

Principes des Systèmes de Gestion de Bases de Données

#1 – Introduction aux Systèmes de Gestion de Bases de Données

Équipe pédagogique BD

2020–2021



Introduction

- **Organisation du cours** : 8 semaines
 - 7h30 cours en amphi
 - 19h30 TD/TP
- **Supports de cours** : Chamilo
(<http://chamilo.grenoble-inp.fr/courses/ENSIMAG4MMPGS1>).
- **Notation** : Examen
- **Équipe Pédagogique** :
 - *Claudia Roncancio*
 - Ugo Comignani
 - Christophe Bobineau
 - Paul-Elliot Anglès d'Auriac
 - Sylvain Bouveret (remplaçant officiel)

- C.J. Date, *An Introduction to Database Systems*, Addison Wesley, 1990,
- C. Delobel, M. Adiba, *Bases de données et Systèmes Relationnels*, Dunod informatique, 1982
- G. Gardarin, *Objet et Relationnel*, Eyrolles, 1999
- W. Kim, *Modern Database Systems*, Addison Wesley, 1995
- Korth H., A. Silberschatz, *Database Systems Concepts*, Mc Graw Hill, 1991
- S. Navathe, R. Elamasri, *Fundamentals of Database Systems*, 2eme. ed., Addison-Wesley Pub, 1994
- J. Ullman, J. Widom, *A First Course in Database Systems*, Prentice-Hall, 1997
- T. Connolly, C. Begg, *Systèmes de Bases de Données*, Editions Reynald Goulet, 2005
- Supports de cours de l'Équipe Bases de Données de Grenoble INP – Ensimag, voir Chamilo

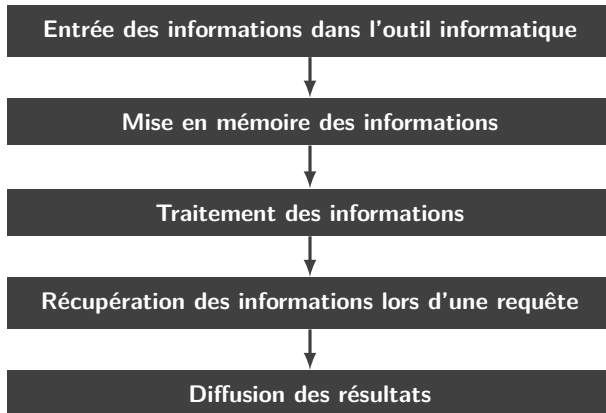
Pourquoi un cours de SGBD ?

Tout organisme / entreprise doit établir un système d'information.

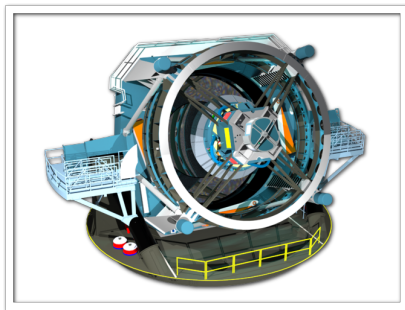
Il a pour but :

- d'identifier les événements qui peuvent surgir dans son environnement et dans son fonctionnement ;
- d'établir des procédures pour avoir la réaction la plus efficace possible lorsqu'un événement surgit ;
- de gérer l'information utile à la gestion de l'organisme (identification, production, diffusion).

- Cette gestion se fait le plus souvent grâce à l'outil informatique.
- Un système d'information n'est jamais définitif. Il doit s'adapter à :
 - l'évolution de l'environnement
 - l'autonomie de plus en plus grande des utilisateurs
 - l'augmentation du flux d'information
 - la mise en place de nouvelles techniques



- Les systèmes documentaires traitent des informations textuelles et fournissent le(s) document(s) susceptibles de contenir la réponse à la question posée. Ils ne donnent pas la réponse directement.
- Les systèmes de gestion de bases de données traitent des informations factuelles et numériques et offrent la réponse directe à la question émise.
- Les systèmes experts traitent des « connaissances » et offrent la réponse directe à la question ou au problème soumis.



Le programme Sloan Digital Sky Survey (2000-) a des archives de 140 Teraoctets (10^{12}).

Son successeur Large Synoptic Survey Telescope (2016-) acquiert ce volume de données tous les 5 jours




Les magasins de la chaîne Walmart gèrent plus d'un million de transactions par heure, qui produisent des masses de données estimées à plus de 2.5 Petaoctets (10^{15})



Le Grand Collisionneur / accélérateur de Hadrons (LHC) produit près de 15 millions de milliards d'octets par an (10^{15}). Ces données nécessitent 70 000 processeurs pour être traitées.



