INFOCEPTS
Data & AI

Applying MCP Principles for Schema-Aware Data Interaction

MODEL CONTEXT PROTOCOL

By Nitish Raman

WHAT IS MCP?

BACKED BY OPENAI, MICROSOFT, GOOGLE, ANTHROPIC — DESIGNED TO SIMPLIFY SCALABLE AGENT DEVELOPMENT.

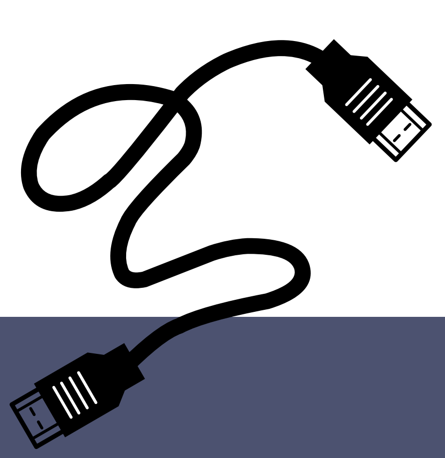
Model Context Protocol = Way to plug **Tools** + **Data** + **Prompts** together for LLMs

📂 **Resources** → What the model can access

🧰 **Tools** → What it can do

🧾 **Prompt Wrappers** → How it is guided, ensure context don't change

🧠 **LLMs use MCP** discover, plan & execute multi-step tasks



🔌The USB-C of AI agents.

- ANTHROP\C

WHY

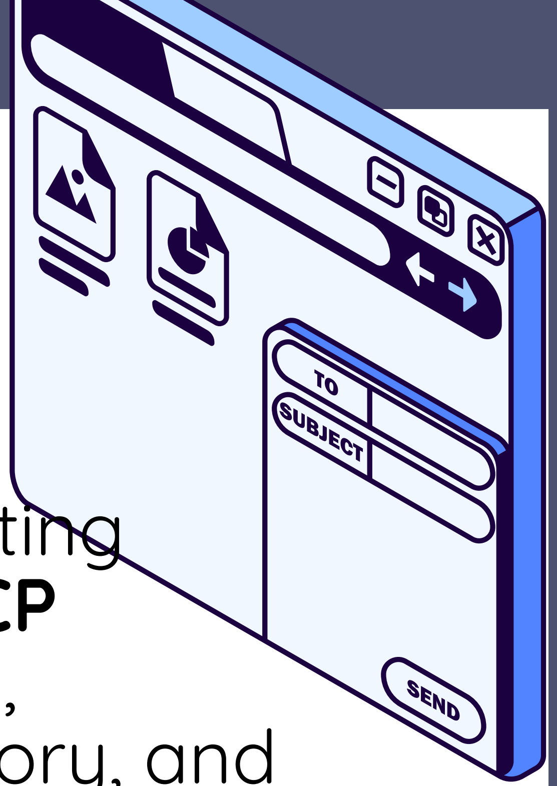
📈 **AI usage is booming**, but enterprise integration is still messy and manual.

🛠️ The old way relies on **hardcoded prompts and toolchains**, limiting flexibility.

✨ **MCP introduce clean, modular standard** for connecting LLMs to enterprise systems.

🚀 It enables **safe, reusable, plug-and-play workflows** at scale.

HOW



🧠 The **LLM acts as brain**, generating plans based on prompts, while **MCP functions as the nervous system**, connecting tools, managing memory, and executing actions dynamically.

🧩 Each layer — tools, wrappers, resources — is **modular and swappable**.

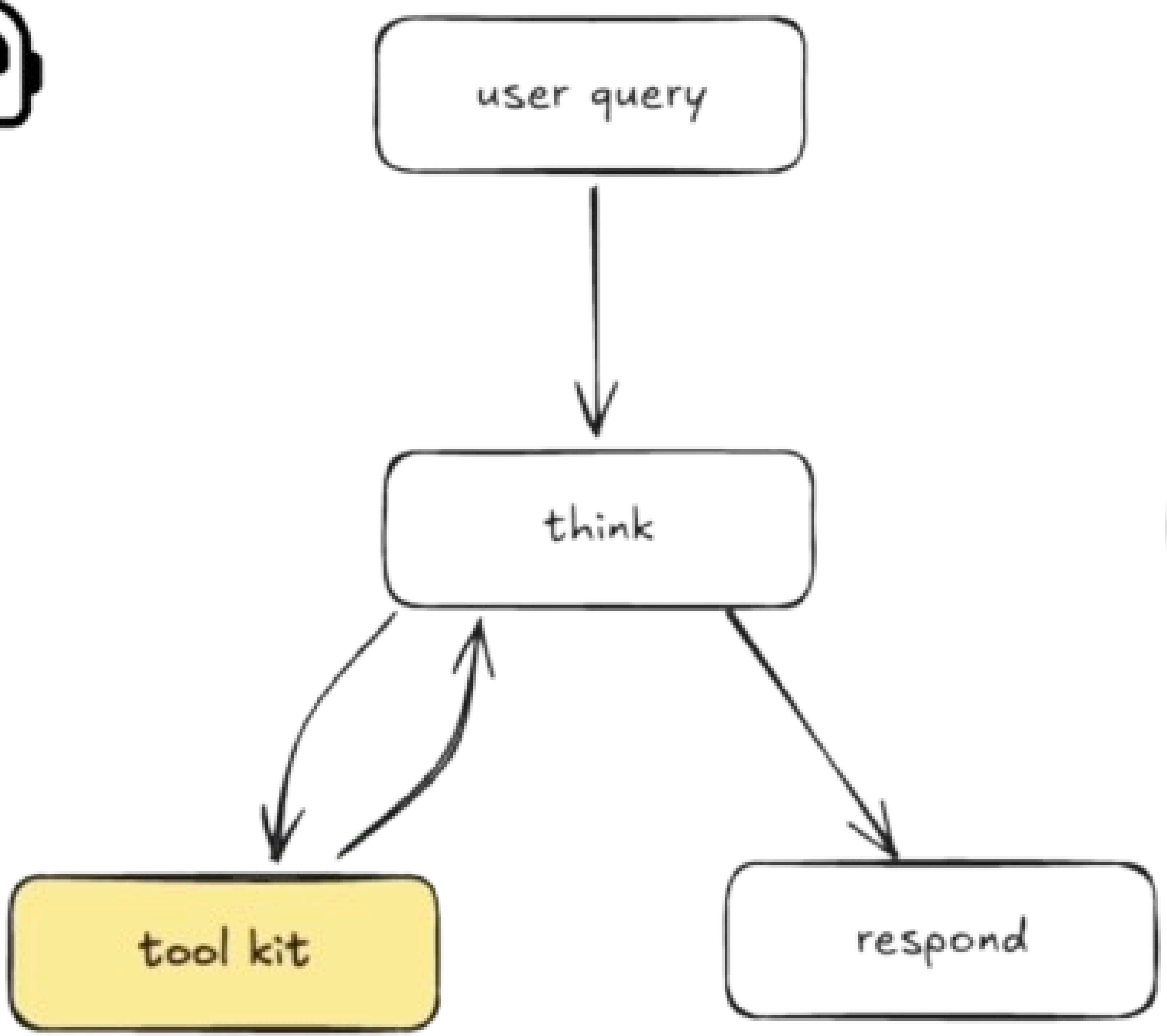
🔄 **Agents stay consistent** even when tools change.

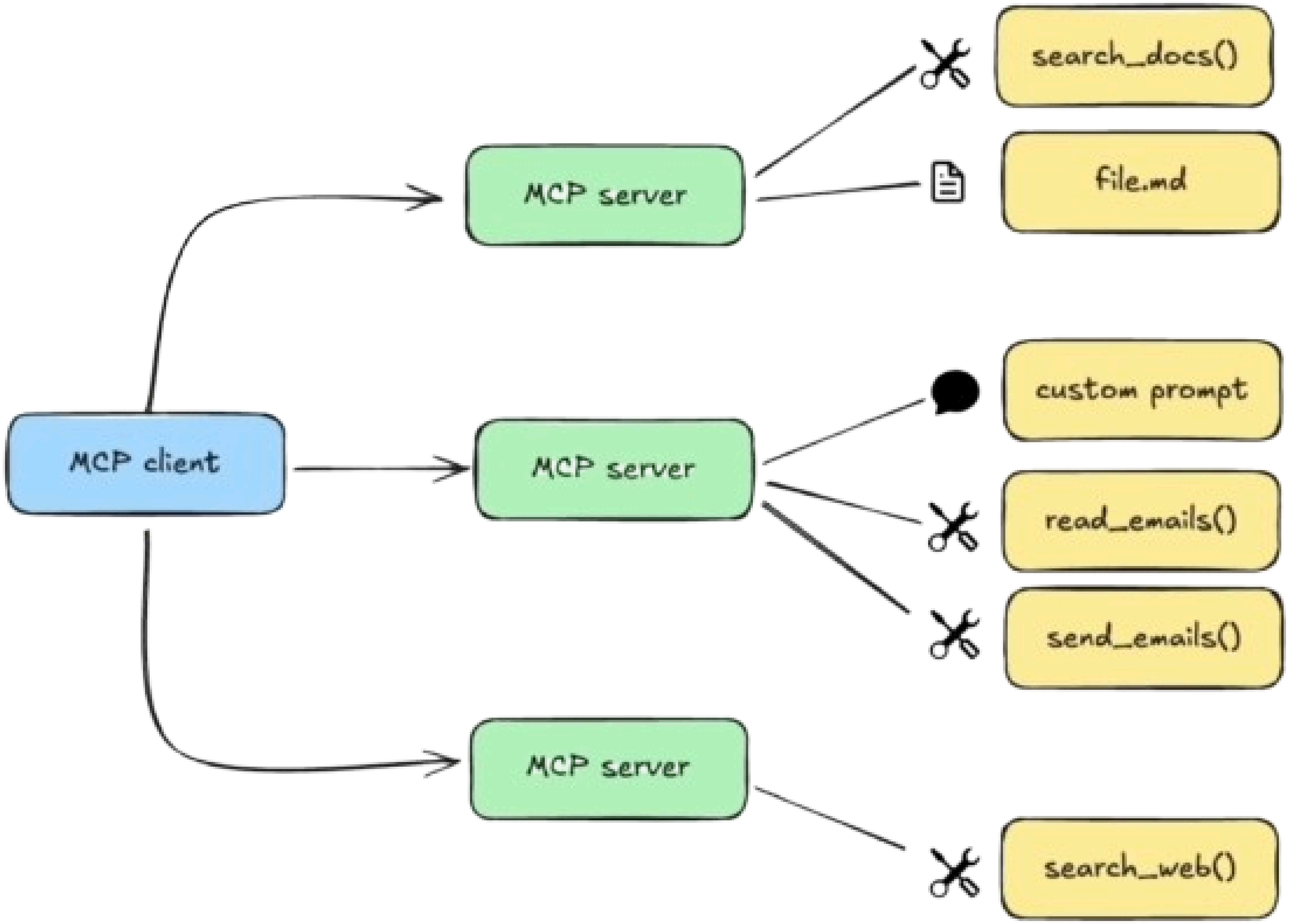
🧠 The **LLM plans**, while MCP c**onnects, discovers, and executes**.

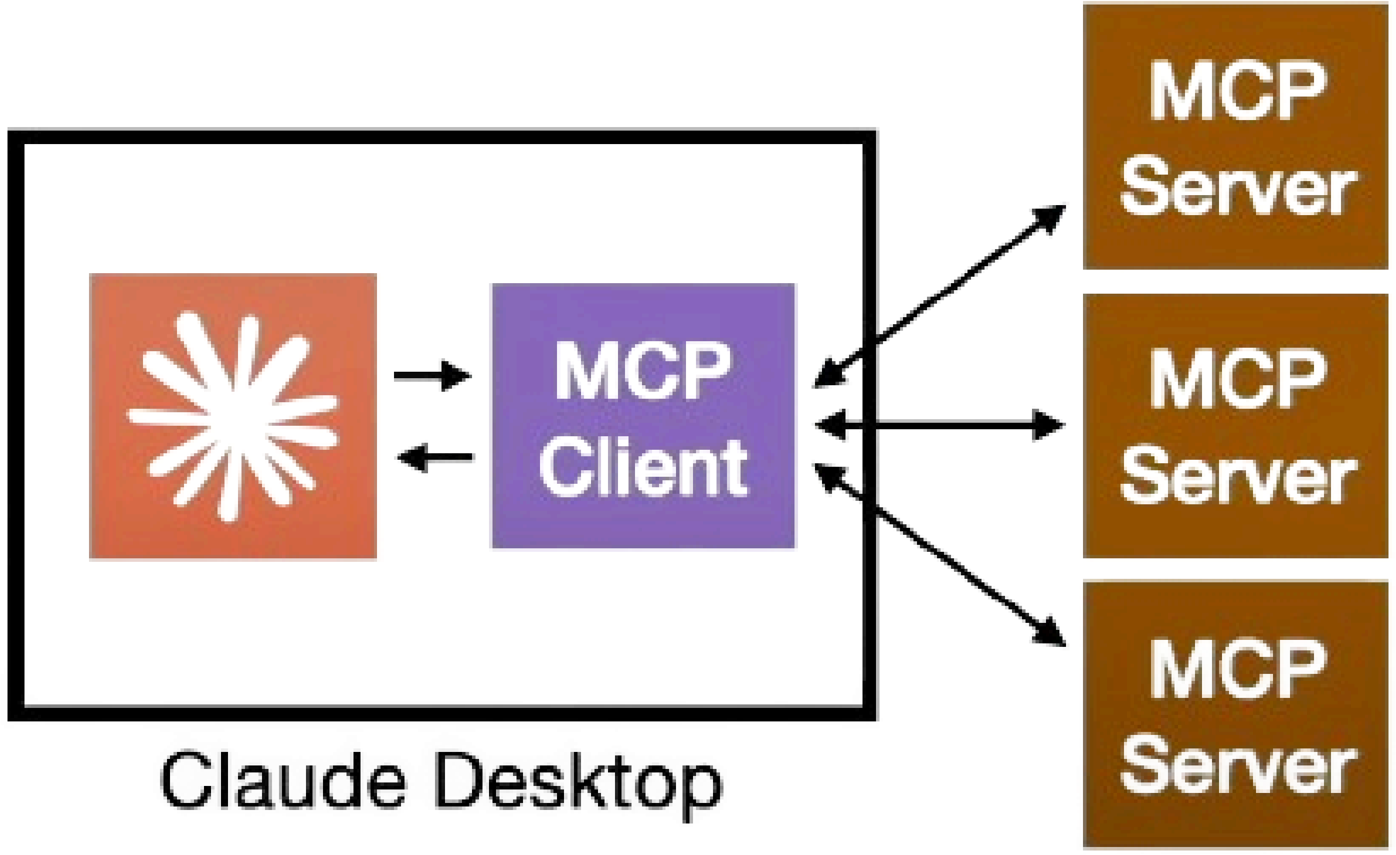
MCP WORKING

🌐 Similar to how **HTTP standardizes communication between browsers & servers**, MCP standardizes how LLMs interact with tools and data — becoming the **backbone of intelligent LLM agents by standardizing how they access tools, data, & context**.

