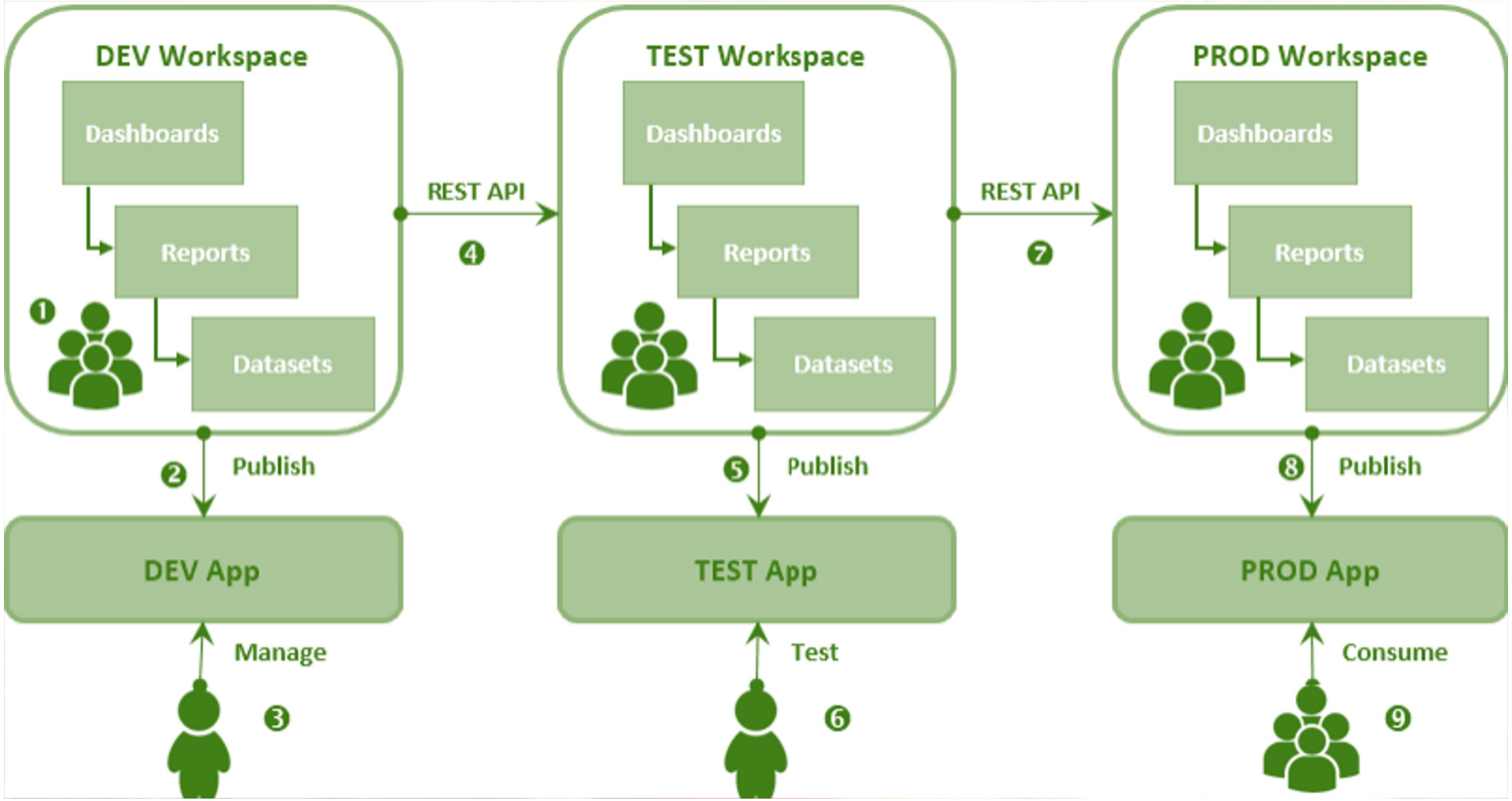



# 12. La gestion des données

# Quelques concepts fondamentaux

**Les données** et les **connaissances** doivent être structurées de manière à pouvoir être :

- stockées et accessibles
- modifiables et ajoutables
- extraites utilement et efficacement (extraire - transformer - charger)
- exploitées par des **humains** et des **ordinateurs** (programmes, bots, IA)

