

Initiation aux bases de données (SGBD)

Walter RUDAMETKIN

Bureau F011

Walter.Rudametkin@polytech-lille.fr
<https://rudametw.github.io/teaching/>

Moi

Je suis étranger...

J'ai un accent...

Je me **trompe beaucoup** en français

(et en info, et en math, et en ...)

- **N'hésitez pas à me corriger**
ou à me demander de répéter

Je commence à enseigner

- J'accepte des critiques (constructifs mais pas que)
et surtout des **recommandations**
- N'hésitez pas à poser des questions

Clause de non-responsabilité

Les transparents de ce cours sont très très très largement inspirés (i.e., copiés) de ceux d'**Anne Etien** (MdC Polytech Lille), qui à son tour s'est inspiré d'Olivier Caron et de Rébecca Deneckere, qui à leur tour ont sûrement..

Intro. aux bases de données

Les **objectifs** des bases de données

Introduction à l'**analyse**

Modélisation de systèmes d'information

Requêtes SQL

Programme

10 heures de cours

10 heures de TP (5 séances)

1 TP noté

Contrôle des connaissances :

- ~~contrôle continu (compte rendu des TP)~~
- **1 contrôle TP individuel (2 heures)**
- ~~examen écrit (2 heures)~~

Plateformes :

- SGBD : **Postgres** sous unix (<http://www.postgresql.org>)
- LibreOffice Calc

Bibliographie

Introduction aux bases de données, Serge Miranda et José-Maria Busta

Les bases de données relationnelles, Serge Miranda et José-Maria Busta

Bases de données, George Gardarin

Des Bases de données à l'Internet, Philippe Mathieu
(2 exemplaires au centre de documentation
Polytech Lille)

et... le support de cours

Historique (1/4)

Au commencement

- Version papier
- Première structuration : le fichier
- plusieurs utilisateurs mais vue unique des données.

Structures de données en mémoire centrale

- Stockage temporaire
- « petits » volumes de données
- Langages de programmation
- Contexte mono-utilisateur

Fichier informatisé

- Stockage persistant sur disque
- « Gros » volume de données
- Langages de programmation et SGF (système de gestion de fichiers)
- Contexte mono-utilisateur (en général)

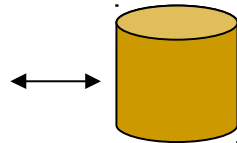
Historique (2/4)

Exemple : Gestion des élèves dans une école d'ingénieurs

Bibliothèque



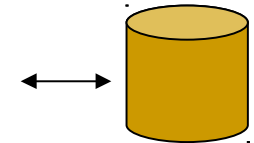
PC DOS



Fichiers
Elèves et
Livres

Direction des études

PC
OS/2



Fichiers
Elèves et
UE

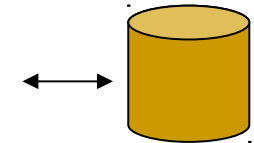
Approche Fichier :

- Information conservée dans les différents services.
- Est-elle sous la **forme souhaitée** ?
- Les informations importantes sont **dupliquées**. Sont-elles identiques ?

Service social



PC DOS



Fichiers
Elèves et
Chambres

Historique (3/4)

Fichier, support informatique (1960) :

Problème de l'approche fichier

- Fichiers de natures différentes, manipulés par des programmes différents, dans des langages différents avec des formalismes différents.
- Difficulté à saisir les **liens** entre les données
- Pas de **partage** de données entre les utilisateurs
- **Centralisation** physique, mais pas logique.
- Dépendance entre les données et les traitements
- Absence de gestion de la sécurité des données
- Pour toute nouvelle application, **où sont** les informations utiles ?

Historique (4/4)

Evolution : SGBD (Systèmes de Gestion de bases de données)

Avantages :

- Description **unique** et **globale** des données
- **Langage commun** de manipulation.
- **Centralisation** (pas de duplication, donc pas d'incohérence possible).
- Indépendance entre données et traitements
- Séparation des descriptions logique et physique des données
- Contrôle sémantique des données
- **Sécurité** (Qui a le droit de lire ou d'écrire, contrôle d'intégrité).
- **Partage** des données

Différences Tableur/BD

Différences sur...	Tableur	Base de données
Utilisation principale	Calculs	Gestion et traitement des données
Structuration des données	Aucune	Structuration et cohérence forte
Contrôles d'intégrité des données	Aucune	Vérification stricte des valeurs possibles de chaque donnée
Accès aux données	Mono-utilisateur	Multi-utilisateurs
Confidentialité des données	Aucun contrôle	Vérification des droits d'accès de chaque utilisateur
Taille des données	- Une table - Quelques dizaines de lignes	- Plusieurs tables - Plusieurs milliers de lignes par table
Traitement sur les données	Quantitatifs	Qualitatifs et quantitatifs
Interrogations des données	Réalisée par des procédures spécifiques	Langage "universel" : SQL

