BDBIO - Gestion de données

Fabien Duchateau

fabien.duchateau [at] univ-lyon1.fr

Université Claude Bernard Lyon 1

2024 - 2025

https://perso.liris.cnrs.fr/fabien.duchateau/BDBIO/

Gestion de données

Prise de conscience de l'importance des informations :

- Organisations (entreprises, collectivités) et individus (web)
- Nombreux domaines (biologie, astronomie, service public, etc.)
- Société de l'information, de la connaissance

Organiser, exploiter, partager les données ⇒ gestion de données

---

Rubin M.R., Taylor M., *The Knowledge Industry in United States : 1960-1980*, Princeton University Press (1984)

Curien N., Muet P.A., *La société de l'information*. Documentation Française (2004)

Exemples de collections de données

- Une banque gère un ensemble de données sur ses clientes, sur les comptes bancaires, sur ses employées, les produits et services proposés, etc.
- La société qui gère les Velovs a un ensemble de données sur les abonnées, la flotte de vélos, les locations en cours, etc.
- Une équipe de biologistes modélise les protéines, les complexes protéiques, les expérimentations réalisées, les interactions découvertes, etc.
- Voire même les catégories d'Aristote !

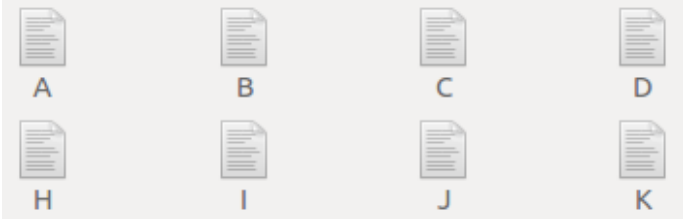
---

http://fr.wikipedia.org/wiki/Cat%C3%A9gories_(Aristote)

Évolution du stockage des données

- Des **fiches papier**...

- En passant par des **fichiers informatiques**...



- Aux **bases de données**

Gestion de données par fichiers informatiques

- Un fichier est une suite d'enregistrements contenant des données logiquement liées
- Chaque application définit et gère ses fichiers
- Intégration étroite entre l'application et les fichiers
  - la manipulation des fichiers est directement intégrée dans le programme

Gestion de données par fichiers informatiques

- Un fichier est une suite d'enregistrements contenant des données logiquement liées
- Chaque application définit et gère ses fichiers
- Intégration étroite entre l'application et les fichiers
  - la manipulation des fichiers est directement intégrée dans le programme

Gestion par fichiers $\Rightarrow$ problèmes de redondance, de cohérence, d'accès aux données, de sécurité, etc.

Plan

Définition

Un système de gestion de bases de données (SGBD, DBMS en anglais) est une application qui permet de définir et manipuler un ensemble de données (les bases de données)

Principes des SGBD

- **Abstraction** = le SGDB est un médiateur entre l'utilisatrice et l'ordinateur et doit donc présenter les données de manière intuitive et permettre de les manipuler à un niveau abstrait (sans avoir à considérer des détails d'implémentation)

- **Universalité** = le SGBD doit permettre de représenter l'ensemble des données d'une entreprise ou organisation quelconque tout en offrant des fonctionnalités variées.

- **Indépendance** = le SGBD distingue ce que voient les utilisatrices (clarté) et ce qui est effectivement géré (efficacité) à travers trois niveaux indépendants

