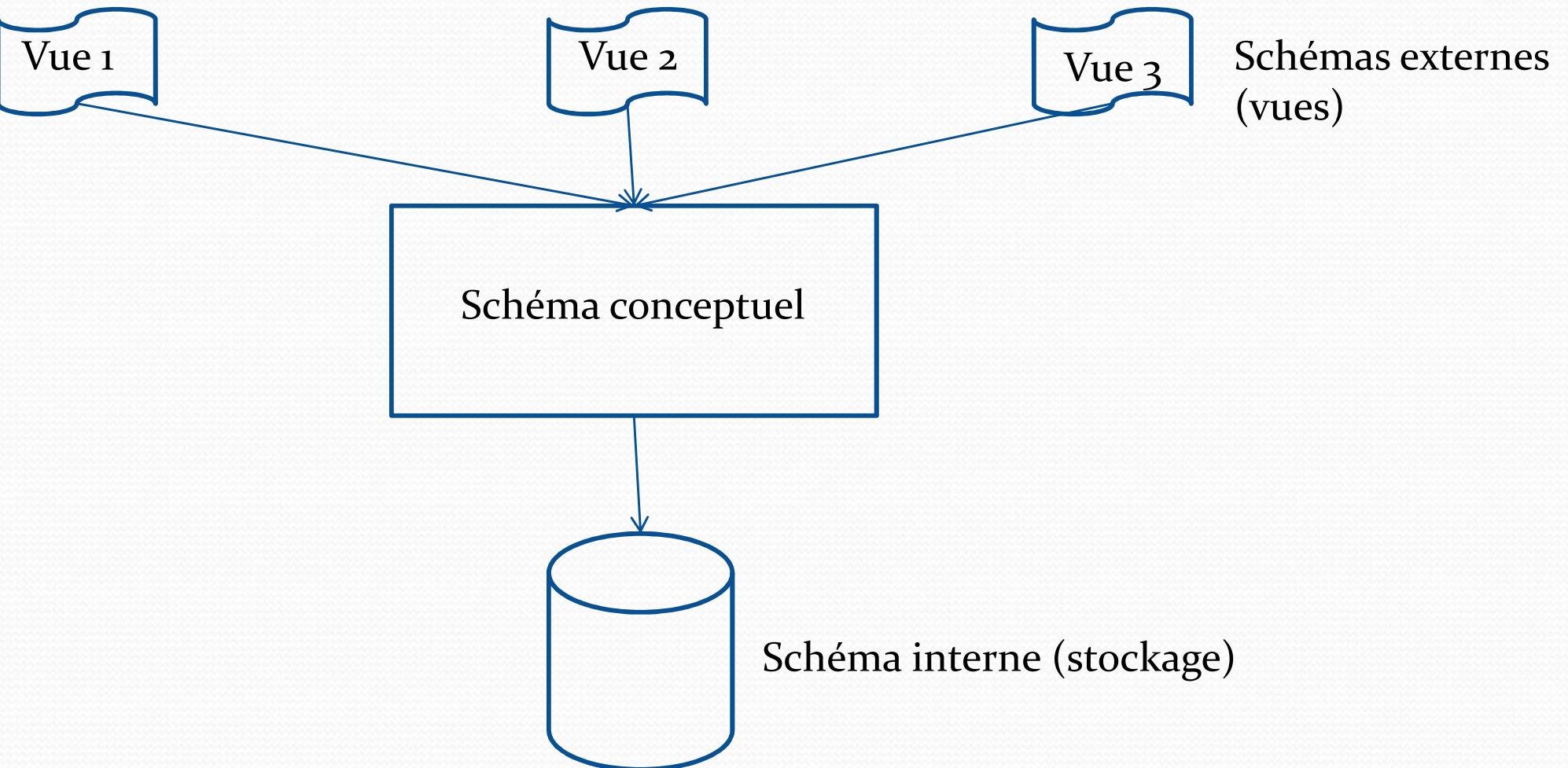
- MySQL
- PostgreSQL
- mSQL
- ...

# Objectifs des SGBD

1. Indépendance physique des données
2. Indépendance logique des données
3. Manipulation simple
4. Gestion des vues
5. Optimisation des questions
6. Gestion de la cohérence
7. Gestion des pannes
8. Concurrency d'accès
9. Gestion de la confidentialité
10. Standards



# Architecture à 3 niveaux



# Indépendance Physique

- Indépendance des programmes d'applications vis à vis du modèle physique :
  - Possibilité de modifier les **structures de stockage** (fichiers, index, chemins d'accès, ...) sans modifier les programmes;
  - Ecriture des applications par des **non-spécialistes des fichiers** et des structures de stockage;
  - Meilleure **portabilité** des applications et **indépendance** vis à vis du matériel.



