

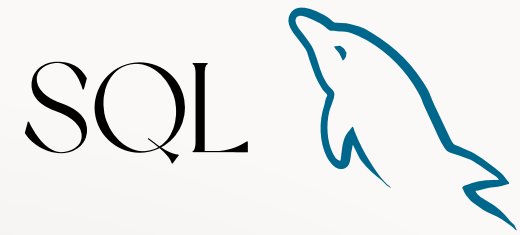


Mongodb is
for devs who don't
want to learn sql

CHANGE MY MIND

DIFFÉRENCE ENTRE MONGO DB ET SQL

INTRODUCTION



Les bases de données SQL sont relationnelles, c'est-à-dire qu'elles utilisent des tables pour organiser les données.

Elles sont connues pour leur stabilité et leur capacité à gérer des transactions complexes

Des exemples de bases de données SQL incluent MySQL, PostgreSQL, Oracle Database et Microsoft SQL Server

INTRODUCTION



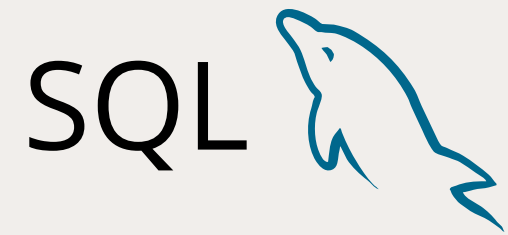
Les bases de données NoSQL, comme MongoDB, sont non relationnelles et ont été conçues pour gérer des ensembles de données non structurées ou semi-structurées

Elles ont gagné en popularité avec l'essor des grandes données et des applications nécessitant une scalabilité horizontale*.

Scalabilité horizontale

Fait référence à la superpuissance de lancer une fête de machines en plus dans le système au lieu de booster une seule machine (scalabilité verticale).

STRUCTURE DES DONNÉES



Les bases de données SQL utilisent des tables composées de lignes et de colonnes. Chaque table a une structure définie, avec des types de données spécifiques pour chaque colonne. Cette structure rigide garantit l'intégrité des données mais peut être moins flexible pour les types de données variés.



MongoDB est une base de données de documents qui stocke les données sous forme de documents BSON (Binary JSON). Chaque document peut avoir une structure différente, offrant une grande flexibilité pour les données hétérogènes.

IMAGE VOLÉ CHEZ LES BACKEND

MONGO



PEOPLE

{ "Id": 1, "FirstName": "Ada", "LastName": "Lovelace", "Email": "ada.lovelace@gmail.com", "Phone": [{ "Home": "+1.123.456.7890" }], { "Work": "+1.111.222.3333" } }
{ "Id": 2, "FirstName": "Grace", "LastName": "Hopper", "Email": "grace.hopper@gmail.com" }
{ "Id": 3, "FirstName": "Kathy", "LastName": "Sierra", "Email": "kathy.sierra@gmail.com" }

SQL



PERSON

Id	FirstName	LastName	Email
1	Ada	Lovelace	ada.lovelace@gmail.com
2	Grace	Hopper	grace.hopper@gmail.com
3	Kathy	Sierra	kathy.sierra@gmail.com

PHONE_NUMBER

PersonId	PhoneId	Phone Number	Type
1	1	+1.123.456.7890	Home
1	2	+1.111.222.3333	Work

LE LANGUAGE

MongoDB

INSERT

```
db.utilisateurs.insertOne({  
  nom: "Dupont",  
  prénom: "Jean",  
  email: "jean.dupont@example.com",  
  âge: 30  
})
```

LE LANGAGE

MongoDB

SELECT

```
// Lire tous les documents de la collection
```

```
db.utilisateurs.find({})
```

```
// Lire les documents avec un critère spécifique
```

```
db.utilisateurs.find({ nom: "Dupont" })
```

```
// Lire un document spécifique
```

```
db.utilisateurs.findOne({ email: "jean.dupont@example.com" })
```


LE LANGAGE

MongoDB

UPDATE

```
// Mettre à jour un seul document
db.utilisateurs.updateOne(
  { email: "jean.dupont@example.com" }, // Critère de sélection
  { $set: { âge: 31 } } // Mise à jour à effectuer
)

// Mettre à jour plusieurs documents
db.utilisateurs.updateMany(
  { nom: "Dupont" }, // Critère de sélection
  { $set: { actif: true } } // Mise à jour à effectuer
)
```

LE LANGUAGE

MongoDB

DELETE

```
// Supprimer un seul document  
db.utilisateurs.deleteOne({ email: "jean.dupont@example.com" })  
  
// Supprimer plusieurs documents  
db.utilisateurs.deleteMany({ nom: "Dupont" })
```

CAS D'UTILISATION

SQL

- Applications transactionnelles : banques, systèmes de paiement
- Systèmes de gestion de contenu
- Applications nécessitant une intégrité et une normalisation strictes des données

CAS D'UTILISATION

Mongo Db

- Applications nécessitant une grande scalabilité et flexibilité des données
- Applications Big Data et analytiques
- Applications mobiles et web avec des schémas de données évolutifs

CONCLUSION

- En résumé. SQL est idéal pour les applications nécessitant une structure de données rigide et des transactions complexes, tandis que MongoDB est parfait pour les applications nécessitant une flexibilité de données et une scalabilité horizontale. Le choix entre les deux dépendra des besoins spécifiques de votre projet.

FIN

Merci à Tom d'avoir prêter sa voix