

Voici une proposition de présentation en 5 diapositives :

---

## Diapositive 1 : Introduction

- **Titre :** Comparaison entre MongoDB (NoSQL) et SQL
  - **Objectifs :**
    - Présenter les caractéristiques principales de MongoDB et des bases de données SQL.
    - Mettre en lumière leurs différences en termes de modèle de données, scalabilité, transactions et cas d'usage.
  - **Contexte :**
    - Choix technologiques en fonction des besoins de l'application (données structurées vs. non structurées, exigences de scalabilité, etc.).
- 

## Diapositive 2 : MongoDB – La base de données NoSQL

- **Modèle de données :**
    - **Document-oriented :** Stocke les données sous forme de documents JSON-like.
    - **Schéma flexible :** Pas de schéma prédéfini, ce qui permet d'évoluer facilement.
  - **Fonctionnalités clés :**
    - **Scalabilité horizontale :** Répartition des données sur plusieurs serveurs (sharding).
    - **Haute disponibilité :** Réplication automatique via les Replica Sets.
    - **Performance :** Optimisée pour la lecture et l'écriture rapide sur de gros volumes de données.
  - **Cas d'usage :**
    - Applications nécessitant une évolution rapide du schéma (startups, applications web modernes, Big Data).
- 

## Diapositive 3 : SQL – Les bases de données relationnelles

- **Modèle de données :**
  - **Table-based :** Données organisées en tables avec lignes et colonnes.
  - **Schéma défini :** Structure rigide et pré-définie, avec relations entre tables.
- **Fonctionnalités clés :**
  - **ACID transactions :** Garantit la fiabilité et l'intégrité des données.
  - **Langage SQL :** Standardisé pour la gestion et la manipulation des données.
  - **Optimisation des requêtes :** Performances élevées pour les jointures et les requêtes complexes.
- **Cas d'usage :**
  - Applications nécessitant une forte cohérence des données (banques, systèmes ERP, e-commerce).

---

## Diapositive 4 : Comparaison directe – MongoDB vs SQL

Critère	MongoDB (NoSQL)	SQL
Modèle de données	Documents JSON-like, schéma flexible	Tables relationnelles, schéma rigide
Scalabilité	Horizontale (sharding)	Verticale principalement (bien que des solutions NoSQL permettent aussi l'horizontal)
Transactions	Supporte désormais des transactions ACID multi-documents, mais historiquement eventual consistency	Transactions ACID robustes
Langage de requête	Requêtes basées sur des documents (JSON, opérateurs spécifiques)	SQL (standardisé et puissant pour les jointures)
Performance	Adapté aux lectures/écritures rapides sur des données non structurées	Optimisé pour les requêtes complexes et relations

---

## Diapositive 5 : Conclusion et recommandations

- **Résumé :**
    - **MongoDB :** Idéal pour des applications nécessitant une grande flexibilité et une scalabilité horizontale. Parfait pour des données non structurées ou en évolution constante.
    - **SQL :** Convient mieux aux systèmes où l'intégrité des données et les transactions complexes sont essentielles.
  - **Recommandations :**
    - **Choix de MongoDB si :**
      - Votre application doit gérer des volumes massifs de données non structurées.
      - Vous recherchez une grande souplesse et une scalabilité horizontale.
    - **Choix de SQL si :**
      - Vous avez besoin d'un schéma strict et de transactions fiables.
      - Votre application repose sur des relations complexes entre les données.
  - **Conclusion :**
    - Le choix entre MongoDB et SQL dépend des exigences spécifiques du projet. Une analyse approfondie des besoins en termes de performance, d'évolutivité et d'intégrité des données est indispensable pour faire le meilleur choix.
-