

# Projet NoSQL: Système de base de données polyglotte

Master 2 MAGE – Informatique pour la Finance

Université Paris Dauphine

Hakim IGUENI, Elias KESRAOUI

Juin 2025



# Bases de données polyglottes

Pour gérer la complexité des données, notre projet explore un système polyglotte combinant plusieurs technologies de stockage pour performance et flexibilité.

Nous utilisons PostgreSQL pour le relationnel, MongoDB pour les documents, et Neo4j pour les graphes, intégrés via un script Python pour une architecture cohérente.



**PostgreSQL**

Données structurées



**MongoDB**

Documents semi-structurés



**Neo4j**

Données graphe

# Architecture du système polyglotte

Notre système d'information touristique adopte une architecture polyglotte pour optimiser la gestion des données, assurant performance, flexibilité et cohérence.

PostgreSQL gère les entités structurées, MongoDB stocke les commentaires, et Neo4j représente le réseau des villes. Le script Python assure l'intégration et la cohérence entre les bases.

## PostgreSQL

Gère les entités structurées (points d'intérêt, hébergements, activités, villes, utilisateurs, avis), garantissant la cohérence via clés étrangères et transactions.

## MongoDB

Stocke les commentaires d'avis avec une structure flexible pour les commentaires imbriqués et les réponses.

## Neo4j

Représente le graphe des villes et leurs connexions (distances, durées), optimisé pour les requêtes de chemin et de voisinage.

# Modèles de données et commandes de manipulation

Cette section présente les modèles de données et commandes de manipulation pour chaque technologie de base de données utilisée dans notre projet polyglotte. Chaque base est optimisée pour des types de données spécifiques.



## PostgreSQL

Modèle relationnel pour données structurées.



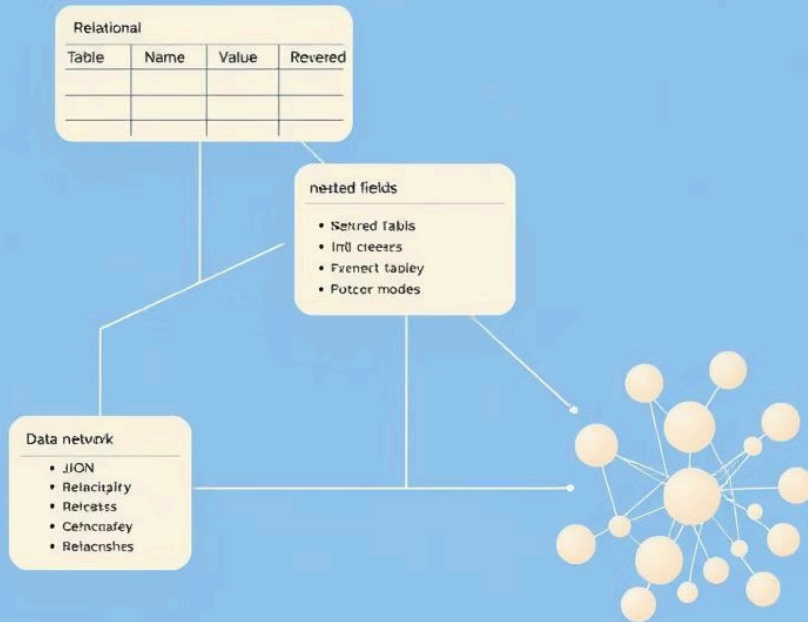
## MongoDB

Modèle document pour données semi-structurées.



## Neo4j

Modèle graphe pour relations complexes.



# Choix de modélisation: avantages et limites

Nos choix de modélisation ont été guidés par la nature des données et les types de requêtes. PostgreSQL excelle pour les données structurées et la cohérence, mais est moins flexible pour les structures imbriquées.

MongoDB offre une grande flexibilité pour les documents semi-structurés, mais manque de contraintes relationnelles fortes. Neo4j est parfait pour les graphes et les requêtes de parcours, mais moins adapté aux grandes quantités de données structurées et nécessite une duplication partielle des données.

Technologie	Avantages	Limites
PostgreSQL	Forte cohérence, requêtes SQL optimisées	Peu adapté aux données flexibles/imbriquées
MongoDB	Flexibilité, représentation hiérarchique	Pas de contraintes relationnelles fortes
Neo4j	Modélisation naturelle des réseaux, performances sur graphes	Moins adapté aux données fortement structurées

# Requêtes et résultats

Notre système polyglotte permet de répondre à diverses requêtes en exploitant les forces de chaque technologie. Nous illustrons ici six requêtes clés, démontrant l'efficacité de notre approche.

