Introduction aux SGBDs

Contenu du chapitre:

2022-2023

- Enjeux des BDs
- Histoire parcellaire des BD
- Principes fondateurs des SGBD

- ✎ Comprendre ce qu'est une BD et un SGBD
- ✎ Connaître les principes qui ont guidé le développement des SGBD.

Table des matières

2022-2023

Introduction aux SGBDs
- Enjeux des BDs
  - Usagers et acteurs des BD
  - Avantages et inconvénients des SGBD (en général)
- Histoire parcellaire des BD
- Principes fondateurs des SGBD

Pourquoi des Bases de données (BD)?

Applications de l'informatique (hier et ajourd'hui):

- **gérer de grandes quantités d'information**
- contrôler des machines
- calculs (physique, notamment balistique), simulation
- communiquer

Bases de donnée (BD)

Ensemble de données structurées en rapport avec un sujet particulier. Ces données informatiques sont reliées entre elles et accessibles à une communauté d'utilisateurs à travers un logiciel dédié: le SGBD.

Système de gestion de bases de donnée (SGBD) (DBMS)

Logiciel permettant de gérer et exploiter une base de donnée. Il s'agit de définir, interroger, et mettre à jour les données.

Applications traditionnelles

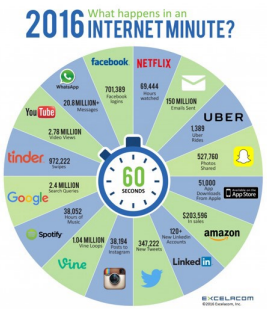
Où rencontre-t-on des BDs?

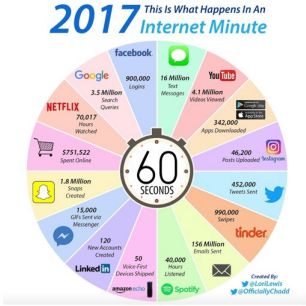
Gestion d'entreprise:
- comptabilité
- stock
- gestion du personnel, clients, fournisseurs
- commandes, réservations

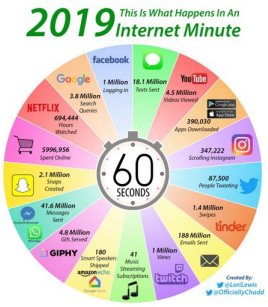
Données scientifiques (médecine, biologie...).
Données de capteurs
Données des administrations (économie, démographie)
Back-end pour des applications de la vie de tous les jours:
sites webs: CMS *joomla, wordpress*
boîtes mails/contacts/certificates (thunderbird,firefox)
Site web (les CMS *joomla, wordpress*)
Archivage des courriels/contacts, certificats (thunderbird,firefox)
CIA

Exemples de la vie ordinaire:
- Commande de billets de trains, avions, produits, réservations

Applications du web actuel







Le marché des SGBD

Quels sont les grands acteurs?

ORACLE

IBM

Microsoft SQL Server

open source:

MySQL

MariaDB

mongoDB.

PostgreSQL

SQLite

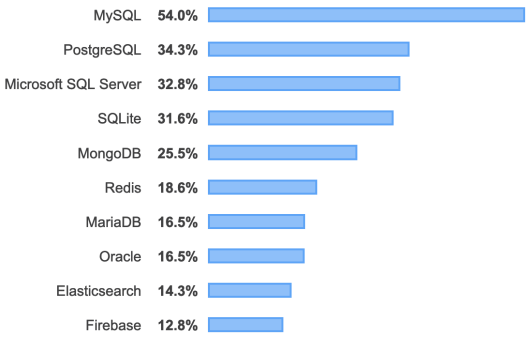
snowflake

... des centaines de systèmes

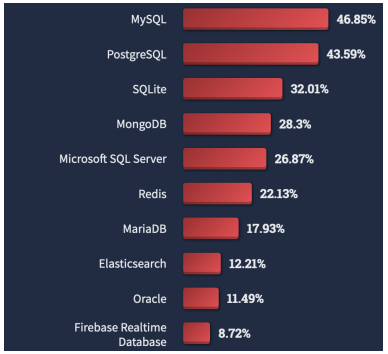
Le marché des SGBD (2)

SGBDs les plus populaires

2019



2022



[https://insights.stackoverflow.com/survey/2019#most-popular-technologies-database]

[https://survey.stackoverflow.co/2022/#most-popular-technologies-database]

