



---

# **BASES DE DONNEES**

Victor FERNANDES



---

# Première Partie

## INTRODUCTION

# Partie I

## Sommaire

---

- Préambule
- Notions de base de données (rappels)
- Formalisme de représentation des données
- Les 12 règles de Codd
- Architecture
- Schémas d'une base de données
- Utilisateurs
- Langages
- Systèmes

# Partie I

## Introduction

- Avant de penser BD, il convient de poser des questions, étudier, analyser les besoins, afin de récolter les informations nécessaires à la conception de la BD.
  
- La méthode d'analyse la plus complète est MERISE
  - Une méthode globale et intégrée
  - permet de repenser l'organisation actuelle en terme de SAI
  - Caractéristique principale => séparation des données / traitements
  
- MERISE est composé de 3 niveaux de modélisation
  - niveau conceptuel
  - niveau organisationnel
  - niveau technique

# Partie I

## Introduction

- **Le niveau conceptuel** répond à la question "QUOI ?"
  - que faire ? avec quelles données ?
  - Le niveau est indépendant des contraintes organisationnelles et techniques.
  - Il comporte le modèle conceptuel des données (MCD)
  
- **Le niveau organisationnel** répond aux questions "QUI, QUAND, OU ?".
  - il comporte le modèle logique des données (MLD)
  - et comporte le modèle organisationnel des traitements (MOT)
    - ✓ qui permet de modéliser les notions de temps et de durée, les postes et leur responsabilités, ainsi que les lieux et leurs ressources
  
- **Le niveau technique** répond à la question "COMMENT ?", en intégrant les moyens techniques, matériels et logiciels
  - Il comporte le modèle physique des données (MPD)
  - et le modèle opérationnel des traitements (MOpT)

# Partie I

## Introduction

	Données	Traitement
Conceptuel	<ul style="list-style-type: none"><li>•Entité</li><li>•Relation</li><li>•Propriété</li></ul> MCD	<ul style="list-style-type: none"><li>•Processus</li><li>•Opération</li><li>•Événement</li><li>•Synchronisation</li></ul> MCT
Organisationnel	<ul style="list-style-type: none"><li>•Entité</li><li>•Relation</li><li>•Propriété</li></ul> MOD	<ul style="list-style-type: none"><li>•Phase</li><li>•Procédure</li><li>•Tâche</li></ul> MOT
Logique	<ul style="list-style-type: none"><li>•Table</li><li>•Attribut</li></ul> MLD	<ul style="list-style-type: none"><li>•Phase</li><li>•Procédure</li><li>•Tâche</li><li>•Fonction</li></ul>
Physique	<ul style="list-style-type: none"><li>•Fichier</li></ul> MPD	<ul style="list-style-type: none"><li>•Programmes</li></ul>

# Partie I

## Notions de base

- **Base de données** est un ensemble de données qui modélise les objets d'une partie du monde, et qui sert de support à une application informatique
  - Capacité de représentation (modèle relationnel) ?
  - Redondance ?
  - Cohérence ?
  - Facilité d'accès ?
  
- **Système de gestion de bases de données (SGBD)** est un logiciel permettant d'interagir avec une base de données
  - Concurrence et cohérence ?
  - Fiabilité ?
  - Sécurité - confidentialité?
  - Performance ?

# Notions de base

## Formalisme de représentation des données

- 1e génération : fin des années 1960
  - modèle hiérarchique et modèle réseau
  - architecture centralisée
  
- 2e génération: fin des années 1970
  - modèle relationnel
  - architectures client-serveur (Oracle, Sybase, *Informix*)
  
- 3e génération : fin des années 1980
  - extensions objet du modèle relationnel
  - architectures distribuées (Oracle8, DB2 Unisersal database,...)
  
- 4e génération : fin des années 1990
  - données « peu structurées », Internet, Web
  - architectures (des applications) N-tiers



# Notions de base

## Le modèle relationnel, les 12 règles de Codd

1. Toutes les informations sur les données sont représentées au niveau logique et non physique (pas besoin d'informations sur la façon dont sont implémentées physiquement les données).
2. Les données sont accessibles uniquement par la combinaison du nom de la table, et du nom de la colonne (pas de chemin à donner).
3. Une valeur spéciale doit représenter l'absence de valeur (NULL).
4. La description de la base de données doit être accessible comme les données ordinaires (un dictionnaire des données est enregistré dans la base).
5. Un langage doit permettre de définir les données, définir des vues, manipuler les données, définir des contraintes d'intégrité, définir des autorisations, de gérer des transactions.
6. On peut faire des mises à jour par les vues lorsque c'est logiquement possible.