Utilisation d'un SGBD

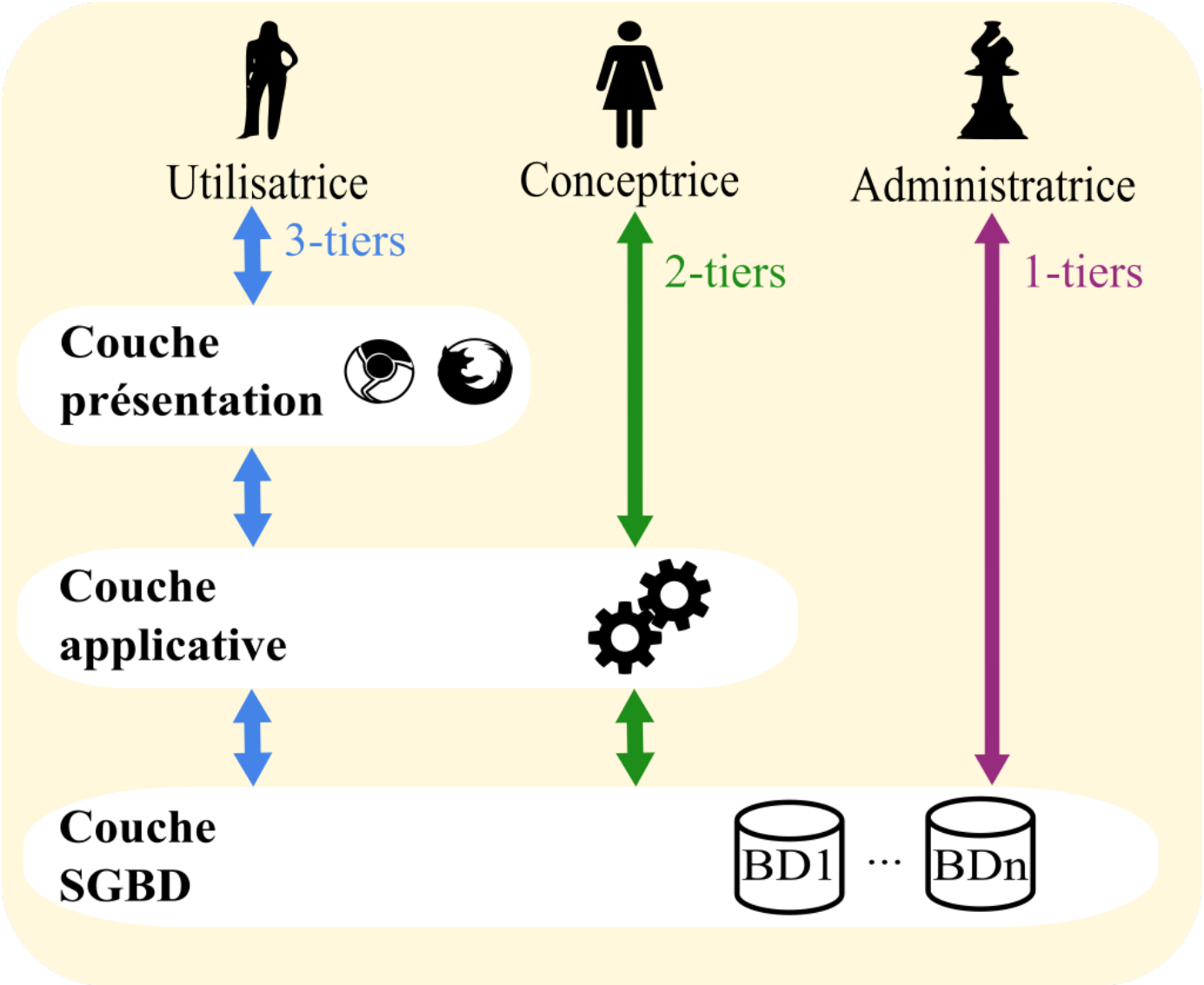
Trois niveaux (idéalement) indépendants pour trois profils :

- Niveau vue/externe (profil utilisatrice)
  - quelles données sont manipulables ?
- Niveau conceptuel/logique (profil conceptrice)
  - quelles est la structure des données stockées ?
- Niveau physique/interne (profil administratrice)
  - comment sont organisées des données sur le support physique ?
  - comment elles sont stockées ?
  - comment accéder rapidement aux données (index) ?

Architectures des SGBD

- Centralisée (un-tiers) = accès direct au SGBD
  - en général profil administratrice
  - utilisation des langages de définition et manipulation du SGBD
- Client-serveur (deux-tiers) = accès au SGBD via une couche applicative (vue abstraite des données)
  - en général profil conceptrice / programmeuse
  - utilisation de bibliothèques logicielles
- Trois-tiers = accès au SGBD via une couche présentation et une couche applicative
  - en général profil utilisatrice (plusieurs vues sur les données)
  - utilisation d'interfaces (e.g., formulaires web)

Architecture des SGBD (2)



Caractéristiques d'un SGBD

- Garantie de l'intégrité (contraintes définies avec les données et préservées par le SGBD)
- Reprise après panne (journalisation)
- Gestion de la concurrence (i.e., incohérences lors d'accès multiples)
- Transactionnel (validation ou annulation d'une suite d'opérations liées)
- Gestion de la sécurité (authentification, privilèges, etc.)