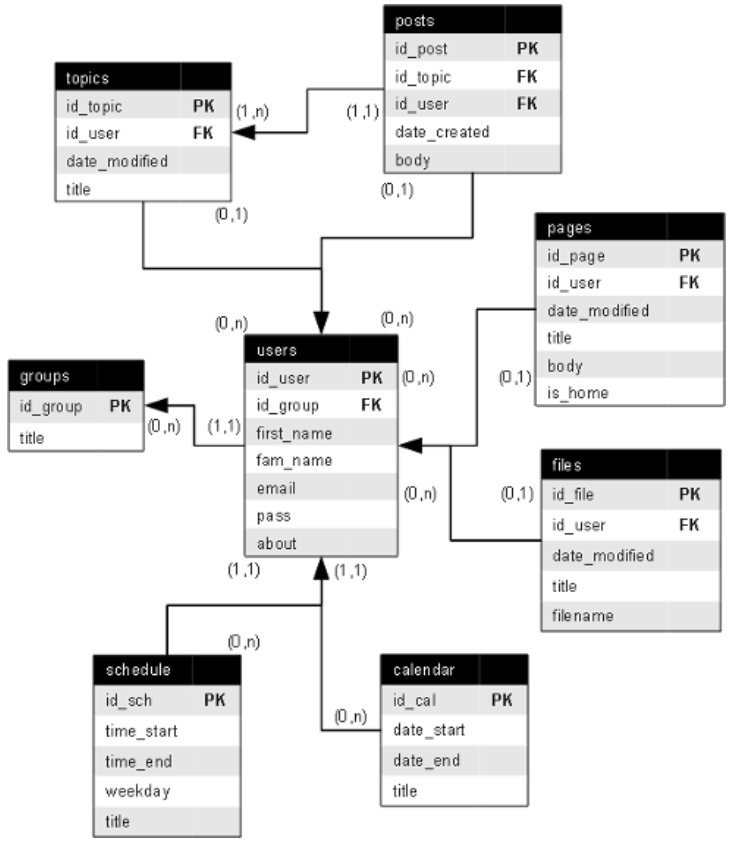
# La modélisation des données

Les modèles de données sont des descriptions **abstraites/logiques** d'un système, utilisant des termes qui sont implémentables en tant que structure d'un type de logiciel de gestion des données.

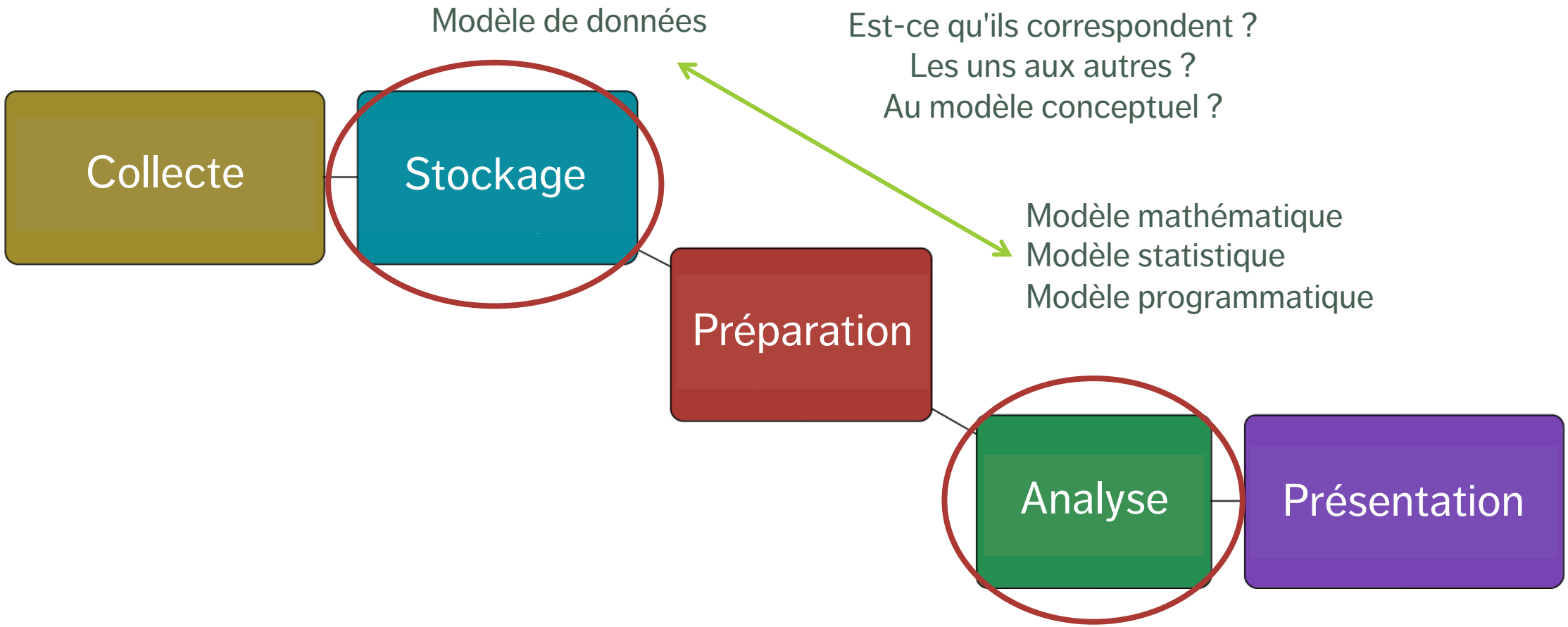
Cela se trouve à mi-chemin entre un **modèle conceptuel** et une **implémentation de banque de données**.

