面试题：设计并实现一个基于 RAG（检索增强生成）的系统

技术要求：

- 数据库：PostgreSQL

- 开发语言：不限

- 大模型接口调用：支持各大模型厂商的接口

功能需求：

1. 数据库设计与初始化：

- 安装 PostgreSQL 数据库，并创建以下数据表：

- `max\_kb\_file`：文件表

- `max\_kb\_dataset`：知识库表

- `max\_kb\_document`：文档表

- `max\_kb\_paragraph`：片段表

- `max\_kb\_embedding`：向量表

2. 对象存储配置：

- 申请并配置一个对象存储账号（如阿里云 OSS、腾讯云 COS、亚马逊云 S3 等）。

3. 知识库管理：

- 创建知识库：用户可通过界面对 `max\_kb\_dataset` 表进行增删改查操作。

4. 文件上传与管理：

- 上传文件：用户通过界面上传文件，文件信息存入 `max\_kb\_file` 表。上传时使用 MD5 校验实现文件去重。

- 文档解析：将上传的 PDF 文件解析为 Markdown 格式，文档信息存入 `max\_kb\_document` 表。解析后的文档上传至对象存储，避免重复解析相同文档。

5. 文档处理：

- 文档拆分：对解析后的 Markdown 文档进行拆分，片段存入 `max\_kb\_paragraph` 表。

- 向量生成：对拆分后的片段生成向量，存入 `max\_kb\_embedding` 表。

6. 相似度检索与排序：

- 相似度检索：用户输入问题，将其向量化，并与 `max\_kb\_embedding` 中的向量进行相似度计算，返回相关片段及相似度。

- 片段重排：对检索结果进行重新排序，筛选出相似度最高的 N 条片段。

7. 模型推理：

- 用户输入问题，系统进行向量检索和片段处理，调用大模型进行推理，最终返回推理结果。

附加要求：

- 用户界面：提供方便用户操作和演示的界面，实现上述功能。

- 代码提交与部署：

- 前端代码需提交至 GitHub，并部署到 Vercel。

- 后端代码需部署到云服务器（云厂商自选，完成此项可加分）。

- 演示视频：完成后需提供一个演示视频。

- 数据库主键统一使用雪花id

- 模型推理推理时因具有上下文能力

测试数据：

- 附件中包含两个 PDF 文件，可用于测试。

数据库表结构建议：

1. `max\_kb\_file` 文件表

```sql

CREATE TABLE max\_kb\_file (

id BIGINT PRIMARY KEY, -- 文件ID

md5 VARCHAR(32) NOT NULL, -- 文件的MD5值，用于校验文件一致性

filename VARCHAR(64) NOT NULL, -- 文件名

file\_size BIGINT NOT NULL, -- 文件大小（字节）

user\_id VARCHAR(32), -- 用户ID，标识上传文件的用户

platform VARCHAR(64) NOT NULL, -- 上传平台（如S3）

region\_name VARCHAR(32), -- 区域名

bucket\_name VARCHAR(64) NOT NULL, -- 存储桶名称

file\_id VARCHAR(64) NOT NULL, -- 文件存储ID

target\_name VARCHAR(64) NOT NULL, -- 文件存储路径

tags JSON, -- 文件标签（JSON格式）

creator VARCHAR(64) DEFAULT '',

create\_time TIMESTAMP WITHOUT TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updater VARCHAR(64) DEFAULT '',

update\_time TIMESTAMP WITHOUT TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

deleted SMALLINT NOT NULL DEFAULT 0,

tenant\_id BIGINT NOT NULL DEFAULT 0

);

```

2. `max\_kb\_dataset` 知识库表

```sql

DROP TABLE IF EXISTS max\_kb\_dataset;

CREATE TABLE max\_kb\_dataset (

id BIGINT PRIMARY KEY,

name VARCHAR NOT NULL,

description VARCHAR,

type VARCHAR,

meta JSONB,

user\_id VARCHAR NOT NULL,

remark VARCHAR(256),

creator VARCHAR(64) DEFAULT '',

create\_time TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updater VARCHAR(64) DEFAULT '',

update\_time TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

deleted SMALLINT DEFAULT 0,

tenant\_id BIGINT NOT NULL DEFAULT 0

);

```

3. `max\_kb\_document` 文档表

```sql

DROP TABLE IF EXISTS max\_kb\_document;

CREATE TABLE max\_kb\_document (

id BIGINT NOT NULL,

name VARCHAR NOT NULL,

char\_length INT NOT NULL,

status VARCHAR NOT NULL,

is\_active BOOLEAN NOT NULL,

type VARCHAR NOT NULL,

meta JSONB NOT NULL,

dataset\_id BIGINT NOT NULL,

hit\_handling\_method VARCHAR NOT NULL,

directly\_return\_similarity FLOAT8 NOT NULL,

files JSON, -- 文件信息（JSON格式）

creator VARCHAR(64) DEFAULT '',

create\_time TIMESTAMP WITHOUT TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updater VARCHAR(64) DEFAULT '',

update\_time TIMESTAMP WITHOUT TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

deleted SMALLINT NOT NULL DEFAULT 0,

tenant\_id BIGINT NOT NULL DEFAULT 0

);

```

4. `max\_kb\_paragraph` 片段表

```sql

DROP TABLE IF EXISTS max\_kb\_paragraph;

CREATE TABLE max\_kb\_paragraph (

id BIGINT NOT NULL,

content VARCHAR NOT NULL,

title VARCHAR NOT NULL,

status VARCHAR NOT NULL,

hit\_num INT NOT NULL,

is\_active BOOLEAN NOT NULL,

dataset\_id BIGINT NOT NULL,

document\_id BIGINT NOT NULL,

creator VARCHAR(64) DEFAULT '',

create\_time TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updater VARCHAR(64) DEFAULT '',

update\_time TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

deleted SMALLINT DEFAULT 0,

tenant\_id BIGINT NOT NULL DEFAULT 0

);

```

5. `max\_kb\_embedding` 向量表

```sql

DROP TABLE IF EXISTS max\_kb\_embedding;

CREATE TABLE max\_kb\_embedding (

id BIGINT PRIMARY KEY,

source\_id BIGINT NOT NULL,

source\_type VARCHAR NOT NULL,

is\_active BOOLEAN NOT NULL,

embedding VECTOR NOT NULL,

meta JSONB NOT NULL,

dataset\_id BIGINT NOT NULL,

document\_id BIGINT NOT NULL,

paragraph\_id BIGINT NOT NULL,

search\_vector TSVECTOR NOT NULL,

creator VARCHAR(64) DEFAULT '',

create\_time TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updater VARCHAR(64) DEFAULT '',

update\_time TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

deleted SMALLINT DEFAULT 0,

tenant\_id BIGINT NOT NULL DEFAULT 0

);

```

---

备注：

- 请确保代码的可读性和规范性，适当添加注释。

- 在实现过程中，可根据需要对数据库结构进行优化和调整。

- 如果有任何问题或需要澄清的地方，请及时提出。