LES AVANTAGES DE L’APPROCHE ORIENTÉ GRAPHE

* Performances accrues :

Traiter des données fortement connectées en évitant les multiples jointures très coûteuses qu’il faudrait mettre en œuvre dans les bases de données relationnelles traditionnelles et ainsi permettre des mises à jour très performantes, même pour un très grand ensemble de données.

Offrir de meilleures performances en termes de rapidité de temps de réponse pour les lectures locales, par parcours de graphe.

* Développements simples :

L’utilisation de langages de requêtes tels que Cypher ou Gremlin destinés au traitement des données connectées facilite les développements. Par exemple, la recherche d’amis prend une seule ligne de code en Cypher.

* Modélisation facile :

Gérer facilement un modèle complexe puisque la base de données ne s’appuie pas sur un schéma rigide.

Permettre une modélisation parfois plus naturelle et plus lisible selon le cas d’utilisation.

Découverte de nouveaux cas d’usages par une représentation naturelle des données.

**Exemple pratique de neo4j**

Base de données native conçu autour d’une optimisation très simple.chaque point de données contient des pointeurs directs(relation) au nœud où il est connecté.Toutes les informations nécessaires pour trouver le nœud suivant sont disponibles dans le nœud lui-même.La couche de stockage native est un graphe connecté.En raison de ce principe,NEO4J n’a pas besoin de calculer les relations entre vos données,les connexions sont déjà stockées directement dans la base de donnée**Les autres bases de données** n’enregistrent pas les pointeurs directs entre les enregistrements.Ils doivent calculer les connexions en recherchant dans une structure de données appelé index.Ce processus doit être exécuté à plusieurs reprises pour trouver chaque connexion . elles sont coûteuse et très lentes

**Inconvénients de l’approche orienté graphe**

-peu évolutif car conçu pour les architectures à serveur unique,la croissance pose un problème.

-pas de langage de requête standard