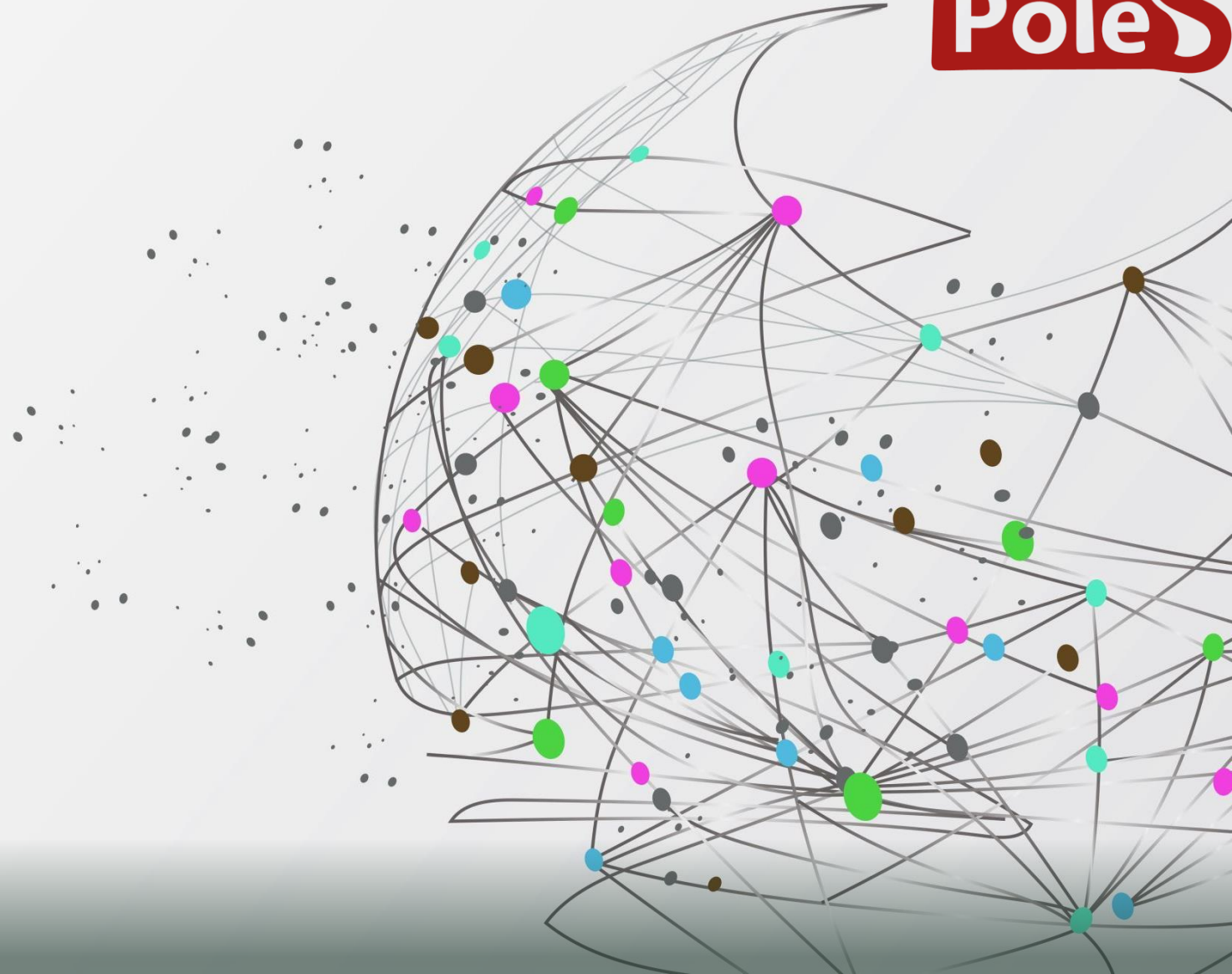
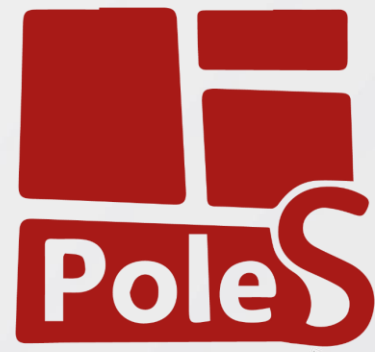




MongoDB/NoSQL
MySQL/SQL



MongoDB c'est quoi ?

MySQL c'est quoi ?



