

Base de données

Chapitre 1 : Introduction aux bases de données

Joel Cavat

2022

Introduction

Objectifs de ce chapitre

Thèmes abordés dans ce chapitre:

- Définitions
- Contexte historique
- Utilités
- Manipulation
- Normalisation et contraintes d'intégrité
- Bases de données en tant que systèmes distribuées
- Conception d'une base de données

Définition

Une base de données est une collection d'informations persistantes organisée de manière à pouvoir être facilement manipulée.

Définition

Un système de gestion de base de données (SGBD) est un système qui permet de gérer une base de données.

- BD: logique & sémantique
- SGBD: physiques & outils
 - moteurs de stockage, langage de requêtes...

SGBD connues: **MySQL**, PostgreSQL, **SQLite**, Oracle Database, Microsoft SQL Server, Access, Redis, **MongoDB**, BigTable, Neo4j, OrientDB, Cassandra, Elasticsearch, ...

Organisation des données

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

- d'**entités** (~ collections)
 - Livres, exemplaires, catégories, emprunteurs. . .
- 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 d'emprunt, nom de l'emprunteur. . .

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

Contexte historique

- 1970, Edgar Frank Codd, description du **modèle relationnel** et de la **normalisation** des données
 - proposer de séparer le stockage physique de sa représentation conceptuelle
 - se baser sur des fondement mathématique pour
 - décrire la représentation des données
 - 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 base de données relationnelles
- 2010, Concept général et émergence du **NoSQL**, rupture de l'approche relationnelle

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
 - respect des règles métiers

Extraire, modifier, transformer rapidement l'information.

Quatre opérations

- Rechercher
- Ajouter
- Supprimer
- Modifier

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

Exemples

- Afficher la liste des étudiants
- Ajouter un enseignant (Jean-Paul, Dubosson, 756-2334.3596.92, 1-juin-1966)
- 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 2%
- Afficher tous les professeurs du département qui ont au moins 6 périodes d'enseignement par semaine
- ...

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

Langage SQL

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

- Standardisé ANSI/ISO
- Mais : nombreux dialectes

Exemple de syntaxe SQL:

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

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 );
```

Stream

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

Formelle (cf. cours d'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

Définition

Les contraintes d'intégrité décrivent des **règles métiers** à respecter. Elles décrivent des situations (ou comportements) 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

Définition (non formelle)

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
Polla	Guillaume	Hepia	Haute école du Paysage, ...
Racloz	Pascal	Hepia	Haute école du Paysage, ...
Doutaz	Véronique	HEIG-VD	Haute école d'ingénierie, ...

Normalisation

Exemple d'informations normalisées:

idEtudiant	nom	prénom	idEcole
18	Polla	Guillaume	1
21	Racloz	Pascal	1
27	Doutaz	Véronique	3

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

- 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 a une date, un patient est suivi par des médecins

Par opposition à 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. . .

Une BD et une application doivent évoluer indépendamment

- séparer leur modélisation respective
- l'un ne doit pas être dépendant de l'autre
- évitez de générer un modèle à partir de l'autre (danger des ORM)¹
- un BD peut servir plusieurs applications, plusieurs vues

¹https://en.wikipedia.org/wiki/Object-relational_impedance_mismatch