

SQL Final Project

LAZY-MAN'S DATABASE MANAGEMENT SYSTEM

組別：第一組

組員：鄭朝鴻、許喬淇、謝欣妤、連震宇

Outline



1. Motivation
2. Objectives
3. System Overview
4. Material and Method
5. Results
6. Future works

Motivation

Issues

- Most people are not familiar with SQL
- Learning SQL language is not an easy task



Idea

Natural language instead of SQL query

Organizing information from data



Objectives

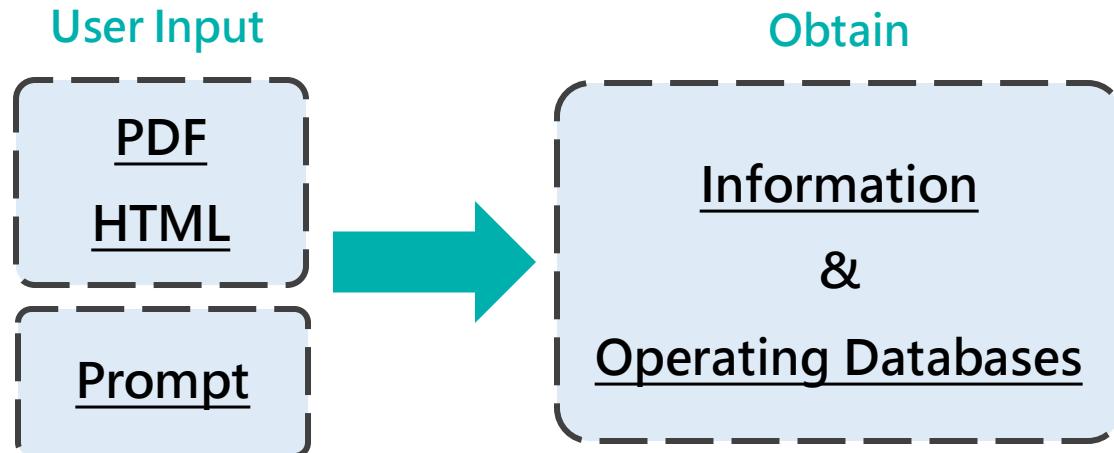
LAZY-MAN'S DATABASE MANAGEMENT SYSTEM



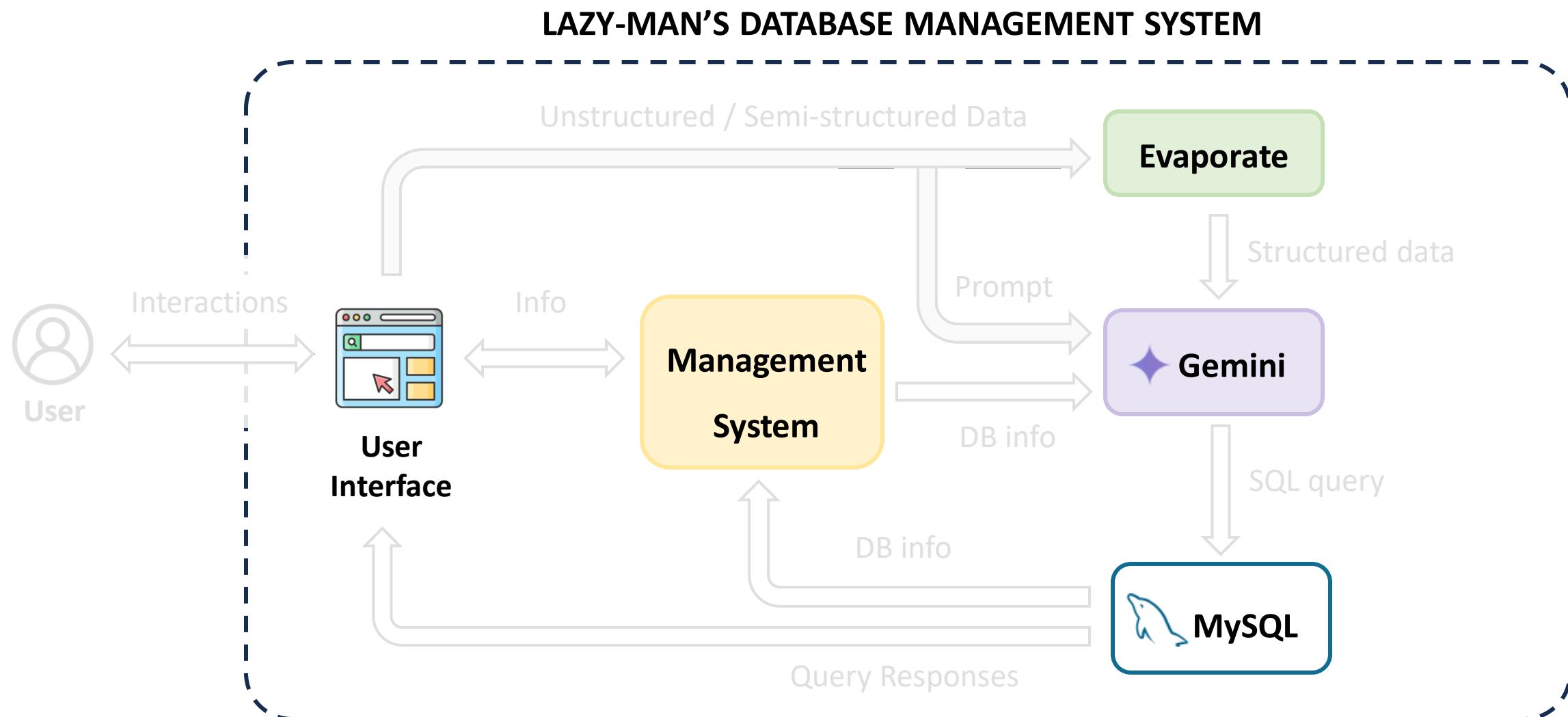
Objectives

LAZY-MAN'S DATABASE MANAGEMENT SYSTEM

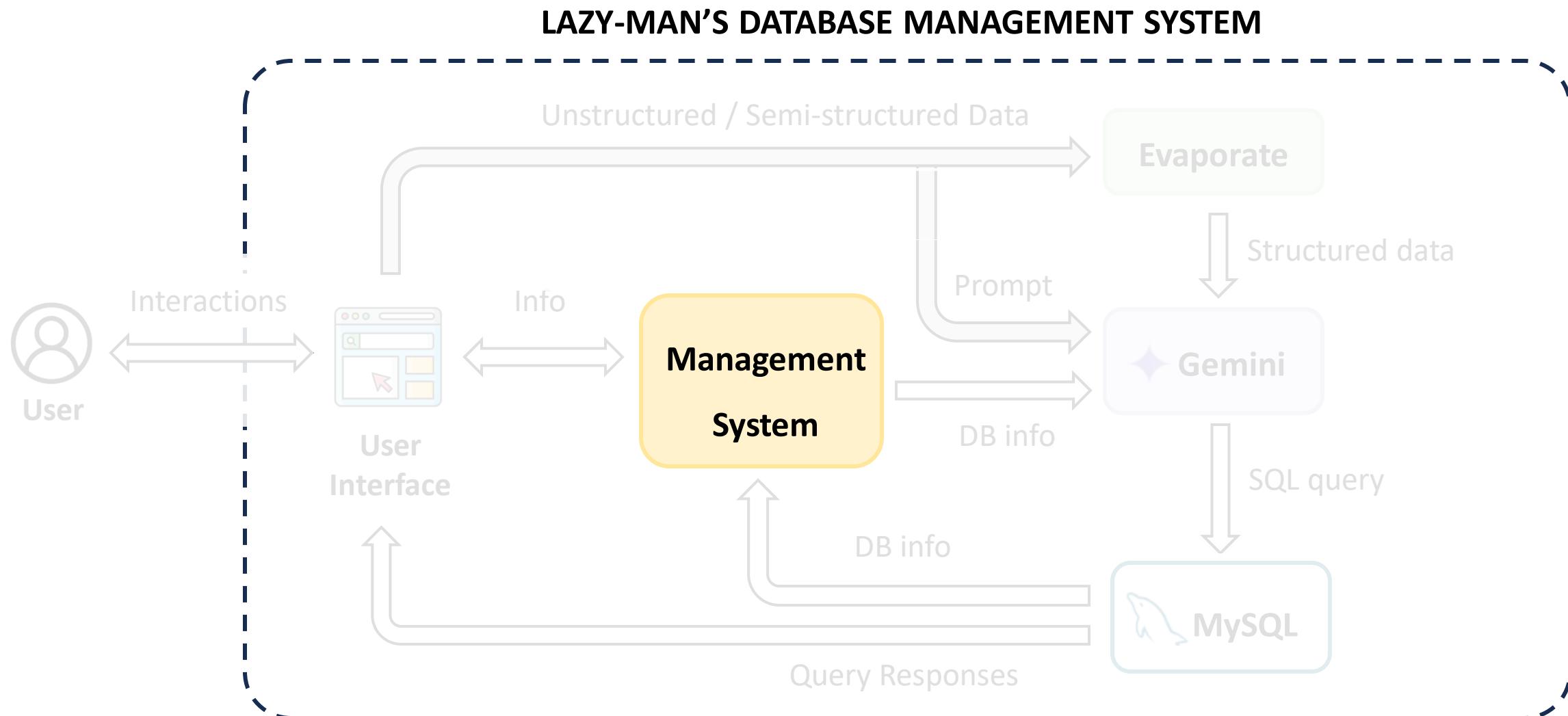
- User-friendly interface
- Easy obtaining information from databases
- Easy access to information from data



