UNIVERSITE BADJI MOKHTAR-ANNABA

DEPARTEMENT D'INFORMATIQUE
M2 SID

Structuration et Manipulation des Bases de Données Complexes

par

Dr. Samira LAGRINI

Année universitaire:202/202

Introduction

- Le **web scraping** est une technique puissante permettant d'extraire des données de diverses sources en ligne, généralement sous des formats structurés tels que CSV, JSON ou XML.
- Ces formats facilitent une analyse initiale des données.
- Lorsque les informations extraites deviennent massives ou incluent des liens entre différentes entités (ex. des utilisateurs, des produits, des commentaires), des hiérarchies, ou encore des dépendances temporelles, **Ces formats simples deviennent insuffisants!!!!**

Introduction

Alors, utiliser **les bases de données relationnelles** pour stocker ces données complexes.

Bases de données relationnelles

- Les bases de données relationnelles permettent de structurer les données sous forme de tables interconnectées par des relations bien définies.
- Chaque table représente une entité spécifique (utilisateurs, produits, commentaires, etc.)
- Les relations entre ces entités sont gérées par des clés primaires et clés étrangères

✓**Clé primaire** : Une ou plusieurs colonne(s) qui identifie de manière unique chaque ligne dans une table.

✓**Clé étrangère** : Une colonne dans une table qui fait référence à la clé primaire d'une autre table, établissant une relation entre ces deux tables.

Bases de données relationnelles

Clé primaire

Table “Commande”

Clé étrangère

Id commande	Produit	Prix	Id client	Livraison ID
100	PC 16’	599	1	200
101	Souris	29	2	201
102	Lampe de bureau	30	2	202
103	Clavier	55	2	203
104	Bureau	300	3	204
105	Chaise Gaming	159	3	205
106	PC Bureau	899	1	206

Clé primaire

Table “Client”

Id client	Prenom	Nom	Adresse
1	Jean	Bonneau	50 av St Marc
2	Ambre	Cerna	13 Grande Rue
3	Jack	Jakson	55 Mac street

Clé Primaire

Table “Livraison”

Livraison ID	Livreur	Numéro de colis
200	Colissimo	22223958
201	DHL	44490009
202	DHL	22009877
203	Chronopost	33466771
204	FedEx	4499008
205	FedEx	557774449
206	Colissimo	112440

Bases de données relationnelles

- La gestion des bases de données relationnelles est assurée par des systèmes de gestion de bases de données (SGBD) tels que:
  - ✓ MySQL,
  - ✓ PostgreSQL
  - ✓ Oracle.
- Ces systèmes permettent l'exécution de requêtes complexes et garantissent la cohérence, l'intégrité et la sécurité des données.

Clé primaire — Table "Commande" — Clé étrangère

Id commande	Produit	Prix	Id client	Livraison ID
100	PC 16'	599	1	200
101	Souris	29	2	201
102	Lampe de bureau	30	2	202
103	Clavier	55	2	203
104	Bureau	300	3	204
105	Chaise Gaming	159	3	205
106	PC Bureau	899	1	206

Clé primaire — Table "Client"

Id client	Prenom	Nom	Adresse
1	Jean	Bonneau	50 av St Marc
2	Ambre	Cerna	13 Grande Rue
3	Jack	Jakson	55 Mac street

Clé Primaire — Table "Livraison"

Livraison ID	Livreur	Numéro de colis
200	Colissimo	22223958
201	DHL	44490009
202	DHL	22009877
203	Chronopost	33466771
204	FedEx	4499008
205	FedEx	557774449
206	Colissimo	112440

Limites des bases de données relationnelles dans le cadre du web scraping

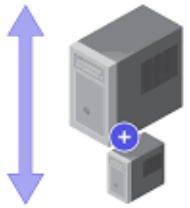
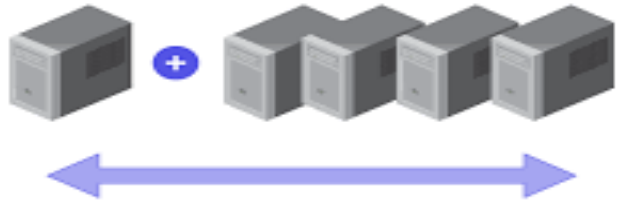
- **Scalabilité limitée**
  - Difficultés à gérer de grands volumes de données.
- **Rigidité du schéma**
  - Schéma des données défini à l'avance, rendant difficile l'adaptation aux données non structurées ou semi-structurées, fréquemment rencontrées lors du web scraping.
- **Faible capacité de modélisation des relations complexes**
  - Difficulté à gérer les relations interconnectées, comme dans les réseaux sociaux ou systèmes de recommandation.
- **Performance réduite avec les données non structurées**
  - Problèmes de stockage et d'analyse des données non structurées (images, flux en temps réel)ou semi-structurées.