Caractéristiques générales, mais pas forcément valable pour chaque SGBD

Fonctionnalités d'un SGBD

Définition d'un schéma (auquel les données se conforment)

- avec un Langage de Description de Données (LDD)

Création, interrogation, mise à jour et suppression de données (CRUD)

- avec un Langage de Manipulation de Données (LMD)

Ces deux langages sont communs aux différentes applications qui utilisent le SGBD :

- cela garantit une indépendance entre données et applications

---

http://fr.wikipedia.org/wiki/CRUD

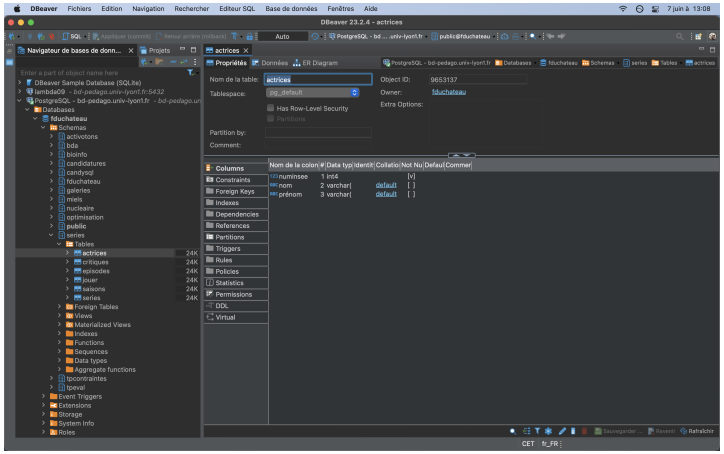
Interactions avec le SGBD

Utilisation du SGBD :

- Interpréteur de commandes
- Interface graphique
- Langage de programmation
  - Java, Python, PHP, C#...
  - via des bibliothèques pour envoyer des requêtes au SGBD.

```
fduchateau=> \d saisons
                        Table "series.saisons"
    Column     |          Type          | Collation | Nullable | Default
---------------+------------------------+-----------+----------+---------
 idsaison      | integer                |           | not null |
 datelancement | date                   |           |          |
 nomsérie      | character varying(42) |           |          |
Indexes:
    "saisons_pkey" PRIMARY KEY, btree (idsaison)
Foreign-key constraints:
    "saisons_nomsérie_fkey" FOREIGN KEY ("nomsérie") REFERENCES series("nomsérie")
Referenced by:
    TABLE "episodes" CONSTRAINT "episodes_idsaison_fkey" FOREIGN KEY (idsaison) REFERENCE

fduchateau=> select * from episodes join saisons using(idsaison);
 idsaison | numéro |           titre           | datelancement |      nomsérie
----------+--------+---------------------------+---------------+---------------------
        1 |      1 | The Skank Reflex Analysis | 2011-09-22    | The Big Bang Theory
        2 |      1 | The Date Night Variable   | 2012-09-27    | The Big Bang Theory
        3 |      1 | Winter is coming          | 2011-04-17    | Game of Thrones
        3 |      2 | The Kingsroad             | 2011-04-17    | Game of Thrones
        4 |      1 | Pilot                     | 2014-03-19    | The 100
        5 |      3 | La Table de Breccan       | 2005-03-01    | Kaamelott
(6 rows)
```



https://fr.wikipedia.org/wiki/L4G

Plan

Définitions

Modèle de données : modèle qui décrit formellement comment les données sont organisées (LDD et LMD)

Exemples : modèle relationnel, modèle de document

Schéma : représentation concrète des concepts d'un domaine

```
R(att1, att2, #att3)
```

```
{ _id: <int>,
nom: <string> }
```

