

Bases de données

Comment gère-t-on les données d'un système d'information?

cedric.buron@isen-ouest.yncrea.fr

Support de Benoît Lardeux, d'après les notes de cours de O. Rossini et C. Vignaud

Organisation du cours (30 hrs)

- Pourquoi a-t 'on besoin d'une base de données?
- Quels types de bases de données existent?
- Quels utilisations faire de ces outils?
- **Modélisation**
- **Manipulation de bases de données relationnelles**
 - Création, modification, administration, interrogations,
 - Mise en pratique utilisant PostgreSQL

— Cours
— TD
— TP

Evaluations

- Devoir surveillé 1 (modélisation)
- Devoir surveillé 2 (modélisation + SQL)
- TP noté

Outils

- Modélisation sur SE Windows: **Java v8** + **JMerise** (version Isen)
- Système de gestion de base de données relationnelle (WSL): **PostgreSQL**



Pourquoi?

Que fait-on des données?

- Différentes situations où on est confronté aux données:
 - Durant l'exécution d'un programme



- Lors du stockage des données pour utilisation plus tard / ailleurs



Exécution

- Principal **lieu de stockage** des données dans un pc
 - Mémoire RAM / ROM
 - Registres CPU
- **Actions** sur les données

Action	Exemple	Qui peut le faire ?
Récupération de données simples	Constantes au sein de l'exécutable	L'exécutable / les bibliothèques
Des calculs	Fonctions / procédures	
Récupération de données conséquences	Récupérer les mesures des capteurs de température	Une base de données
Exécuter des algorithmes d'agrégation	Déterminer le client ayant acheté le plus de produit dans le mois	

Stockage

- Principal **stockage** des données
 - Disques durs, stockage flash
- **Actions** sur les données

Action	Exemple	Qui peut le faire ?
Persistence des données	Assurer la sauvegarde d'une configuration	Un fichier
Mise à disposition partagée	Partager le catalogue de produits avec les collègues	Une base de données
Mise en correspondance des différentes données	Retrouver les données de tous les capteurs d'un même bâtiment	
Sélectionner une politique de stockage efficace	Stocker un booléen et 200 000 booléens n'est pas identique	

Définitions

- **Base de données (data base)**
Collection partagée de données en relation logique et une description de ces données, conçues pour satisfaire les besoins d'information d'une organisation
- **Système de gestion de base de données (data base management system)**
Système logiciel qui permet de définir, créer, mettre à jour une base de données et d'en contrôler l'accès

Propriétés ACID

- Ensemble de propriétés qui garantissent qu'une transaction informatique (opération sur les données) est exécutée de façon fiable
- **Atomicité**
Transaction complète, finalisée, sinon rien
- **Cohérence**
Chaque transaction amènera le système d'un état valide à un autre état valide
- **Isolation**
Chaque transaction s'exécute comme si elle était seule sur le système sans dépendance avec les autres
- **Durabilité**
Une transaction étant terminée, le changement est bien enregistré

Inconvénient des systèmes de fichiers

- **Avant:** stockage avec des fichiers plats (ex. fichiers texte) ou fichiers type Excel
- Isolement des données
- Dédoublage des données (redondance)
- Dépendance des programmes aux données
- Incompatibilité des formats de fichier
- Interrogation directe difficile
- Pas de prise en compte de la sécurité et de l'intégrité des données

Utilité d'une base de données

Du point de vue de l'administrateur du système

- Gérer la persistance des données à long terme
 - Responsable des choix des politiques de stockage des données
 - Une table? Plusieurs?
- Faciliter la maintenance d'un ensemble de données important
 - Outils de maintenance, supervision, backup, réplication, etc
- Abstraction sur le stockage des données