🤔 If HTTP connects browsers to the web, MCP can connect LLMs to the world.


user query
think
tool kit
respond


MCP client
MCP server
MCP server
MCP server
search_docs()
file.md
custom prompt
read_emails()
send_emails()
search_web()



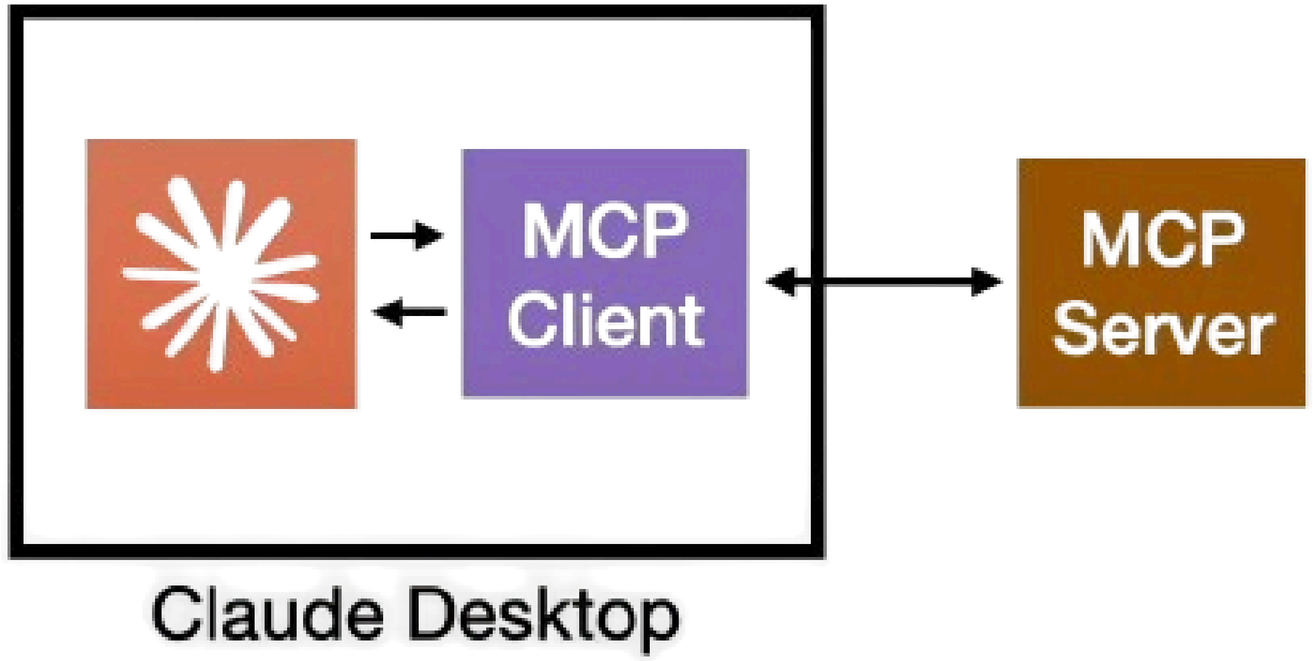
Client Responsibilities

🔍 Discover server capabilities

💿 Receive data from servers

🪛 Manage LLM tool execution

Typically don't need to built this



3 Key Services

📄 **Prompt** = prompt templates

📚 **Resource** = data, filesystem, database

🪛 **Tool** = function, API, image processing

2 Default Transports

Stdio (local)

*to communicate via HTTP with Server-Sent Events (SSE) (remote)

BENEFITS OF MCP

- 🧩 **Interoperability** — connect any tool or resource
- 🔄 **Reusability** — same agent, many tasks
- ⚙️ **Less prompt engineering** — wrappers do the work
- 📈 **Scalable agent design** — standard flows
- 🔐 **Safer, auditable workflows** — logs + control

OpenAI GPTs (Function Calling)

ChatGPT uses structured function calls & context files to trigger tools like calculators, file readers, or APIs.

It reflects MCP by combining resources, tools, and wrappers in a standard, reusable way.

Claude by Anthropic

Claude agents reason through tasks using tool-calling and self-planning logic.

This matches MCP's goal of giving LLMs autonomy with structured context and tool access.

LangChain + LangGraph (Open Source)

ChLangChain lets developers register tools and memory for LLM workflows.

LangGraph orchestrates tool sequences. Together, they model MCP's modular, discoverable pipeline.

CHALLENGES

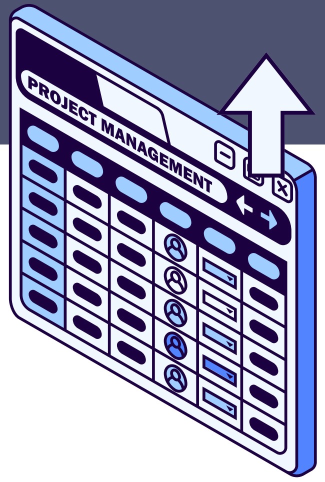
LLMs are great at text — but struggle with structured files (CSV, DB, XML)

Analysts still rely on manual SQL, Excel transformations

Goal: Can we help LLMs understand & use real-world datasets?

AI IS POWERFUL—BUT IT NEEDS CONTEXT + TOOLS TO BE USEFUL.

PROJECT



Key Feature

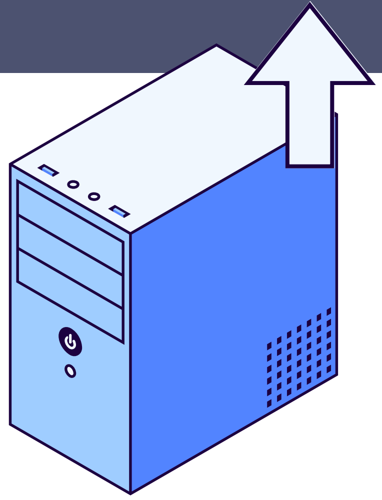
📁 **Upload dataset** (CSV, Excel, JSON, ZIP, DB).

🧠 **LLM-powered** column summary & key/tag inference.

💬 **Prompt to SQL** with chat history.

📊 **Chart builder** for visualisation.

🛢 **Optional Supabase** export.



Tech Stack

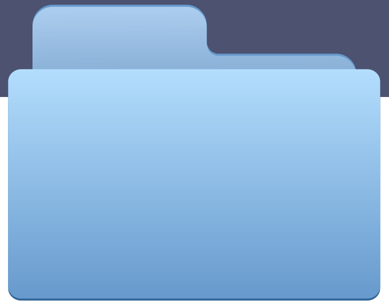
💻 UI built using **Streamlit**.

⚙️ Backend powered by **FastAPI**.

🧬 Vector search via **ChromaDB** with SentenceTransformers.

🤖 LLM integration with **Ollama**, **OpenAI**, or **GPT4All**.

📀 Data storage handled through **SQLite**.



Folder Overview

```
project_root/
├── app/              # Python logic modules
├── pages/            # Streamlit interface tabs
├── mcp_server/       # Output
├── sample_datasets/  # Example
                        Input
├── home.py
├── run.bat           # Launch script
└── README.md
└── requirements.txt
```

▶️To launch the MCP Server, run the `run.bat`file located in the root folder(`mcp_server_project`

STREAMLIT PIPELINE

Output

📦 dataset.db (SQLite database)
📊 CSV/Excel preview and ingest

🧾 schema_description.txt
🗂️ metadata.json with tags, primary keys, foreign keys
🧠 Chroma vector index

📝 query_log.txt & chat_log.txt

📈 Optional ER diagram image

1⃣ **Upload & Preview**: Select & preview dataset (CSV, Excel, JSON, ZIP, DB)

2⃣ **Describe Schema**: Generate LLM-powered schema summary, PK/FK tags

3⃣ **Prompt → SQL**: Convert natural prompt into SQL and view result

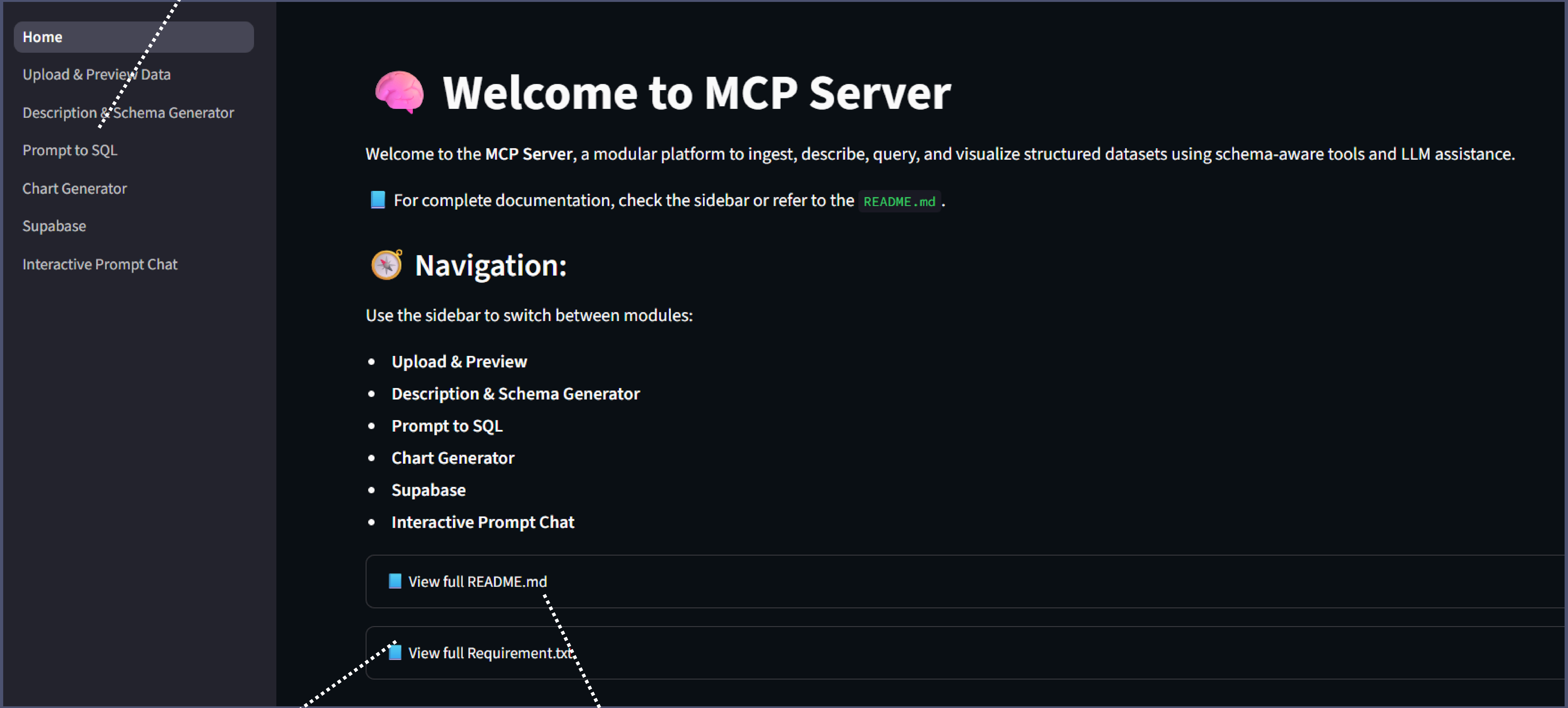
4⃣ **Chart Generator**: Select chart type and render using schema columns

5⃣ **Supabase**: Pull Supabase/PostgreSQL for analysis

6⃣ **Interactive Chat**: Ongoing chat with schema memory

📥 Upload → 📄 Parse File → 🧠 LLM Describe → 🧬 Vectorize → 💬 ChatSQL → 📊 Visualize

Navigate to Different Mcp Server



🧠 Welcome to MCP Server

Welcome to the **MCP Server**, a modular platform to ingest, describe, query, and visualize structured datasets using schema-aware tools and LLM assistance.

📘 For complete documentation, check the sidebar or refer to the `README.md`.