Les données elles-mêmes concernent les **instances** – le modèle, quant à lui, concerne les **types d'objets**.

Une autre option à envisager : les **ontologies**.



# Un pipeline de données automatisé



[Representation of Langerhans cells in the *Cell Ontology*, A. M. Masci *et al.* ]

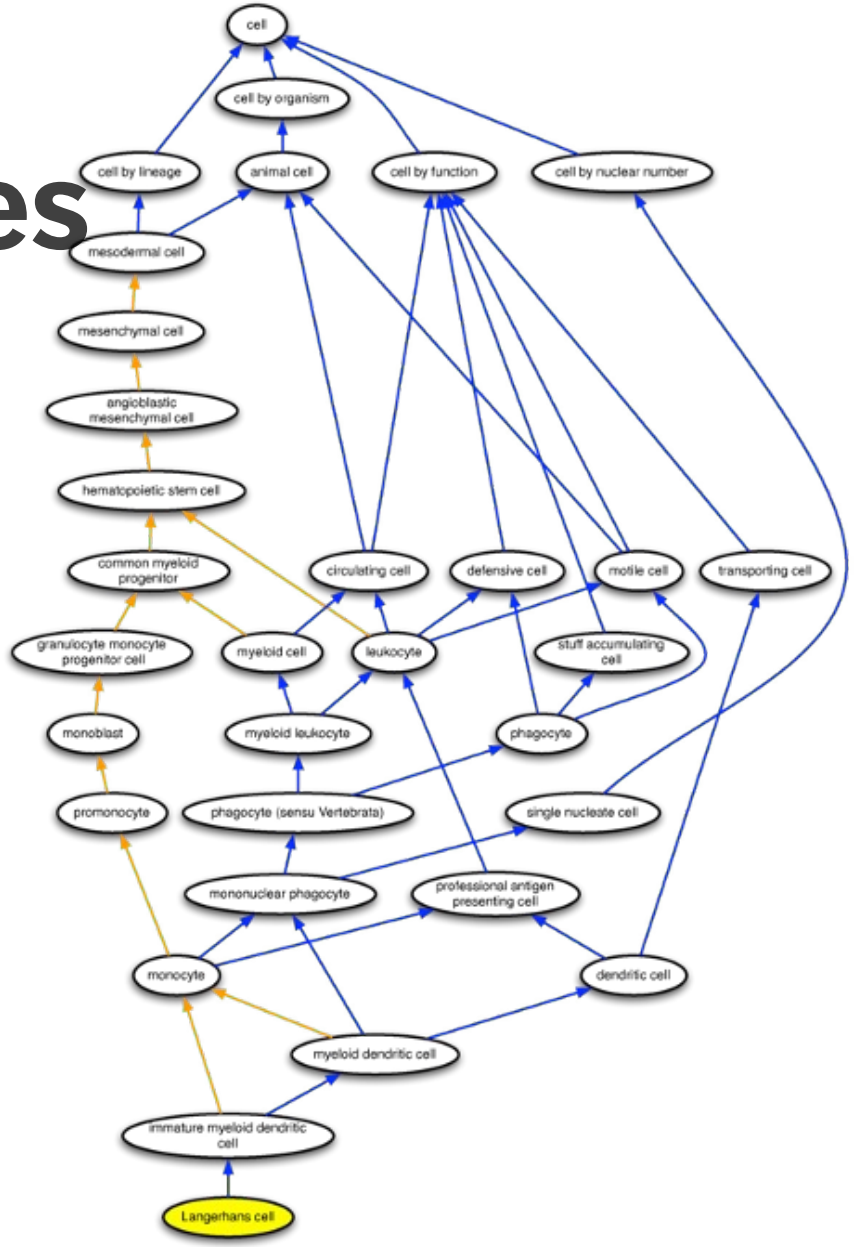
# Métadonnées contextuelles

Nous perdons quelque chose lorsque nous passons de notre modèle conceptuel à