# Notions de base

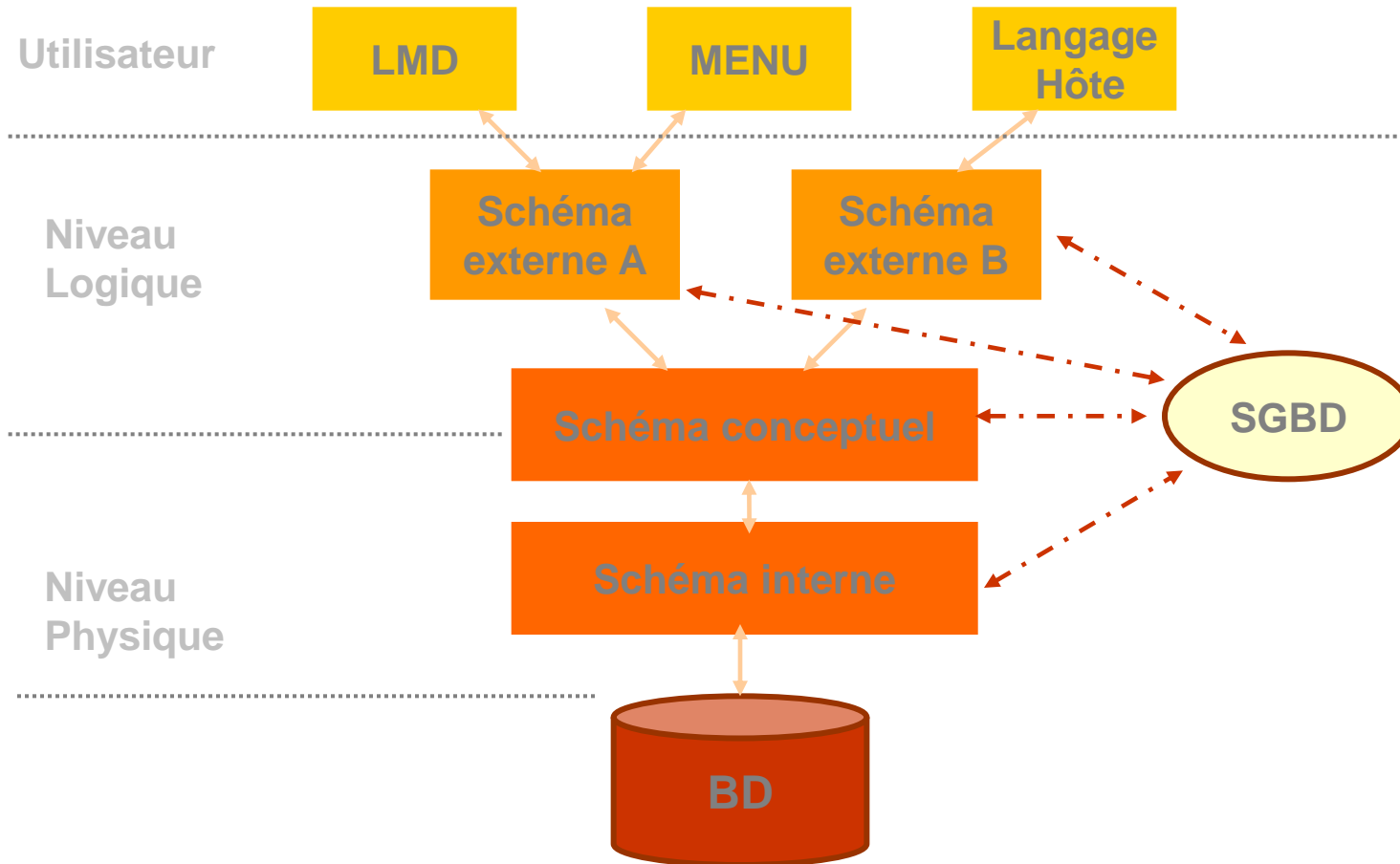
## Les 12 règles de Codd (suite)

7. Le langage doit comporter des ordres effectuant l'insertion, la mise à jour et la suppression des données (un seul ordre pour effectuer chacune de ces fonctions)
8. Indépendance des programmes vis-à-vis de l'implantation physique des données.
9. Indépendance des programmes vis-à-vis de l'implantation logique des données.
10. Les contraintes d'intégrité doivent pouvoir être définies dans le langage relationnel et enregistrées dans le dictionnaire des données.
11. Indépendance vis-à-vis de la répartition des données sur divers sites.
12. On ne peut jamais contourner les contraintes (d'intégrité ou de sécurité) imposées par le langage du SGBD en utilisant un langage de plus bas niveau (par exemple le C).