# Indépendance logique

- Les applications peuvent définir des **vues logiques** de la BD
- Possibilité pour chaque application **d'ignorer** les besoins des autres (bien que partageant la même BD).
- Possibilité **d'évolution de la base de données** sans réécriture des applications :
  - ajout de champs, ajout de relation, re-nommage de champs.
- Possibilité **d'intégrer des applications existantes** sans modifier les autres.
- Possibilité de limiter les conséquences du partage :  
**Données confidentielles.**

# Manipulation simple

- La manipulation se fait via un langage **déclaratif**
  - La question déclare l'objectif sans décrire la méthode
  - Le langage suit une norme commune à tous les SGBD
  - **SQL : Structured Query Language**
- Sémantique
  - Logique du 1er ordre
- Syntaxe (aperçu !)  
SELECT <structure des résultats>  
FROM <relations>  
WHERE <conditions>



# Des vues multiples des données

- Les vues permettent d'implémenter l'indépendance logique en permettant de créer des **relations virtuelles**
- Vue = Question stockée
- Le SGBD stocke la **définition** et non le résultat
- Exemple :
  - la vue des patients sétifiens
  - la vue des projets de chaque service (chaque employeur ne peut voir que les projets de son service)
  - La vue des services statistiques
  - ...

# Exécution et Optimisation

- Traduction **automatique** des questions déclaratives en programmes procéduraux :
  - ➔ Utilisation de l'algèbre relationnelle
- Optimisation **automatique** des questions
  - ➔ Utilisation de l'aspect déclaratif de SQL
  - ➔ Gestion centralisée des chemins d'accès (index, hachages, ...)
  - ➔ Techniques d'optimisation poussées
- Economie de l'astuce des programmeurs
  - milliers d'heures d'écriture et de maintenance de logiciels.



# Intégrité Logique

- Objectif : Détecter les **prises à jour erronées**
- Contrôle sur les données élémentaires
  - Contrôle de types: ex: Nom alphabétique
  - Contrôle de valeurs: ex: Salaire mensuel entre 15000 et 500000
- Contrôle sur les relations entre les données
  - Relations entre données élémentaires:
    - Prix de vente > Prix d'achat
  - Relations entre objets:
    - Un employeur ne doit être rattaché qu'à un seul service.

# Contraintes d'intégrité

- **Avantages :**

- **simplification** du code des applications
- **sécurité renforcée** par l'automatisation
- **mise en commun** des contraintes

- **Nécessite :**

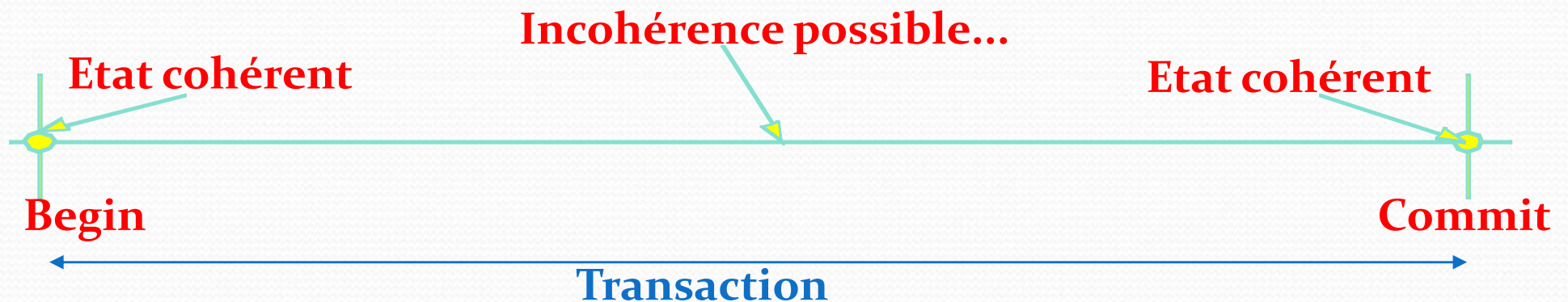
- un langage de définition de contraintes d'intégrité
- la vérification **automatique** de ces contraintes



# Intégrité Physique

- **Motivations : Tolérance aux fautes**
  - Transaction Failure : Contraintes d'intégrité, Annulation
  - System Failure : Panne de courant, Crash serveur ...
  - Media Failure : Perte du disque
  - Communication Failure : Défaillance du réseau
- **Objectifs :**
  - Assurer l'**atomicité** des transactions
  - Garantir la **durabilité** des effets des transactions commises
- **Moyens :**
  - Journalisation : Mémorisation des **états successifs** des données
  - Mécanismes de reprise

# Transaction



**Begin**

Compte1 = Compte1 - 3000

Compte2 = Compte2 + 3000

**Commit T1**



# Atomicité et Durabilité

## ATOMICITE

Begin

Compte1 = Compte1 - 3000

Compte2 = Compte2 + 3000

Commit T1

Panne

→ Annuler le débit !!

