

# DB design for Contiq

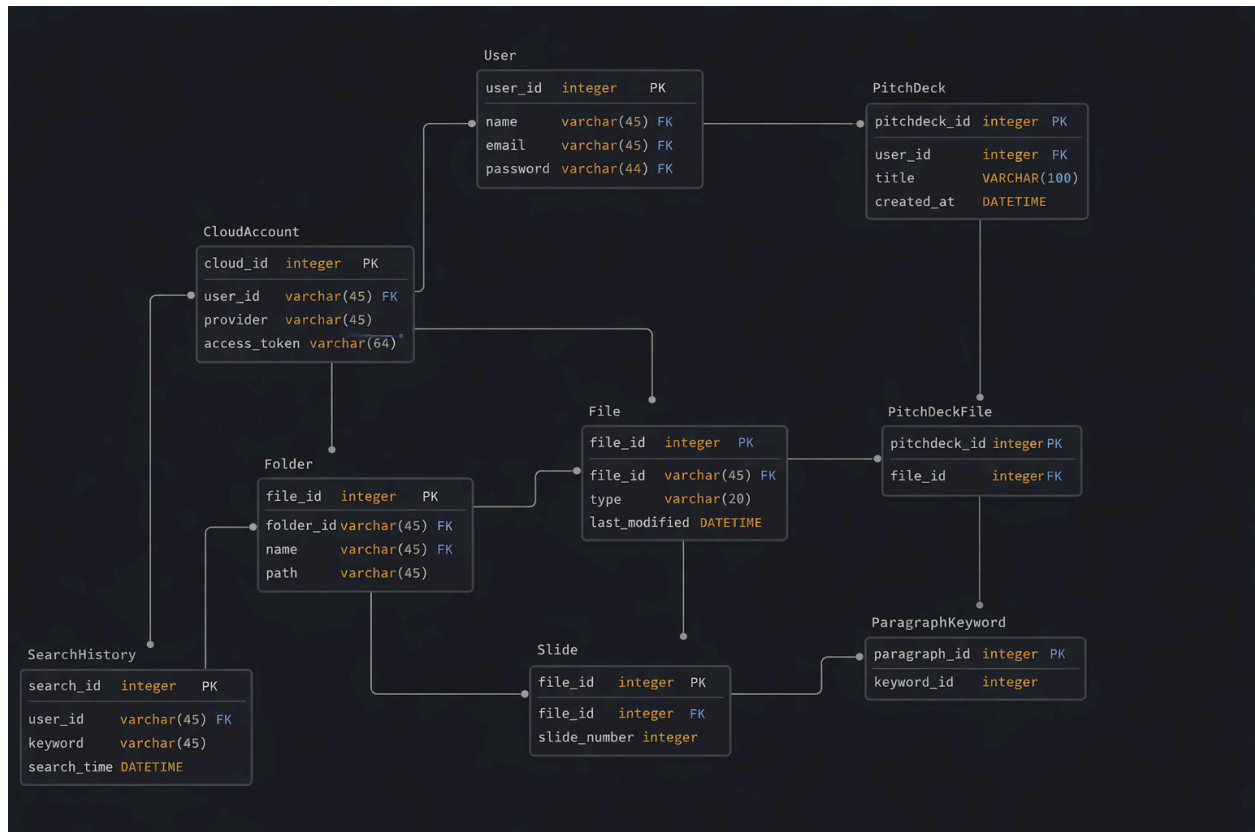
**Problem Statement :** Contiq product. Contiq connects to cloud storage which enables the user to find and fetch the PPT easily.

Contiq is a cloud-based content management system that helps users easily find, search, and reuse presentation content stored across different cloud platforms such as Google Drive, Dropbox, and TeraBox.

The main objective of the system is to reduce the time required to search for relevant presentation material and allow users to quickly create meaningful pitch decks by reusing existing files.

The system allows users to:

- Sign up and log in
- Connect cloud storage accounts
- Synchronize folders and files
- Search inside PPT files using keywords
- Navigate directly to specific slides and paragraphs
- Create pitch decks by grouping selected files



## API Design for Contiq

<https://app.swaggerhub.com/apis/zemosotechnologies-dab/Contiq-1/1.0.0>

Swagger, also called OpenAPI, is used to **define, document, and test the APIs** of our system in a standard format.

Instead of writing backend code first, we designed the API structure first using OpenAPI. This helps frontend, backend, and testers understand the system clearly.

In my project **Contiq**, the system connects to cloud storage, manages files and pitch decks, and allows users to search content inside

documents.

So I designed the Swagger APIs according to our **database and system flow**, not using generic examples like Petstore.

Swagger helps us to:

- Clearly define all API endpoints.
- Understand what data is required and what response is returned.
- Test APIs easily using Swagger UI.
- Keep frontend and backend teams aligned.

### **Main API Modules**

I divided the APIs into logical resources:

1. **User APIs** – to create and manage users.
2. **Cloud Account APIs** – to connect Google Drive, Dropbox, etc.
3. **Folder and File APIs** – to fetch folders and files from cloud storage.
4. **Pitchdeck APIs** – to create and manage pitch decks.
5. **Search APIs** – to search keywords inside files and slides.

Each API is connected to a corresponding table in the database.

### **Example**

For example:

- `/users` → manages users
- `/cloud-accounts` → stores connected cloud accounts
- `/files` → returns files from cloud folders
- `/pitchdecks` → creates pitch decks from selected content

- /search → searches keywords and returns matching paragraphs and slides