Des requêtes de proximité avec Neo4j aux activités liées à un point d'intérêt avec PostgreSQL, chaque exemple montre comment les bases combinées répondent à des besoins variés dans un contexte touristique.

1

## Villes proches (Neo4j)

Trouver les villes à moins de 10 km.

2

## Activités POI (PostgreSQL)

Lister les activités d'un point d'intérêt.

3

## Hébergements ville (PostgreSQL)

Récupérer les hébergements d'une ville.

4

## Distance/Temps (Neo4j)

Informations de trajet direct entre villes.

5

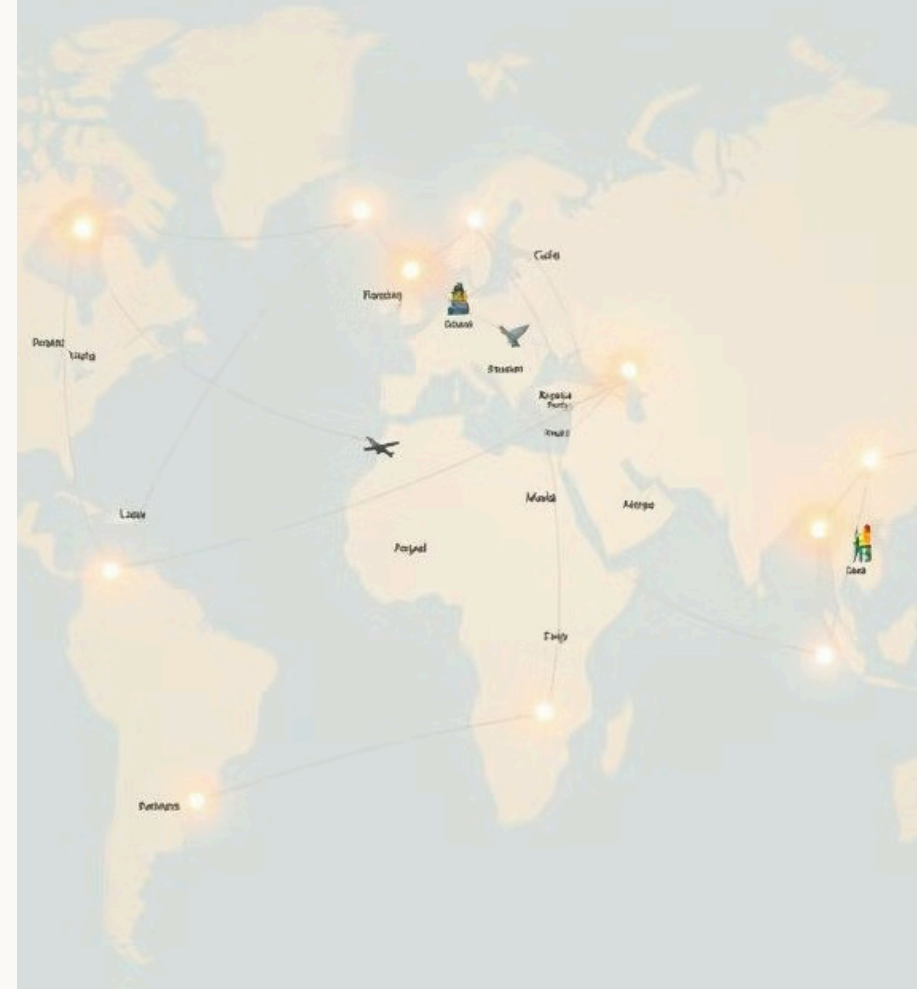
## Activités par date (PostgreSQL)

Filtrer les activités selon les dates.

6

## Itinéraires (Neo4j)

Lister les chemins possibles entre villes.



# Bilan du Projet NoSQL

**35-40**

**Heures de travail**

Intense effort par personne.

- Maîtrise de la gestion des bases de données hétérogènes et modélisation avancée.
- Expertise approfondie en requêtes croisées et en architecture de systèmes distribués.

**4**

**Compétences Clés**

Développées durant ce projet.