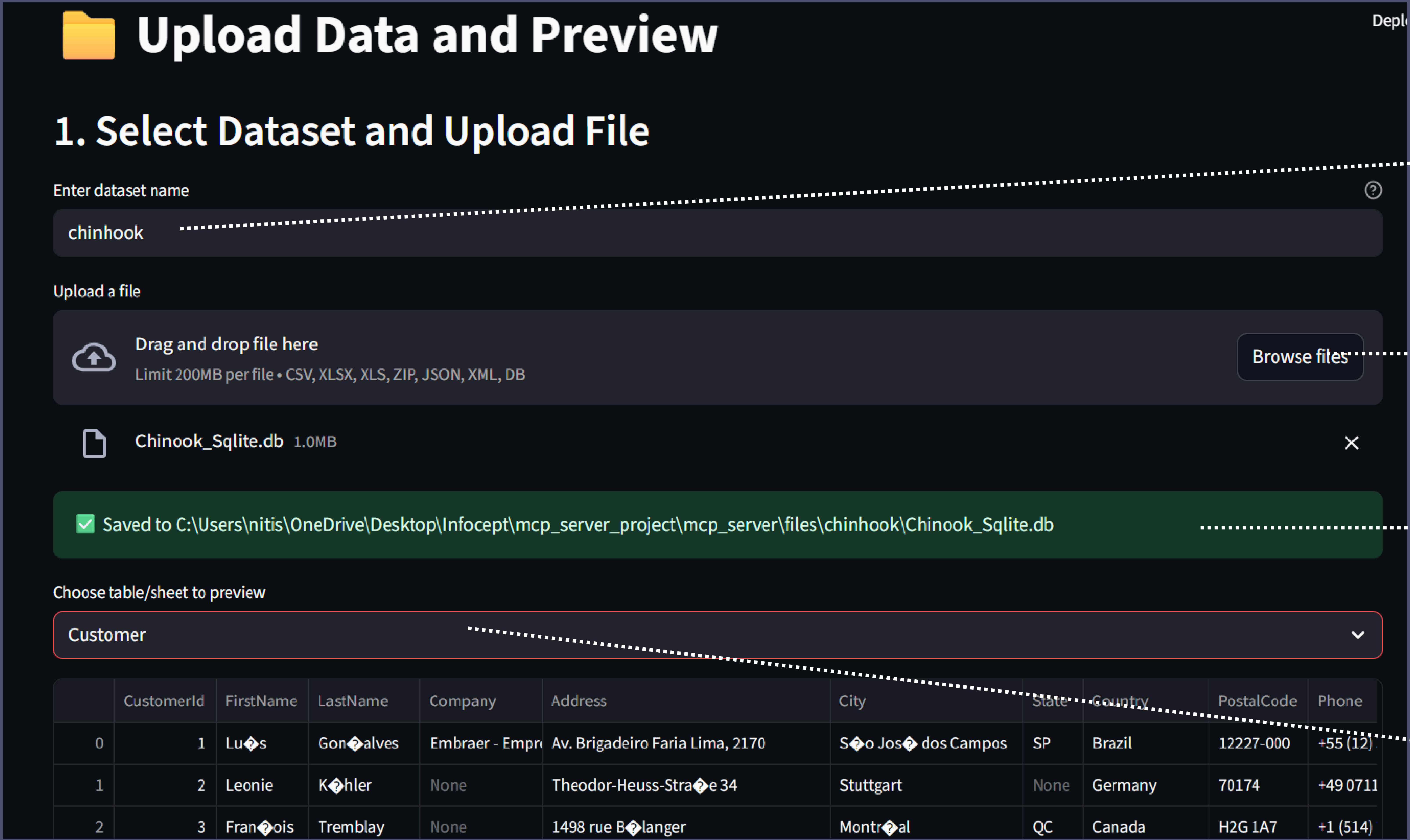
🧭 Navigation:

Use the sidebar to switch between modules:

- **Upload & Preview**
- **Description & Schema Generator**
- **Prompt to SQL**
- **Chart Generator**
- **Supabase**
- **Interactive Prompt Chat**

Click to see Requirement

Click for Readme Text



📁 Upload Data and Preview

1. Select Dataset and Upload File

Enter dataset name

chinhook

Upload a file

Drag and drop file here

Limit 200MB per file • CSV, XLSX, XLS, ZIP, JSON, XML, DB

Chinook_Sqlite.db 1.0MB

✅ Saved to C:\Users\nitis\OneDrive\Desktop\Infocept\mcp_server_project\mcp_server\files\chinhook\Chinook_Sqlite.db

Choose table/sheet to preview

Customer

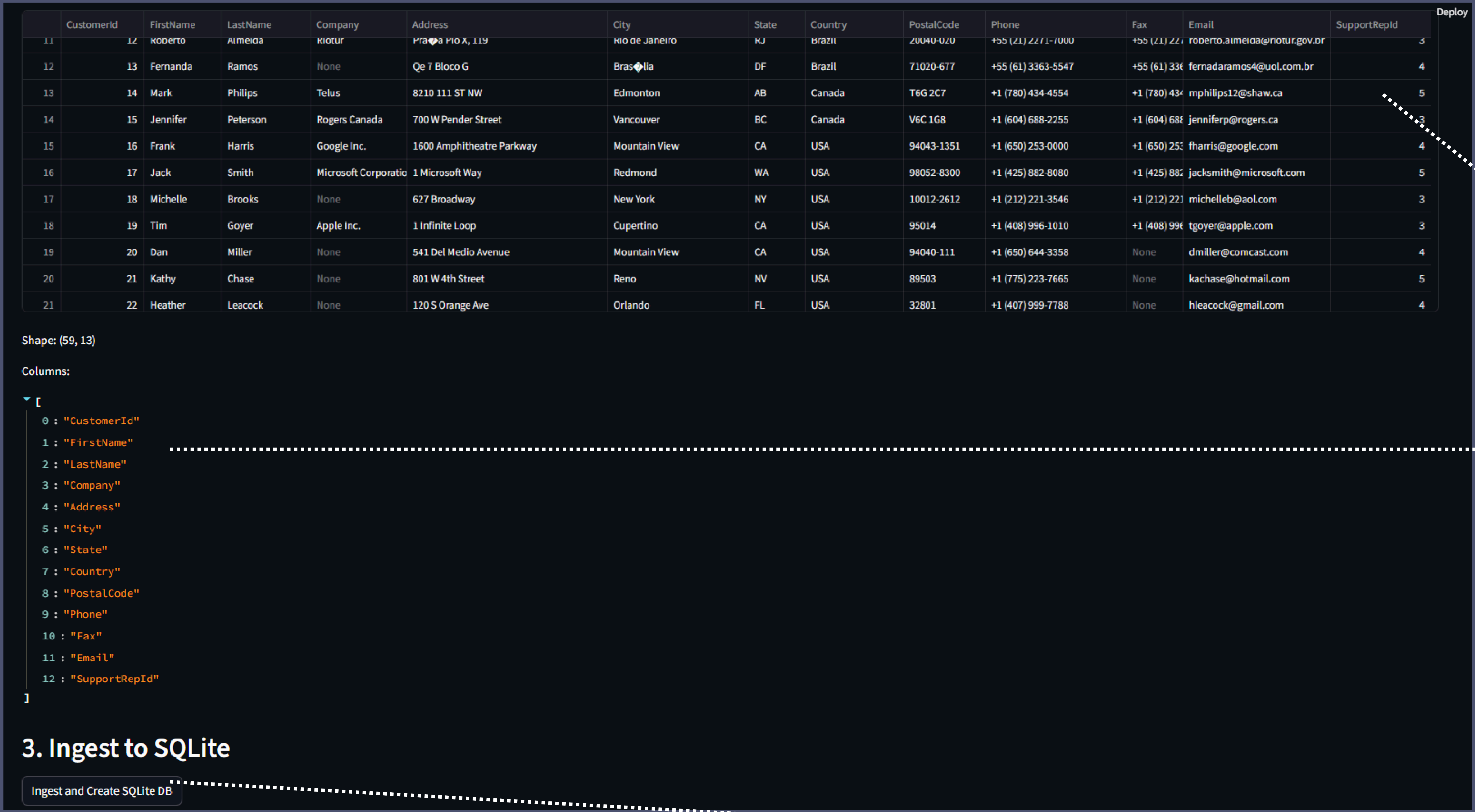
	CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	Phone
0	1	Lu�s	Gon�alves	Embraer - Empr	Av. Brigadeiro Faria Lima, 2170	S�o Jos� dos Campos	SP	Brazil	12227-000	+55 (12)
1	2	Leonie	K�hler	None	Theodor-Heuss-Stra�e 34	Stuttgart	None	Germany	70174	+49 0711
2	3	Fran�ois	Tremblay	None	1498 rue B�langer	Montr�al	QC	Canada	H2G 1A7	+1 (514)

Give Dataset Name

Browse & Upload File

Output Path

Click to Select Table



	CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	Phone	Fax	Email	SupportRepId
11	12	Roberto	Almeida	Riotur	Pra�a Pio X, 119	Rio de Janeiro	RJ	Brazil	20040-020	+55 (21) 2271-7000	+55 (21) 227	roberto.almeida@riotur.gov.br	3
12	13	Fernanda	Ramos	None	Qe 7 Bloco G	Bras�lia	DF	Brazil	71020-677	+55 (61) 3363-5547	+55 (61) 336	fernadaramos4@uol.com.br	4
13	14	Mark	Philips	Telus	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	+1 (780) 434-4554	+1 (780) 434	mphilips12@shaw.ca	5
14	15	Jennifer	Peterson	Rogers Canada	700 W Pender Street	Vancouver	BC	Canada	V6C 1G8	+1 (604) 688-2255	+1 (604) 688	jenniferp@rogers.ca	3
15	16	Frank	Harris	Google Inc.	1600 Amphitheatre Parkway	Mountain View	CA	USA	94043-1351	+1 (650) 253-0000	+1 (650) 253	fharris@google.com	4
16	17	Jack	Smith	Microsoft Corporatio	1 Microsoft Way	Redmond	WA	USA	98052-8300	+1 (425) 882-8080	+1 (425) 882	jacksmith@microsoft.com	5
17	18	Michelle	Brooks	None	627 Broadway	New York	NY	USA	10012-2612	+1 (212) 221-3546	+1 (212) 221	michelleb@aol.com	3
18	19	Tim	Goyer	Apple Inc.	1 Infinite Loop	Cupertino	CA	USA	95014	+1 (408) 996-1010	+1 (408) 996	tgoyer@apple.com	3
19	20	Dan	Miller	None	541 Del Medio Avenue	Mountain View	CA	USA	94040-111	+1 (650) 644-3358	None	dmiller@comcast.com	4
20	21	Kathy	Chase	None	801 W 4th Street	Reno	NV	USA	89503	+1 (775) 223-7665	None	kachase@hotmail.com	5
21	22	Heather	Leacock	None	120 S Orange Ave	Orlando	FL	USA	32801	+1 (407) 999-7788	None	hleacock@gmail.com	4

Shape: (59, 13)

Columns:

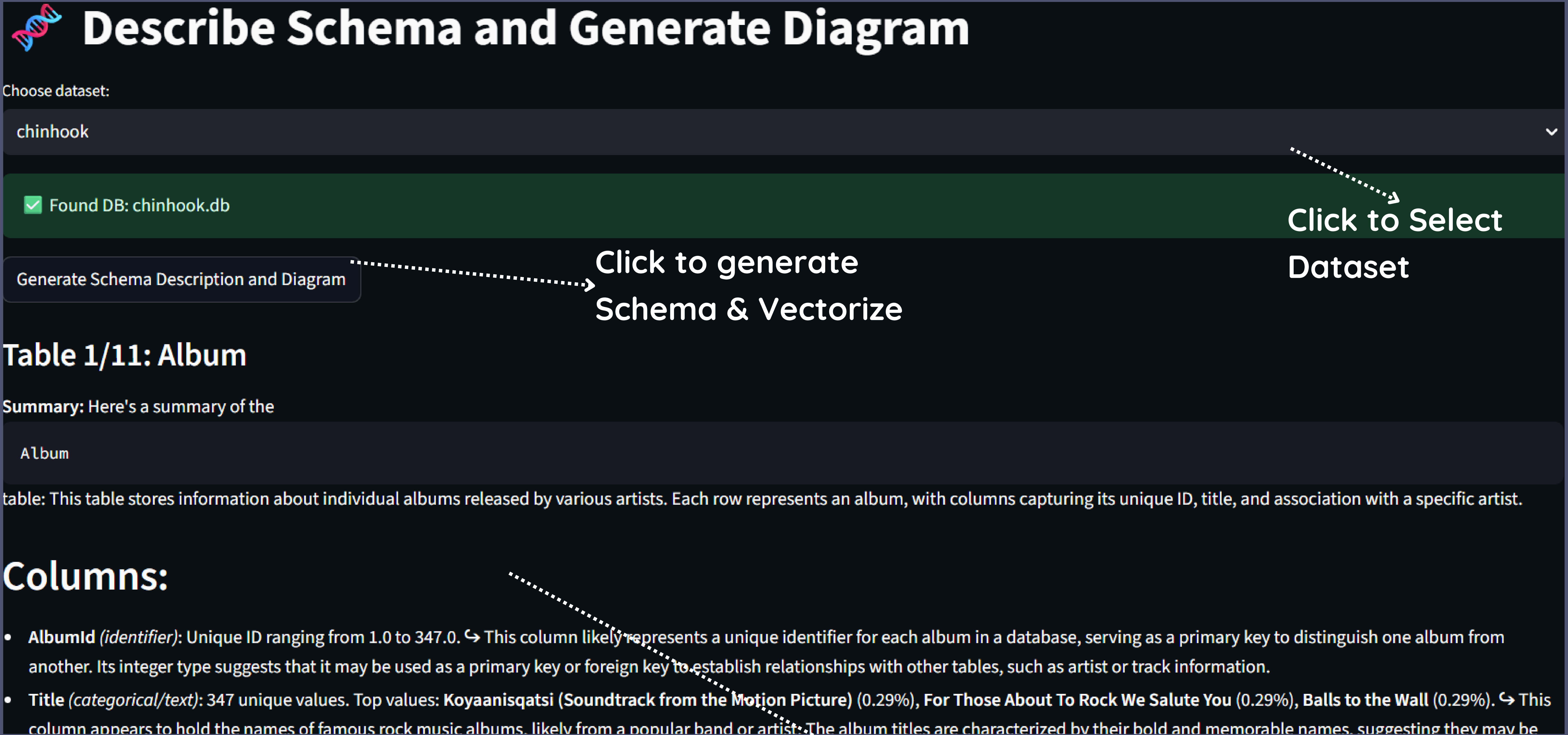
```
[
  0 : "CustomerId"
  1 : "FirstName"
  2 : "LastName"
  3 : "Company"
  4 : "Address"
  5 : "City"
  6 : "State"
  7 : "Country"
  8 : "PostalCode"
  9 : "Phone"
  10 : "Fax"
  11 : "Email"
  12 : "SupportRepId"
]
```

3. Ingest to SQLite

Table Preview

Column List

Click to Save as db File



Describe Schema and Generate Diagram

Choose dataset:

chinhook ⌄

☑ Found DB: chinhook.db

Generate Schema Description and Diagram

Table 1/11: Album

Summary: Here's a summary of the

```
Album
```

table: This table stores information about individual albums released by various artists. Each row represents an album, with columns capturing its unique ID, title, and association with a specific artist.