un modèle de type spécifique – p. ex. le modèle de données ou de connaissances.

Une façon de conserver le contexte est de fournir des **métadonnées** (riches, si possible) – des données sur nos données!

Les métadonnées sont essentielles lorsqu'il s'agit de mettre en œuvre des stratégies pour travailler d'un ensemble de données à l'autre.

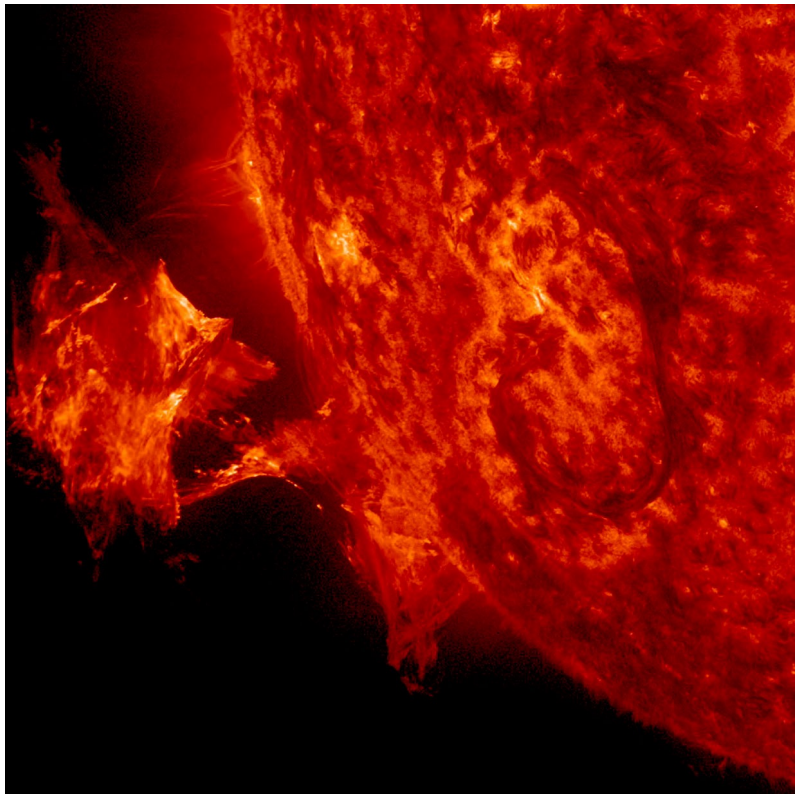
Les **ontologies** peuvent aussi jouer un rôle ici!