System Overview



System Overview



Materials and Methods

- User Interface

- Allow users to
select between databases

Show an overview
of the selected database



- Allow users to input questions with natural language,
and import data files

Materials and Methods

- User Interface



Allow users to
select between databases

Show an overview
of the selected database



Allow users to input questions with natural language,
and import data files

Materials and Methods

- User Interface



Allow users to
select between databases

Show an overview
of the selected database



**Allow users to input questions with natural language,
and import data files**

Materials and Methods



- Gemini is an LLM developed by Google
- We utilize Gemini Pro to convert natural language to SQL query
- Proficient in code-related tasks and is also free of charge

Materials and Methods

- Management System

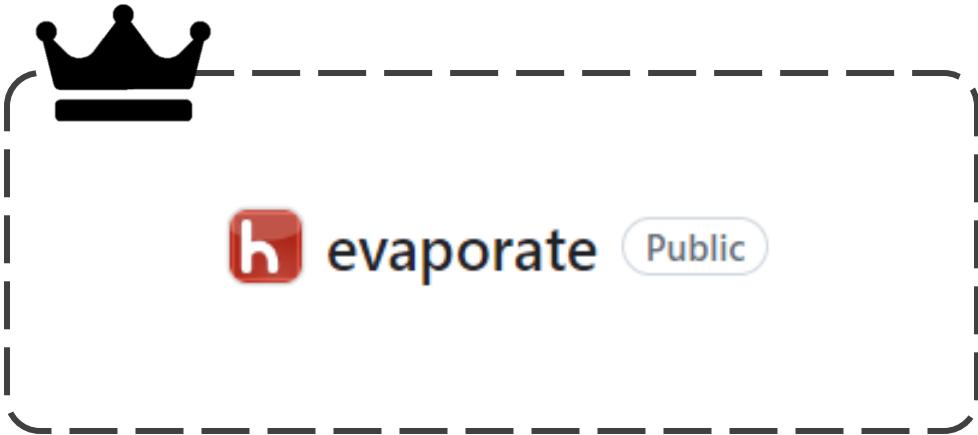
Before user give prompt

- Connecting to MySQL
- Using Mysqldump to acquire DB data by outputting schemas and tables as individual files
- Displaying the DB list on the user interface through Streamlit

After user give prompt

- Confirming whether any table names are mentioned in given prompt and only reading the needed DB data files
- Inputting the given data to Evaporate and obtain the structured data
- Inputting the extracted DB data, prompt, and structured data into Gemini and obtain the SQL query
- Inputting the SQL query to MySQL and obtain the response
- Renewing the changed database data files
- Displaying the response outputted by MySQL on the UI

Materials and Methods



Input

Data lake: A collection of semi-structured documents (e.g. HTML, TXT, XML)



EVAPORATE-CODE+

1 Schema Identification

Prompt LLM: List all attributes about the player mentioned in this document.



Filtered Attributes

- ✓ position
- ✓ name
- ✓ draft year
- ✗ headline

2 Function Synthesis

Prompt LLM: Write a function to extract **draft year** from the document.



Function Candidates

```
def draft_year(doc)
```

3 Function Aggregation

Prompt LLM: Extract **draft year** directly from the document.



Filter: Compare function and LLM outputs, filtering out functions that disagree.

```
def draft_year(doc)
```

Filtered Functions

	def → 2017	Estimate Quality	def → 1999	def → 2017
def	→ 2017	●	●	→ 2017
def	→ 1999	●	●	→ 2017
def	→ 2017	●	●	→ 2017

Output

Tables: A structured view of the data in the input documents.



name	draft year	position
Jayson Tatum	2017	Power Forward
Anthony Davis	2012	Center
Kevin Durant	2007	Small Forward
Steph Curry	2009	Point Guard
Joel Embiid	2014	Center
Nikola Jokic	2014	Center
Luka Doncic	2018	Point Guard
LeBron James	2003	Small Forward

