Nouveaux paradigmes de base de données
BD5 (NoSQL)

IUT La Rochelle
BUT INFO3 Développeur IA

M. Sakkari

Plan

1. **Les bases de Données Orientées Document**
2. **Les bases de Données Orientées Graphe.**
3. **Les bases de Données Orientées Clé-Valeur**
4. **Les bases de Données Orientées colonnes**

Objectifs

A l'issue de cette formation l'étudiant connaîtra les principales solutions NoSQL : leur typologie, leurs possibilités et leurs limites. Il saura évaluer l'opportunité de mettre en œuvre une solution NoSQL dans ses projets. Il connaîtra les solutions permettant de traiter de forts volumes de données.

Evaluation

1. **TPs notés**
2. **Learning Paths**
3. **Mini-projets**
4. **Examen pratique de fin de formation**

Les bases de Données Orientées Document

I. Le NoSQL

- Le terme « NoSQL » désigne les différents types de bases de données non relationnelles.
- Les bases de données NoSQL, qui signifie « not only SQL » sont des types de base de données qui ont commencé à émerger depuis 2009, pour répondre **aux nouveaux besoins** (big Data).
- Le nom peut sembler comme une opposition aux bases de données SQL, sa fonction première n'étant pas de remplacer les bases de données relationnelles, mais de proposer une alternative.

I. Pourquoi les SGBDR ne Suffisent Plus ?

1. Ajouter plus de puissance à un seul serveur (vertical scaling) devient de moins en moins efficace et plus coûteux.

I. Pourquoi les SGBDR ne Suffisent Plus ?

2. Les SGBDR ne sont pas conçus pour être facilement répartis sur plusieurs serveurs (Déploiement Horizontal /Horizontal scaling).

I. Pourquoi les SGBDR ne Suffisent Plus ?

3. Modifications de schéma complexes et coûteuses en temps, en particulier pour les applications dynamiques.
4. Difficulté à gérer des données qui ne rentrent pas dans un modèle tabulaire.

Rappel :

Données structurées : Les données structurées sont des informations qui ont été formatées et transformées en un modèle de données bien défini

➔ Les exemples typiques de données structurées sont les noms, les adresses, les numéros de carte de crédit, etc.

I. Pourquoi les SGBDR ne Suffisent Plus ?

Rappel :

Les données semi-structurées sont des données qui ne sont pas conformes aux normes des données structurées classiques(qui ne résident pas dans une base de données rationnelle), mais qui possèdent des propriétés organisationnelles facilitant leur analyse

➔ Un fichier de données client délimité par des tabulations

Données non structurées : Les données non structurées sont des données qui manquent de structure ou d'architecture identifiable.

➔ pages Web, vidéos, ...

I. Pourquoi les SGBDR ne Suffisent Plus ?

5. Les opérations complexes telles que les jointures peuvent devenir très coûteuses en termes de performance avec des volumes de données massifs.

```
--BD5 exemple :compter les amis de 3ème niveau
select u1.nom,count(*) from user as u1
inner join user_friends as uf1 on u1.id = uf1.user1
inner join user_friends as uf2 on uf1.user2 = uf2.user1
inner join user_friends as uf3 on uf2.user2 = uf3.user1
group by u1.nom
```

I. Pourquoi les SGBDR ne Suffisent Plus ?

6. **Transactions ACID** : Si elles garantissent la cohérence, elles peuvent devenir un goulot d'étranglement dans des environnements très concurrents.

   ACID est un ensemble de propriétés garanties par les bases de données relationnelles pour assurer la fiabilité des transactions. Cela inclut :

   Atomicité: Chaque transaction est tout ou rien, Cohérence: Toute transaction valide amène la base de données d'un état valide à un autre, Isolation: Les transactions concurrentes ne s'interfèrent pas mutuellement et Durabilité: Une fois une transaction validée, elle reste enregistrée même en cas de panne.

I. Pourquoi les SGBDR ne Suffisent Plus ?

6. **Transactions ACID :** Dans un environnement où de nombreuses transactions sont effectuées simultanément, garantir l'isolation (I de ACID) peut ralentir les performances, car les transactions doivent attendre que d'autres transactions se terminent pour garantir que les données restent cohérentes.

   Exemple : Un site e-commerce lors d'une vente flash où des milliers de clients essaient d'acheter le même produit en même temps.