# Notions de base

## Architecture

### ■ Architecture



# Notions de base

## Architecture (suite)

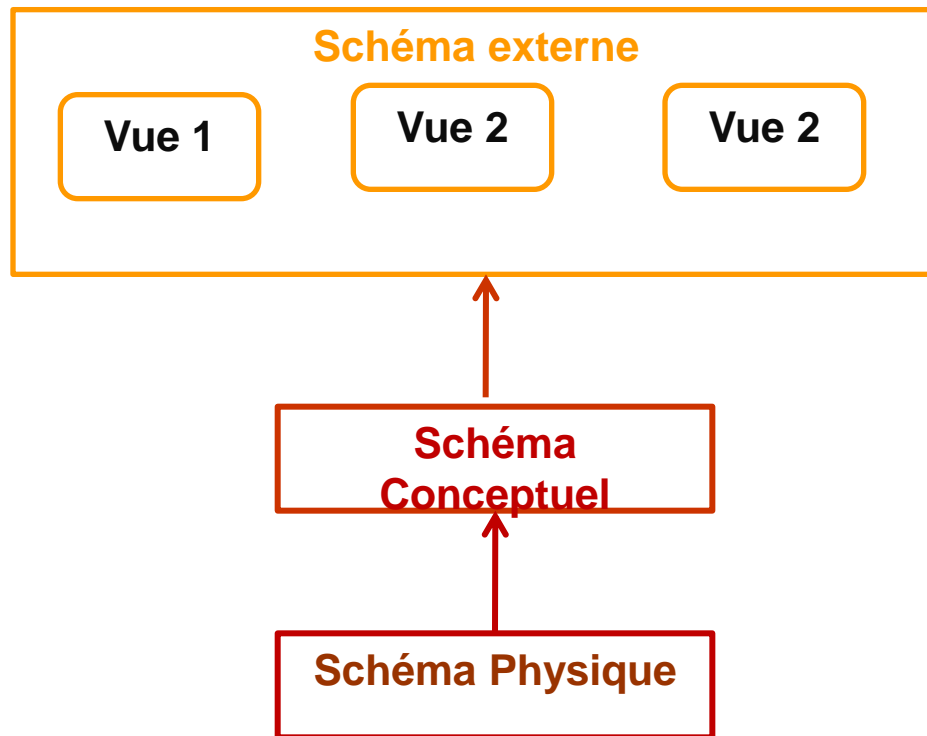
- Niveau interne :
  - dépendant du système d'exploitation
  - répartition sur disques.
  - optimisation physique des accès
  - spécifique au SGBD
  
- Niveau conceptuel :
  - lié au formalisme de représentation
  - constitue un catalogue des données
  - nécessairement complet
  - relativement portable
  
- Niveau externe :
  - vision limitée et/ou abstraite conforme aux besoins
  - fortement liée à l'application
  - extensible selon les besoins

# Notions de base

## Architecture

### ■ Niveau d'abstraction des données

Exemple: Relation employe (nom,prenom,adresse,salaire)



#### Niveau logique externe:

Vue1 :nom,adresse

Vue2: nom salaire

#### Niveau logique central:

nom,prenom,adresse,salaire

#### Niveau physique:

Taille nom: 30 octets

Taille prenom:20 Octets

Taille adresse:40 Octets

Taille salaire:4 octets

# Notions de base

## Schémas d'une base de données

- Le **schéma conceptuel** est central. Il est exprimé en termes de :
  - tables
  - colonnes
  - clés primaires
  - liens (clés étrangères)
  
- SQL fournit un langage pour exprimer ce schéma
  - CREATE TABLE ...
  - PRIMARY KEY ...
  - FOREIGN KEY ... REFERENCES ...
  
- Un SGBD relationnel doit fournir un « catalogue »
  - une base de données est auto-descriptive
  - le catalogue a lui-même un schéma conceptuel
  
- On peut utiliser d'autres « formalismes » : un modèle *entités-associations* est un modèle graphique équivalent, plus « simple » à manipuler/communiquer

# Notions de base

## Schémas d'une base de données (suite)

- Le **schéma physique** est spécifique au SGBD. Il décrit :
  - les structures physiques : hachage, indexation, etc.
  - les paramètres dynamiques : bloc de données, segment d'allocation (extent), libération, etc.
  - les tailles de partitions
  - etc.
  
- SQL fournit un niveau intermédiaire :
  - notions d'index
  - syntaxe ouverte aux extensions
  
- Le **schéma externe** est une projection du schéma conceptuel, à destination de l'utilisateur.
  
- SQL fournit la notion de vue : pseudo-table, éventuellement construite à partir de tables « réelles », consultable et modifiable (parfois)

# Notions de base

## Indépendance des données

- 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 changements de structure au niveau physique.