Instances : ensemble de données conforme à un schéma

att1	att2	att3
1	a	x
2	b	y

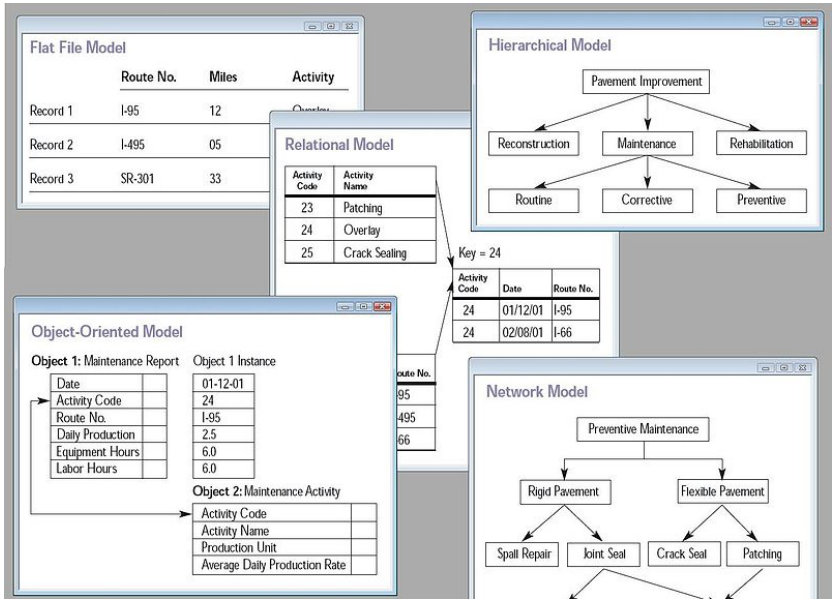
```
[ { _id: 123, nom: "Ana" },
{ _id: 234, nom: "Bob" }, ... ]
```

---

http://en.wikipedia.org/wiki/Data_model
E. F. Codd. *Data Models in Database Management*. SIGMOD Record (1981)