   → Chaque commande de produit doit vérifier la disponibilité du stock, réserver l'article, puis enregistrer la transaction

I. Pourquoi les SGBDR ne Suffisent Plus ?

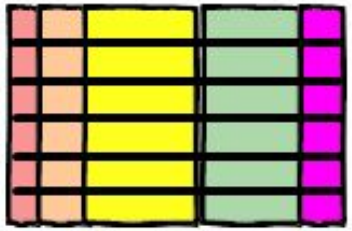
→ Isolation : Pour éviter que deux clients n'achètent le même article en même temps, le système doit verrouiller l'enregistrement du stock jusqu'à ce que la transaction soit terminée. Pendant ce temps, d'autres transactions doivent attendre

→ Résultat : Si beaucoup de clients passent des commandes simultanément, cela peut créer un goulot d'étranglement, car le système ne peut traiter qu'un nombre limité de transactions à la fois pour garantir la cohérence des données. Les clients peuvent alors rencontrer des délais, voire des échecs de transaction.

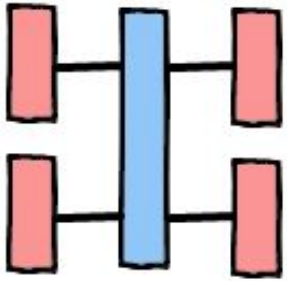
II. L'Émergence de NoSQL

1. Scalabilité Horizontale : Conçues pour être réparties sur des clusters de serveurs, répondant aux besoins de Big Data
2. Flexibilité du Schéma : Capacité à gérer des données non structurées et à évoluer avec les besoins de l'application.
3. Modèles de Données Diversifiés : Clé-valeur, document, colonne, graphe, adaptés à des cas d'utilisation spécifiques.
4. Adoption Croissante : Utilisation par des entreprises nécessitant de gérer des volumes de données massifs, comme Facebook, Netflix, ...

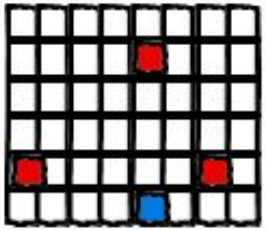
Relationnelle



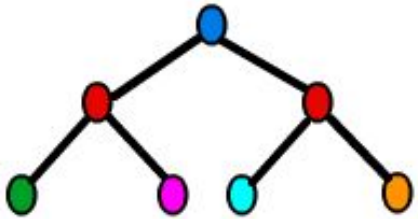
OLAP



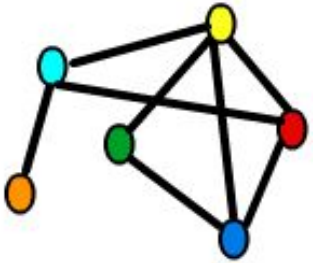
Orienté colonne



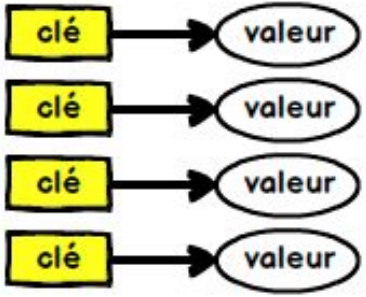
Document



Graphe



Clé-Valeur



III. **Les bases de données orientée document**

- Une base de données orientée document est un type de base de données NoSQL qui stocke des données sous forme de documents, généralement en formats JSON, BSON, ou XML.
  → BSON (Binary JSON) est une représentation binaire de JSON. Il est utilisé principalement par MongoDB pour stocker les documents de manière plus efficace en termes de performance et d'espace

III. Les bases de données orientée document

Structure des Données :

- **Documents :** Les données sont stockées dans des documents qui contiennent des paires clé-valeur. Chaque document est une entité autonome, similaire à un objet ou un enregistrement.
- **Collections** : Les documents sont groupés en collections, l'équivalent des tables dans les bases de données relationnelle

III. Les bases de données orientée document

Caractéristiques Principales :

- **Flexibilité du Schéma** : Les documents au sein d'une même collection peuvent avoir des structures différentes, ce qui permet une grande flexibilité.
- **Indexation** : Les documents peuvent être indexés sur des champs spécifiques pour améliorer les performances des requêtes.

Exemples de SGBD Documentaires : MongoDB, CouchDB, Firebase.