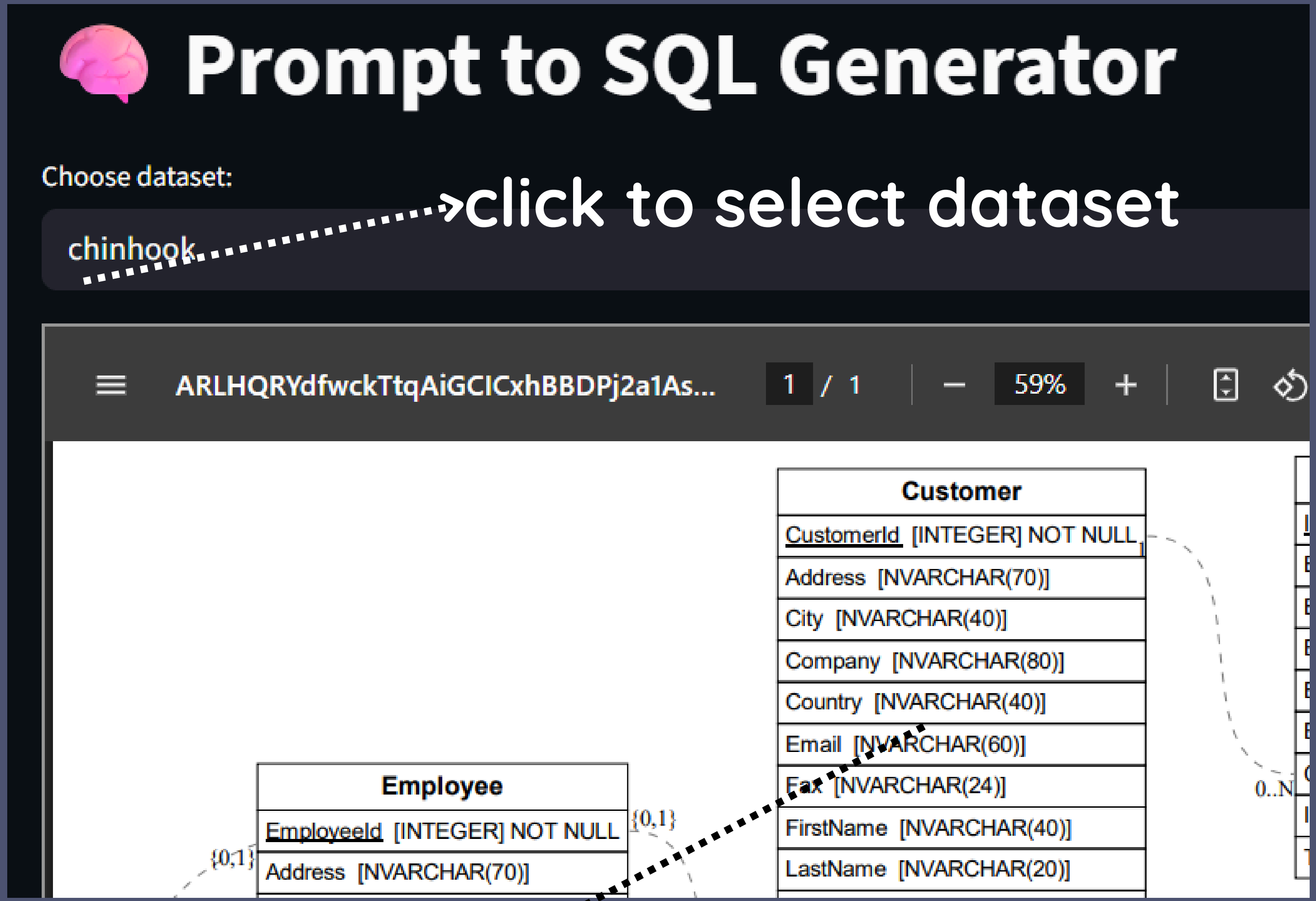
Columns:

- **AlbumId** *(identifier)*: Unique ID ranging from 1.0 to 347.0. ↪ This column likely represents a unique identifier for each album in a database, serving as a primary key to distinguish one album from another. Its integer type suggests that it may be used as a primary key or foreign key to establish relationships with other tables, such as artist or track information.
- **Title** *(categorical/text)*: 347 unique values. Top values: **Koyaanisqatsi (Soundtrack from the Motion Picture)** (0.29%), **For Those About To Rock We Salute You** (0.29%), **Balls to the Wall** (0.29%). ↪ This column appears to hold the names of famous rock music albums, likely from a popular band or artist. The album titles are characterized by their bold and memorable names, suggesting they may be

Click to Select Dataset

Click to generate Schema & Vectorize

LLM Powered Description of Tables & Columns



Prompt to SQL Generator

Choose dataset:

chinhook

Customer

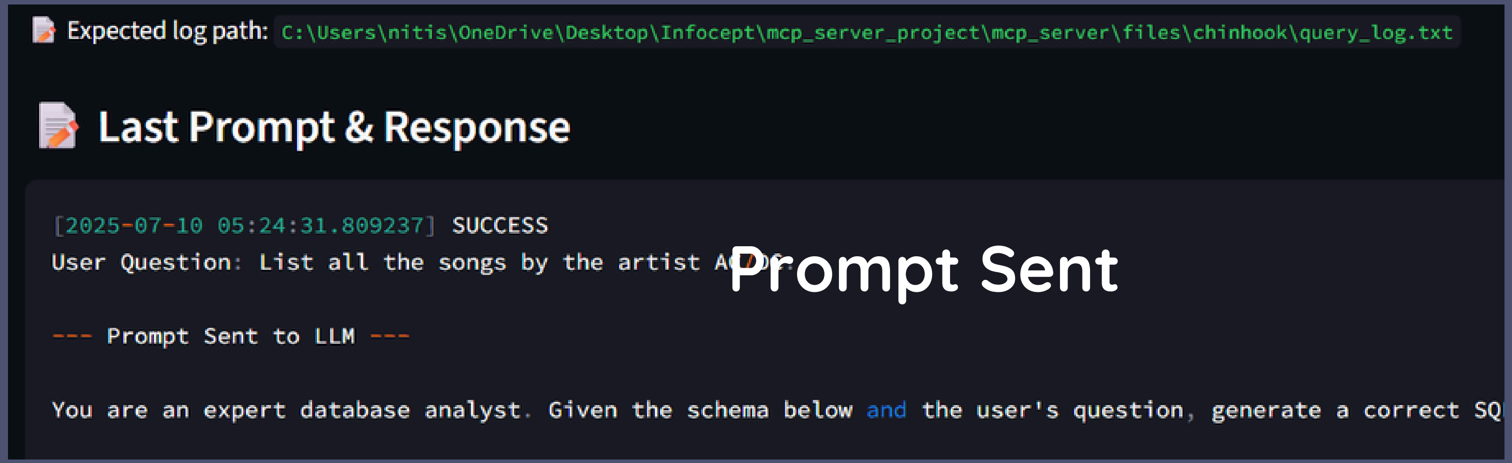
CustomerId [INTEGER] NOT NULL
Address [NVARCHAR(70)]
City [NVARCHAR(40)]
Company [NVARCHAR(80)]
Country [NVARCHAR(40)]
Email [NVARCHAR(60)]
Fax [NVARCHAR(24)]
FirstName [NVARCHAR(40)]
LastName [NVARCHAR(20)]

Employee

EmployeeId [INTEGER] NOT NULL
Address [NVARCHAR(70)]

click to select dataset

ER Diagram to understand structure



Expected log path: `C:\Users\nitis\OneDrive\Desktop\Infocept\mcp_server_project\mcp_server\files\chinhook\query_log.txt`

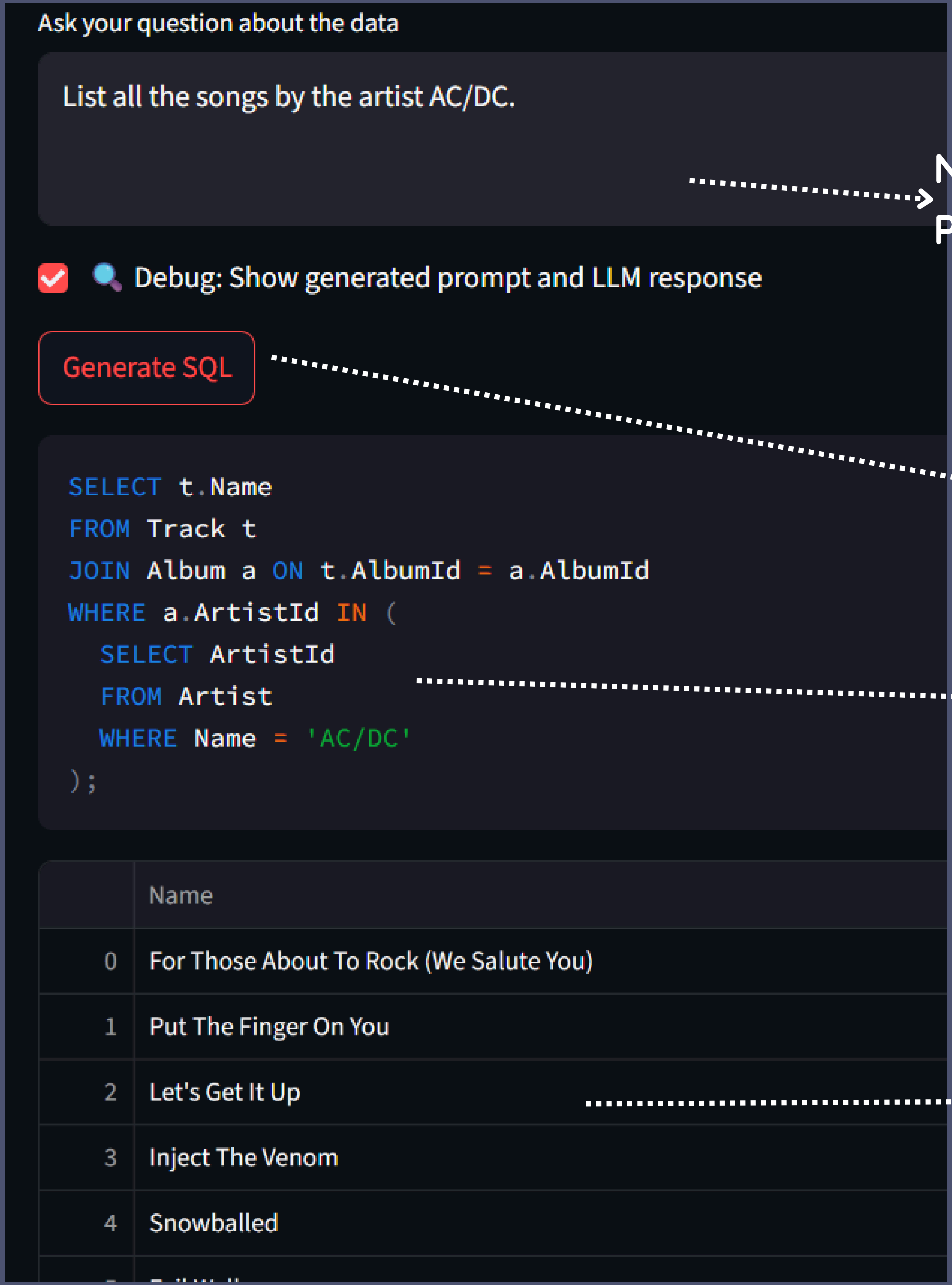
Last Prompt & Response

```
[2025-07-10 05:24:31.809237] SUCCESS
User Question: List all the songs by the artist AC/DC.

--- Prompt Sent to LLM ---

You are an expert database analyst. Given the schema below and the user's question, generate a correct SQ
```

Prompt Sent



Ask your question about the data

List all the songs by the artist AC/DC.