Materials and Methods

Evaporate-Direct

Input document to the LLM and prompt it to extract values directly

prompt

list

Document file



List all attributes that have mentioned in the list

Cost more



2. Extraction



Evaporate-Code

Input a small sample of documents to the LLM and prompt it to write code to do the extraction

prompt

Document file



Write a Python function
`get_<attribute>` _ to extract
the <attribute> from the
Document file



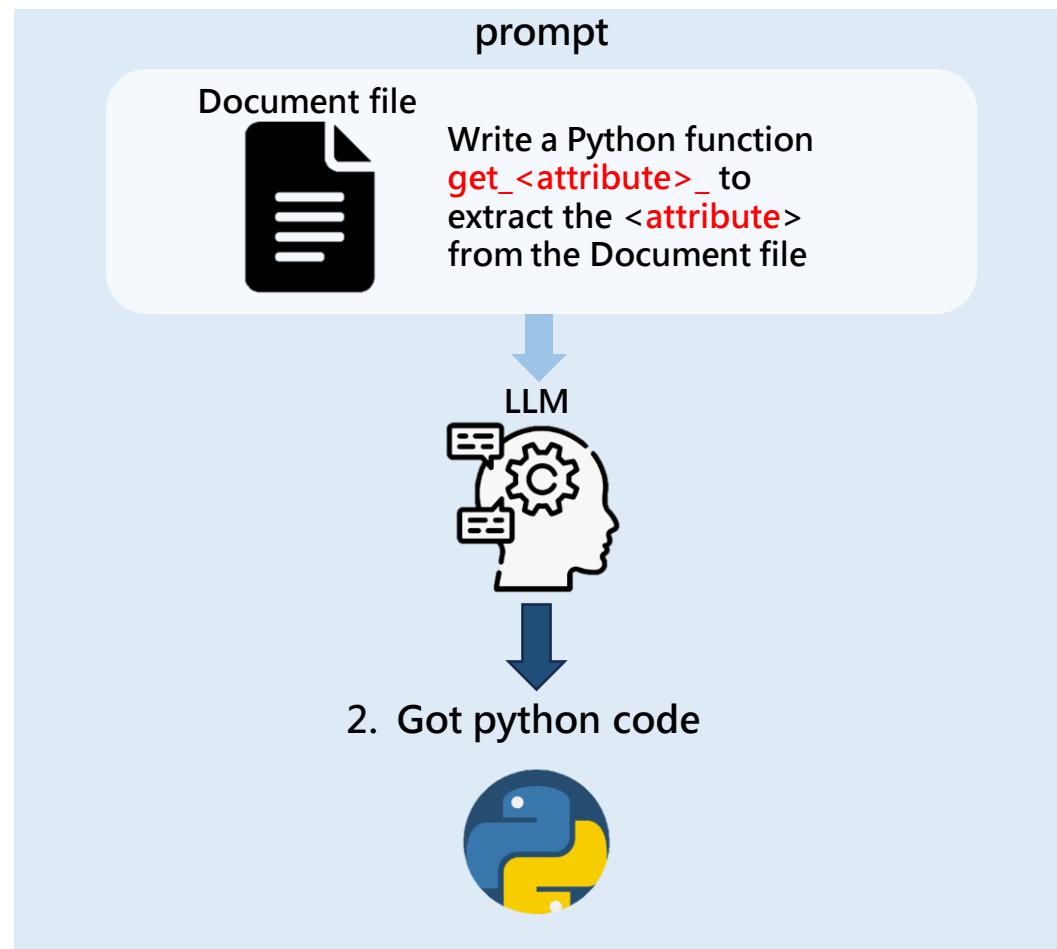
2. Got python code



Materials and Methods

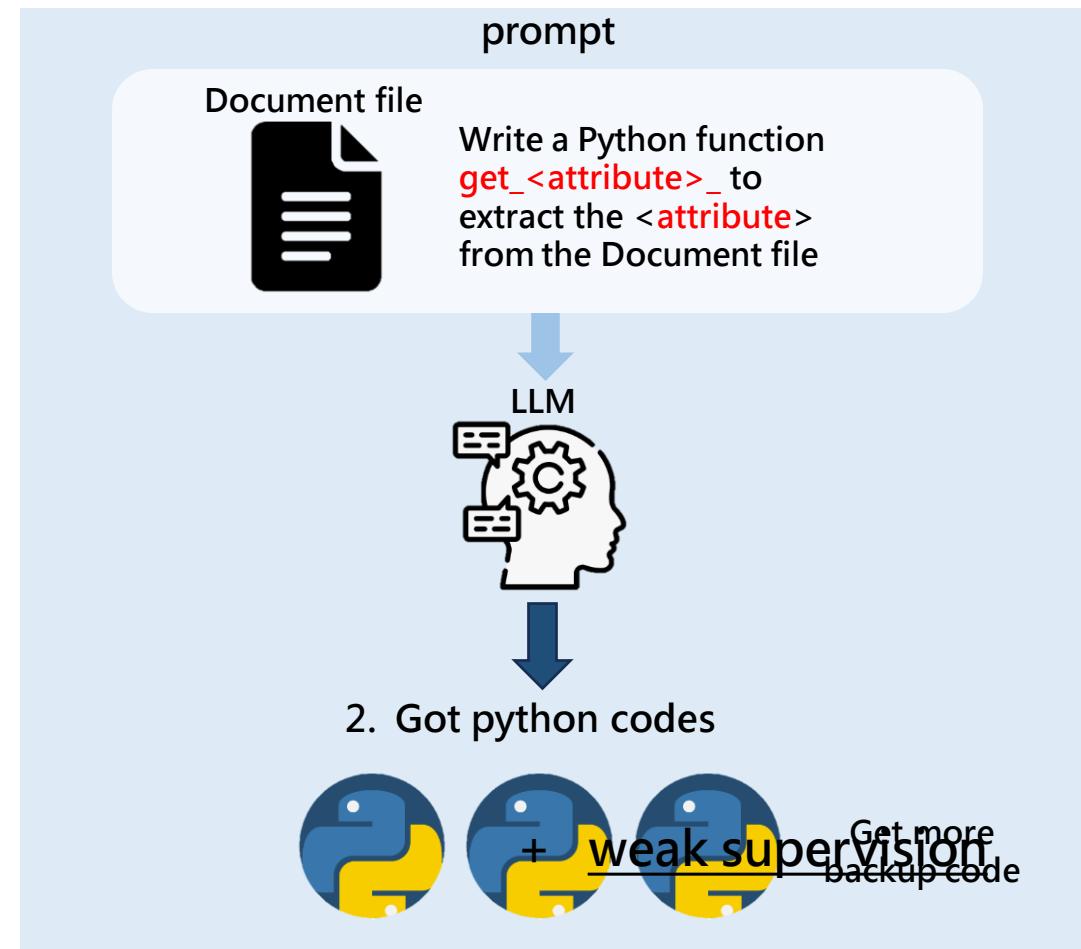
Evaporate-Code

Input a small sample of documents to the LLM and prompt it to write code to do the extraction