Facebook quant à lui stocke 250 Petaoctets (10^{15}) de données, 600 Teraoctets de nouvelles données chaque jour ; utilise plus de 10 Petaoctet (10^{15}) de données par jour pour ses traitements.

- Un Boeing 737 génère 240 To pour un vol intra US.
- 5,7 milliards de  / jour des utilisateurs Facebook.
- Google stocke pour chaque utilisateur connecté : info. perso , IP, services utilisés (type, usage et appareil utilisé), requêtes de recherche, localisation...
(voir <https://www.google.com/policies/privacy/#infocollect>)
- Plus de 98% de l'information existante a été créée ces cinq dernières années

L'univers digital double tous les 18 mois

0.79 Zettaoctets (10^{21}) en 2009 \Rightarrow 35 Zettaoctets en 2020 ($\times 44$)

Ce volume excède de loin notre capacité de traitement.

Déluge de données



Objectif général du cours : Bien utiliser un système de gestion de bases de données (SGBD) et comprendre son fonctionnement.

- Introduction SGBD et modèles de données
- Bases de données relationnelles ✗
 - Modèle relationnel ✗
 - Algèbre relationnelle ✗
 - SQL ✗
- Transactions ✗
- Conception de bases de données ✗
 - Analyse, dépendances, normalisation ✗
 - Modèle entité-associations, traduction en relationnel ✗

Qu'est-ce qu'un Système de Gestion de Bases de Données ?

Système de Gestion de Bases de Données (SGBD) : système permettant de stocker et d'accéder à de l'information.

Un simple ensemble de fichiers en fait ?



Élèves



Enseignants



Cours



Notes



Filières

Pas tout-à-fait...

Accès aux données :

1. Accès complexe à une information disparate et potentiellement stockée à plusieurs endroits.
→ **besoin d'un mécanisme d'interrogation complexe**
2. Accès inefficace si quantité de données conséquente
→ **besoin d'un accès optimisé aux données**
3. Données potentiellement sensibles (en terme de sécurité)
→ **besoin de mécanismes de contrôle d'accès**

Cohérence de l'information :

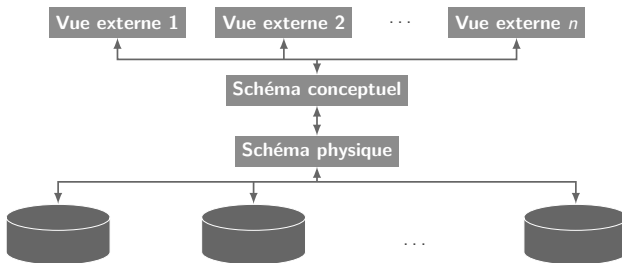
4. Pas de vérification de la cohérence de l'information entrée dans la base
→ **Expression et vérification de contraintes nécessaires**
5. Information potentiellement redondante \Rightarrow risque d'incohérence
→ **redondance subie** ✗
→ **redondance contrôlée (sauvegarde)** ✓
6. Accès concurrents de plusieurs utilisateurs \Rightarrow problèmes de cohérence
→ **besoin de mécanismes de gestion des accès concurrents**
7. Panne pendant une mise-à-jour massive ?
→ **besoin de garantir l'atomicité des mises-à-jour**
(atomique = indivisible, pas de mise à jour à moitié faite)

8. Stockage des données (fichiers, format) : influe sur leur interrogation.

→ **besoin d'assurer l'indépendance physique des données**

9. Différents types d'utilisateurs \Rightarrow différents points de vue sur les données.

→ **besoin d'assurer l'indépendance logique des données**

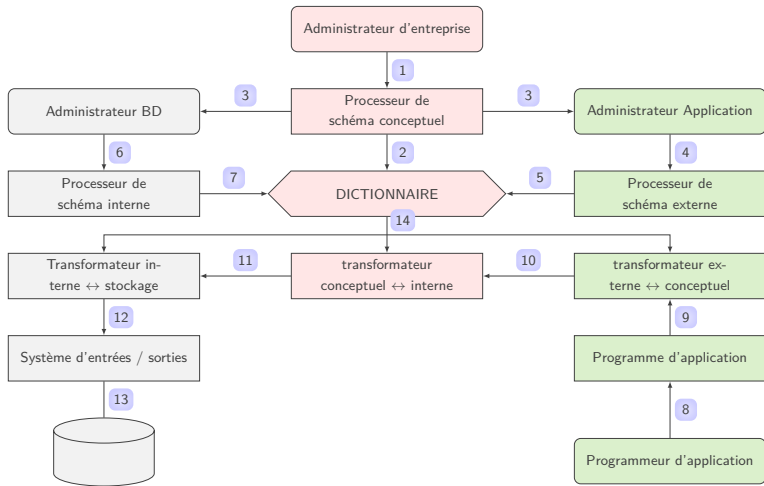


Système de Gestion de Bases de Données

Un SGBD est un système de stockage de données, qui permet de résoudre tous ces problèmes.