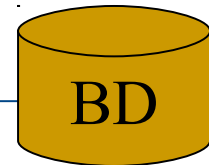
Le cycle de vie d'une base de données

Définir la structure (ou schéma)

- Langage de description

Collecter les données

Les **stocker** sur disque



Interroger

- Langage d'interrogation

Manipuler

- Langage de manipulation / programmation

Les objectifs des b. de données

- **Indépendance** physique des données
- Indépendance logique des données
- Manipulation des données par des **non-informaticiens**
- **Efficacité** des accès aux données
- **Administration** centralisée des données
- **Non redondance** des données
- **Cohérence** des données
- **Partageabilité** des données
- Sécurité des données

Objectif : Indépendance Physique

Représentation du monde réel sous forme indépendante de la structure physique

Contre-exemple : telle valeur est située à telle adresse en mémoire.

En informatique, les utilisateurs de SGBD n'ont pas à se soucier de :

- Structures de stockage, fichiers, listes, tableaux, disque dur, système de sauvegarde

Objectif : Indépendance Logique

Idée : Chaque utilisateur doit pouvoir ne connaître qu'une partie de la sémantique des données et ne voir qu'une partie des données

Avantages :

- On peut modifier (virtuellement) la structure
- On ne dispose que des informations pertinentes
- Sécurité (on ne peut pas accéder à des données qu'on ne connaît pas.
- Plusieurs représentations sont possibles destinées à un utilisateur ou un groupe d'utilisateurs

Objectif : Manipulation des données par des non-informaticiens

Les utilisateurs peuvent :

- consulter les données
- mettre à jour

Il faut donc définir un langage le plus simple et le plus intuitif possible :

- QBE (Query By Example) : simple pour des requêtes simples (graphique)
- SQL (Structured Query Language) : standard de fait, très puissant

Objectif : Efficacité des accès aux données

Réservé aux utilisateurs **informaticiens** :

- Offrir un langage de programmation et une bibliothèque (API) permettant la gestion de bases de données.
- Bibliothèques propriétaires : PL-SQL (Oracle), VBA (Ms Access), libPostgres (C) , php et postgres, ...
- Standards (de fait) : ODBC (Microsoft), JDBC (Sun JavaSoft)

Algorithmes d'optimisation de requêtes

Support réseau

Objectif : Administration cohérente des données

L'administration est souvent **centralisée**.
Le rôle de l'administrateur de bases de données :

- Contrôle efficace des données, sauvegarde
- Résoudre les conflits
- Optimiser les accès aux données
- Prise en compte de la charge (nombre d'utilisateurs connectés, charge disque,...)

Objectif : Non redondance des données

Éviter la perte de place en mémoire
Éviter les mises à jour multiples

Exemple :

Personne (nom, prénom, age, qualité)

Adresse (nom, prénom, rue , ville, code postal)

Personne (**numID**, nom, prénom, age, qualité)

Adresse (**numID**, rue , ville, code postal)

À prendre en compte lors de la conception de la base

Objectif : Cohérence des données

Notion d'intégrité

- de domaine, référentielle,...

Fournir des outils de contrôle automatique

Objectif : Partageabilité des données

Permettre aux applications de partager les données de la base dans le temps mais aussi simultanément.

L'utilisateur n'a pas à se soucier des autres accès !

→ **systemes de transactions**

Objectif : Sécurité des données

Fournir des mécanismes pour définir des droits d'accès.

Les données doivent être protégées contre les accès non autorisés ou mal intentionnés.

Les droits d'accès peuvent également dépendre de la valeur des données :

- Exemple : un employé pourra connaître les salaires des personnes qu'il dirige mais pas des autres employés de l'entreprise.

La qualité d'un SGBD, c'est aussi

- Gérer de grandes quantités de données (data-warehouse, data-mining) (GIS5)
- Gérer des données réparties sur plusieurs serveurs
- Accepter un grand nombre de connexions
- Exécuter des requêtes avec un temps de réponse le plus court possible
- Proposer des mécanismes de reprise après panne (dans un état cohérent), système de réplication,...
- Avoir une bonne ergonomie
- Être à un prix abordable

Les SGBD relationnels

SGBDR représentent **80%** du marché

Le SGBD parfait n'existe pas !

Les Pros :

- Oracle, Informix, Sybase, DB2,...

Les semi-pros :

- Postgres, MS-SQL server, ...

et les autres :

- Ms Access, MySQL, MariaDB