# Chapitre 6: Bases de Données Vectorielles et RAG

## 6.1 Comprendre les bases de données vectorielles

### Principes fondamentaux

Les bases de données vectorielles sont une nouvelle génération de bases de données conçues pour stocker et interroger des données sous forme de vecteurs numériques de haute dimension, appelés “embeddings”. Ces vecteurs représentent la signification sémantique des données (texte, images, audio, etc.), permettant des recherches basées sur la similarité plutôt que sur des correspondances exactes de mots-clés.

Le principe fondamental est simple : plus deux vecteurs sont proches dans l’espace vectoriel, plus les données qu’ils représentent sont sémantiquement similaires.

### Différence avec les bases de données traditionnelles

Les bases de données traditionnelles (relationnelles ou NoSQL) sont optimisées pour stocker et récupérer des données structurées (nombres, chaînes de caractères) basées sur des requêtes exactes. Elles ne sont pas conçues pour comprendre le sens ou le contexte des données.

| Caractéristique | Bases de données traditionnelles | Bases de données vectorielles |
| --- | --- | --- |
| **Type de données** | Structurées (texte, nombres) | Non structurées (embeddings) |
| **Type de recherche** | Correspondance exacte (mots-clés) | Recherche par similarité sémantique |
| **Cas d’usage** | Transactions, catalogage | Recherche sémantique, recommandation, RAG |
| **Structure** | Tables, documents JSON | Collections de vecteurs de haute dimension |

### Représentation vectorielle des données

Pour utiliser une base de données vectorielle, les données non structurées (texte, images, etc.) doivent d’abord être converties en vecteurs numériques à l’aide d’un modèle d’embedding. Ce processus transforme des concepts complexes en points dans un espace mathématique, où la distance entre les points reflète leur relation sémantique.

## 6.2 Exemples pratiques de bases de données vectorielles

Plusieurs solutions de bases de données vectorielles sont disponibles, chacune avec ses propres caractéristiques :

* **Pinecone :** Une solution cloud gérée, facile à utiliser et très performante.
* **FAISS (Facebook AI Similarity Search) :** Une bibliothèque open source développée par Meta, très efficace pour la recherche de similarité à grande échelle.
* **Weaviate :** Une base de données vectorielle open source avec des fonctionnalités de recherche sémantique avancées.
* **ChromaDB :** Une base de données vectorielle open source conçue pour être simple à utiliser et à intégrer dans des applications d’IA.

*Tableau comparatif des différentes solutions, avec leurs avantages, inconvénients et cas d’usage recommandés.*

## 6.3 Retrieval-Augmented Generation (RAG)

### Principes de fonctionnement

Le Retrieval-Augmented Generation (RAG) est une technique qui combine la puissance des LLMs avec des connaissances externes provenant d’une base de données. Au lieu de se fier uniquement aux informations apprises lors de son entraînement, le LLM peut accéder à des données à jour et spécifiques à un domaine pour générer des réponses plus précises et contextuelles.

Le processus RAG se déroule en trois étapes : 1. **Recherche (Retrieval) :** Lorsqu’une question est posée, le système recherche les informations les plus pertinentes dans une base de données vectorielle. 2. **Augmentation :** Les informations récupérées sont ajoutées au prompt initial de l’utilisateur. 3. **Génération (Generation) :** Le LLM utilise le prompt augmenté (question + informations pertinentes) pour générer une réponse précise et informée.

### Avantages par rapport aux LLMs standards

* **Réduction des hallucinations :** Les réponses sont basées sur des données factuelles, ce qui réduit le risque d’informations incorrectes.
* **Connaissances à jour :** Le RAG permet d’intégrer des informations récentes sans avoir à ré-entraîner le LLM.
* **Spécificité du domaine :** Les entreprises peuvent utiliser leurs propres données pour créer des systèmes d’IA experts dans leur domaine.

## 6.4 Implémentation de solutions RAG pour les PME

### Architecture simplifiée

Pour une PME togolaise, une architecture RAG de base peut être mise en place avec des outils open source : 1. **Source de données :** Documents internes de l’entreprise (PDF, Word, etc.). 2. **Modèle d’embedding :** Un modèle open source pour convertir le texte en vecteurs. 3. **Base de données vectorielle :** Une solution comme ChromaDB ou FAISS. 4. **LLM :** Un modèle open source comme Mistral pour la génération de réponses.

*Schéma illustrant l’architecture d’un système RAG simple.*

### Étapes d’implémentation

1. **Collecte et préparation des données :** Rassembler les documents pertinents.
2. **Création des embeddings :** Convertir les documents en vecteurs.
3. **Indexation :** Stocker les vecteurs dans la base de données vectorielle.
4. **Mise en place de l’interface utilisateur :** Créer une interface simple pour poser des questions.
5. **Déploiement :** Héberger l’application sur un serveur local ou dans le cloud.

## 6.5 Applications pour la gestion des connaissances locales

Le RAG est particulièrement pertinent pour le contexte togolais, car il permet de créer des systèmes d’IA qui “connaissent” les spécificités locales :

* **Préservation du savoir traditionnel :** Créer une base de connaissances sur les plantes médicinales, les techniques agricoles ancestrales, etc.
* **Documentation des pratiques commerciales locales :** Capitaliser sur l’expertise des entrepreneurs locaux.
* **Création de bases de connaissances spécifiques au Togo :** Développer des systèmes d’information sur la législation, les procédures administratives, ou les opportunités de marché au Togo.

*Exemple de projet : Création d’un chatbot pour les entrepreneurs togolais, capable de répondre à des questions précises sur la création et la gestion d’entreprise au Togo, en se basant sur des documents officiels et des guides pratiques.*