☑ 🔍 Debug: Show generated prompt and LLM response

```sql
SELECT t.Name
FROM Track t
JOIN Album a ON t.AlbumId = a.AlbumId
WHERE a.ArtistId IN (
  SELECT ArtistId
  FROM Artist
  WHERE Name = 'AC/DC'
);
```

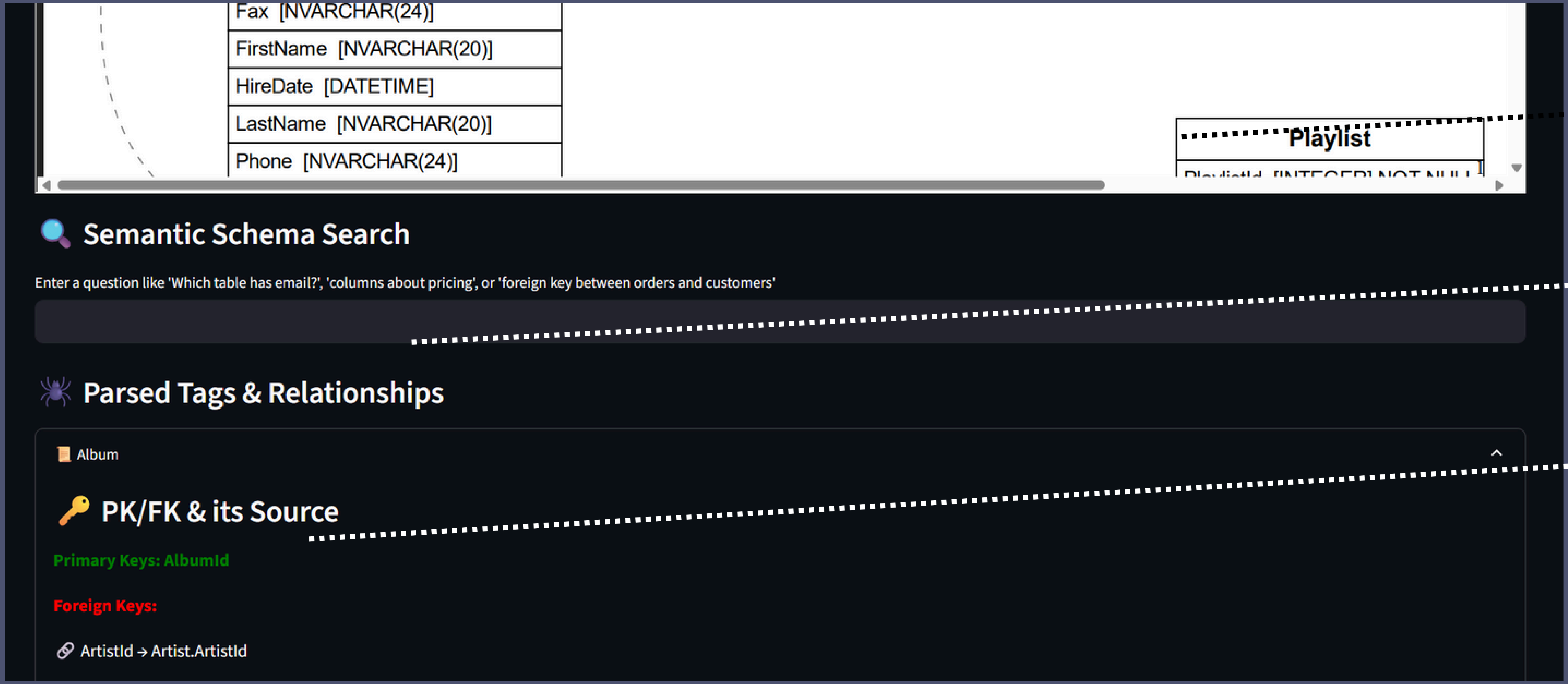
	Name
0	For Those About To Rock (We Salute You)
1	Put The Finger On You
2	Let's Get It Up
3	Inject The Venom
4	Snowballed

Natural Language Prompt by User

Click to generate SQL

Generated SQL

Result



🔍 Semantic Schema Search

Enter a question like 'Which table has email?', 'columns about pricing', or 'foreign key between orders and customers'

Parsed Tags & Relationships

Album

🔑 PK/FK & its Source

Primary Keys: AlbumId

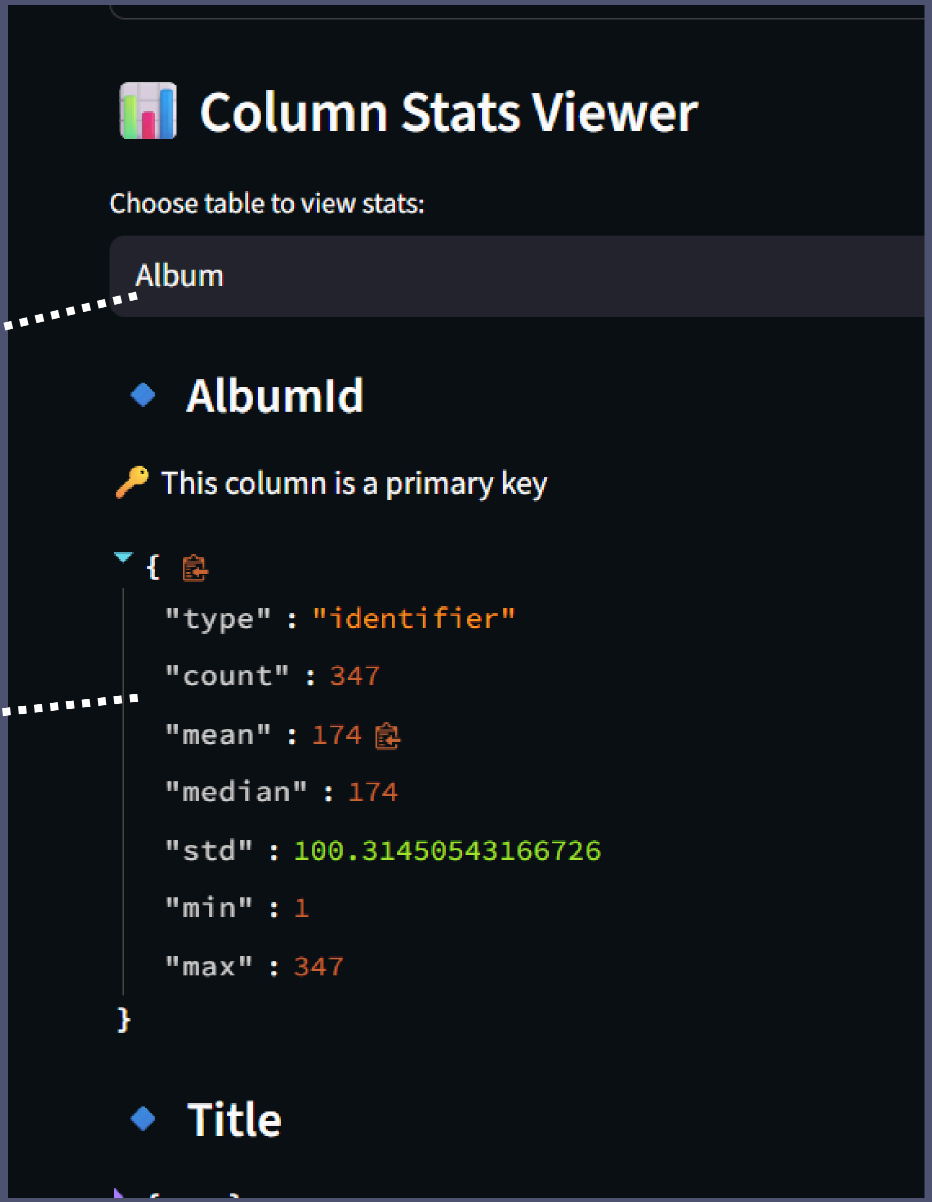
Foreign Keys:

🔗 ArtistId → Artist.ArtistId

ER Diagram using LLM

Semantic Search for Vector Db

Tags & Relationship for each table using LLM



📊 Column Stats Viewer

Choose table to view stats:

Album

◆ AlbumId

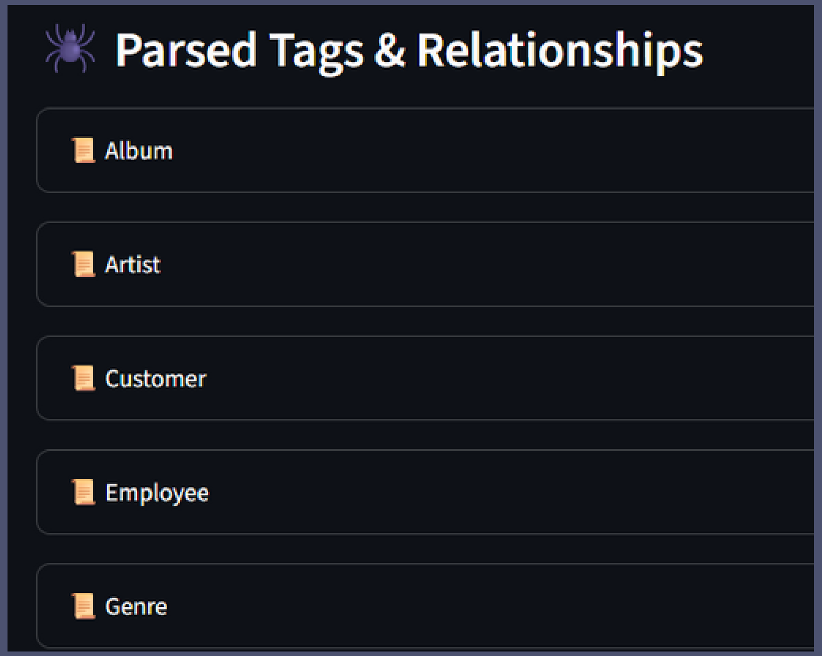
🔑 This column is a primary key

```
{
  "type" : "identifier"
  "count" : 347
  "mean" : 174
  "median" : 174
  "std" : 100.31450543166726
  "min" : 1
  "max" : 347
}
```

◆ Title

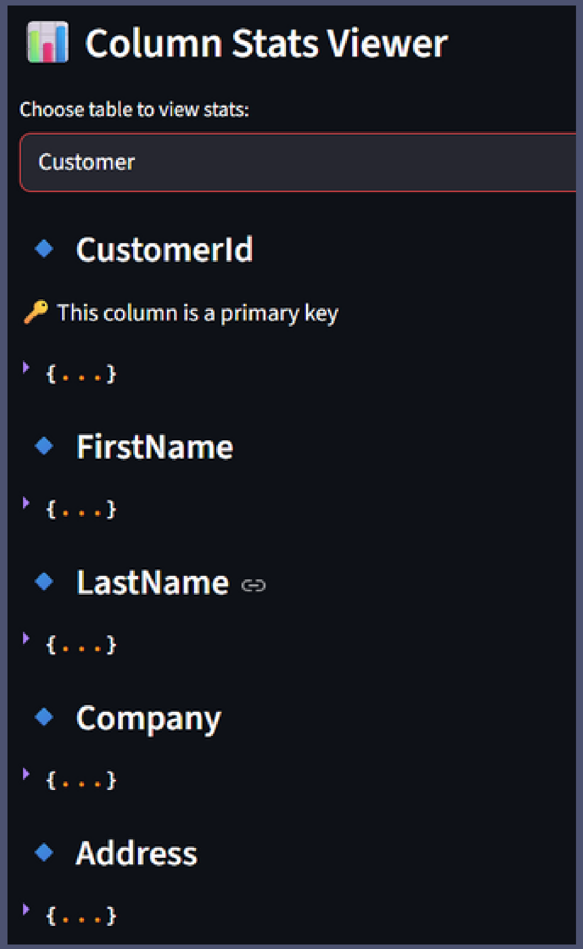
Click to Select Table

Browse to see Column Statistics



Parsed Tags & Relationships

- Album
- Artist
- Customer
- Employee
- Genre



📊 Column Stats Viewer

Choose table to view stats:

Customer

◆ CustomerId

🔑 This column is a primary key

```
{...}
```

◆ FirstName

```
{...}
```

◆ LastName

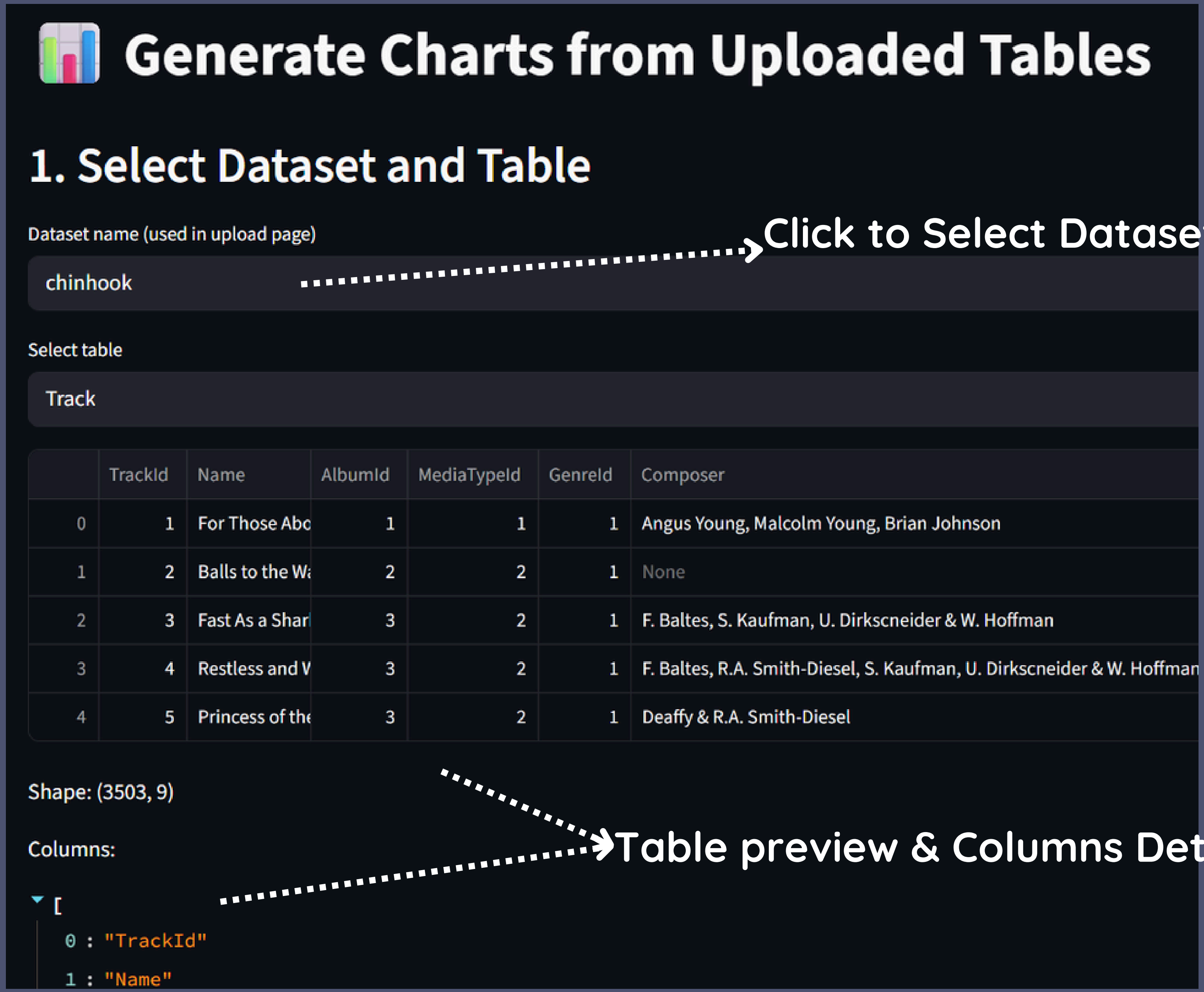
```
{...}
```

◆ Company

```
{...}
```

◆ Address

```
{...}
```



Generate Charts from Uploaded Tables

1. Select Dataset and Table

Dataset name (used in upload page)