III. Les bases de données orientée document

Exemple Document 1 dans une Collection users

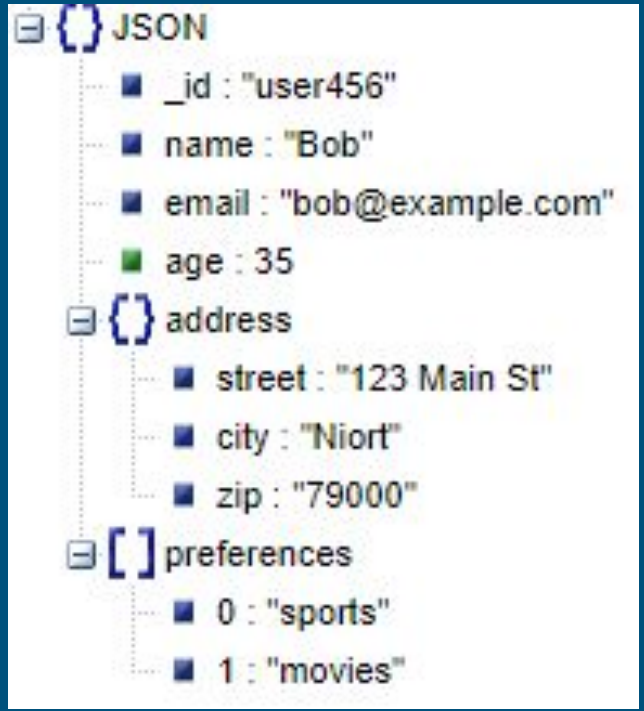
```
{
  "_id": "user123",
  "name": "Alice",
  "email": "alice@example.com",
  "age": 28
}
```

III. Les bases de données orientée document

Exemple Document 2 dans la Collection users

```
{
        "_id": "user456",
        "name": "Bob",
        "email": "bob@example.com",
        "age": 35,

        "address": {
                "street": "123 Main St",
                "city": "Niort",
                "zip": "79000"
                        },
        "preferences": ["sports", "movies"]
}
```



III. Les bases de données orientée document

1. Les différentes terminologies et concepts SQL ainsi que la terminologie et les concepts MongoDB correspondants.

Le SQL	Le MongoDB
Table	Collection
Ligne	document ou document BSON
Colonne	Field
index	index
Jointure	$lookup, embedded documents
clé primaire	clé primaire (_id)
aggregation	aggregation pipeline
transactions	transactions

2. Les exécutables de base de données et les exécutables MongoDB correspondants.

	MongoDB	MySQL	Oracle	PostGreSQL
server	mongod	mysql	oracle	PostGreSQL
client	mongosh	mysql	sqlplus	PGAdmin

3. Termes, fonctions et concepts d'agrégation SQL et opérateurs d'agrégation MongoDB correspondants

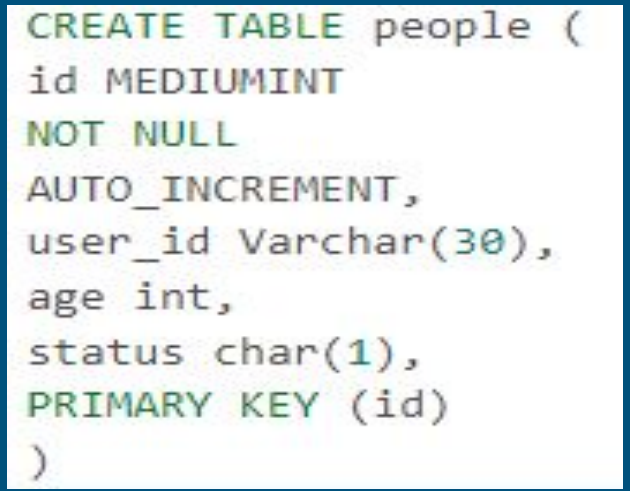
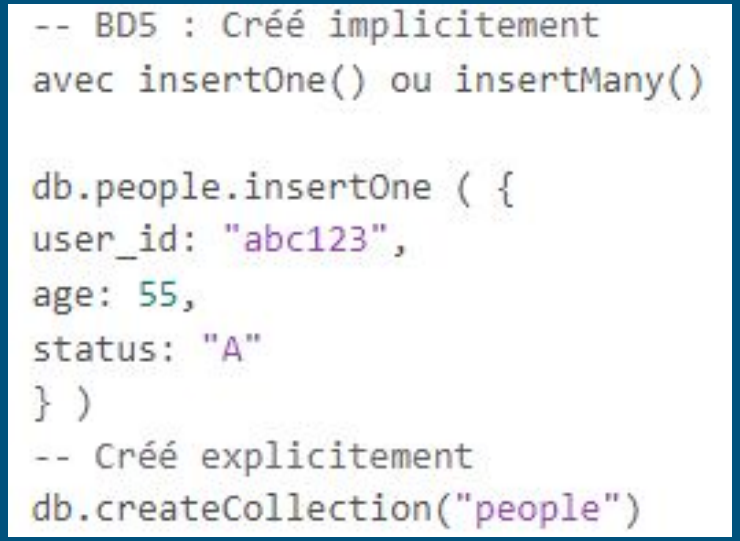
Le SQL	Le MongoDB
WHERE	$match
GROUP BY	$group
HAVING	$match
SELECT	$project
ORDER BY	$sort
LIMIT	$limit
SUM()	$sum
COUNT()	$sum, $sortByCount