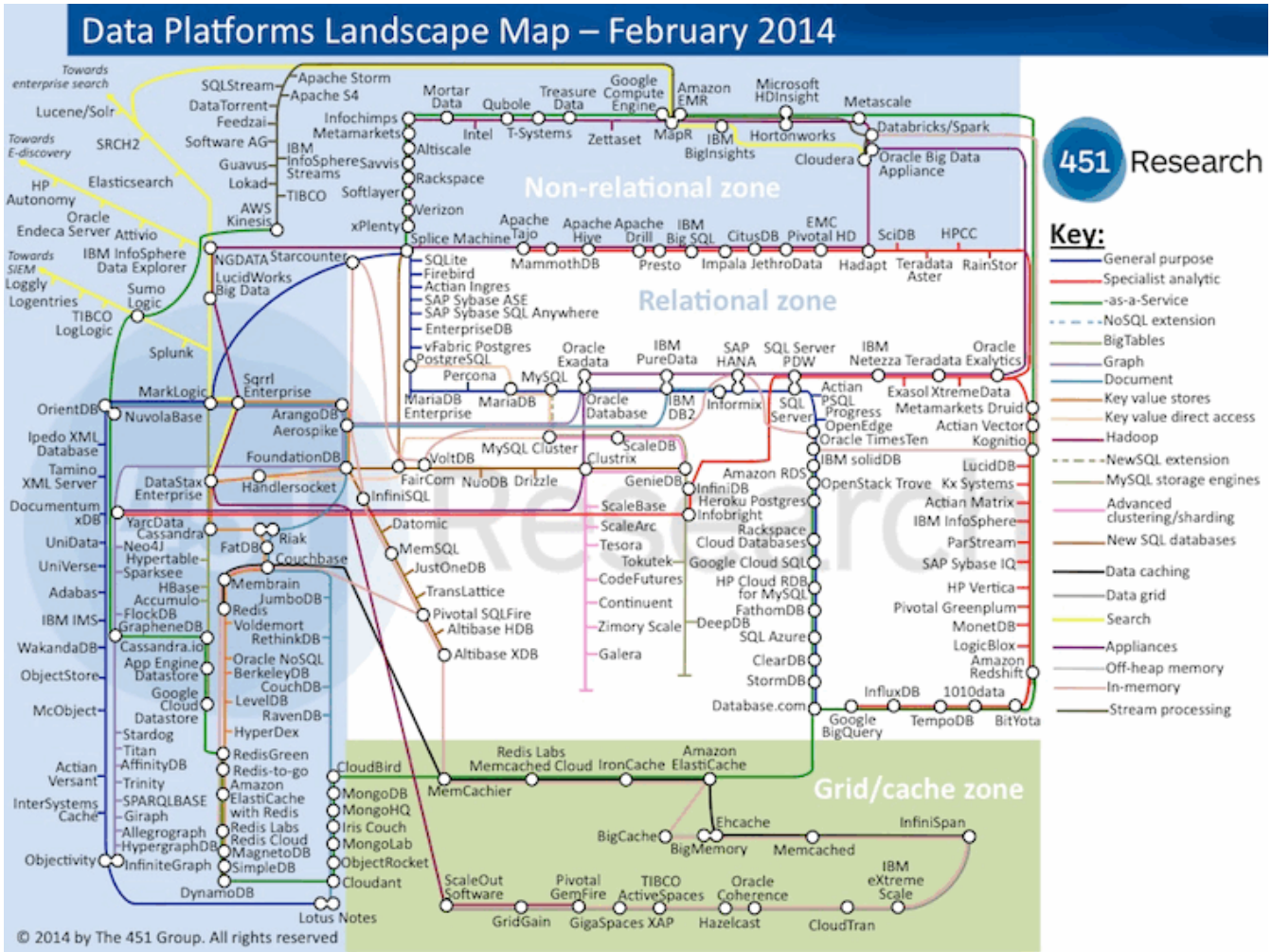
Historique des modèles de SGBD

- Années 1980 : modèles hiérarchique, réseau, entité-attribut-valeur, relationnel, etc.
- Années 1990 : dominance du modèle relationnel, modèles dimensionnel, objet, etc.
- Années 2000 : phénomène du web et du "Big Data", modèles post-relationnels (document, graphe, colonne, etc.)



http://en.wikipedia.org/wiki/Database_model

Aperçu des SGBD existants



https://451research.com/451-research-data-platform-map

Modèles génériques vs spécifiques

Pourquoi ne pas utiliser des modèles de données spécifiques à la bioinformatique (FASTA, REFSEQ, SAM pour Sequence Alignment/Map, etc.) ?

- Simplicité / complexité
- Performances, obsolescence
- Utilisation d'outils existants (parsers, SGBD, langages, etc.)
- Compréhension par des personnes "non-bioinformatiques"
- ...

Aperçu des modèles étudiés

Dans cet enseignement, étude de plusieurs modèles :

- Relationnel
- Document, notamment XML et JSON
- Graphe RDF

---

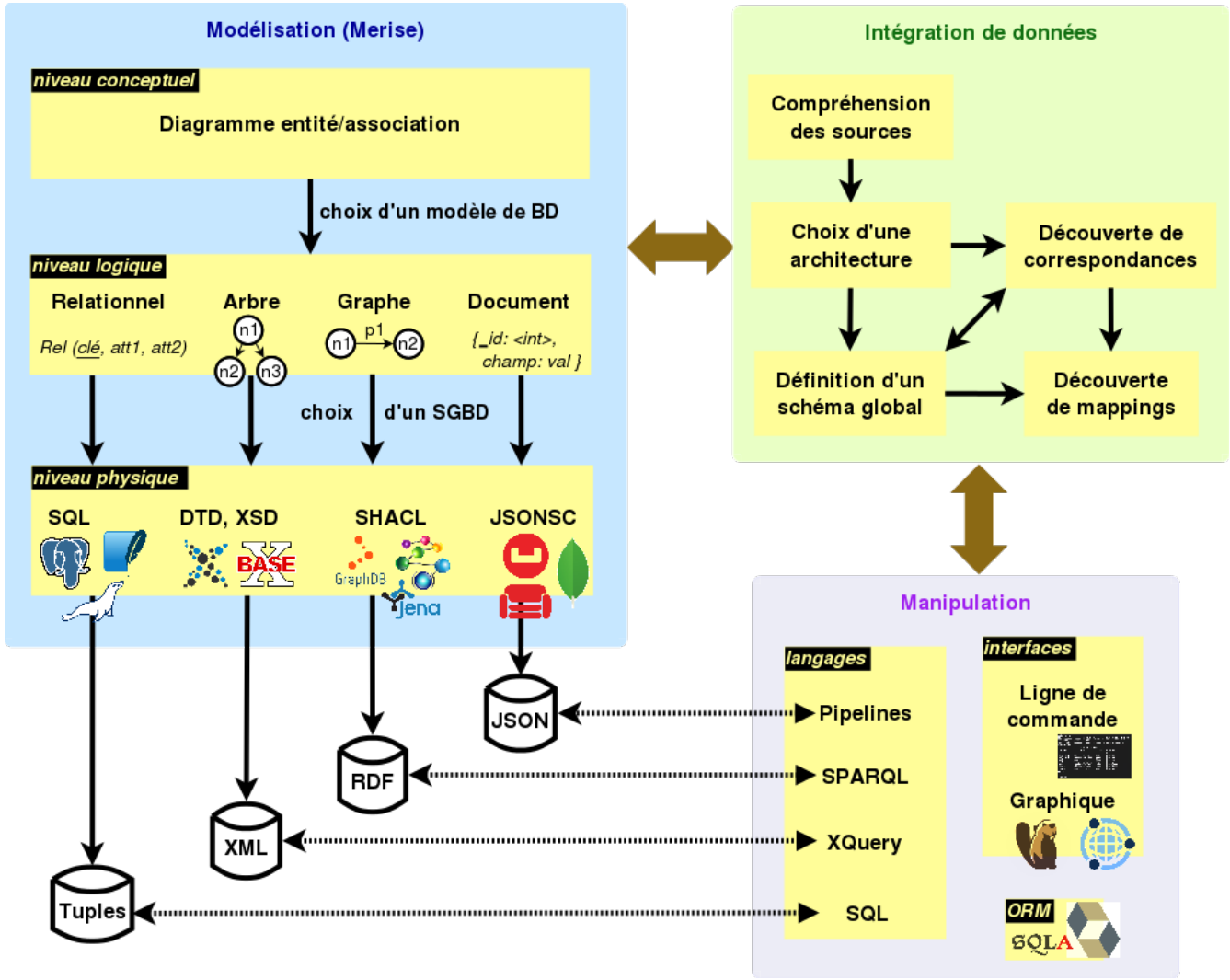
http://en.wikipedia.org/wiki/Relational_model
http://en.wikipedia.org/wiki/XML_database
http://en.wikipedia.org/wiki/Document-oriented_database
http://en.wikipedia.org/wiki/Resource_Description_Framework
http://db-engines.com/en/ranking