# Notions de base

## Utilisateurs

- **Concepteur :**
  - Chargé de la construction du schéma conceptuel
  
- **Programmeur :**
  - Doit connaître les structures logiques
  - Construit les vues externes
  
- **Administrateur**
  - Responsable des structures physiques
  - Doit fournir les schémas externes
  - Responsable des optimisations spécifiques au SGBD
  
- **Utilisateur :**
  - Dispose d'une vue abstraite de la base de données
  - Interagit via une application (IHM)
  - Accès limité à son domaine de compétence

# Notions de base

## Langages

- Langage de définition de données (DDL)
  - description des tables et colonnes
  - liens, contraintes d'intégrités référentielles
  - éventuellement déclencheurs
  
- Langage de manipulation de données (DML)
  - insertion, modification suppression
  - Langage de requête (QL)
  - accès complet aux données
  - contrainte d'efficacité
  
- Langage procédural
  - bases de données actives
  
- SQL est le standard adopté (sauf pour la partie procédurale)

# Notions de base

## Systemes

- Les SGBD relationnels sont des systèmes logiciels élaborés, qui fournissent un certain nombre de services:
  
- Un SGBD doit garantir la confidentialité :
  - applications majoritairement multi-utilisateurs
  - utilisateurs (propre au SGBD ou non)
  - droits d'accès, privilèges, rôles, etc.
  - SQL fournit le modèle de base : GRANT, REVOKE
  
- Un SGBD doit garantir la sécurité des données :
  - système de transactions, atomicité des opérations complexes
  - gestion des accès simultanés, de grain fin (ligne)
  - journalisation pour la reprise en cas d'erreur
  - distribution de charge
  - ...

# Notions de base

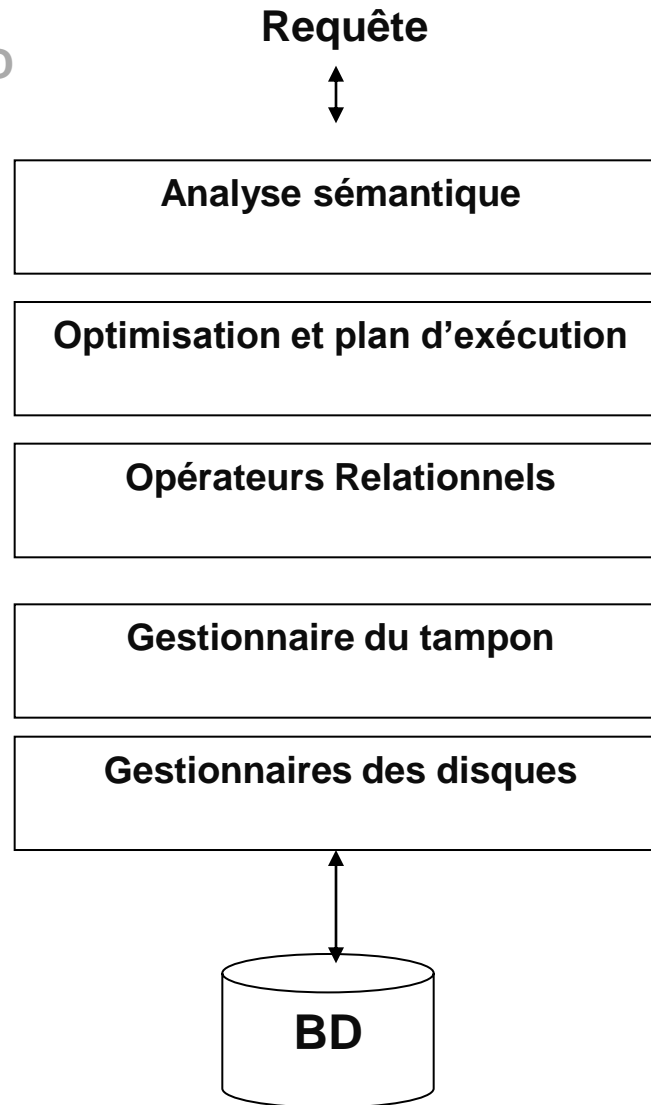
## Systemes (suite)

- Un SGBD doit assurer la performance des opérations :
  - utilisation optimale du matériel
  - optimisation à tout niveau : stockage, accès, etc.
  - autoriser les compromis : performance/sécurité, etc.
  
- L'architecture est majoritairement client-serveur :
  - un serveur, donc administration centralisée
  - des clients divers : applications spécifiques, moteurs de grille d'écran, « frameworks » Web, etc.
  - une interface homogène : SQL (du texte)
  
- Tous ces dispositifs sont spécifiques aux SGBD :
  - impossible d'édicter des règles générales
  - administrateurs souvent experts sur un seul produit

# Notions de base

## Systemes (suite)

Architecture SGBD





# Deuxième Partie

## MODELE ET OPERATIONS