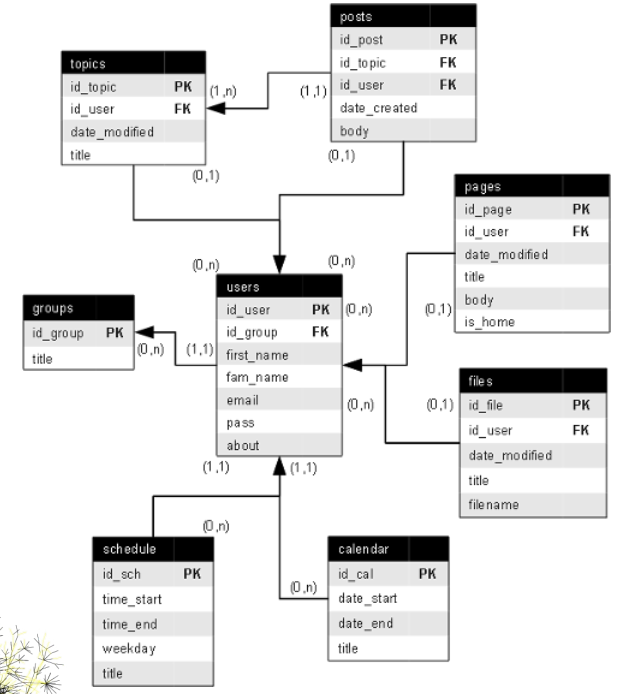
# Les données (non) structurées

La disponibilité croissante de données non structurées et de grands objets binaires (**blob)** est l'une des principales motivations de certains des nouveaux développements dans les types de bases de données et autres stratégies de stockage de données :

- **données structurées :** étiquetées, organisées, discrètes, selon une structure limitée et prédéfinie
- **données non structurées :** non organisées, pas de modèle de données à structure spécifique prédéfinie
- **données blob :** grand objet binaire – images, audio, multimédia

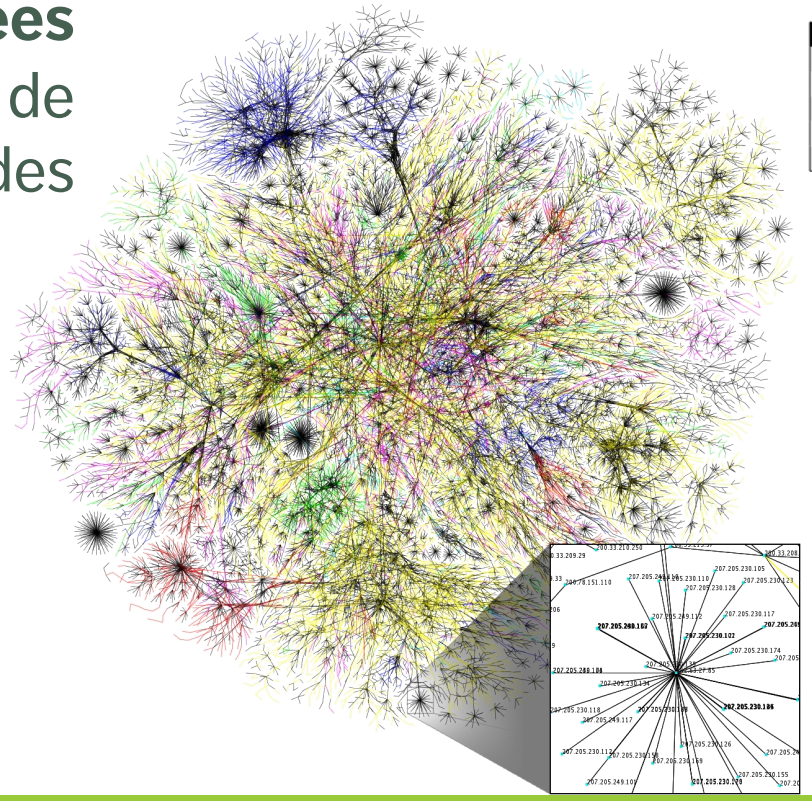


# La modélisation des données




Différentes options sont actuellement populaires en termes de **données** fondamentales et de stratégies de modélisation ou de structuration des **connaissances :**

- paires valeur-clé (e.g., JSON)
- triples (e.g., RDF)
- bases de données graphiques
- bases de données relationnelles
- feuilles de calcul



# Les mémoires et les bases de données

## Base de données relationnelle :

- largement soutenue, bien comprise, fonctionne bien pour de nombreux types de systèmes et de cas d'utilisation. Base toutefois difficile à changer une fois mise en œuvre; ne gère pas bien les liens.

## Magasins de clés-valeurs :

- peuvent prendre n'importe quel type de données; nul besoin de beaucoup de renseignements sur la structure ; si vous avez beaucoup de valeurs manquantes, ces mémoires ne prendront pas de place ; peuvent toutefois être désordonnées et mystérieuses; difficile d'y trouver des données.

## Bases de données graphiques :

- rapides et intuitives si vous utilisez des données fortement axées sur les liens; pourraient être la seule option si vos données sont ainsi parce que les bases de données traditionnelles peuvent ralentir énormément ; sont souvent trop spécialisées ; pas encore supportées à grande échelle.