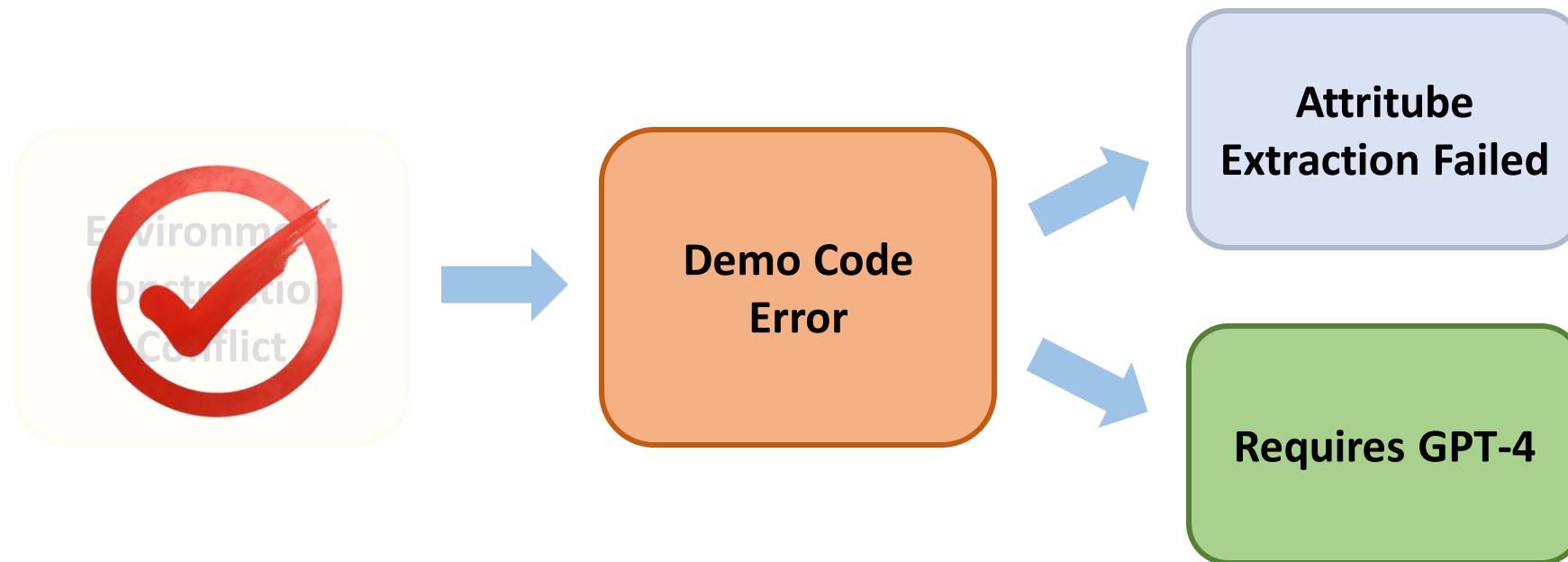
Evaporate-Code+

Base on Evaporate-Code



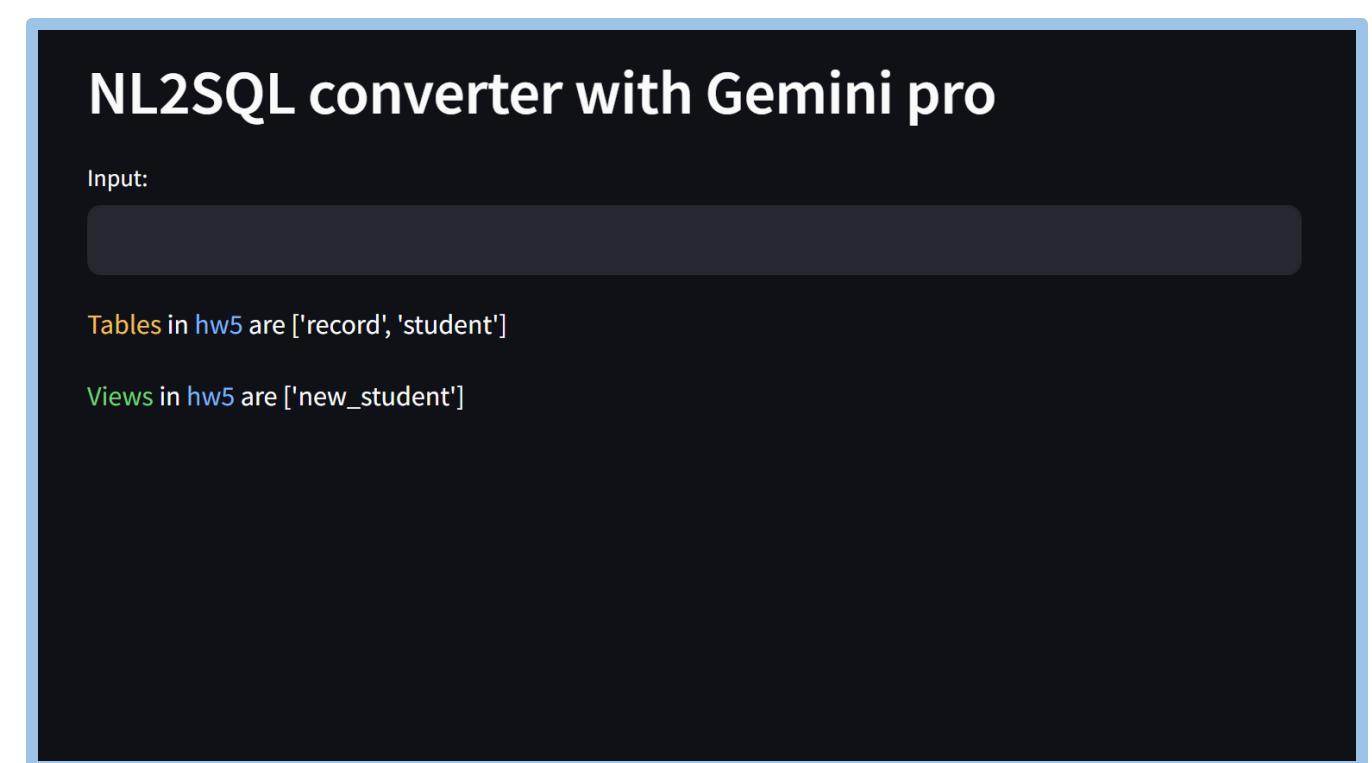
