

warning "Mise à jour du backend (octobre 2026)" L'inférence GitHub Models a été retirée le 30 juillet 2026. Les références ci-dessous sont conservées uniquement comme contexte historique pendant la migration de ce lab. Utilisez le Lab 013 ? Backends LLM gratuits pour configurer Foundry Local ou Ollama avant de continuer.

Lab 039 : Comparaison des bases de données vectorielles

Niveau : L300 Parcours : ? RAG Durée : ~50 min ? Coût : Free ? Toutes les options ont des niveaux gratuits ou un mode local

Ce que vous apprendrez

Comprendre les différences clés entre les principales bases de données vectorielles

Comparer pgvector, Chroma, Qdrant et Azure AI Search sur la même tâche Évaluer chacune sur

: complexité de mise en place, vitesse de requête, filtrage, intégration cloud Choisir la

bonne base de données vectorielle pour les besoins de votre agent

Introduction

Le choix d'une base de données vectorielle est l'une des décisions d'architecture les plus importantes pour un agent basé sur RAG. Le « meilleur » choix dépend de la stack existante de votre équipe, des exigences de mise à l'échelle, des besoins de filtrage et de la stratégie cloud.

Les candidats :

Base de données | Type | Option gratuite | Idéal pour | pgvector | Extension PostgreSQL |

Azure free tier / local | Équipes utilisant déjà PostgreSQL | Chroma | Embarqué / serveur |

Entièrement open-source | Développement local, petits projets | Qdrant | BD vectorielle

dédiée | Qdrant Cloud free tier | Production à grande échelle, filtrage avancé | Azure AI

Search | Service Azure | Free tier (1 index) | Azure natif, recherche hybride, entreprise |

Prérequis

[diagram or code sample]

Pas de clés API nécessaires pour Chroma (local) et Qdrant (mode local). GitHub Token requis pour les embeddings : [diagram or code sample]

Configuration : Fonction d'embedding partagée

Les quatre tests utilisent les mêmes données produit OutdoorGear et le même modèle d'embedding :

[diagram or code sample]

Option A : Chroma (local, sans configuration)

Chroma est la façon la plus simple de démarrer ? pur Python, s'exécute en mémoire ou sur disque :

[diagram or code sample]

Option B : Qdrant (mode serveur local)

Qdrant offre un filtrage avancé et passe à l'échelle à des centaines de millions de vecteurs :

[diagram or code sample]

[diagram or code sample]

Option C : pgvector (Azure PostgreSQL ou local)

Voir le Lab 031 pour la configuration complète de pgvector. Comparaison rapide :

[diagram or code sample]

Option D : Azure AI Search (recherche hybride)

Azure AI Search prend en charge de manière unique la recherche hybride : recherche vectorielle + BM25 par mots-clés combinée avec un re-classement sémantique :

[diagram or code sample]

Cadre de décision

[diagram or code sample]

Tableau comparatif

| pgvector | Chroma | Qdrant | Azure AI Search | Configuration | Moyenne (BD nécessaire) | Minimale | Facile (Docker) | Moyenne (Azure) | Dev local | ? (Docker) | ? (en mémoire) | ? (Docker) | ? (Azure uniquement) | Recherche hybride | ? (manuelle) | ? | ? | ? (intégrée) | Filtrage | SQL WHERE | Métadonnées basiques | Avancé | OData complet | Échelle | Modérée (100M) | Grande (entreprise) | Intégration Azure | ? (PG managé) | ? | Partielle | ? (native) | Coût (niveau gratuit) | Free tier PG | Gratuit | Qdrant Cloud free | 1 index gratuit |

? Quiz de connaissances

question "1. Qu'est-ce que la recherche hybride et pourquoi est-elle souvent meilleure que la recherche vectorielle pure ?" La recherche hybride combine la recherche vectorielle (sémantique) avec la recherche par mots-clés (BM25) et classe les résultats en utilisant les deux signaux. La recherche vectorielle excelle dans la compréhension sémantique (« abri chaud » ? sac de couchage) mais peut manquer les correspondances exactes (un identifiant produit spécifique). BM25 est excellent pour les correspondances exactes de mots-clés mais manque les synonymes. Les combiner surpasse l'un ou l'autre seul, en particulier pour les noms de produits, les SKU et la terminologie spécialisée.

question "2. Pourquoi choisiriez-vous pgvector plutôt qu'une base de données vectorielle dédiée comme Qdrant ?" Si vous avez déjà PostgreSQL comme base de données principale, pgvector ajoute la recherche vectorielle sans ajouter un autre service à exploiter, maintenir et payer. Les données cohabitent avec vos données relationnelles ? vous pouvez faire un JOIN entre les enregistrements produit et leurs embeddings dans une seule requête. Pour la plupart des applications de moins de 1M de vecteurs, les performances de pgvector sont excellentes. Choisissez Qdrant quand vous avez besoin de > 100M de vecteurs ou d'un filtrage très avancé.

question "3. Qu'est-ce que l'index IVFFlat dans pgvector et quand devriez-vous utiliser HNSW à la place ?" IVFFlat (Inverted File Index with Flat quantization) : rapide à construire, utilise moins de mémoire, bon pour les jeux de données qui ne changent pas fréquemment. Utilise une recherche approximative ? le paramètre lists contrôle le compromis rappel/vitesse. HNSW (Hierarchical Navigable Small World) : meilleur rappel, requêtes plus rapides, mais utilisation mémoire plus élevée et construction plus lente. Utilisez IVFFlat pour les jeux de données < 1M qui ne changent pas beaucoup ; utilisez HNSW pour les jeux de données fréquemment mis à jour ou quand le rappel est critique. Les deux nécessitent pgvector ? 0.5.0.

Résumé

Pour le scénario du hub d'apprentissage OutdoorGear (< 10K produits, infrastructure Azure, équipe connaissant SQL) :

Recommandé : pgvector sur Azure Database for PostgreSQL Flexible Server.

Pas de nouveau service à apprendre SQL + vectoriel dans une seule requête Niveau gratuit disponible Prêt pour la production avec les migrations du Lab 031

Prochaines étapes

Construire la configuration pgvector : ? Lab 031 ? Recherche sémantique pgvector

Application RAG complète : ? Lab 022 ? RAG avec GitHub Models + pgvector RAG d'entreprise avec évaluation : ? Lab 042 ? RAG d'entreprise *(bientôt disponible)*

Kit d'accompagnement

Le build du site génère des téléchargements PDF et ZIP stables. Le ZIP contient le guide du lab et tous les fichiers d'accompagnement déclarés.

[Télécharger le PDF généré](#) [Télécharger le ZIP généré](#)

Supporting file: CHEATSHEET.md

Fichiers d'accompagnement: Lab 039 : Comparaison des bases de données vectorielles
Commencez par le parcours local ou gratuit documenté. Consignez les hypothèses et les preuves de validation. Gardez les services hébergés facultatifs sauf exigence explicite du lab.

Supporting file: README.md

Lab 039 : Comparaison des bases de données vectorielles
Utilisez ces fichiers avec le guide du lab et le kit généré.