Utilité d'une base de données

Du point de vue du développeur d'applications

- Permettre l'accès aux données
 - API et/ou langage de récupération des données
 - Gestion des accès aux données
 - Permet (éventuellement) d'ajouter des procédures au sein du système de gestion des données
- Permettre la modification des données
 - Définition de la structure des données
 - Inclus une validation des données
 - Ajout / modification / suppression des données
- Abstraction sur l'accès aux données

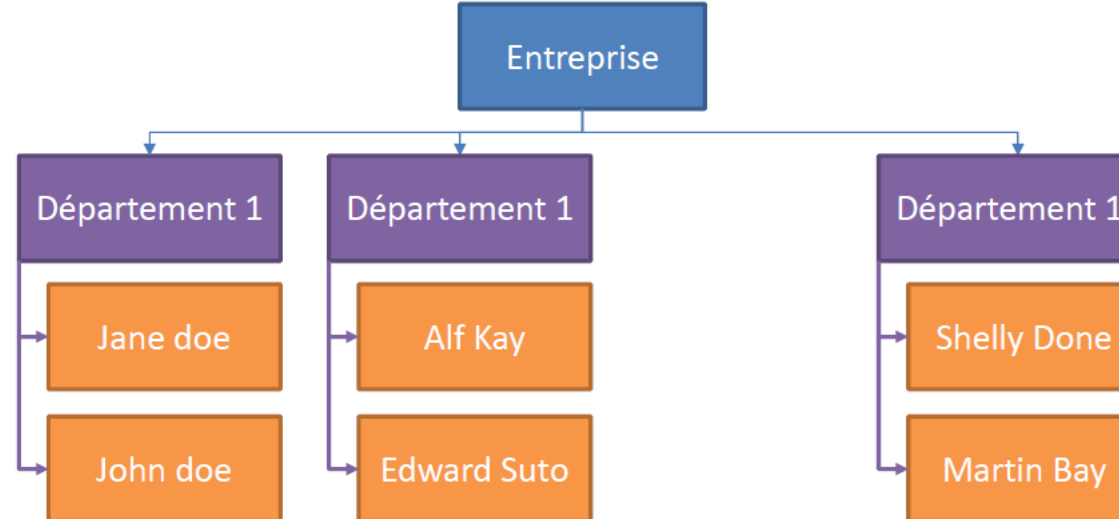
Types de bases de données

Différentes situations

- Donc **différentes solutions**
- Quelques **cas archétypaux**
 - Une base de données d'une application ne s'exécutant que pour un utilisateur / contexte
 - E.g. Collection de musiques / podcasts sur un ordinateur de bureau
 - Une application mobile / embarquée
 - E.g. Configuration d'une application Android, capteur de température intelligent
 - Une application d'entreprise avec plusieurs employés qui y accèdent simultanément
 - Une application distribuée sur internet avec plusieurs millions de clients
 - Stockage des données de consommation d'essence des voitures d'un constructeur
 - Annuaire d'entreprise...

Base de données hiérarchique

- **Cas classique:** annuaire d'entreprise
- **Exemple LDAP:**
 - Lightweight Directory Access Protocol (interrogation et modifications des services d'annuaires)
 - Les données peuvent être représentées par une arborescence



Base de données hiérarchique

- **Atouts**
 - Très performant pour accéder aux données
 - Les nœuds de l'arborescence peuvent être enrichis
 - Le nombre de niveaux est paramétrable
- **Inconvénients**
 - Stockage et requêtage spécialisé pour l'arborescence: La structure ne peut donc être utilisée que pour des données suivant ce principe
- **Cas d'usage**
 - Annuaire
 - Enregistrement d'une zone DNS

Base de données « clé-valeurs »

- **Cas classique:** configuration, mise en mémoire cache
- Exemples: Berkeley DB, Memcache, Redis, HBase
- **Besoins**
 - Récupérer une donnée uniquement par un identifiant
 - La donnée ne peut pas être découpée au sein de la base
 - Pas de traitement sur les données au sein de la base