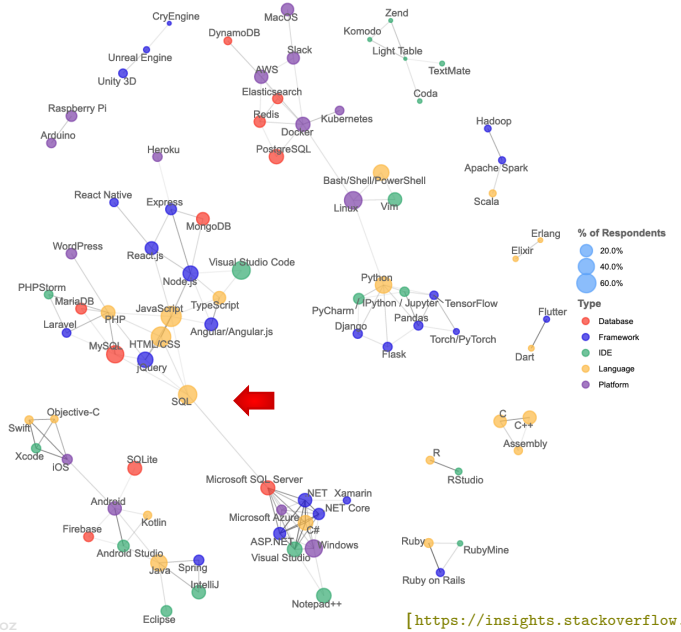
Les métiers des SGBD

Administrateur BD
installer, configurer administrer (droits d'accès), optimiser

Concepteur BD
concevoir l'organisation/schéma de la BD
à travers des applications ou plus directement

. . . de nos jours (ère du "Big Data"), surtout utile au sein d'un ensemble de compétences autours des données: les métiers liés à l'analyse de donnée, combinent souvent des aspects BD et l'apprentissage.
Ces métiers évoluent. Certaines tâches (ex: optimisation) sont en voie d'être automatisées. Les déploiements se font de plus en plus sur le Cloud, en utilisant l'infrastructure et les logiciels des géants du web.

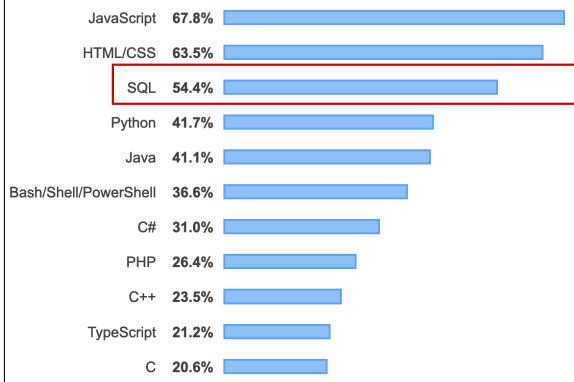
Interactions entre SQL et autres technologies.



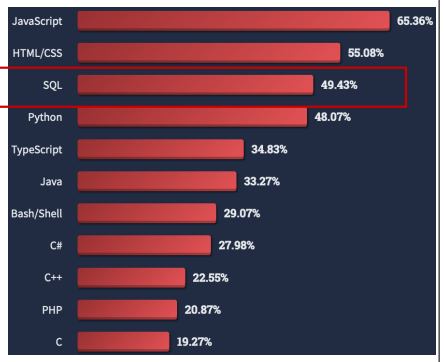
[https://insights.stackoverflow.com/survey/2019]

Popularité de SQL (développeurs sur stackoverflow)

2019



2022



[https://insights.stackoverflow.com/survey/2019#most-popular-technologies]

[https://survey.stackoverflow.co/2022/#most-popular-technologies-language]

Pourquoi un SGBD?

Supposons que vous voulez gérer

un site web de vente en ligne (chocolats).

Clients	**Achats**	**Produit**
adresse	nom_client	description
nom	nom_produit	prix
CB	quantité	. . .
	date	
	adresse_livraison	

Quel que soit l'ordinateur, les données seront gérées par un *système de fichier*.

produits.txt

achats.txt

et les tâches peuvent êtres implémentées par un programme en Java, C, Python. . .

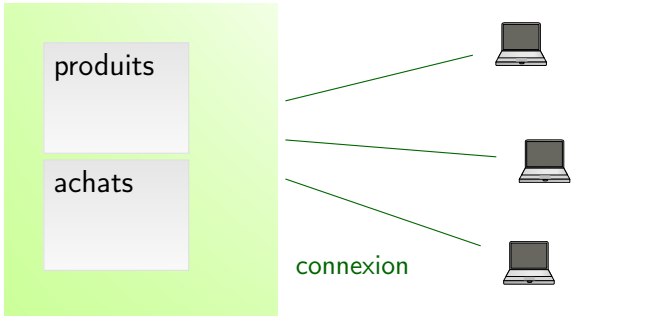
Système de fichier+programmes?