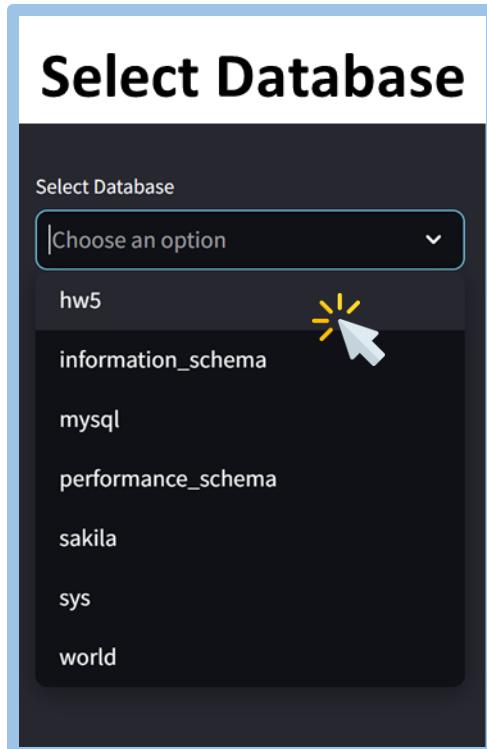
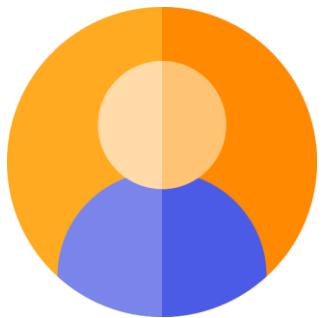
Results