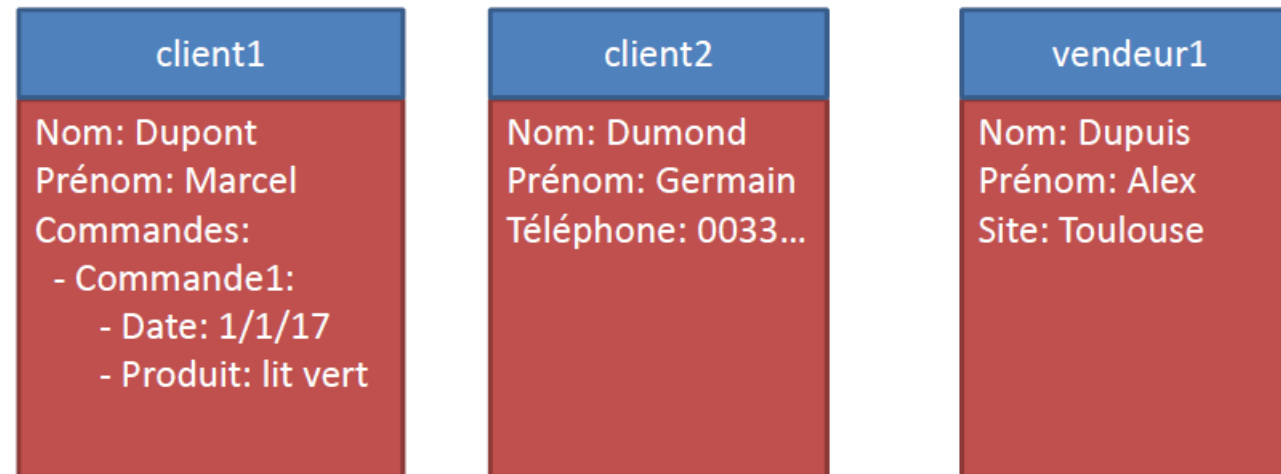
Clés	Données
app.themes	Darkmoon, Monokai, Maui
user.preferredLocale	sv
user.currentView	customers/dashboard

Base de données « clé-valeurs »

- **Atouts**
 - Accès aux informations extrêmement rapide
 - Très facile à répartir sur plusieurs machines pour une mise à l'échelle horizontale
- **Inconvénients**
 - La base de donnée ne doit contenir aucune logique
 - Les données doivent uniquement être retrouvées par le même identifiant
- **Cas d'usage**
 - Cache d'une application
 - Paniers dans les boutiques en ligne
 - Données de session

Base de données documents

- **Cas classique**: état d'application en ligne
- **Exemples**: MongoDB, couchDB, Firebase
- Si on souhaite manipuler des **documents isolés** les uns des autres