## DURABILITE

Begin

Compte1 = Compte1 - 3000

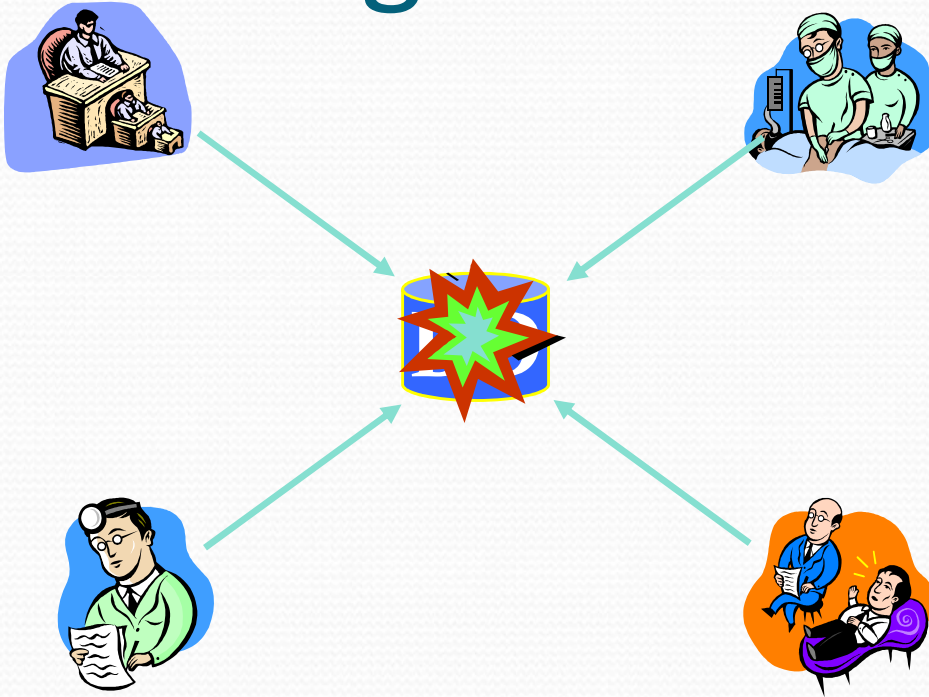
Compte2 = Compte2 + 3000

Commit T1

Crash disque

→ Restaurer les données telles qu'elles étaient avant la transaction

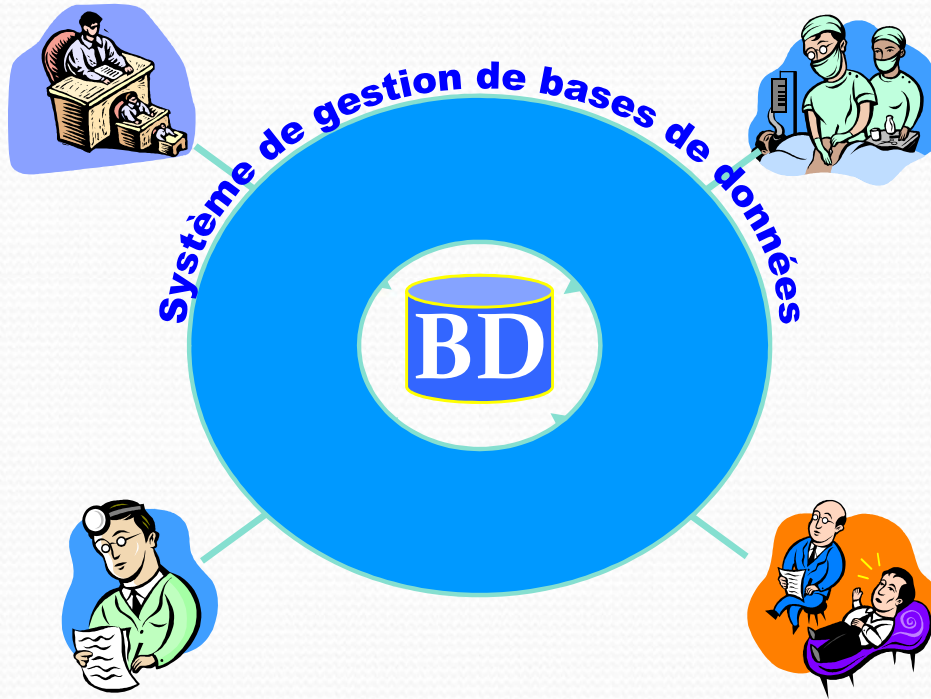
# Partage des données



- Accès concurrent aux mêmes données  
➔ Conflits d'accès !!



# Isolation et Cohérence



- Le SGBD gère les accès concurrents
  - ➔ Chacun à l'impression d'être seul (Isolation)
  - ➔ Cohérence conservée (Pas de mises à jour conflictuelles)

# Confidentialité

- Objectif : Protéger les données de la BD contre des accès non autorisés
- Deux niveaux :
  - Connexion restreinte aux **utilisateurs répertoriés** (mot de passe)
  - **Privilèges** d'accès aux objets de la base
- Objets : Relation, **Vue**, autres objets (procédures, etc.)



# Standardisation

- L'approche bases de données est basée sur plusieurs standards
  - Langage SQL (SQL<sub>1</sub>, SQL<sub>2</sub>, SQL<sub>3</sub>)
  - Communication SQL CLI (ODBC / JDBC)
  - Transactions (X/Open DTP, OSI-TP)
- Force des standards
  - Portabilité
  - Interopérabilité
  - Applications multisources...