Architecture des Données - SIA (Sources Islamiques Authentiques)

1. Stratégie de Backup (Disaster Recovery)

1.1 Politique de Sauvegarde Recommandée

Conformément aux recommandations du document Databasus, voici la stratégie de backup à mettre en place :

Fréquence	Rétention	Contenu
Horaire	24 heures	Différentiel (logs WAL)
Journalière	7 jours	Snapshot complet
Hebdomadaire	1 mois	Snapshot + exports JSON
Mensuelle	1 an	Archive complète chiffrée

1.2 Commandes de Backup PostgreSQL

```
# Backup complet avec compression
pg_dump -Fc -Z9 -h $DB_HOST -U $DB_USER -d $DB_NAME > backup_$(date +%Y%m%d_%H%M%S).du
mp

# Backup des données uniquement (JSON exportable)
pg_dump -h $DB_HOST -U $DB_USER -d $DB_NAME --data-only --format=plain > data_export.s
ql

# Restauration
pg_restore -h $DB_HOST -U $DB_USER -d $DB_NAME backup.dump
```

1.3 Backup Automatisé (Script)

Voir `/scripts/backup-db.sh` pour le script complet.

1.4 Stockage des Backups

Options recommandées :
- **S3/Scaleway** : Stockage objet économique
- **Google Drive** : Pour les petits volumes
- **NAS local** : Pour les environnements on-premise

---

2. Schéma de Base de Données Optimisé

2.1 Extension pgvector

Pour la recherche sémantique avancée avec embeddings vectoriels :

```sql
-- Activer pgvector
CREATE EXTENSION IF NOT EXISTS vector;

-- Table avec embeddings
CREATE TABLE document_chunks_v2 (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  content TEXT NOT NULL,
  content_arabic TEXT, -- Texte arabe original
  embedding vector(384), -- Dimension pour all-MiniLM-L6-v2 ou arabic-bert
  source VARCHAR(50) NOT NULL, -- 'coran', 'hadith', 'imam'
  reference VARCHAR(255) NOT NULL,
  metadata JSONB,
  created_at TIMESTAMP DEFAULT NOW()
);

-- Index pour recherche vectorielle rapide
CREATE INDEX idx_embedding ON document_chunks_v2
USING ivfflat (embedding vector_cosine_ops) WITH (lists = 100);
```

2.2 Structure JSONB pour Hadiths Complets

```
CREATE TABLE hadiths (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),

  -- Identifiants
  hadith_number INTEGER,
  book_number INTEGER,
  chapter_number INTEGER,

  -- Textes
  text_arabic TEXT NOT NULL,
  text_french TEXT,
  text_english TEXT,

  -- Chaîne de transmission
  isnad TEXT, -- Chaîne de narrateurs
  narrator_chain JSONB, -- [{"name": "...", "tier": 1}]

  -- Classification
  grade VARCHAR(50), -- 'sahih', 'hasan', 'daif', etc.
  graded_by VARCHAR(255), -- 'Al-Albani', 'Ibn Hajar', etc.

  -- Catégorisation
  source VARCHAR(100) NOT NULL, -- 'Sahih Al-Bukhari', 'Sahih Muslim', etc.
  book_name VARCHAR(255),
  chapter_name VARCHAR(255),
  themes TEXT[], -- ['prière', 'foi', 'comportement']

  -- Métadonnées complètes en JSONB
  metadata JSONB DEFAULT '{}',

  -- Embeddings pour recherche sémantique
  embedding_arabic vector(768), -- Pour modèles arabes
  embedding_french vector(384), -- Pour modèles français

  -- Horodatage
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Index pour recherche full-text arabe
CREATE INDEX idx_hadith_arabic_fts ON hadiths
USING gin(to_tsvector('arabic', text_arabic));

-- Index JSONB pour requêtes sur métadonnées
CREATE INDEX idx_hadith_metadata ON hadiths USING gin(metadata);

-- Index sur les thèmes
CREATE INDEX idx_hadith_themes ON hadiths USING gin(themes);
```