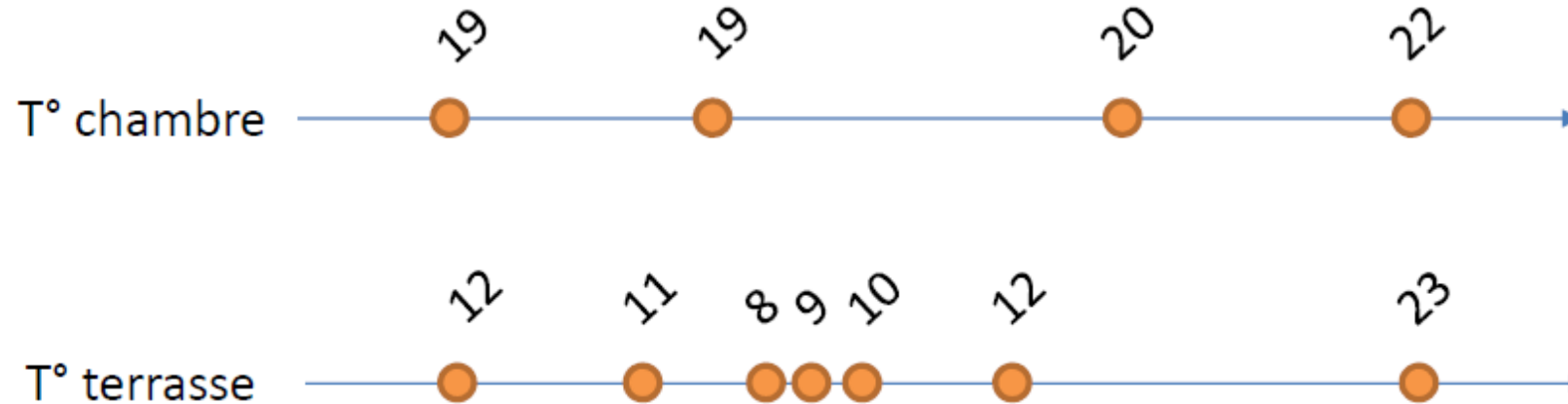


Base de données documents

- **Atouts**
 - Permet de s'abstraire d'un système de fichiers
 - Modèle adapté si les objets manipulés sont peu liés
 - Les documents n'ont pas de schéma défini, c'est un système très flexible
 - Les objets peuvent avoir des structures différentes
- **Inconvénients**
 - Les documents n'ont pas de schéma défini, c'est au développeur d'assurer la validité des données
- **Cas d'usage**
 - Application e-commerce
 - Document de travail au sein d'une application bureautique

Base de données « séries chronologiques »

- **Cas classique**: enregistrement des données temporelles
- **Exemple**: OpenTSDB, InfluxDB
- Représenter l'évolution d'une donnée plus ou moins complexe au cours du temps
 - Valeur boursière d'une entreprise
 - Température d'un capteur

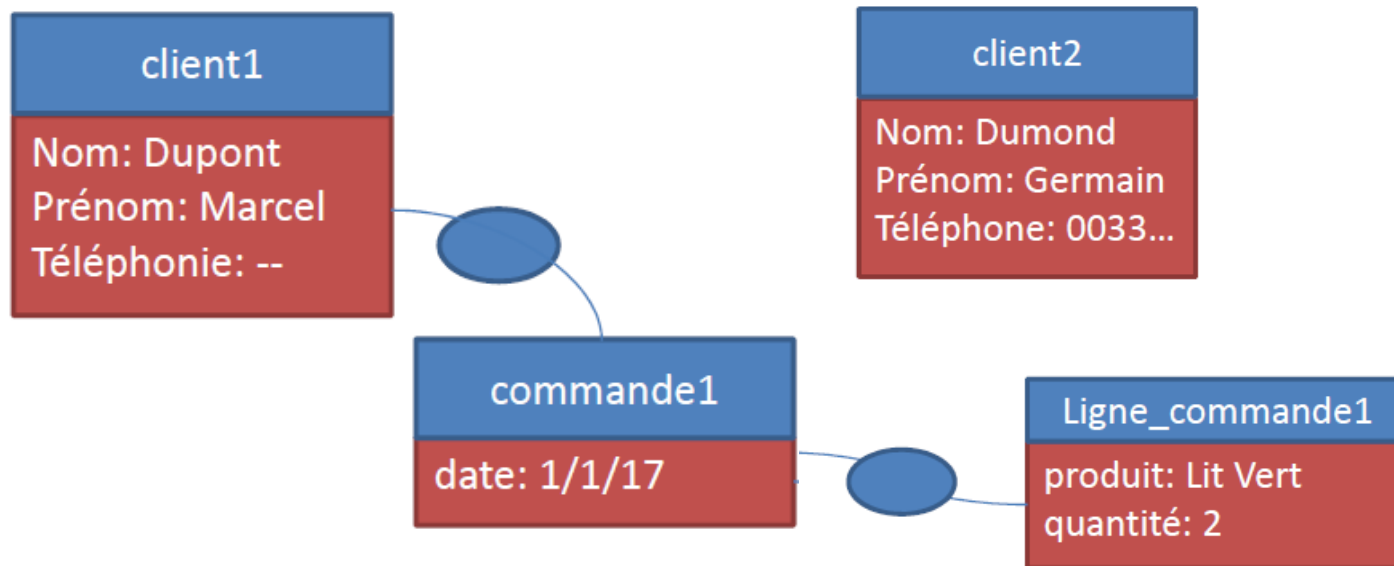


Base de données « séries chronologiques »