Bases de données non relationnelles (NoSQL)

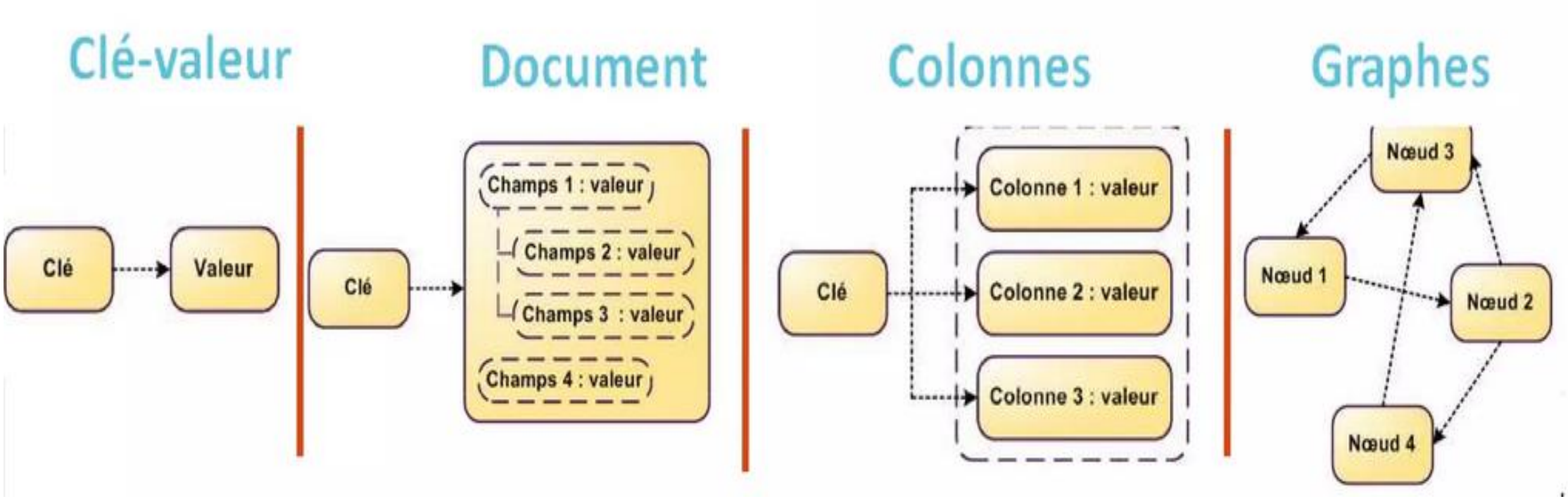
Les BDD NoSQL (Not Only SQL):

- Solution aux limitations des bases relationnelle
- Conçues pour gérer données structurées, semi-structurées et non structurées.
- Ne suivent pas un schéma rigide (plutôt dynamique) et offrent une flexibilité pour modéliser des relations complexes
- Adaptées aux volumes de données massifs

Principales différences entre BDD SQL et NoSQL

Critère	SQL (Bases de données relationnelles)	NoSQL (Bases de données non relationnelles)
structuration de données	Structuré, données organisées en tables interconnectées,	Variée: stockent les données sous forme de documents, de paires clé-valeur ou de graphiques
Schéma	Schéma fixe (défini à l'avance).	Schéma dynamique ou sans schéma.
Langage de requête	Utilise SQL (Structured Query Language) pour les requêtes.	Utilise des langages spécifiques à chaque type de BDD NoSQL.
Scalabilité	**Scalabilité verticale** Ajouter de la puissance à un serveur unique (plus de RAM, de CPU, etc.) pour accroître ses capacités de stockage. 	**Scalabilité horizontale** (Ajouter des serveurs Supplémentaires) en cas de données massives. 
Types de données	Bien adapté aux données structurées	Adapté aux données structurées, non structurées ou semi-structurées (documents JSON, fichiers, données multimédia) et fortement interconnectées.
Performance	Moins performant avec de très grands volumes de données ou des relations complexes.	Très performant pour gérer de grands volumes et des données distribuées.
Flexibilité	Moins flexible : les changements de schéma nécessitent souvent des migrations complexes.	Plus flexible : adaptabilité facile aux données variables ou modifications fréquentes du schéma.