3. Termes, fonctions et concepts d'agrégation SQL et opérateurs d'agrégation MongoDB correspondants

Le SQL	Le MongoDB
join	$lookup
SELECT INTO NEW_TABLE	$out
MERGE INTO TABLE	$merge
UNION ALL	$unionWith

Pour les commandes MongoDB, consultez la documentation officielle, pas ChatGPT.
https://www.mongodb.com/docs/manual/reference/command/

4. Create

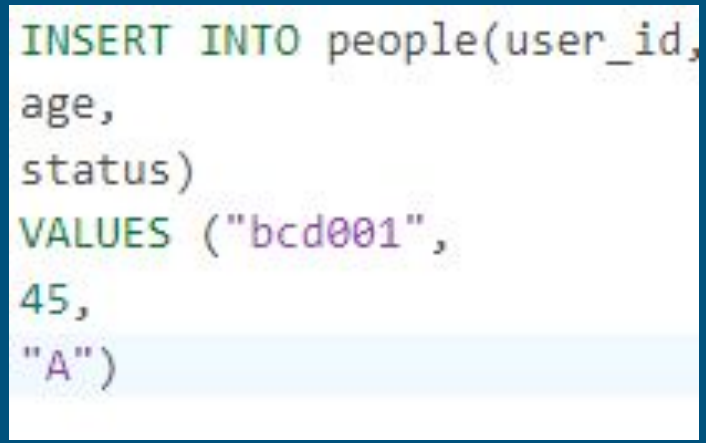
Le SQL	Le MongoDB
	

5. Alter

Le SQL	Le MongoDB
ALTER TABLE people ADD join_date DATETIME	db.people.updateMany ({ }, { $set: { join_date: new Date() } })

- Les collections ne décrivent ni n'imposent la structure de leurs documents ; c'est-à-dire qu'il n'y a aucune modification structurelle au niveau de la collection.
- Lles opérations updateMany() peuvent ajouter des champs aux documents existants à l'aide de l'opérateur $set.

6. Insert

Le SQL	Le MongoDB
	À vous de jouer !

7. Select

Le SQL	Le MongoDB
• SELECT * FROM people • SELECT * FROM people WHERE status = "A" • SELECT * FROM people WHERE status != "A" • SELECT user_id, status FROM people WHERE status = "A" • SELECT * FROM people WHERE status = "A" OR age = 50	À vous de jouer !

8. Exercice

Soit la base de données suivante permettant de représenter des articles en général et illustrée avec l'article As we may think de Vannevar Bush publié en 1945.

```
CREATE TABLE article (
  id INTEGER PRIMARY KEY,
  title VARCHAR(255) UNIQUE NOT NULL,
  publication DATE,
  licence TEXT
);

CREATE TABLE page (
  number INTEGER,
  article INTEGER NOT NULL REFERENCES article(id),
  title VARCHAR(255) NOT NULL,
  PRIMARY KEY (number, article)
);
```

8. Exercice

```
CREATE TABLE paragraph (
  id INTEGER PRIMARY KEY,
  page INTEGER NOT NULL,
  article INTEGER NOT NULL,
  content TEXT,
  FOREIGN KEY (page, article) REFERENCES page(number, article)
);
INSERT INTO article VALUES (1, 'As we may think', '1945-07-01', 'Copyheart');
INSERT INTO page VALUES (1, 1, 'Introduction');
INSERT INTO page VALUES (2, 1, 'Of what lasting benefit has been man''s use of science');
INSERT INTO paragraph VALUES (1, 1, 1, 'This has not been a scientist''s war; it has been a war in which all have had a part');
INSERT INTO paragraph VALUES (2, 1, 1, 'For the biologists, and particularly for the medical scientists, there can be little indecision');
```

8. Exercice

1. Question 1 : Le relationnel est il adapté à la représentation de document ? Expliquez.
2. Question 2 : Proposer une représentation imbriquée en JSON de cet article.
3. Question 3 : Quel est le type de moteur Nosql le plus adapté dans ce cas ?

Fin

Merci Pour Votre Attention