- **Atouts**
 - Très efficace pour stocker les données représentant un événement
 - Permet de faire des calculs sur les valeurs (fenêtrage, échantillonnage) et d'extrapoler des valeurs sur des périodes sans événements
- **Inconvénients**
 - Souvent les données sont simplement numériques
 - Les données n'ont pas d'identifiant autre que la date
- **Cas d'usage**
 - Suivi de grands nombres de valeurs venant de plusieurs capteurs
 - Suivi de compteurs de réservations dans un avion/hotel

Base de données relationnelle

- **Cas classique:** application mettant en lien plusieurs types de données
- **Exemples:** MySQL, PostgreSQL, MS SQL, Oracle, IBM DB2
- Si on souhaite manipuler des objets liés les uns aux autres par des relations



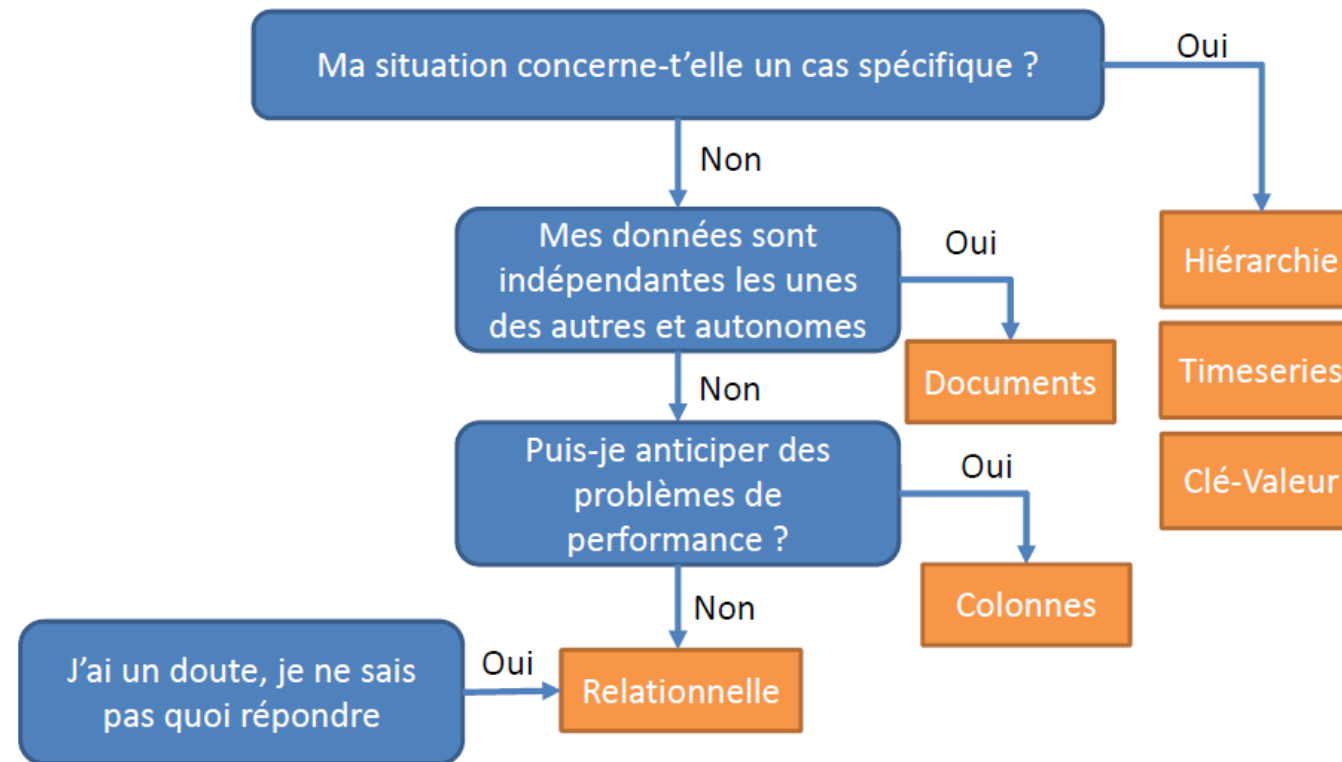
Base de données relationnelles

- **Atouts**
 - Modèle extrêmement adaptable aux applications
 - Permet une bonne évolutivité et de bonnes performances
 - Les types de données gérées sont validés et l'état de la base est en permanence cohérent
- **Inconvénients**
 - Gestion des migrations des types de données parfois complexe
 - Performances dans les situations de forte charge peu efficaces
 - Réplication complexe à mettre en œuvre
- **Cas d'usage**
 - 99% des applications où des objets sont liés les uns aux autres

Récapitulatif

Type	Point fort majeur	Faiblesse majeure	Cas d'utilisation
Hiérarchique	Performance des accès et opérations adaptées à une situation donnée	Structure imposée	Dès que vous entrez exactement dans cette situation
Clé-valeur			
Time Series			
Document	Extrêmement adaptable à la situation	La rigueur est à gérer par le développeur	
Relationnelle	Cohérence des données Validation des données	Passage à l'échelle horizontal plus complexe Le développeur doit accepter l'aide du DBA	Votre premier choix, n'utilisez un autre que si vous avez une excellente raison (ce qui ne veut pas dire choix par défaut)
Colonne	Adaptabilité intermédiaire, tolérance à un manque de cohérence locale	Rigueur équivalente aux documents	Second choix si vous sentez un problème de performance