Types de Bases de Données NoSQL



Bases de Données Clé-Valeur

- Basique, les données sont **représentées** par un **couple clé/valeur**
- La valeur peut être une **simple chaîne de caractères**, ou un objet complexe ou tout autre type de données.

Clé	Valeur
nom-auteur1	Dupont
prenom-auteur1	Jean
nom-auteur2	Martin
prenom-auteur2	Anne
nom-auteur3	Smith
prenom-auteur3	Paul
titre-livre392649	Bases de données NoSQL
prix-livre392649	15
nom-livre392649-auteur1	Dupont
…	…

Cas d'Utilisation :

- Stockage des sessions utilisateur dans une application web.
- les profils, préférences d'utilisateur,
- les données de panier d'achat
- …..

Exploitation

Leur exploitation est basée sur 4 opérations (CRUD):

- C reate : créer un nouvel objet avec sa clé → create(key, value)
- R ead : lit un objet à partir de sa clé → read(key)
- U pdate : met à jour la valeur d'un objet à partir de sa clé →Update (key, value)
- D elete: supprime un objet à partir de sa clé → delete(key)

Bases de Données Clé-Valeur

- Exemple d'implémentation : Redis, Riak, DynamoDB (d'Amazon)

❖Avantages

✓ Performances très élevées en lecture et en écriture

✓Une scalabilité horizontale considérable

❖Inconvénients

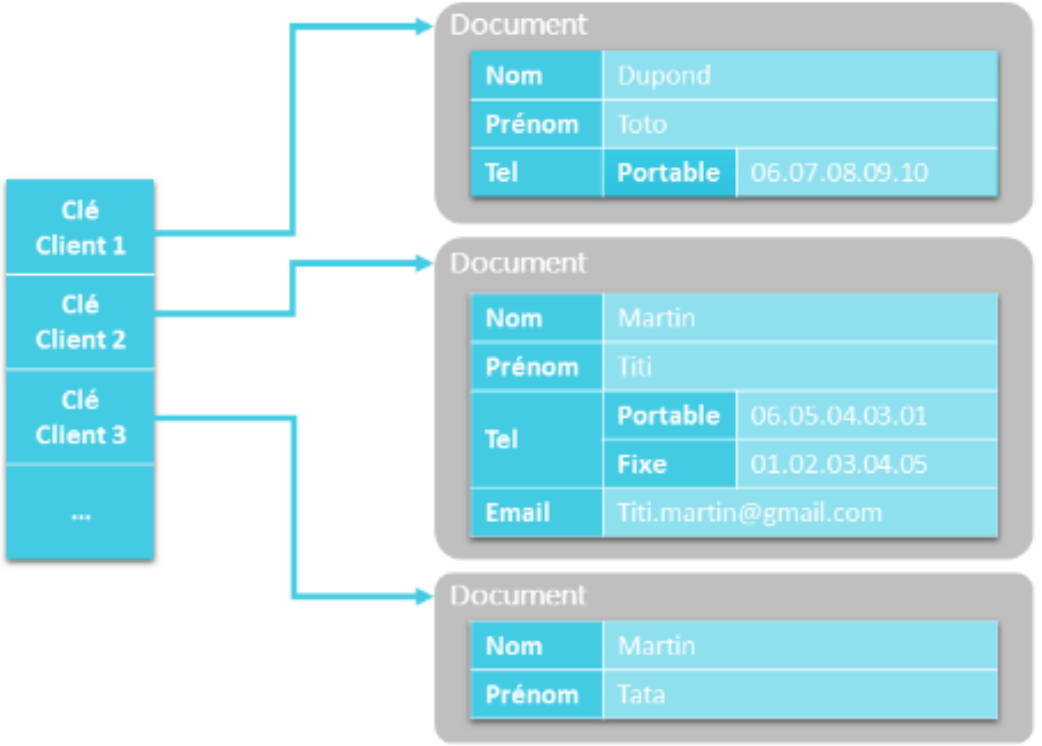
✓ pauvre pour les données complexes

✓ Interrogation seulement sur clé

✓ Limité pour les requêtes complexes et les relations entre données.

