**Diapositive 1 : Introduction à SQL et NoSQL**

* **SQL (Bases de données relationnelles) :**
  + Exemples : MySQL, PostgreSQL, SQL Server.
  + Données structurées et tabulaires stockées dans des lignes et des colonnes.
  + Utilise le langage de requête structuré (SQL) pour définir et manipuler les données.
  + Basé sur un schéma avec une structure de données fixe.
* **NoSQL (MongoDB) :**
  + Une base de données NoSQL utilise un stockage orienté document.
  + Sans schéma : documents de type JSON (format BSON).
  + Conçu pour la flexibilité et l'évolutivité.
  + Populaire pour les applications modernes nécessitant des données non structurées.

**Diapositive 2 : Fonctionnalité de SQL**

* **Structure des données :**
  + Tableaux avec lignes et colonnes.
  + Les relations sont explicitement définies.
* **Requête :**
  + Syntaxe SQL puissante pour les opérations CRUD.
  + Conforme à ACID pour la cohérence des transactions.
* **Cas d'utilisation :**
  + Systèmes bancaires, ERP et applications nécessitant des relations complexes.
* **Points forts :**
  + Intégrité et fiabilité des données.
  + Communauté forte et support des outils.

**Diapositive 3 : Fonctionnalités de MongoDB**

* **Structure des données :**
  + Documents BSON (de type JSON).
  + Données hiérarchiques stockées dans des collections.
* **Requête :**
  + Requêtes dynamiques avec MongoDB Query Language (MQL).
  + Cadre d'agrégation pour le traitement des données.
* **Cas d'utilisation :**
  + E-commerce, analyses en temps réel, IoT.
* **Points forts :**
  + Évolutivité horizontale.
  + Flexibilité pour les données semi-structurées ou non structurées.

**Diapositive 4 : MongoDB vs. SQL (tableau de comparaison)**

| **Fonctionnalité** | **Bases de données SQL** | **MongoDB** |
| --- | --- | --- |
| **Modèle de données** | Relationnel (Tableaux) | Orienté Document (Collections) |
| **Schéma** | Schéma fixe | Flexible (sans schéma) |
| **Langage de requête** | SQL | Langage de requête MongoDB (MQL) |
| **Évolutivité** | Verticale | Horizontal |
| **Transactions** | Conforme à ACID | Prend en charge ACID (pour les collections) |
| **Cas d'utilisation** | Données structurées | Données non structurées/semi-structurées |
| **Performance** | Optimisé pour les jointures | Optimisé pour les lectures/écritures |

**Diapositive 5 : Conclusion**

* **Quand choisir SQL :**
  + Données structurées avec des relations complexes.
  + Applications nécessitant une cohérence élevée des données.
* **Quand choisir MongoDB :**
  + L’évolutivité et la flexibilité sont essentielles.
  + Applications avec des données dynamiques ou non structurées.
* **Réflexion finale :**
  + Les deux bases de données excellent dans leurs domaines respectifs. Choisissez en fonction des besoins spécifiques du projet.