chinhook

Click to Select Dataset

Select table

Track

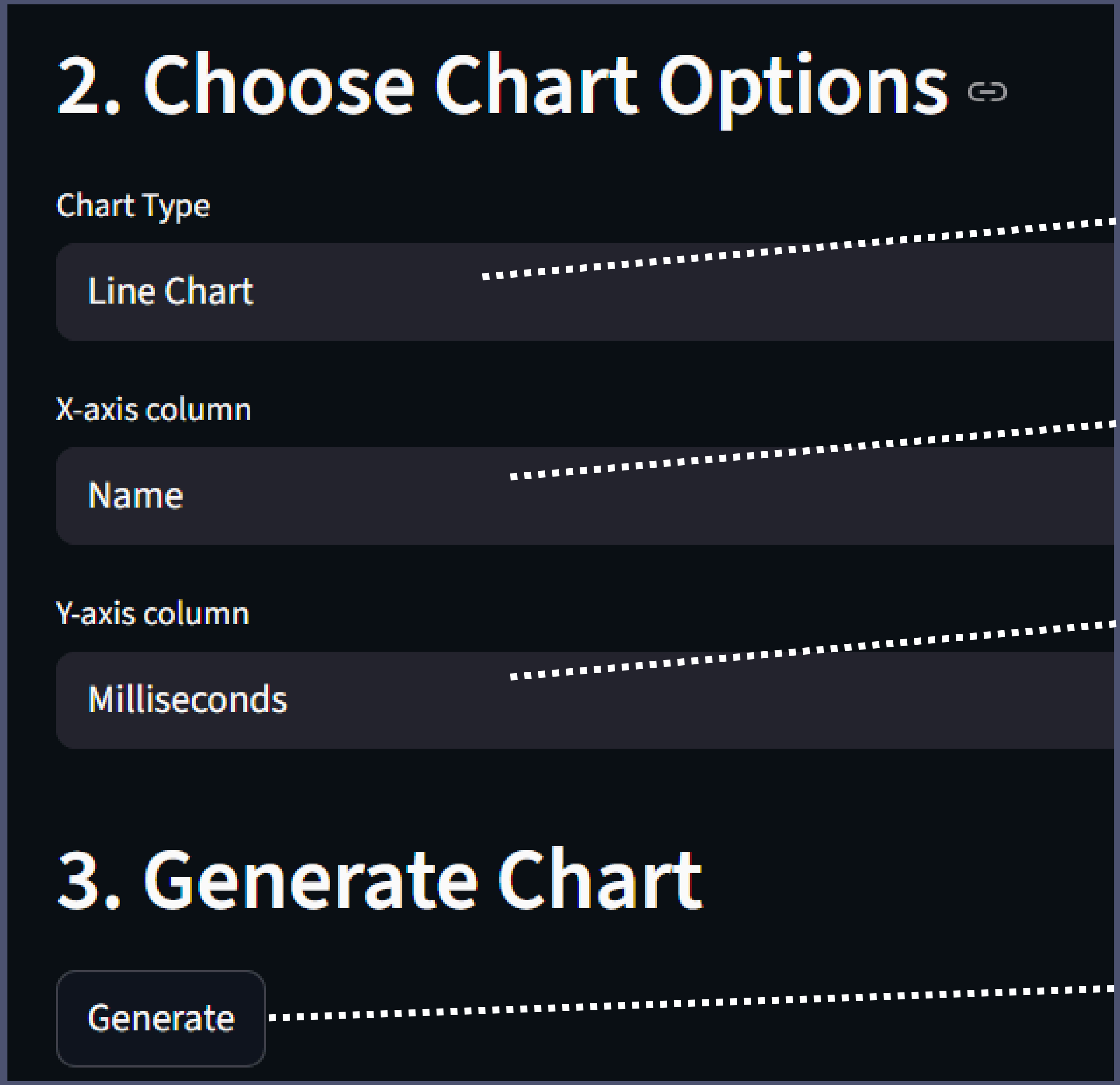
	TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer
0	1	For Those Abo	1	1	1	Angus Young, Malcolm Young, Brian Johnson
1	2	Balls to the Wa	2	2	1	None
2	3	Fast As a Shar	3	2	1	F. Baltes, S. Kaufman, U. Dirkscneider & W. Hoffman
3	4	Restless and V	3	2	1	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. Dirkscneider & W. Hoffman
4	5	Princess of the	3	2	1	Deaffy & R.A. Smith-Diesel

Shape: (3503, 9)