Choix du type de base de données

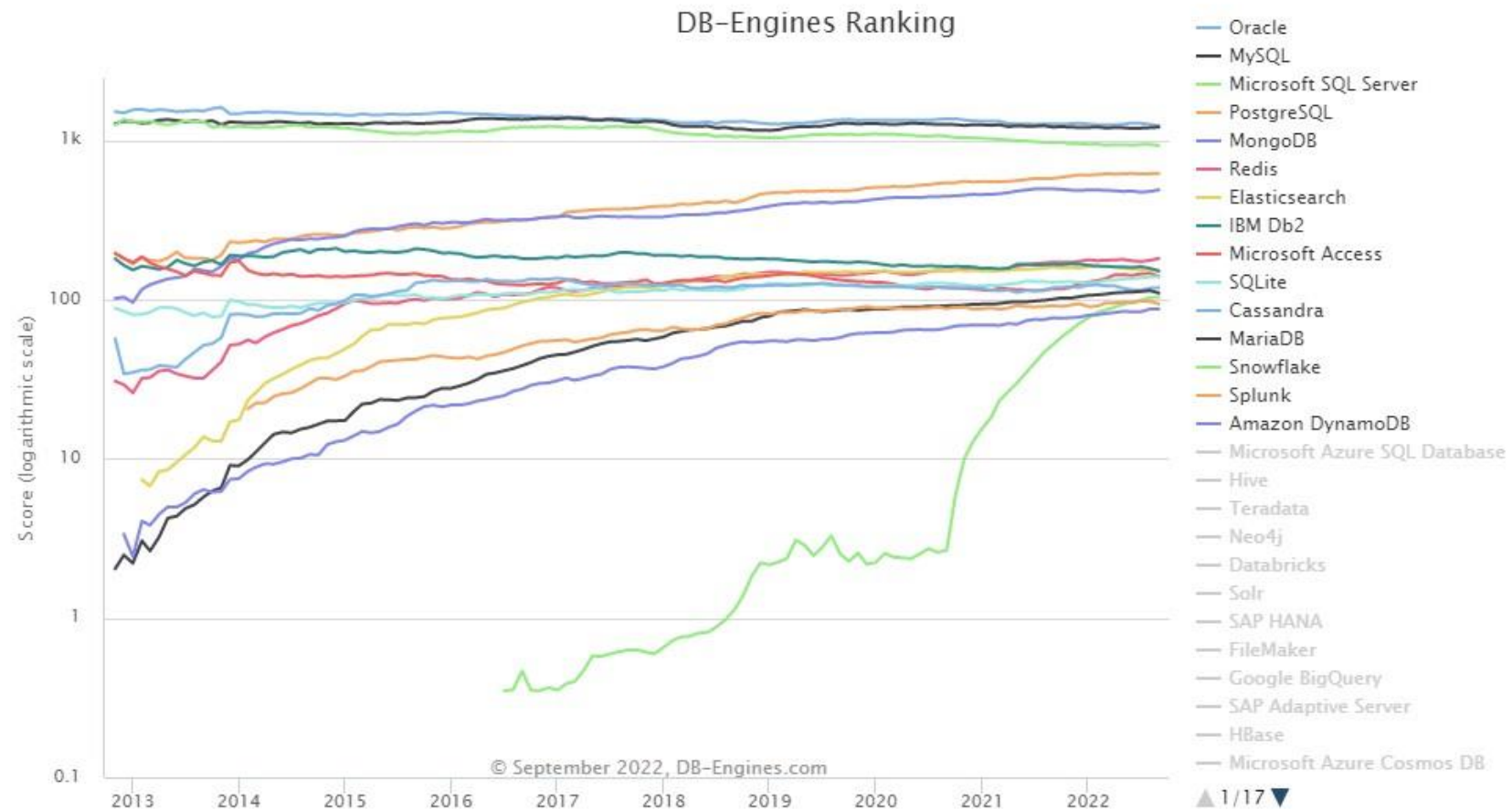


Etat des lieux

Historique

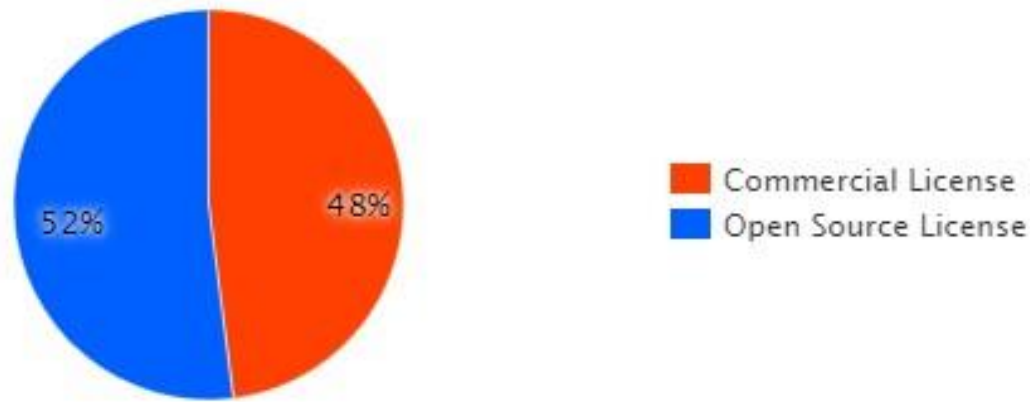
- **Inception**
 - Programme Apollo de la NASA (60's)
 - IBM IMS (Information Management System): base de données hiérarchique
- **Le modèle relationnel**
 - Proposé par E.F Codd, chercheur IBM, en 1970 et complété par C. J. Date (IBM)
 - SQL (Structured Query language) défini en 1974 par D. Chamberlain (IBM)
 - 90-00's: Concentration et apparition de l'**open source** (mySQL, PostgreSQL)
- **Bases de données orientées objets** (90's)
 - Reste confidentiel (<2% du marché)
- Dans les 2010's, essor du **No SQL** (clef-valeur, document, graphe...)