Problèmes à prendre en compte:

- pannes du systèmes
- commandes simultanées
- performance (50GB dans un fichier?)
- maintenance: durée de vie donnée≈10ans, programmes évoluent
- documenter la structure des données
- éviter les redondances dans les données (et programmes), assurer leur cohérence.
- contrôle d'accès
- garantir l'intégrité

L'approche "BD" traditionnelle

Architecture client-serveur



Le serveur est juste un (gros) programme C que vous n'avez pas eu à écrire. Gère les données, évalue requêtes et mises à jour envoyées depuis les clients.

Fonctionnalités des SGBD (en bref)

1. Stockage *persistant* des données dans les fichiers (implémente un modèle de donnée): (gros volumes) *disques, buffer, dictionnaires...*
2. Performance des requêtes *index, optimisation de requête...*
3. Multi-utilisateur (accès concurrents) ***transactions, verrous, MVCC...***
4. Fiabilité du SGBD (reprise sur panne...) *logs, backups...*
5. Sécurité (contrôle d'accès) *droits d'accès...*
6. Facilité d'utilisation (**requêtes**, maintenance, évolution)
7. Fiabilité des données (cohérence) ***normalisation, contraintes d'intégrité (FK)***

Fonctionnalités des SGBD (exemples)

Accès concurrents:

- on ne vend pas la même place de train à 2 personnes
- le système ne plante pas lorsque 1000 utilisateurs essaient d'acheter un billet de train

Sécurité:

- l'étudiant Dupond peut voir ses notes aux examens, mais pas celle de Durand.
- le responsable de filière peut modifier les notes, mais pas les étudiants

Quand est-ce qu'un SGBD relationnel *n'est pas* adapté

NoSQL (ou simples fichiers) plus appropriés quand:

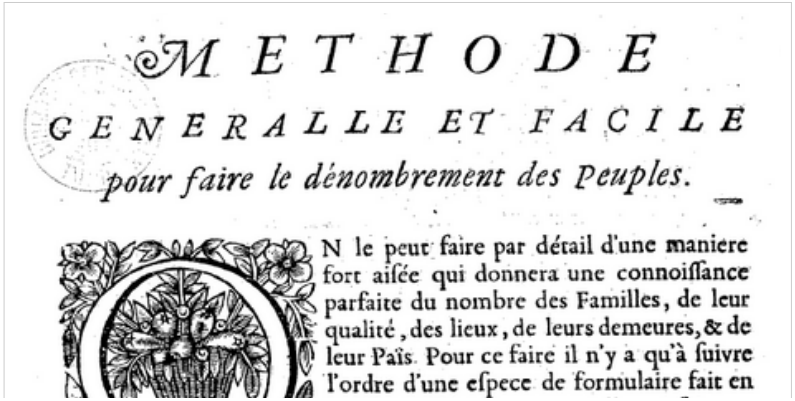
- trop de données: $> 1TB$
- on exige des délais (latence) très courts
- objets IOT avec mémoire très limitée
- les données et requêtes se représentent mal en relationnel (graphes, séries temporelles).
- la pile logicielles s'intègre mieux avec d'autres technologies

Table des matières

2022-2023

Introduction aux SGBDs
- Enjeux des BDs
- Histoire parcellaire des BD
- Principes fondateurs des SGBD

Les recensements: collecte et aggrégation de données



METHODE
GENERALLE ET FACILE
pour faire le dénombrement des Peuples.

N le peut faire par détail d'une maniere fort aiſée qui donnera une connoiſſance parfaite du nombre des Familles, de leur qualité, des lieux, de leurs demeures, & de leur Païs. Pour ce faire il n'y a qu'à ſuivre l'ordre d'une eſpece de formulaire fait en

Chaſtellenie de .....

