Database Design Document

Author: Dhawanit Bhatnagar

Date: 25-July-2025

Version: 2.0

---

1. Purpose

This document describes the detailed database design of the Document Management System. It covers all tables, fields, relationships, constraints, and indexing strategies to ensure a scalable, maintainable, and high-performing database structure.

---

2. Database Overview

The system uses **PostgreSQL** as its relational database. The design follows normalization principles to avoid redundancy, while still being optimized for read-heavy operations.

Key features:

- **Referential integrity** with foreign keys.
- **Clear relationships** between Users, Documents, and Ingestion Logs.
- **Scalability** with indexing and future support for sharding or replication.

---

3. Entity-Relationship Model (ERD)

3.1 Entities and Relationships:

1. **Users** → Upload and manage documents.
2. **Documents** → Uploaded files with metadata.
3. **Ingestion Logs** → Track ingestion attempts and statuses for each document.

3.2 Relationships:

- A **User** can upload many **Documents** (1:N).
- A **Document** belongs to exactly **one User**.
- A **Document** can have many **Ingestion Logs** (1:N).
- A **User** can trigger ingestion on many **Documents** (1:N).

---

4. Table Definitions

4.1 Users Table

Column Name	Data Type	Constraints
id	UUID	Primary Key, auto-generated
username	VARCHAR(100)	NOT NULL, UNIQUE
email	VARCHAR(255)	NOT NULL, UNIQUE
password	TEXT	NOT NULL (hashed password)
role	ENUM	Values: admin, editor, viewer, NOT NULL
can_trigger_ingestion	BOOLEAN	DEFAULT FALSE
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

4.2 Documents Table

Column Name	Data Type	Constraints
id	UUID	Primary Key, auto-generated
title	VARCHAR(255)	NOT NULL
description	TEXT	
file_path	TEXT	NOT NULL
file_type	VARCHAR(100)	
storage_location	ENUM	Values: local, s3, NOT NULL
status	ENUM	Values: uploaded, ingested, NOT NULL

Column Name	Data Type	Constraints
user_id	UUID	Foreign Key → users(id), NOT NULL
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

4.3 Ingestion Logs Table

Column Name	Data Type	Constraints
id	UUID	Primary Key, auto-generated
document_id	UUID	Foreign Key → documents(id), ON DELETE CASCADE
triggered_by	UUID	Foreign Key → users(id), ON DELETE SET NULL
status	ENUM	Values: pending, in-progress, completed, failed, cancelled
response_message	TEXT	
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

5. Relationships

- **Users → Documents (1:N)**

  A user can upload multiple documents, but each document belongs to a single user.

- **Documents → Ingestion Logs (1:N)**

Multiple ingestion attempts can be recorded for each document.

- **Users → Ingestion Logs (1:N)**

  A user (admin/editor) can trigger multiple ingestions.

- **Cascade Rules:**
  - Deleting a user sets triggered_by in ingestion logs to NULL (history preserved).
  - Deleting a document cascades to remove its ingestion logs.

---

6. Constraints and Indexing

- **Primary Keys** on all id columns.
- **Unique constraints** on email and username.
- **Indexes**:
  - documents.user_id for fast document lookup by user.
  - ingestion_logs.document_id for quick ingestion history retrieval.
- **Foreign Key constraints** to ensure referential integrity.

---

7. Future Enhancements

- Add **document_versions** table to track different file versions.
- Add **tags** table for document categorization (many-to-many relationship).
- Implement **full-text search indexing** on title and description for faster searching.
- Add **notifications table** to store status updates for real-time alerts.
- Use **partitioning or sharding** for handling very large ingestion logs.