Columns:

```
[
  0 : "TrackId"
  1 : "Name"
```

Table preview & Columns Detail



2. Choose Chart Options

Chart Type

Line Chart

Click to select Chart Type

X-axis column

Name

Click to select X-axis Column

Y-axis column

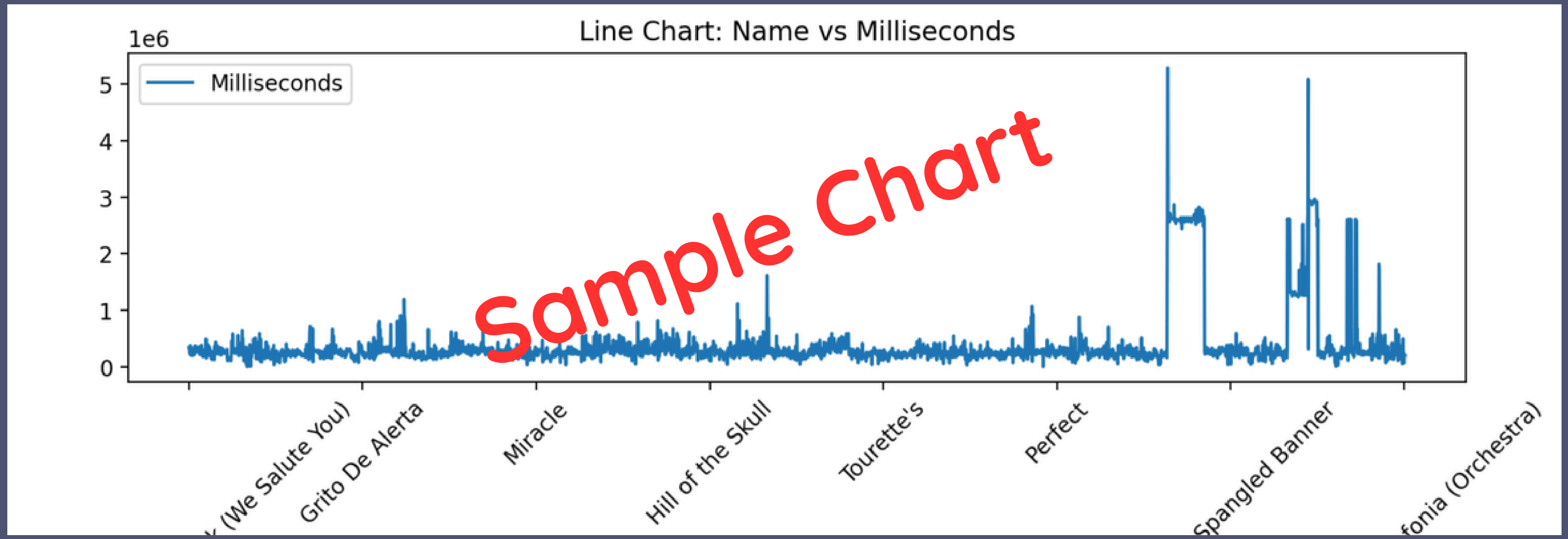
Milliseconds

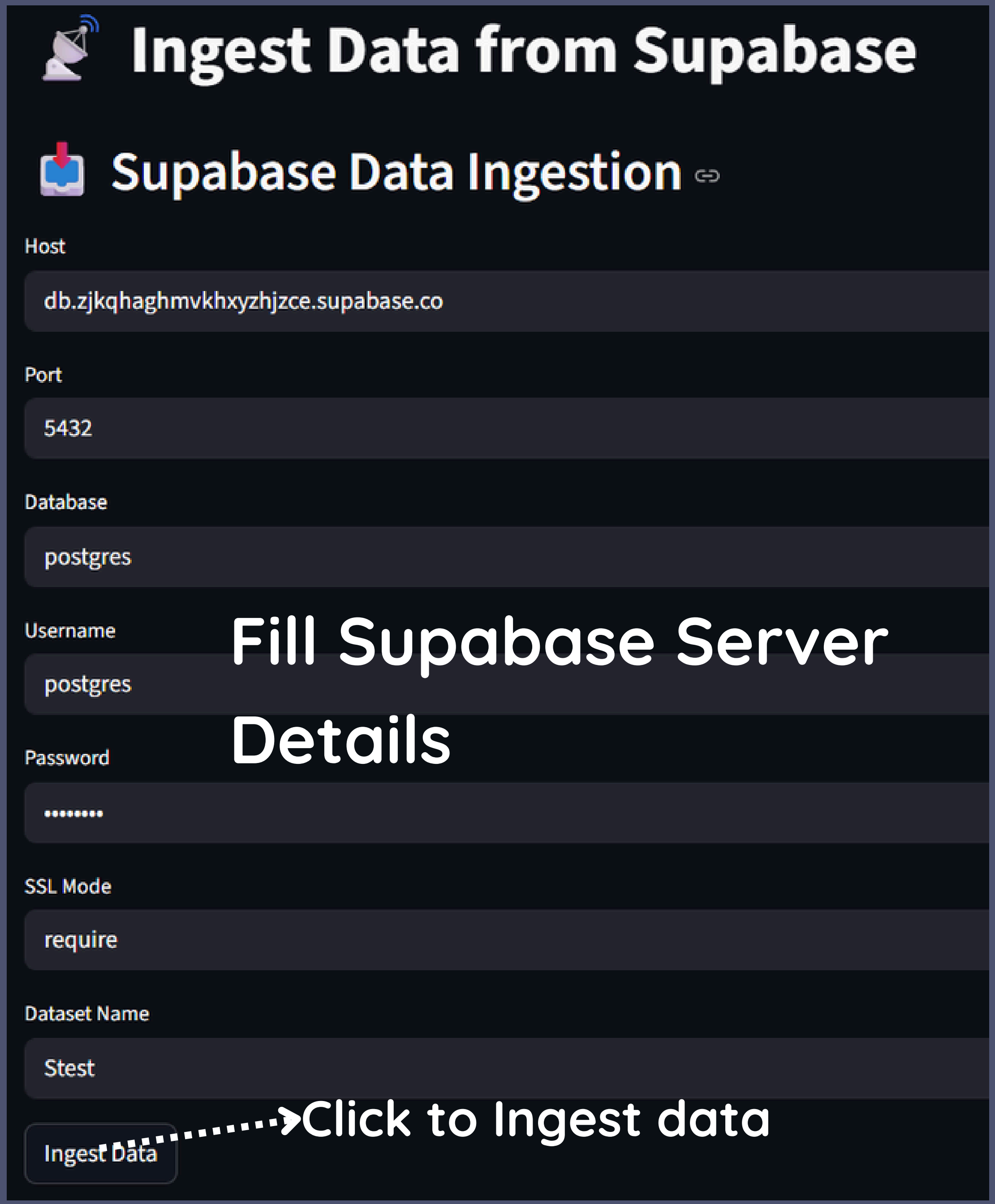
Click to select Y-axis Column

3. Generate Chart

Generate

Click to generate Chart





Ingest Data from Supabase

Supabase Data Ingestion

Host

db.zjkqhaghmvkhxyzhjzce.supabase.co

Port

5432

Database

postgres

Username

postgres

Password

••••••••

SSL Mode

require

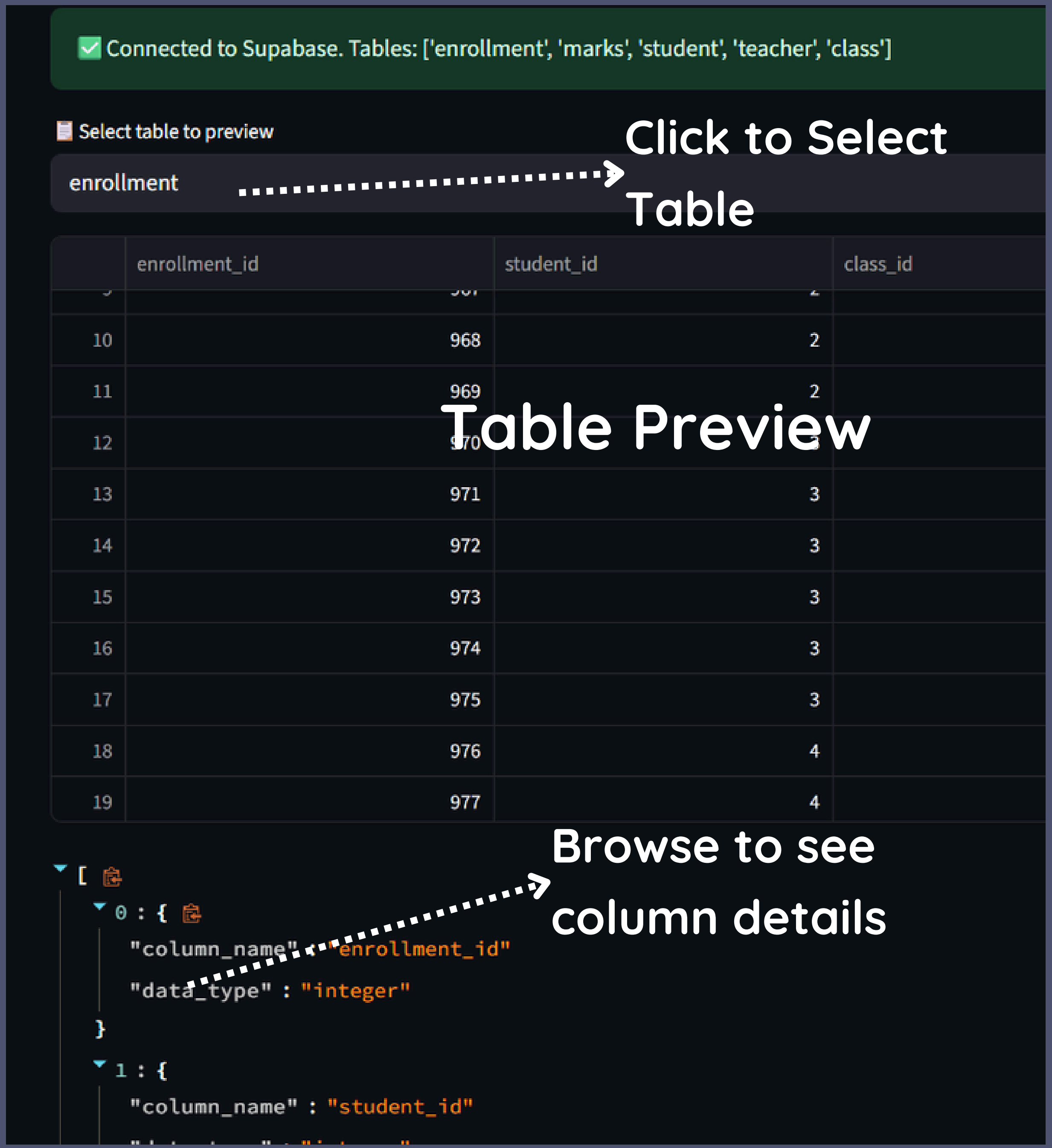
Dataset Name

Stest

Ingest Data

Fill Supabase Server Details

Click to Ingest data



✅ Connected to Supabase. Tables: ['enrollment', 'marks', 'student', 'teacher', 'class']

Select table to preview

enrollment

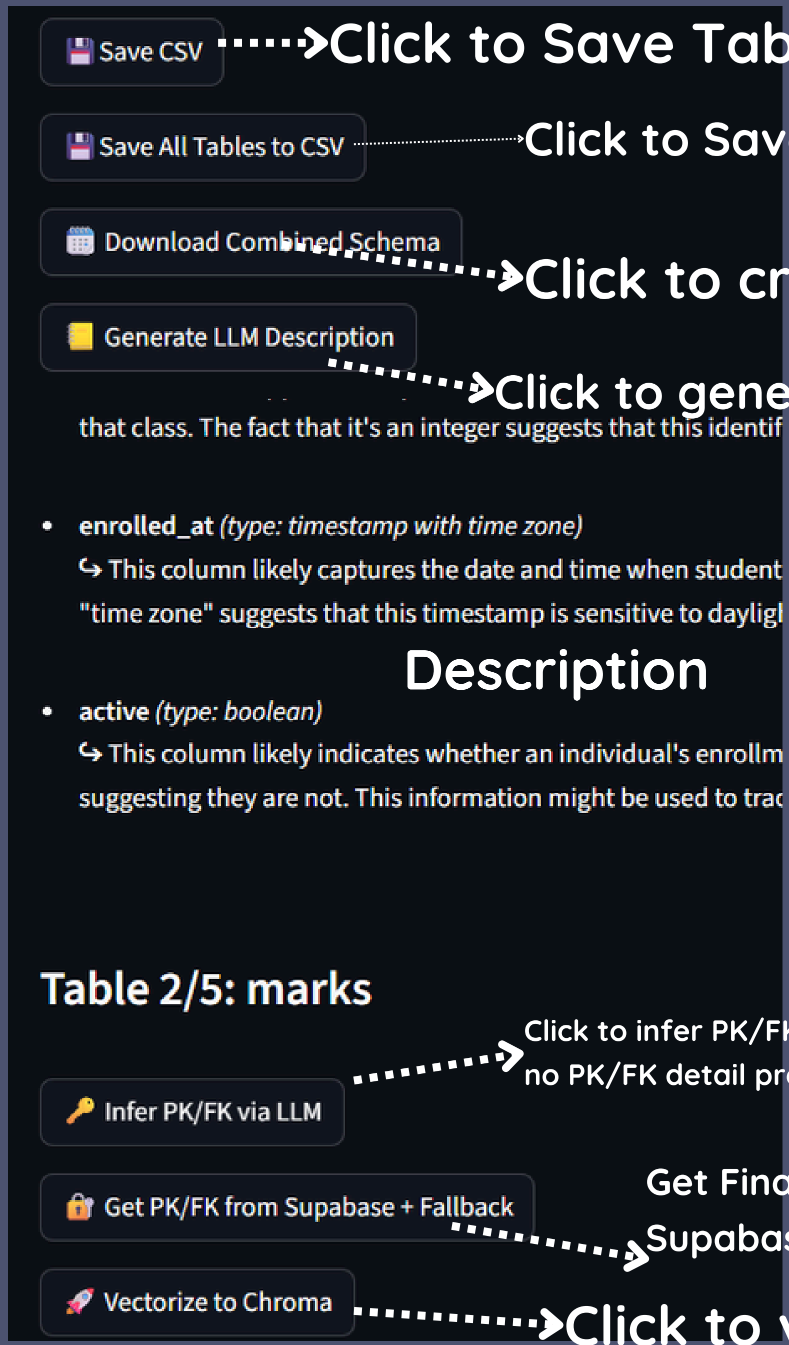
Click to Select Table

	enrollment_id	student_id	class_id
10	968	2	
11	969	2	
12	970	2	
13	971	3	
14	972	3	
15	973	3	
16	974	3	
17	975	3	
18	976	4	
19	977	4	

Table Preview

```
[
  0 : {
    "column_name" : "enrollment_id"
    "data_type" : "integer"
  }
  1 : {
    "column_name" : "student_id"
```

Browse to see column details



Save CSV

Click to Save Table as CSV

Save All Tables to CSV

Click to Save all Tables as CSV

Download Combined Schema

Click to create Schema

Generate LLM Description

Click to generate description

that class. The fact that it's an integer suggests that this identif

- **enrolled_at** *(type: timestamp with time zone)*
  ↪ This column likely captures the date and time when student
  "time zone" suggests that this timestamp is sensitive to dayligl
- **active** *(type: boolean)*
  ↪ This column likely indicates whether an individual's enrollm
  suggesting they are not. This information might be used to trac

Description

Table 2/5: marks

Infer PK/FK via LLM

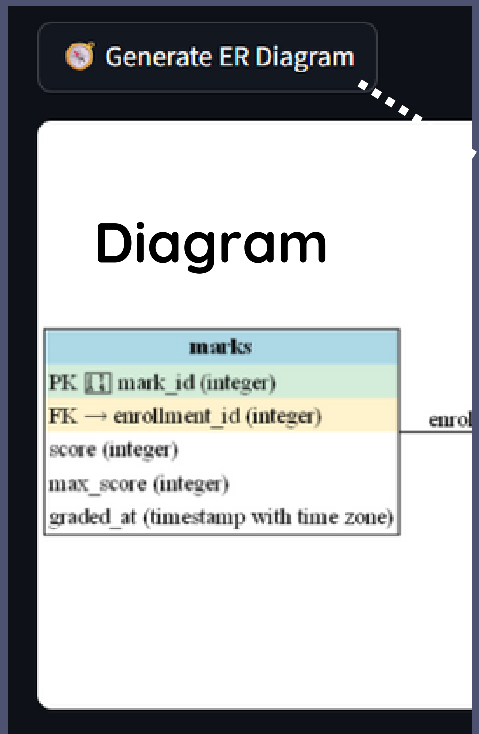
Click to infer PK/FK with LLM(when no PK/FK detail present)

Get PK/FK from Supabase + Fallback

Get Final PK/FK either from Supabase or else use Inferred

Vectorize to Chroma

Click to vectorize db

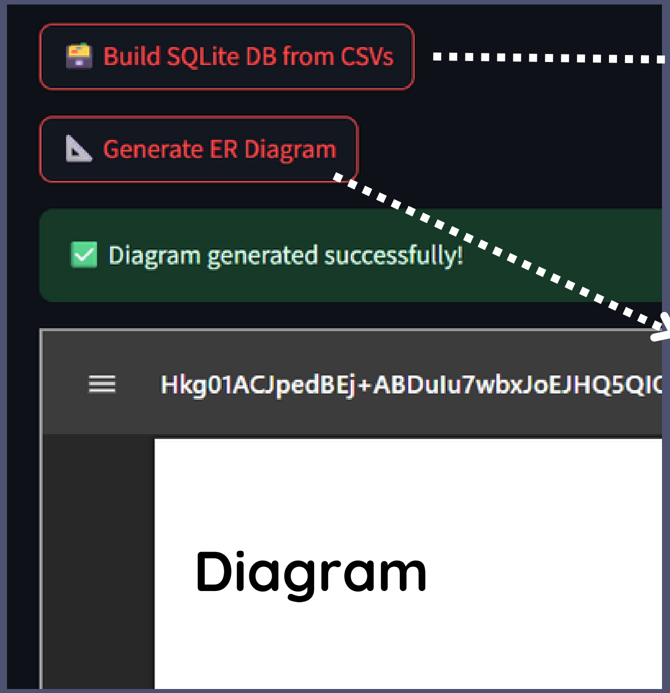


Generate ER Diagram

Diagram

marks
PK mark_id (integer)
FK → enrollment_id (integer)
score (integer)
max_score (integer)
graded_at (timestamp with time zone)
enrol

Click to generate diagram using supabase



Build SQLite DB from CSVs

Click to save db file

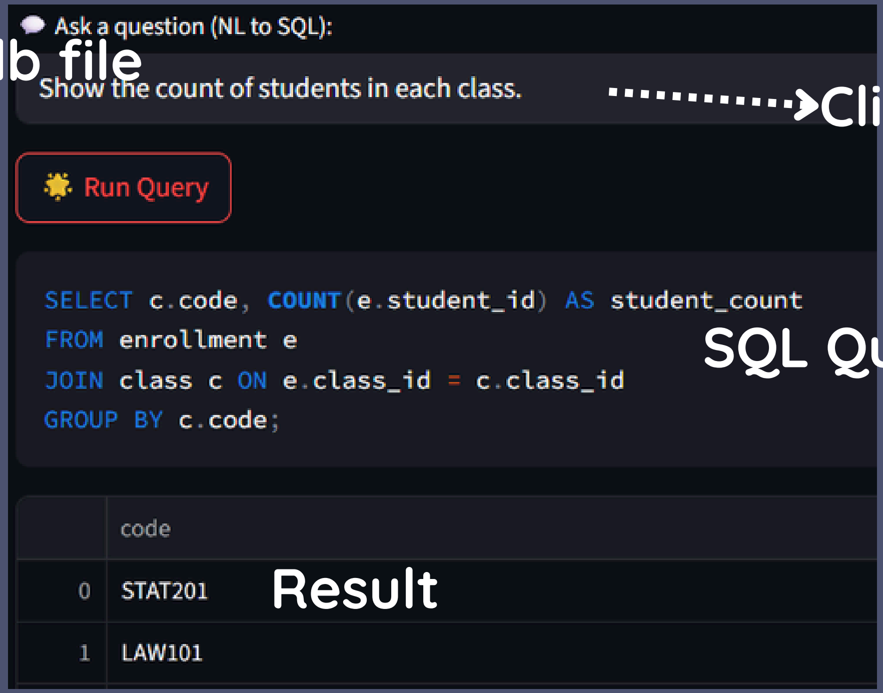
Generate ER Diagram

✅ Diagram generated successfully!

Hkg01ACJpedBEj+ABDuIu7wbxJoEJHQ5QI

Diagram

Click to generate diagram using ERalchemy



Ask a question (NL to SQL):

Show the count of students in each class.