Parroiſſe de S. N. *Ruë Fr.*

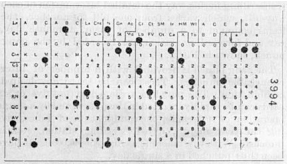
	Maiſons	Noms & qualitez.	Hommes	Femmes	Grands Garçons	Grandes Filles	Petits Garçons	Petites Filles	Vallets	Servantes	Nomb. des Familles
	1	Mr le Conte de ..... Seigneur du lieu, y reſidant actuellement.	1	1	2	0	0	0	6	2	12

Vauban, 1686

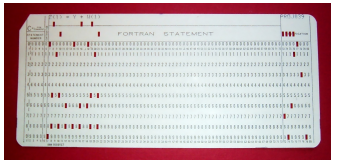
Automatiser le recensement: cartes perforées (Punch cards)

H.Hollerith: mécanographie (machine à carte perforée) pour le recensement US de 1890.

Fonde une société à l'origine d'IBM.



Cartes perforées: principal support de stockage/traitement de données de 1900 à 1950.



80 column card (12lines)

Un centre d'archives en 59: 2000 cartes par carton: total de l'entrepôt≅ 4GB.

Hardware: Stockage des données

Le disque dur

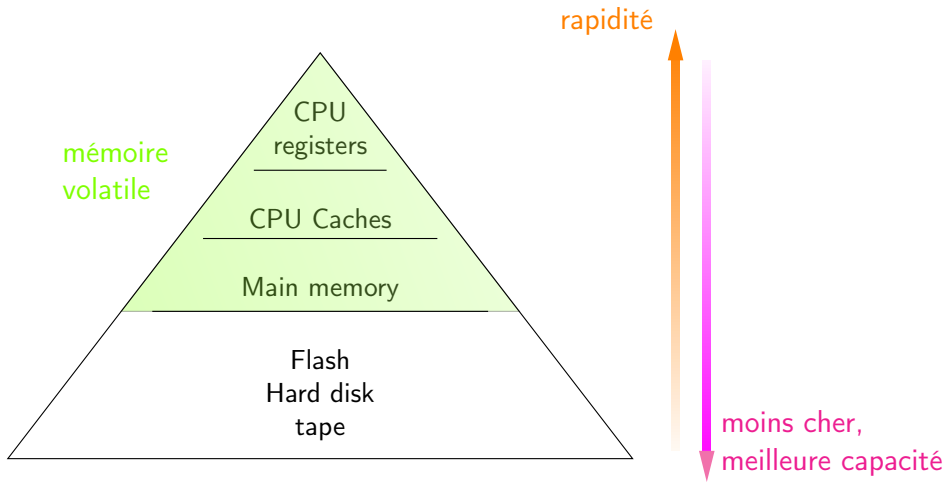
Inventé en 1956 (IBM).

Plusieurs disques solidaires tournant rapidement (milliers tr/min). Couche ferromagnétique. Multiples pistes concentriques par disque. 1 tête de lecture par disque, sur coussin d'air.

≈ 100€
qques TB
100gr.

Principal support de stockage (secondary storage) depuis 1960.
Le support pour lequel les SGBD ont été conçus.

Hardware: Hiérarchie de mémoires



Sur les premiers ordinateurs les fréquences CPU ≈ memory bus et accès mémoire. Mais CPU devenu beaucoup plus rapide que les accès mémoire.

Mémoire vive sur un PC: 4-10GB.

Historique des SGBD

- 1960: Début des SGBD, app. de gestions suivent modèles:
  - ⋆ *hiérarchiques* (IMS, par IBM)
  - ⋆ *en réseau* (IDS pour General Electric)

  Principe: haute performance.

- 1970: modèle relationnel par Codd à IBM San Jose (Almaden). Modèle simple. Principe: gain de productivité. Permet éliminer redondances et inconsistances. Fondements théoriques (logique) attirent universitaires et industriels:

  SystemR (IBM,74), Ingres (UC Berkeley,75), Oracle (79)

Historique des SGBD (2)

- 1980-90: modèle relationnel devient paradigme dominant pour les SGBD. *SQL s'enrichit, nouveaux types de données (images, texte), gros volumes de données, requêtes analytiques (Entrepôts de données). Installation simplifiée (intégration PC). Tentatives: BD objets.*

- 2000: GAFA déploient leurs datacenters, économie du Web. *Tentatives: BD XML, multimédia. Recherches par mot-clef, Syst. de recommandation. BD fédérées: intégrer des informations de plusieurs sources hétérogènes.*

- 2010: Le web comme source de données. Stockage Cloud. Nouveaux hardware (FPGA, GPGPU, SSD). Stockage/traitement des données revisité:
  - in-memory DB (colonnes...)
  - **NoSQL (Not only SQL):** architecture massivement parallèles (Map/Reduce, systèmes clef/valeur...).
  - **AI** pour le "big data"

SGBDs: prix Turing

Bachman 1973

For his outstanding contributions to database technology

Designed Integrated Data Store(IDS) by 1963. Ideas: store data in single place. Data Manipulation Language. Highly efficient.