2.3 Structure pour le Coran

```sql
CREATE TABLE quran_verses (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),

  -- Références
  surah_number INTEGER NOT NULL,
  verse_number INTEGER NOT NULL,
  juz_number INTEGER,
  hizb_number INTEGER,
  page_number INTEGER, -- Page Mushaf Madina

  -- Textes
  text_arabic TEXT NOT NULL,
  text_french TEXT,
  text_transliteration TEXT,

  -- Métadonnées de la sourate
  surah_name_arabic VARCHAR(100),
  surah_name_french VARCHAR(100),
  surah_name_english VARCHAR(100),
  revelation_type VARCHAR(20), -- 'mecquoise' ou 'medinoise'

  -- Catégorisation thématique
  themes TEXT[],
  related_verses JSONB, -- [{"surah": 2, "verse": 255, "relation": "tafsir"}]

  -- Embeddings
  embedding vector(384),

  -- Index unique
  UNIQUE(surah_number, verse_number)
);

-- Index composite pour navigation
CREATE INDEX idx_quran_nav ON quran_verses(surah_number, verse_number);
```

3. Sources de Données Recommandées

3.1 Hadiths (Objectif : 50,000+ entrées)

Collection	Nombre	Source API/Dataset
Sahih Al-Bukhari	~7,275	sunnah.com API
Sahih Muslim	~7,500	sunnah.com API
Sunan Abu Dawud	~5,274	sunnah.com API
Jami At-Tirmidhi	~3,956	sunnah.com API
Sunan An-Nasa’i	~5,761	sunnah.com API
Sunan Ibn Majah	~4,341	sunnah.com API
Muwatta Malik	~1,832	sunnah.com API
Total	**~36,000**	

3.2 Coran (6,236 versets)

Source	Format	Lien
Tanzil.net	XML/JSON	https://tanzil.net/download
QuranEnc	JSON multi-langues	https://quranenc.com
Al-Quran Cloud	API REST	https://alquran.cloud/api

3.3 Ouvrages des Imams

Ouvrage	Auteur	Extraits
Riyad as-Salihin	An-Nawawi	~1,900 hadiths commentés
Al-Adab al-Mufrad	Al-Bukhari	~1,300 hadiths
Ihya’ Ulum al-Din	Al-Ghazali	Extraits sélectionnés
La Risala	Al-Qayrawani	Jurisprudence Maliki

4. Migration vers pgvector

4.1 Prérequis

```
# Sur Supabase (déjà inclus)
# Sur PostgreSQL auto-hébergé :
sudo apt install postgresql-15-pgvector
```

4.2 Script de Migration

```
// scripts/migrate-to-pgvector.ts
import { PrismaClient } from '@prisma/client'
import OpenAI from 'openai'

const prisma = new PrismaClient()
const openai = new OpenAI({ baseURL: 'https://apps.abacus.ai/v1' })

async function generateEmbedding(text: string): Promise<number[]> {
  const response = await openai.embeddings.create({
    model: 'text-embedding-3-small',
    input: text
  })
  return response.data[0].embedding
}

async function migrateChunks() {
  const chunks = await prisma.documentChunk.findMany()

  for (const chunk of chunks) {
    const embedding = await generateEmbedding(chunk.content)

    await prisma.$executeRaw`
      UPDATE document_chunks
      SET embedding = ${embedding}::vector
      WHERE id = ${chunk.id}
    `
  }
}
```

5. Options d'Hébergement

5.1 Comparatif

Service	pgvector	Gratuit	Prix/mois	Notes
Supabase	✅	500MB	$25+	Recommandé pour MVP
Neon	✅	512MB	$19+	Serverless, auto-scale
Render	✅	-	$7+	Simple, bon rapport qualité/prix
AWS RDS	✅	-	$15+	Enterprise, SLA garanti
DigitalOcean	✅	-	$15+	Simple, bonne perf

5.2 Recommandation pour SIA

Phase MVP (actuel) : Abacus AI hosted DB (gratuit)
Phase Production : Supabase Pro ($25/mois) avec pgvector
Phase Scale : AWS RDS ou DigitalOcean avec réplicas

---

6. Estimation des Volumes

Donnée	Lignes	Taille estimée
Versets Coran	6,236	~5 MB
Hadiths (6 collections)	36,000	~50 MB
Embeddings (384 dim)	42,236	~65 MB
Ouvrages Imams	5,000	~10 MB
Total initial	**~50,000**	**~130 MB**
Objectif 1 an	**500,000**	**~1.5 GB**

---

7. Checklist de Production

- [ ] Activer pgvector sur la base de données

- [ ] Migrer vers le schéma optimisé JSONB
- [ ] Importer les 6 collections de hadiths majeurs
- [ ] Générer les embeddings pour tous les documents
- [ ] Configurer les backups automatiques (journalier minimum)
- [ ] Tester la restauration de backup
- [ ] Mettre en place le monitoring (temps de requête, espace disque)
- [ ] Documenter les procédures de recovery

Document généré le 17 janvier 2026 - SIA v1.0 Alpha