Popularité des bases de données



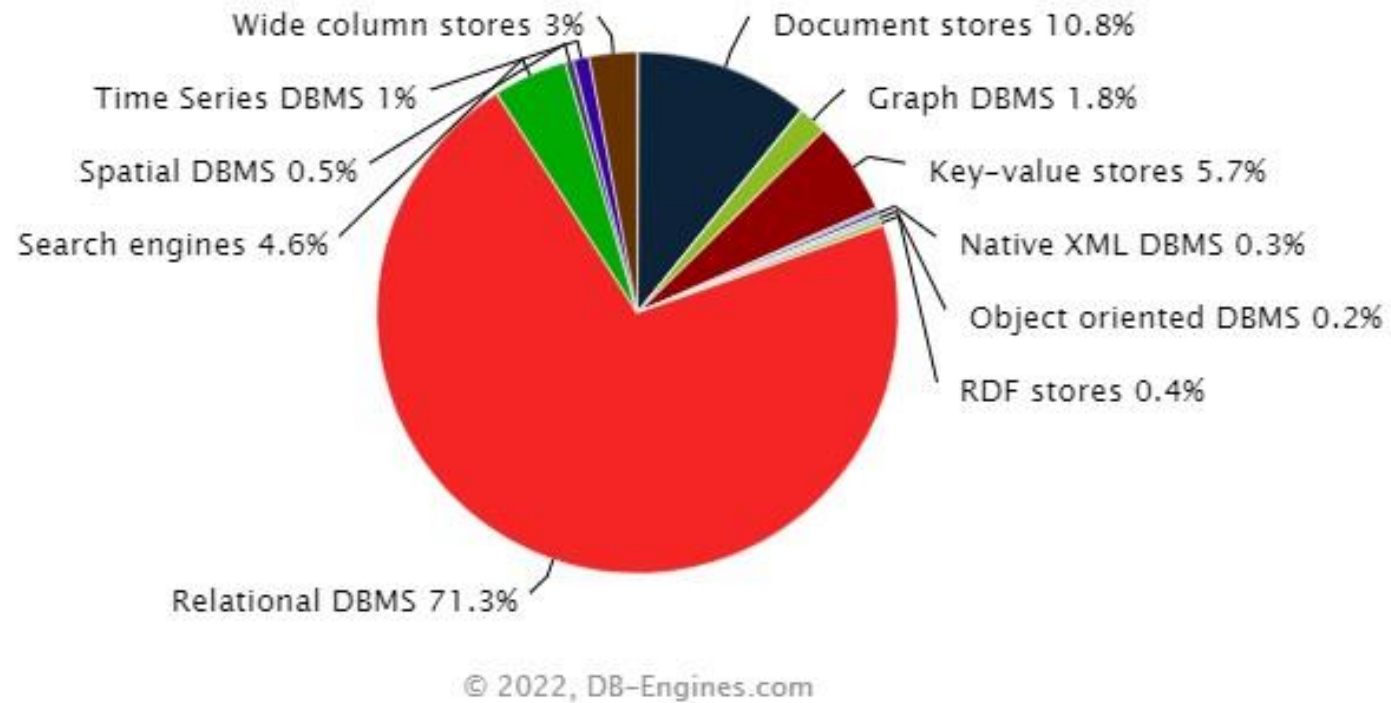
<https://db-engines.com/en/ranking>

Parts de marché des éditeurs de bases de données



© 2022, DB-Engines.com

Parts de marché des éditeurs de bases de données



Question?