Tout SGBD s'appuie sur un *modèle de données*, permettant de représenter en mémoire des données du monde réel.

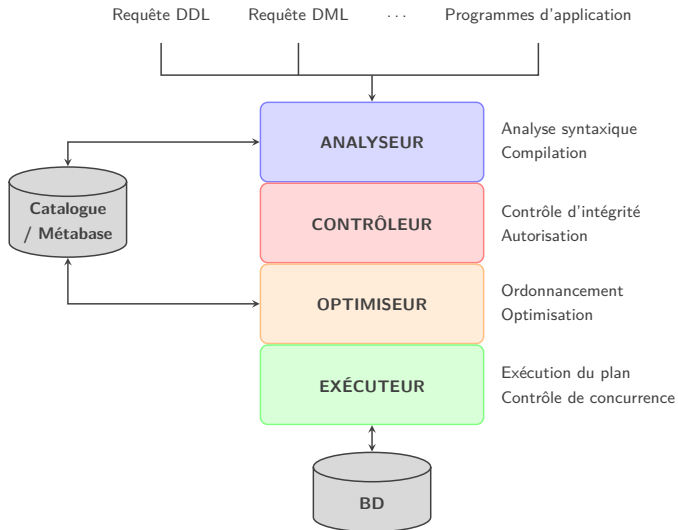
- Avant 1969 : la préhistoire des SGBD
 - Fichiers, modèles hiérarchique et réseau...
- 1970 : le modèle relationnel (Codd)
- 1970 – 2000 : Modèle relationnel, SQL, Transactions, Extensions, ...
Modèle Données Objet
 - RDBMS DB2, INGRES, Oracle, SQL Server; OLTP ; Distributed DB
 - Active Spatial Temporal Multimedia Deductive Object & OR
 - Warehouse, OLAP, Mining, Parallel
- après 2000 : XML, NoSQL & NewSQL,... (nouveaux « outils »)
 - WEB-Data Mgt XML, XQuery
 - Stream, Data Events
 - NoSQL, NewSQL, Cloud



1. Définition de schémas conceptuels (format source). Exemple : CREATE ENTITY, CREATE RELATIONSHIP ;
2. Définitions de schémas conceptuels (format objet) et rangement dans le dictionnaire ;
3. Définitions de schémas conceptuels (format édition) : consultation ;
4. Définitions des schémas externes (format source). Exemple : DEFINE VIEW ;
5. Définitions de schémas externes (format objet) et rangement dans le dictionnaire ;
6. Définitions de schémas internes (format source). Exemple : CREATE INDEX ;
7. Définitions de schémas internes (format objet) et rangement dans le dictionnaire ;

8. Manipulation de données externes (format source). Exemple : commandes RETRIEVE, APPEND, MODIFY, DELETE sur des objets externes ;
9. Manipulation de données externes (format objet) ;
10. Manipulation de données conceptuelles (format objet) : compilation des commandes RETRIEVE, APPEND, MODIFY, DELETE sur des objets conceptuels ;
11. Manipulation de données internes (format objet) : généré par le processeur de transformation conceptuel à interne afin de manipuler des données internes ;
12. Interface du système de stockage de données (lire ou écrire une page) ;
13. Interface mémoires secondaires : E/S sur disque ;
14. Interface d'accès au dictionnaire de données.

Une architecture plus réaliste ?



- l'administrateur de la base de données, qui s'occupe de l'installation globale du SGBD et des outils qui gravitent autour, et qui s'assure que le tout fonctionne de manière efficace ;
- le concepteur de la base de données (ou administrateur de données) dont le rôle est d'établir le schéma conceptuel de la base ;
- le développeur d'applications, en charge de programmer des applications qui interagissent directement avec le SGBD par l'envoi de requêtes d'interrogation et / ou de mise à jour des données ;
- l'utilisateur « avisé », qui interagit directement avec le SGBD grâce au langage d'interrogation (SQL) :
- l'utilisateur profane, qui ne voit les données qu'au travers un des programmes développés par le développeur d'applications.