Bases de Données Orientées Document

- ❑ Stockent les données sous forme de documents (**JSON**, **BSON** (Binary JSON), ou **XML)**.
- Elles sont basées sur le modèle « clé-valeur » mais la valeur est un document en format semi-structuré (**JSON** ou **XML,..)**
- Un document est composé de champs et des valeurs associées
- Il n'est pas nécessaire de définir au préalable les champs utilisés dans un document.



Exemple de Systèmes : MongoDB, CouchDB, RavenDB, Couchbase,

Bases de Données Orientées Document

Cas d'Utilisation

- Gestion de contenu web (CMS) où chaque document représente un article avec des champs (titre, auteur, contenu, tags).
- Applications de réseaux sociaux où les profils d'utilisateurs ont des champs variés.

Avantages

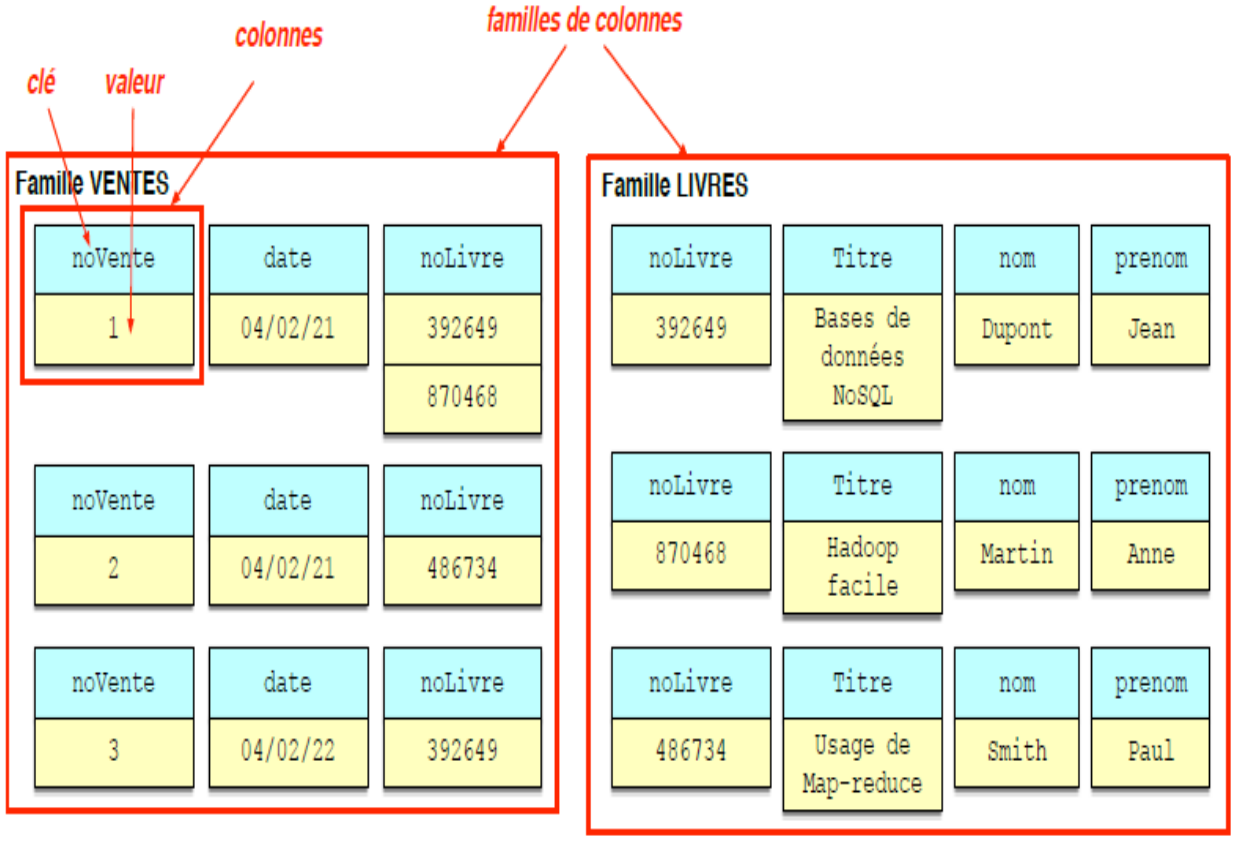
- Flexibilité des documents sans schéma fixe.