- Evapoarte progress status



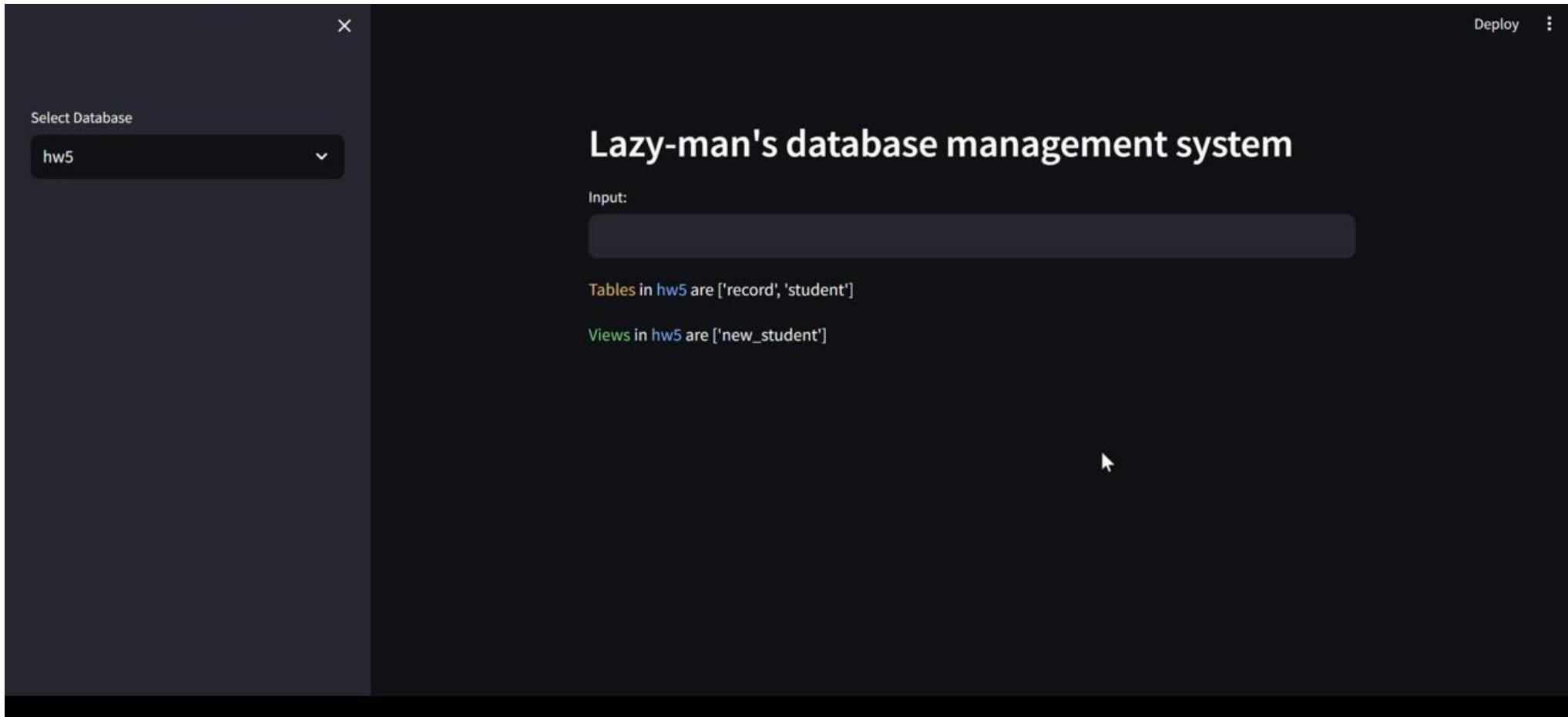
Results

- User Interface



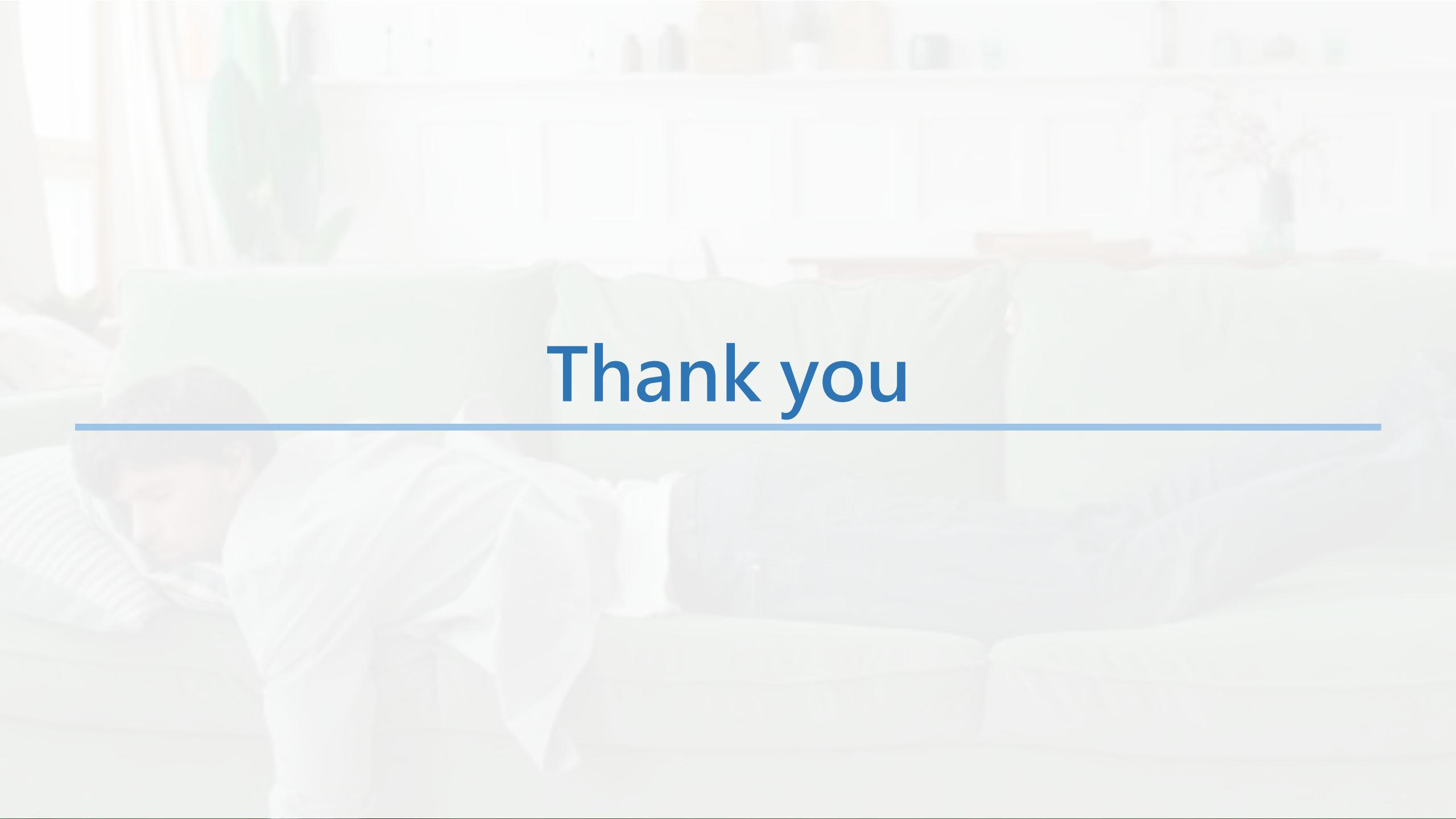
Results

- Lazy-man's demo



Future works

- Combining Evaporate into our lazy-man interface
- Integrating other extensions that can **convert data to structured data**
- Adding **query editor** function on the user interface



A faint, grayscale photograph of a person in academic attire, possibly a graduation gown and cap, is visible in the background. The person appears to be standing outdoors with some foliage or trees behind them.

Thank you