- **Description des données**, grâce à un *DDL*, permettant :
 - à l'administrateur des données (entreprise), de décrire le schéma conceptuel de la base,
 - à l'administrateur de la base de données, de décrire les correspondances entre les structures physiques et le schéma conceptuel,
 - à l'administrateur d'application, de décrire les correspondances entre les vues et le schéma conceptuel ;
- **Recherche de données**, grâce à un *DML* ;
- **Mise-à-jour des données**, grâce au *DML* ;
- **Transformation de données**, c'est-à-dire d'établir une correspondance entre des données décrites à deux niveaux différents (interne, conceptuel, externe) ;
- **Contrôle d'intégrité**, grâce au *DDL* ;
- **Gestion des transactions**, grâce à un *CL* ;
- **Gestion des droits d'accès**, grâce au *CL*.

À propos de modèles de données

Modèle = représentation formelle abstraite (simplifiée) de la réalité.

Modèle de données = représentation formelle abstraite des données du monde réel à manipuler.

Permet de décrire :

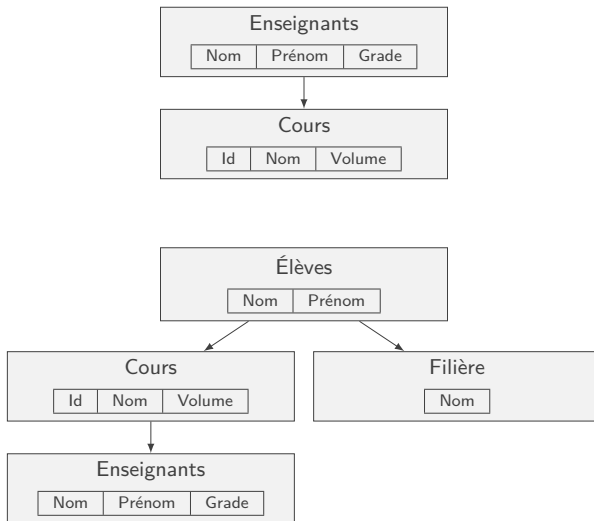
1. *la structure des données* (types d'objets, relations entre ces types, etc.)

Exemple : « un étudiant est un objet qui possède un numéro INSEE, un nom, un prénom, une adresse... »

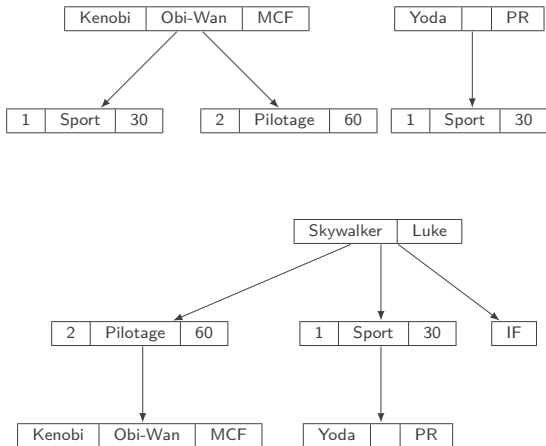
2. *les occurrences des données*

Exemple : « Dark Vador est un étudiant possédant le numéro INSEE 1770901142555, le nom 'Vador', le prénom 'Dark', l'adresse vador@imag.fr »

- **Hiérarchique** : arbres d'enregistrements
- **Réseau** : graphes d'enregistrements
- **Relationnel** : relations (tables)
- **Semi-structuré** : arbres, graphes

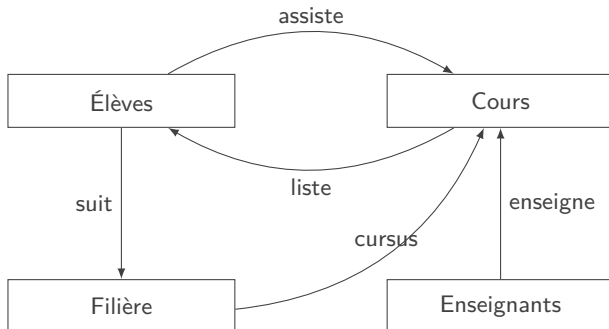


Modèle hiérarchique : exemple d'instanciation



Un modèle défini par le CODASYL.

Des articles et des liens (\approx pointeurs)



Dans ce cours, nous nous focaliserons uniquement sur

le modèle relationnel

- développé par Codd en 1970
- très simple conceptuellement (tableaux 2D)
- très puissant grâce à l'*Algèbre Relationnelle*
- modèle sur lequel s'appuie la norme *SQL2*, standard encore à l'heure actuelle

Modèles structurés (hiérarchique, réseau, relationnel...) : séparation claire schéma / données

Ce n'est plus le cas pour les **modèles semi-structurés**.

