

# Base de données – Chapitre 1

---

## Chapitre 1 : Introduction aux bases de données

**Stéphane Malandain – sbd – 2024**  
**(d'après le cours de Joël Cavat )**

L'avenir est à créer

# Introduction

---

# Objectifs

---

## Thèmes abordés :

- Définition
- Historique
- Utilités
- Manipulation
- Normalisation et contraintes d'intégrités
- Bases de données en tant que systèmes distribuées
- Conception d'une base de données

# Définition

---

## Définitions

1. Une base de données (BD) est une collection d'informations persistantes organisée de manière à pouvoir être facilement manipulée.
2. Un système de gestion de base de donnée (SGBD) est un système qui permet de gérer une base de données
  - BD : logique et sémantique
  - SGBD : physique et outils

## SGBD connus

---

MySQL, PostgreSQL, SQLite, Oracle Database, IBM Db2,  
Microsoft SQL Server, Access Redis, MongoDB, BigTable,  
OrientDB, Cassandra, Elasticsearch, ...

## Organisation des données

---

Pour une activité donnée (*ex. une bibliothèque*), l'information est généralement organisée en terme :

- d'**entités** ( = collections) : *Livres, emprunteurs, catégories, ...*
- de **liens** entre les entités : un emprunteur emprunte un exemplaire d'un livre, un livre appartient à une catégorie, ...
- de **structure** (caractéristique) des liens et entités : *Titre du livre, date de l'emprunt, nom de l'emprunteur, ...*

Elle doit respecter les **contraintes** du domaine d'activité (*ex : un livre ne peut pas être emprunté deux fois durant la même période*)

# Historique

---

- 1970, Edgar Frank Codd, description du modèle relationnel et de la **normalisation** des données. Il propose de séparer le stockage physique de sa représentation conceptuelle. Il se base sur des fondements mathématiques pour décrire la représentation des données et les requêtes (indépendamment d'un langage de programmation impératif )
- 1974, Don Chamberlin, langage SQL pour l'exploitation de bases de données relationnelles
- 1980, essor des bases de données relationnelles
- 1990, naissance des BD analytiques et objet-relationnel (Essor de la POO)
- 1990 à nos jours, dominance des bases de données relationnelles
- 2010, Concept général et émergence du NoSQL, rupture de l'approche relationnelle

# Utilité

---

## Utilité d'une BD

- Persistance et pérennité
- Partage et centralisation
- Sécurité (gestion des droits)
- Efficacité (optimisation des requêtes, du stockage, de l'indexation. . . )
- Source de confiance (cohérence des données, éviter les redondances et respect des règles métiers )



# Manipulation des données

---

Extraire, modifier, transformer rapidement l'information

## Quatre opérations :

- Rechercher
- Ajouter
- Supprimer
- Modifier

Acronyme **CRUD** : Create, Read, Update et Delete

# Manipulation des données

---

## Exemples

- Afficher la liste des étudiants
- Ajouter un enseignant (Pierre, Martin, 756-2348.4096.92, 01/06/1968)
- Supprimer les étudiants ayant terminé leurs études avant 2010
- Afficher tous les étudiants qui ont cours le vendredi entre 13h00 et 18h00
- Augmenter le salaire des enseignants de 10%
- Diminuer le salaire du directeur de 20%
- Afficher tous les professeurs qui ont au moins 6 périodes d'enseignement par semaine
- ...

# Manipulation des données

---

## Langage de requêtes :

Ces opérations sont effectuées à l'aide d'un **langage déclaratif** ou d'une API.

## Définition :

Un langage déclaratif permet de décrire l'objectif sans préciser comment l'obtenir.

# Manipulation des données

---

## Langage SQL

Le langage SQL est un langage de requêtes structuré et déclaratif permettant de manipuler les bases de données relationnelles.

- Standardisé ANSI / ISO
- Mais : de nombreux dialectes ...

### Exemple de syntaxe SQL :

```
1  SELECT lastname
2  FROM Student
3  WHERE age >= 18;
```

# Manipulation des données

---

## API et bibliothèques déclaratives

Certaines bases de données mettent à disposition une API ou une bibliothèque pour manipuler les données.

Exemple de la bibliothèque de MongoDB :

```
1 db.students.find(  
2   { age: { $gte: 18 } },  
3   { lastname: 1 }  
4 );
```

# Manipulation des données

---

## Stream

```
1  students.stream()  
2      .filter( s -> s.age() >= 18 )  
3      .map( s -> s.lastname() );
```

# Manipulation des données

---

## Formelle (cf algèbre relationnelle)

Algèbre relationnel

$$\pi_{\{lastname\}}(\sigma_{age \geq 18}(Students))$$

Relational calculus

$$\{s.lastname \mid s \in Students \wedge s.age \geq 18\}$$

## Contrainte d'intégrité

---



# Contraintes d'intégrité

---

## Définition

Les contraintes d'intégrité décrivent des **règles métiers** à respecter. Elles décrivent des situations (ou comportement) admissibles ou non admissibles pour le domaine d'application (contraintes définies dans un cahier des charges par ex.)

## Exemples

- chaque entité possède un identifiant
- une personne a un nom
- un professeur enseigne au minimum un cours
- un livre est classé dans une catégorie
- l'email d'une personne est facultatif
- un siège sur un vol ne peut pas être attribué à plusieurs clients
- le salaire d'un professeur ne peut jamais diminuer

# Normalisation

---

# Normalisation

## Définition

Une base de données **normalisée** est modélisée de manière à éviter des redondances, des incohérences et des anomalies

Exemple d'informations dénormalisées :

nom	prénom	école	détails
Malandain	stephane	hepia	Haute école du paysage, ...
Rekik	yassin	hepia	Haute école du paysage, ...
Martin	pierre	HEIG-VD	Haute école d'ingénierie

## Normalisation

Exemple d'informations normalisées :

idEtudiant	prénom	nom	idEcole
4	stephane	malandain	1
12	yassin	rekik	1
21	pierre	martin	4

idEcole	école	détails
1	hepia	Haute école du paysage, ...
2	HEM	Haute école de musique
3	HEIG-VD	Haute école d'ingénierie

# Modélisation d'une base de données

---

# Modélisation d'une base de données

---

## Modélisation d'une base de données

- centrée sur les données (*data-centric*)
- identifier les données à persister, les entités et les relations entre elles
- on s'intéresse à la **structure**
- ex: une consultation à une date, un patient est suivi par des médecins

# Modélisation d'une base de données

---

Par opposition la modélisation d'une logique métier :

- centrée sur le domaine, la logique métier (*domain-centric*)
- identifier les concepts et les fonctionnalités
- on s'intéresse au **comportement**
- La persistance est un détail d'implémentation dont on se préoccupe le plus tard possible
- ex : une consultation doit être validée par un chef de clinique, ne doit pas être en conflit avec d'autres consultations. . .

# Modélisation d'une base de données

---

Une BD et une application doivent évoluer indépendamment.

- Séparer leur modélisation respective.
- Ne pas être dépendant l'un de l'autre.
- Evitez de générer un modèle à partir de l'autre (Danger des ORM)<sup>1</sup>.
- une BD peut servir plusieurs applications, plusieurs vues.

---

<sup>1</sup> [https://en.wikipedia.org/wiki/Object-relational\\_impedance\\_mismatch](https://en.wikipedia.org/wiki/Object-relational_impedance_mismatch)