Click to vectorize db

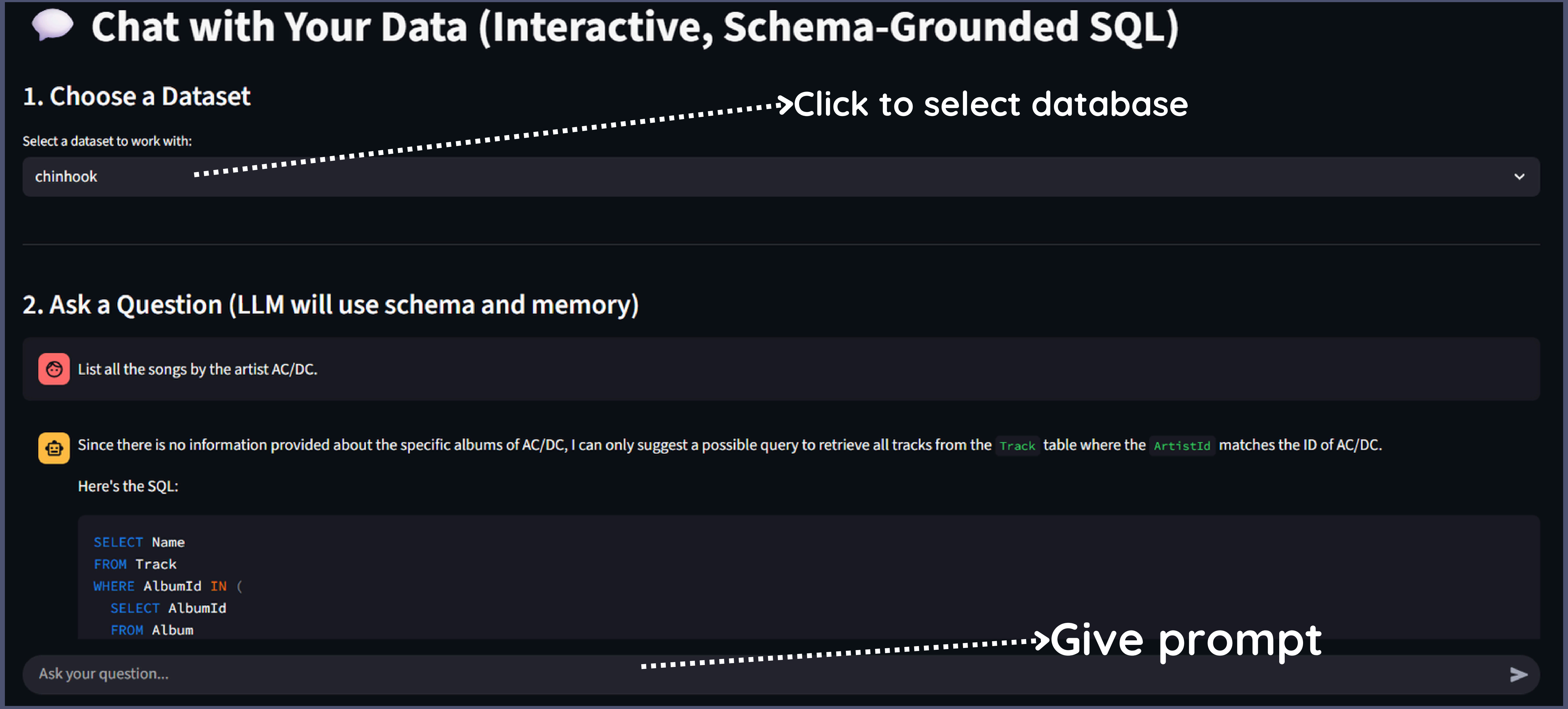
Run Query

```sql
SELECT c.code, COUNT(e.student_id) AS student_count
FROM enrollment e
JOIN class c ON e.class_id = c.class_id
GROUP BY c.code;
```

SQL Query

	code
0	STAT201
1	LAW101

Result



💬 Chat with Your Data (Interactive, Schema-Grounded SQL)

1. Choose a Dataset

Select a dataset to work with:

chinhook

Click to select database

2. Ask a Question (LLM will use schema and memory)

List all the songs by the artist AC/DC.

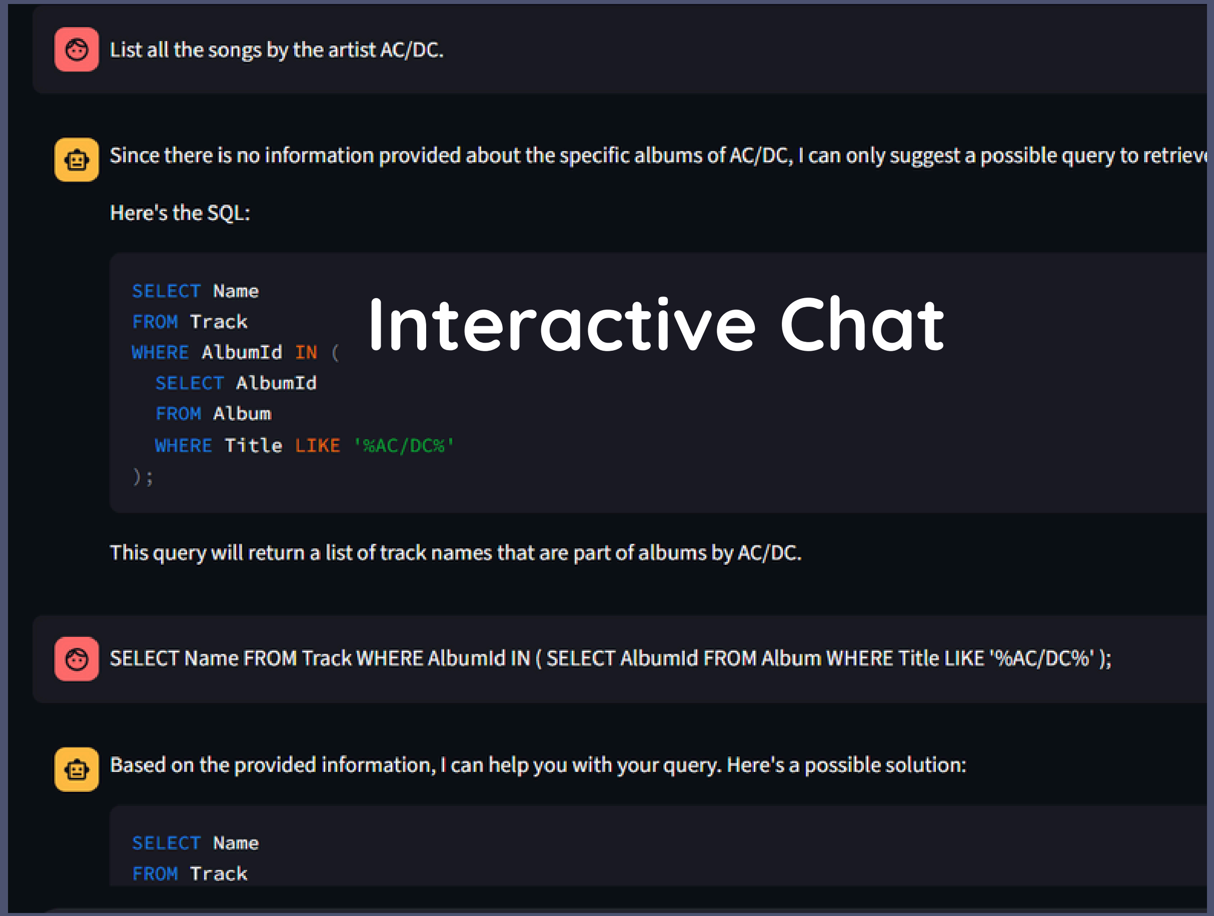
Since there is no information provided about the specific albums of AC/DC, I can only suggest a possible query to retrieve all tracks from the `Track` table where the `ArtistId` matches the ID of AC/DC.

Here's the SQL:

```
SELECT Name
FROM Track
WHERE AlbumId IN (
  SELECT AlbumId
  FROM Album
```

Ask your question...

Give prompt



List all the songs by the artist AC/DC.

Since there is no information provided about the specific albums of AC/DC, I can only suggest a possible query to retriev

Here's the SQL:

```
SELECT Name
FROM Track
WHERE AlbumId IN (
  SELECT AlbumId
  FROM Album
  WHERE Title LIKE '%AC/DC%'
);
```

This query will return a list of track names that are part of albums by AC/DC.

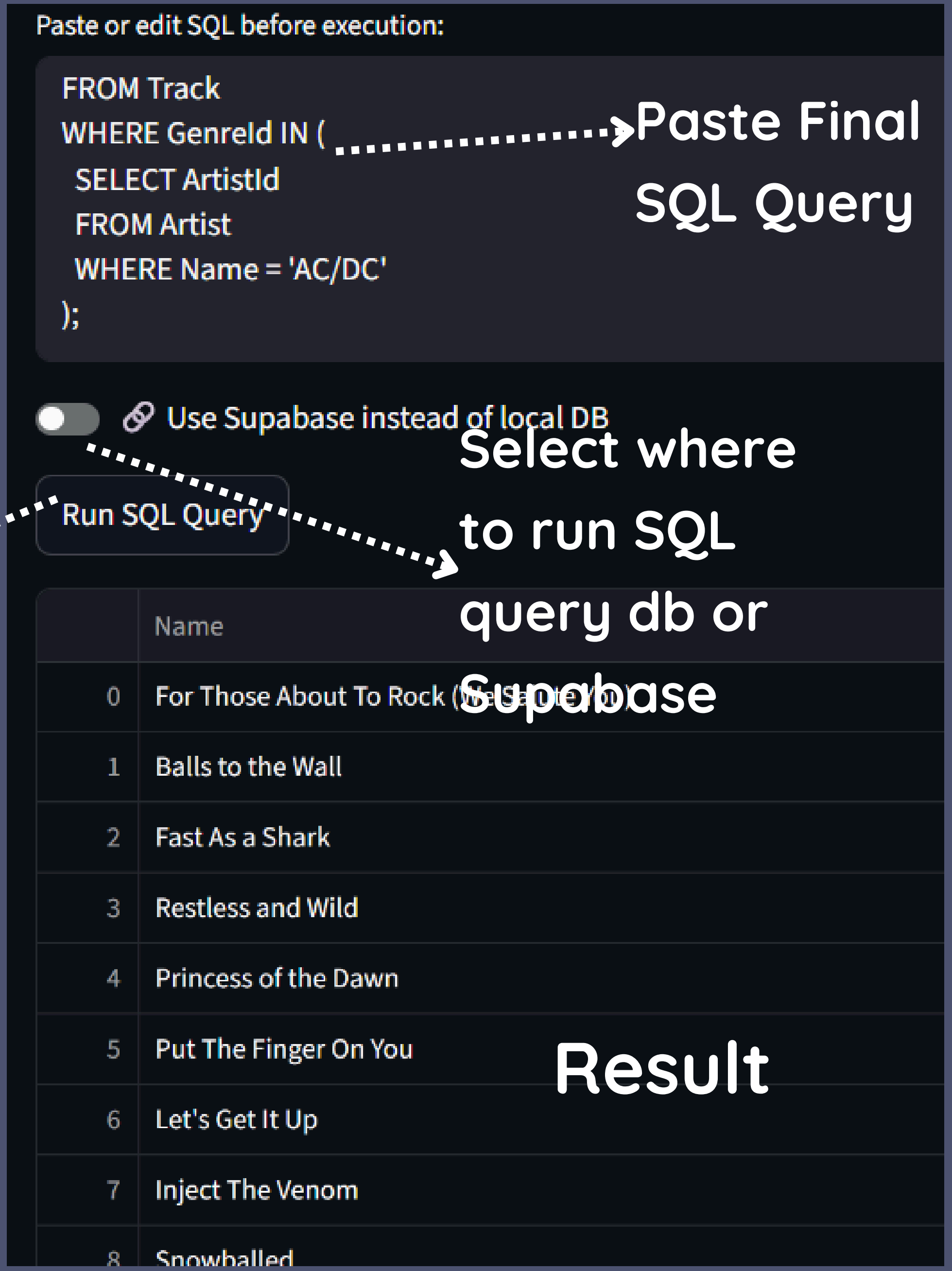
SELECT Name FROM Track WHERE AlbumId IN (SELECT AlbumId FROM Album WHERE Title LIKE '%AC/DC%');

Based on the provided information, I can help you with your query. Here's a possible solution:

```
SELECT Name
FROM Track
```

Interactive Chat

Click to Run query



Paste or edit SQL before execution:

```
FROM Track
WHERE GenreId IN (
  SELECT ArtistId
  FROM Artist
  WHERE Name = 'AC/DC'
);
```

Paste Final SQL Query

○ 🔗 Use Supabase instead of local DB

Run SQL Query

Select where to run SQL query db or Supabase

	Name
0	For Those About To Rock (We Salute You)
1	Balls to the Wall
2	Fast As a Shark
3	Restless and Wild
4	Princess of the Dawn
5	Put The Finger On You
6	Let's Get It Up
7	Inject The Venom
8	Snowballed

Result

🚀 What This Project Demonstrates

🧠 **LLMs + Schema Context** → Smarter Structured Data Interaction

🧩 **Modular Architecture** → Plug-&-play MCP-aligned tools

📊 **Prompt-to-SQL** → Empower analysts with minimal coding

🔌 **Open Backends** → ChromaDB + Ollama for flexibility

🌱 Future Enhancements

🧠 Error-aware SQL correction & explanations

📁 Add support for Parquet, Avro, remote APIs

🔄 Live database sync & streaming query support

🔐 User roles and access control features
📈 Auto-chart recommendations from prompt or data type

🔗 Integration with BI tools (Power BI, Tableau)

Important Links

Team:
Sudharsanam R
Sanket Ninawe
Nitish Raman

GitHub

Github

Supabase Server

Host	Port	Database	Password	SSL Mode
db.zjkqhaghmvkhxyzhjzce.supabase.co	5432	postgres	p@55w0rd	require

MCP Documentation

MCP Point of View

MCP Server Documentation

Q&A

Thank You