Objectif : représenter des structures de données

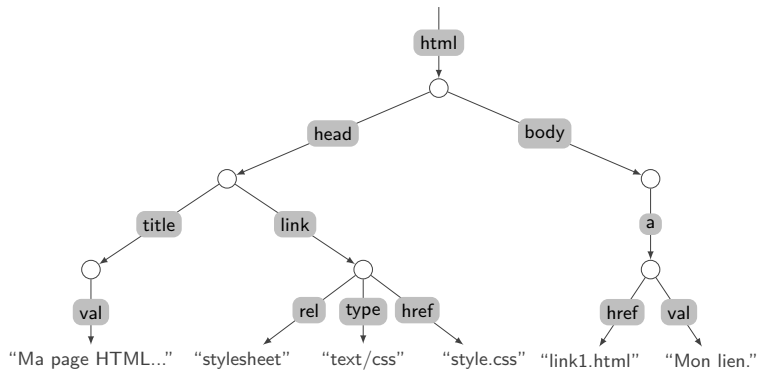
- **Irrégulières** : on peut comparer des données dans des formats différents (e.g. une chaîne de caractères avec un n -uplet)
- **Implicites** : données et structures (grammaire, schéma) sont mélangées
- **Partielles** : coexistence de données structurées et non structurées

- Pas de séparation entre les données et le schéma : données sans schéma ou auto-descriptives
- **Idée** : ensemble de paires (étiquette — valeur)
- Modèle sous-jacent : **arbre** ou **graphe orienté**
 - nœuds = objets / valeurs complexes
 - libellés sur les arcs
 - valeurs atomiques sur les feuilles
- **Flexibilité** : pas de restriction sur les libellés des arcs sortants d'un nœud (ni sur le nombre de successeurs)
- **Typage possible** (implicite ou par annotation)

Un fichier HTML...

```
<html>
  <head>
    <title>Ma page HTML...</title>
    <link rel="stylesheet" type="text/css" href="style.css" /
  >
</head>
<body>
  <a href="link1.html">Mon lien.</a>
</body>
</html>
```

L'arbre DOM du fichier HTML



XML est un langage de représentation de données s'appuyant sur une syntaxe à balises similaire à HTML (HTML : un sous-langage de XML).

Structure d'un document XML :

- **prologue** dont la présence est facultative, mais fortement conseillée

```
<?xml version="1.0" standalone="yes" ?>
```

- **arbre d'éléments**, dont la présence est obligatoire

```
<document>  
  <salutation> Monsieur </salutation>  
</document>
```

Dans le domaine de l'échange de données sur le Web, XML tend à être progressivement remplacé par JSON (*JavaScript Object Notation*).

```
{ "BARS": [
  { "NAME": "Joe's Bar",
    "BEER": [
      { "NAME": "Bud",
        "PRICE": "2.50" },
      { "NAME": "Miller",
        "PRICE": "2.75" } ]
  },
  { "NAME": "Sue's Bar",
    "BEER": [
      { "NAME": "Bud",
        "PRICE": "2.50" },
      { "NAME": "Miller",
        "PRICE": "3.00" } ]
  }
]
```

Tendance très forte à l'heure actuelle : **NO-SQL** (Not Only SQL)

- Il ne s'agit pas de dire que SQL ne devrait pas être utilisé, ni qu'il est mort...
- Il s'agit de reconnaître que pour certains problèmes particuliers, d'autres solutions de stockage sont plus adaptées.

Tendances : taille, connexion, données semi-structurées, architecture...

- **Clef-valeur** : Dynamo, Voldemort, Riak, Redis, Cassandra,...
- **BD orientées document** : MongoDB, CouchDB, Redis...
- **BD orientées colonnes** : Big Table, Hbase, Hypertable, Cassandra,...
- **BD orientées graphes** : Neo4J, FlockDB, Pregel...

Nous ne traiterons cette année que les bases de données **relationnelles** (SQL).

- Les SGBD relationnels restent encore largement majoritaires à l'heure actuelle
- Les problématiques fondamentales restent les mêmes quels que soient les modèles de données

Le mot de la fin

- Introduction SGBD et modèles de données ✓
- Bases de données relationnelles ✗
 - Modèle relationnel ✗
 - Algèbre relationnelle ✗
 - SQL ✗
- Transactions ✗
- Conception de bases de données ✗
 - Analyse, dépendances, normalisation ✗
 - Modèle entité-associations, traduction en relationnel ✗

- Problèmes liés au stockage persistant des données
- Définition d'un SGBD et architecture basique
- Fonctions et utilisateurs d'un SGBD
- Définition d'un modèle de données
- Principaux modèles de données