Codd 1973

For his fundamental and continuing contributions to the theory and practice of database management systems.

Early parallel programming. relational model: data modeling, normalization, relational algebra, connection to logics. OLAP.

Gray 1998

For seminal contributions to database and transaction processing research and technical leadership in system implementation.

Foundations of transaction processing. GIS. Fault-tolerant DBMS.

Stonebraker 2014

For fundamental contributions to the concepts and practices underlying modern database systems.

INGRES. Postgres. Vertica. VoltDB. SciDB.

Table des matières

2022-2023

Introduction aux SGBDs
- Enjeux des BDs
- Histoire parcellaire des BD
- Principes fondateurs des SGBD
  - Le modèle relationnel: premier aperçu
  - Schéma et Instance de base de donnée
  - Architecture à 3 niveaux ANSI SPARC

Représentation des données: le modèle relationnel

Le modèle relationnel

représente les données sous formes de tables.

- modèle de BD largement dominant
- formalisé par E.F.Codd (@IBM, Turing'81)
- *données organisées en tables: 1 table= 1 relation*

Ex:

schéma de la table

NSS	NOM	PRENOM	ADRESSE
1111	GROZ	Benoit	75016
2222	COHEN	Sarah	75008
3333	BIDOIT	Nicole	75014

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

Communications of the ACM, 13(6), 1970

Représentation des données: le modèle relationnel

SQL

langage permettant d'interroger et modifier les données dans le modèle relationnel

- défini en 74, standardisé par la suite
- enrichi par version successives (dernière version: SQL 2016)
- langage par excellence pour interroger une BD relationnelle.
- *déclaratif* : décrit le résultat souhaité, pas les opérations à effectuer

```
SELECT nom,prenom
FROM client
WHERE nom = 'GROZ'
  OR nom = 'BIDOIT';
```

Le résultat est une relation:

NOM	PRENOM
GROZ	Benoit
BIDOIT	Nicole

Schéma vs Instance

Schéma

décrit l'organisation des données. Dans le modèle relationnel: le schéma d'une table donne le nom et type (domaine des valeurs possibles) de chaque colonne. (plus éventuellement des contraintes supplémentaires sur les valeurs possibles)

Ex: Clients(id:int, nom:string, prenom:string, adresse:string)

- L'utilisateur utilise en général le schéma pour formuler ses requêtes.
- Cette description des données est elle-même une (méta)donnée stockée dans la base par le SGBD.

Instance

les données dans la base (qui sont organisées comme indiqué par le schéma).

On peut voir une instance comme l'état courant (ou un état "possible") de la relation, alors que le schéma fait abstraction du contenu détaillé.

Ex: une instance de la base client ci-dessus est présentée au transparent 27.

Schéma vs instance: représentation graphique

Une instance d'une base:

Cours

cours	horaire	bât	salle	nb_étudiants
Prog Web	...	640	C101	24
Algo	...	640	C101	38

Salles

bât	salle	type
640	C101	TP
620	A101	Amphi
120	D101	Amphi

Schéma de la base:

Cours	
cours	TEXT
horaire	TEXT
bât	INT (FK)
salle	TEXT (FK)
nb_étudiants	INT

→

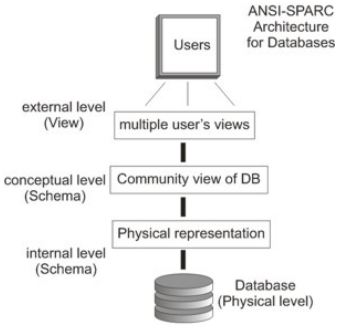
Salles	
bât	INT
salle	TEXT
type	TEXT

3 Niveaux d'abstraction d'un SGBD

externe: gère l'accès des applications aux données; définit la *vision* de la base qu'a chaque groupe d'utilisateurs

logique: définit la structure des données: le schéma

physique: définit l'organisation et le stockage des données sur le disque: fichiers et indexs utilisés



modèle de conception ANSI-SPARC, proposé en 1975, très largement adopté

objectif: pouvoir modifier un niveau indépendemment des autres

Cycle de vie d'une BD.

- Modélisation des besoins.
- Conception des schémas, initialisation
- Manipulation (interrogation, mises à jours)
- Maintenance (optimisations, corrections, évolution)
  le réglage (tuning) peut-être difficile!