MONGODB

- SGBD (Système de gestion de bases de données)
- Bases de données non relationnelles (pas d'interactions entre-elles)
- Utilise des fichiers Bson (Binary Json) pour stocker les informations de la BDD (les données sont traitées en JSON et converties en BSON une fois sur le serveur MongoDB)
- Utilise des serveurs MongoDB pour stocker la BDD

• MYSQL

- SGBD (Système de gestion de bases de données)
- Bases de données relationnelles
- Utilise des tables relationnelles structurées pour stocker les informations de la BDD
- Utilise des serveurs MySQL pour stocker la BDD



Pourquoi utiliser MySQL



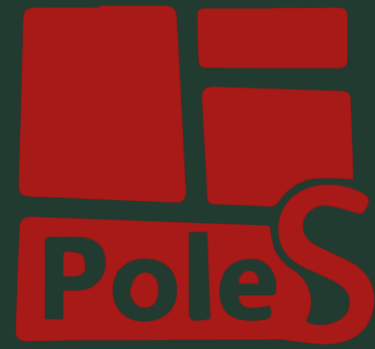
MySQL va permettre de gérer les relations entre toutes les entités utiles de notre application.

Prenons le cas d'un e-commerce :

- Relations entre entités : MySQL permet de gérer les relations entre les clients, leurs commandes, les produits associés à chaque commande.
- Gestion des transactions : Il nous assure le bon déroulement des transactions en garantissant que les paiements soient correctement enregistrés.
- Gestion des stocks : Lorsqu'une commande est passée, MySQL peut mettre à jour le stock des produits en temps réel, ce qui nous évite les erreurs de vente lorsque le stock est vide.



Pourquoi utiliser MongoDB

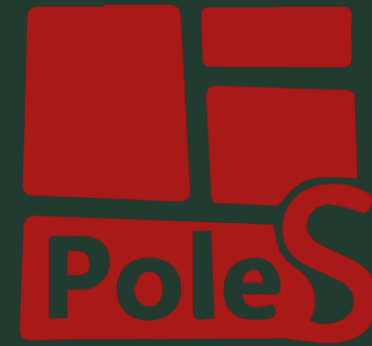


MongoDB va permettre de gérer de grandes quantités de données qui n'ont pas de liens entre elles

Prenons le cas d'un reseau social :

- Flexibilité des données : MongoDB va nous permettre de stocker des données variées sous forme de documents JSON. Idéal pour les profils où les informations peuvent varier d'une personne à l'autre (photos, posts, amis)
- Gestion des publications : Les posts, commentaires, réactions peuvent être stockés dans des documents distincts, ce qui facilite la gestion et la récupération de ces données.
- Performances/rapidité : MongoDB permet de gérer des volumes élevés en lectures et écritures des fichiers JSON, ce qui est pratique pour les interactions en temps réel comme les notifications ou les flux de publications sur les reseaux (type twitter tiktok ou encore instagram)

Attention !



On peut très bien faire un reseau social en SQL et un e-commerce en NoSQL selon les besoins spécifiques de notre application, les avantages et inconvénients diffèrent cependant :

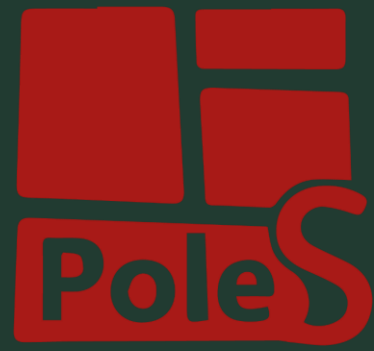
Avantage d'un reseau social en SQL : Une base de données relationnelle telle que MySQL est excellente pour garantir les bonnes relations entre les tables des utilisateurs, des amis, des abonnés, des groupes etc...

Inconvénient d'un reseau social en SQL : Les reseaux sociaux génèrent de grandes quantités de données de requêtes en temps réel, ce qui peut poser des problèmes de performances avec des bases de données relationnelles

Avantages d'un e-commerce en NoSQL : Bien adapté pour stocker des données non relationnelles comme le details des produits, le panier d'un acheteur etc.. De plus il peut gérer de grandes quantités de données avec des millions de produits et de transactions.

Inconvénient d'un e-commerce en NoSQL : MongoDB offre un support pour les transactions multi-documents (qui ne sont pas reliés entre eux), ce qui peut poser des problèmes lors de transactions complexes par exemple ou même encore de mises à jour de stock

Bases de données Hybrides



Absolument rien de vous empêche de créer votre application et profiter des avantages des deux SGBD, vous pouvez très bien utiliser MySQL pour les données nécessitant des interactions entre elles telles que les relations entre différents utilisateurs pour un réseau social et MongoDB pour des données n'en nécessitant pas comme la gestion des médias (photos, vidéos, publications)

Ou bien encore MySQL pour un e-commerce pour la gestion des relations entre l'utilisateur et sa commande, la gestion des stocks des produits, les transactions financières et MongoDB pour les détails des avis, le stockage des informations du panier (si besoin)