Inconvénients

- Moins efficace pour les opérations complexes de jointure.

Bases de Données Orientées Colonne

- ❑ Stockent les données dans des colonnes au lieu de lignes.
- ❑ Les colonnes peuvent varier en nombre et en nom d'une ligne à l'autre, et peuvent changer dans le temps.
- ❑ Il n'y a pas d'espace mémoire consommé par des valeurs NULL, contrairement aux cas des BdD relationnelles



Les **principaux concepts** associés :

- **Colonne :** entité de base représentant un **champ de donnée** Chaque colonne est **définie par un couple clé / valeur**

Famille de colonnes : permet de **regrouper plusieurs colonnes**

Bases de Données Orientées Colonne

Utilisations Principales

- Idéales pour l'**analyse en temps réel** de données massives,
- Parfaites pour stocker des **données chronologiques** issues de capteurs.
- Utilisées pour gérer les **catalogues de produits**, les stocks et les prix dans des systèmes à grande échelle avec des données distribuées.
- Adaptées pour le **suivi des appels** et la gestion des données des utilisateurs dans les réseaux de télécommunication.
- ….

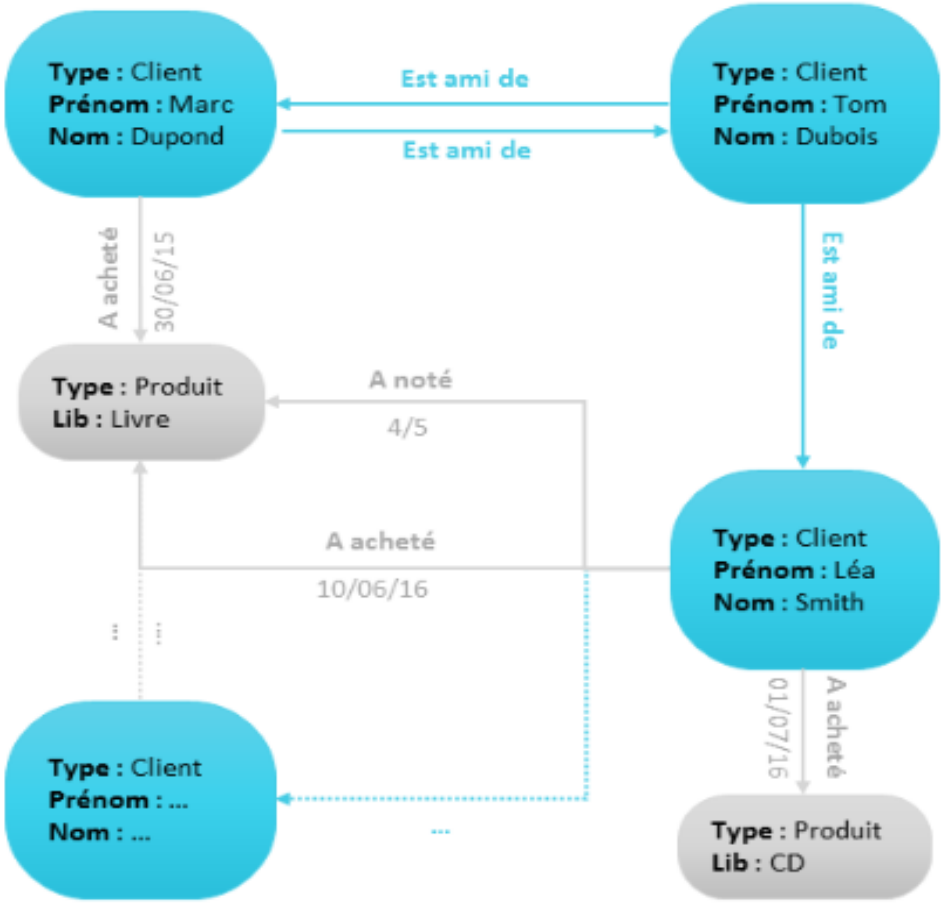
Implémentations les plus connues

- **SimpleDB** de Amazon, Cassandra, HBase

Bases de Données Orientées Graphe

- Stockent des données sous forme de graphe (des nœuds pour les entités(ex. une personne, un produit, une ville) et des arcs pour les relations(ex. "AMI DE", "ACHÈTE", "SUIT")).
- Chaque nœud contient des propriétés (ou attributs), comme le nom, l'âge, ou la catégorie.
- Les arêtes peuvent également avoir des propriétés, comme la date ou l'intensité de la relation.