# Les fichiers “plats” et les feuilles de calcul

## Pour :

- très efficace si vous recueillez des données une seule fois, sur un type particulier d’objet
- certains types d’analyse exigent que vous ayez toutes les données en un seul endroit
- facile à lire dans un logiciel et à effectuer des opérations sur l'ensemble des données

## Cons :

- très difficile de gérer l'intégrité des données si l'on collecte continuellement des données
- pas idéal pour les données de systèmes impliquant de multiples types d'objets et de relations
- il peut être très difficile d'effectuer des opérations d'interrogation de données

# Quelques outils et mots-clés

- MongoDB, ArangoDB
- Magasin de documents
- JSON, YAML
- API, GraphQL
- Données interreliées
- Web sémantique
- Langage d'ontologie Web (OWL)
- Protégé
- SQL, etc.

# La mise en œuvre du modèle

Pour mettre en œuvre votre modèle de données/connaissances, il faut avoir accès à un **logiciel de stockage et de gestion des données**.

Cela peut constituer un défi pour les particuliers : ces logiciels fonctionnent généralement sur des **serveurs**.

Les serveurs sont utiles car ils permettent à plusieurs utilisateurs d'accéder **simultanément** à une même base de données, à partir de différents programmes clients, mais il est difficile de "jouer" avec les données.

C'est là que **SQLite** entre en jeu.

# Le rôle du logiciel de gestion des données

Les logiciels de gestion des données offrent aux utilisateurs un moyen facile d'interagir avec leurs données.

Il s'agit essentiellement d'une interface entre les **personnes** et les **données**.

Grâce à cette interface, les utilisateurs peuvent :

- ajouter des données à leur collection de données
- extraire des sous-ensembles de données de leur collection en fonction de certains critères
- supprimer ou modifier des données dans leur collection

# Un peu de terminologie

**Auparavant :**

- base de données
- entrepôt de données
- mini-entrepôt de données
- système de gestion des données
- (SQL)

**Maintenant :**

- lac de données
- bassin de données
- marais de données ?
- cimetière de données ?
- (NoSQL)

De plus en plus : on fait une distinction entre l'**entrepot de données** et le **logiciel de gestion des données**.

# Du modèle de données à la mise en œuvre

Une fois que le mode de données (logique) est achevé :

1. **instancier le modèle** dans le logiciel choisi (par exemple, créer des tables dans MySQL)
2. **télécharger/charger les données**
3. **interroger les données :**
   - les bases de données relationnelles traditionnelles utilisent le **langage de requête structuré** (SQL : Structured Query Language)
   - d'autres utilisent des langages de requête différents (AQL, moteurs sémantiques, etc.) ou s'appuient sur des programmes informatiques sur mesure (par exemple, écrits en R, Python)

# La gestion des bases de données

Une fois les données collectées, il faut aussi les **gérer**.

Fondamentalement, cela signifie que la base de données doit être **maintenue**, afin que les données soient

- **précises**
- **exactes**
- **cohérentes**
- **complètes**

Ne laissez pas votre lac de données se transformer en marais de données !

# Sevvices en nuage (Cloud Services)

1. Stocker de **grandes** quantités de données
2. Exécutez des processus coûteux et avancés en **cliquant sur un bouton**
3. **Flexible** et **évolutif**
4. Permettre le traitement des données **en code bas**

# Nuage vs. accès local

**Nuage (Cloud)**

- sans intervention manuelle
- paiement à la consommation
- propriétaire douteux

**Accès local (On-Premise)**

- auto-entretenu
- tous les coûts sont absorbés
- sécurité entièrement contrôlée

# Lectures suggérées

La gestion des données

*Data Understanding, Data Analysis, Data Science*

**Data Science Basics**

Getting Insight From Data

- Structuring and Organizing Data

**Data Engineering and Management**

Data Management

- Databases
- Data Modeling
- Data Storage

Reporting and Deployment

- Reports and Products
- Cloud and On-Premise Architecture

# Exercices

La gestion des données

1. Votre organisation possède-t-elle des données ? Si oui, sont-elles hébergées localement ou sur le cloud ? Comment y accède-t-on ? Comment sont-elles structurées ?

2. Complétez tous les exercices précédents que vous n'avez pas eu l'occasion de terminer.