Aperçu des modèles étudiés (2)

Modèle	Sérialisation	LDD	LMD	Exemples SGBD
Relationnel	n-uplets	SQL	SQL	SQLite, MariaDB, PostgreSQL
Arbre	XML	DTD, XML Schema	XPath, XQuery	XBase, existDB
Document	JSON	?	?	MongoDB, CouchDB
Graphe (triplets)	RDF/XML, N3, Turtle, N-triples	RDFS OWL	SPARQL, RQL	AllegroGraph, BlazeGraph

Détails sur les modèles vus en BDBIO. La sérialisation représente le mode de stockage, le LDD est le langage de description de données (schéma), et le LMD est le langage de manipulation de données.

Aperçu des notions de BDBIO



Plan

Cycle de vie d'une base de données

Étapes du cycle de vie d'une base de données :

1. Définition des spécifications (besoins)
2. Réalisation d'un modèle au niveau conceptuel
3. Réalisation d'un modèle au niveau logique
4. Réalisation d'un modèle au niveau physique
5. Utilisation de la BD, maintenance

Motivation

Imaginons que l'on gère des candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

Jeu de données francisé inspiré du cours Databases de Stanford

Motivation

Imaginons que l'on gère des candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- *Bob* candidate une nouvelle fois ?

Jeu de données francisé inspiré du cours Databases de Stanford

Motivation

Imaginons que l'on gère des candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- *Bob* candidate une nouvelle fois ?
- l'effectif de l'université *INSA* change ?

Jeu de données francisé inspiré du cours Databases de Stanford

Motivation

Imaginons que l'on gère des candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- *Bob* candidate une nouvelle fois ?
- l'effectif de l'université *INSA* change ?
- l'on veut ajouter une université qui n'a pas encore de candidatures ?

Jeu de données francisé inspiré du cours Databases de Stanford

Motivation

Imaginons que l'on gère des candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- *Bob* candidate une nouvelle fois ?
- l'effectif de l'université *INSA* change ?
- l'on veut ajouter une université qui n'a pas encore de candidatures ?

⇒ **Mauvaise modélisation !**

Jeu de données francisé inspiré du cours Databases de Stanford

Objectifs de la modélisation

Conception de base de données = modélisation et implémentation des spécifications pour la partie données, en démarrant par un modèle abstrait (niveau conceptuel) jusqu'à l'obtention d'un script pour implémenter la BD dans un SGBD donné (niveau physique)

Pourquoi modéliser ?

- Structurer et organiser les données d'un domaine
- Optimiser les performances
- Garantir la cohérence et la non-redondance
- Difficulté à modéliser directement un problème

---

http://en.wikipedia.org/wiki/Data_modeling

Étapes de la modélisation (Merise)

Niveau conceptuel = modèle avec une forte abstraction (indépendant des SGBD), qui organise les données issues des spécifications (e.g., diagramme entité / association, diagramme UML de classes)

Niveau logique = modèle qui se rapproche du type de SGBD choisi (e.g., relationnel, graphe, document)

Niveau physique = modèle de création des données pour un SGBD particulier (e.g., MariaDB, PostgreSQL, MongoDB)