Systèmes : Neo4j, OrientDB,



Bases de Données Orientées Graphe

Les bases de données orientées graphes sont conçues pour stocker et gérer des données qui sont fortement interconnectées:

- **Réseaux sociaux**

  Stockage des utilisateurs, des relations (amis, abonnés), et des interactions (commentaires, likes).

- **Systèmes de recommandation**

  Stockage des connexions entre utilisateurs et produits pour fournir des recommandations basées sur les relations d'intérêt.

- **Gestion des identités et accès**

Stockage des relations entre utilisateurs et ressources, permettant de gérer les permissions d'accès à différents systèmes.

Bases de Données Orientées Graphe

Avantages

- ❑ Modélisation Naturelle des Relations Complexes
- ❑ Flexibilité du Schéma
  - Pas de schéma rigide, ce qui permet d'ajouter de nouveaux types de nœuds ou de relations sans modifier la structure existante.
  - Bien adaptées aux données hétérogènes et en évolution
- ❑ Adaptées aux Données Interconnectées

Inconvénients

- Nécessitent souvent une infrastructure spécialisée et peuvent être plus coûteuses à déployer et à maintenir par rapport aux bases de données relationnelles.

Comparaison entre les Types de BDD NoSQL

Type de Base de Données NoSQL	Structure des Données	Avantages	Cas d'Utilisation	Limitations
Clé-Valeur (Key-Value)	Paires clé-valeur	Rapide, simple, scalable	Caching, gestion de sessions, stockage temporaire	Pas adapté aux données complexes ou relations
Document (Document Stores)	Documents JSON ou BSON	Flexible, supporte les données semi-structurées	Applications web, **Content Management System (**CMS),	Moins efficace pour les jointures complexes
Colonne (Column Stores)	Colonnes regroupées dans des familles	Hautes performances pour les écritures/lectures massives	Big data, logs, systèmes de recommandation	Pas adapté aux requêtes complexes ou relations
Graphe (Graph Databases)	Nœuds, relations, propriétés	Idéal pour les relations complexes et interconnectées	Réseaux sociaux, détection de fraude, recommandations	Moins efficace pour des données non liées

Manipulation des Bases de Données Complexes

- La manipulation des données dans ces systèmes inclut un ensemble d'opérations :
- **Insertion**: ajouter de nouvelles données dans la base de données.
- **Mise à jour:** modifier des données existantes dans la base de données.
- **Suppression**: retirer des données de la base
- **interrogation des données:** récupérer des informations stockées dans la base de données en fonction de critères spécifiques

Conclusion

- Les bases NoSQL offrent une flexibilité accrue pour gérer des données hétérogènes et semi-structurées.
- Le choix entre ses types de bases dépend également des besoins d'évolutivité et des caractéristiques des données
- Les bases de données MongoDB et Neo4j sont particulièrement adaptées aux données extraites via le web scraping, qui sont souvent non structurées et nécessitent une modélisation flexible des relations complexes

Travaux pratiques

TP

- ❑ Scraper des informations à partir d'un site web de votre choix (par exemple, un site d'e-commerce pour collecter des informations sur les produits(nom du produit, prix, catégorie, note et commentaires des utilisateurs).
- ❑ Nettoyer et préparer les données extraites.
- ❑ Structurer et stocker les données dans une base de données NoSQL (MongoDB)(Installez **MongoDB** sur votre machine ainsi que de la bibliothèque **PyMongo** pour interagir avec la base de données.
- ❑Réaliser des requêtes simples pour manipuler et analyser les données stockées.
  - •Trouver tous les produits dont le prix est supérieur à 50 $.
  - •Compter le nombre de produits par catégorie.
  - •……


Merci de
votre
attention