

# Using LLMs to Query Data

ODSC East 2025

Rami Krispin, May 15th, 2025

# Rami Krispin

Senior Manager - Data Science & Engineering

Author | Open Source | Docker Captain 



**Running LLMs with Docker Desktop**

The coming release of Docker Desktop from Docker, Inc is going t...

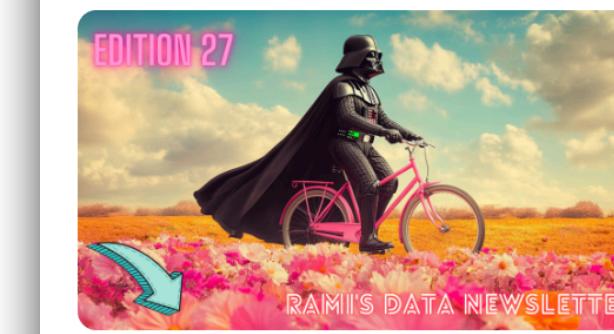
3 min read



**The skforecast Project, AI Engineering and New Learnin...**

Happy Saturday! A quick update - the newsletter is moving from...

3 min read



**The Optuna Project, Advanced Topics in Cryptography, Beyo...**

This week's agenda: Open Source of the Week - the Optuna project New...

3 min read



**The Point of Engineering**

This week...

4 min read



# Agenda

- Introduction
- Prompt Engineering 101
- Setting the Environment
- Data
- AI Agent

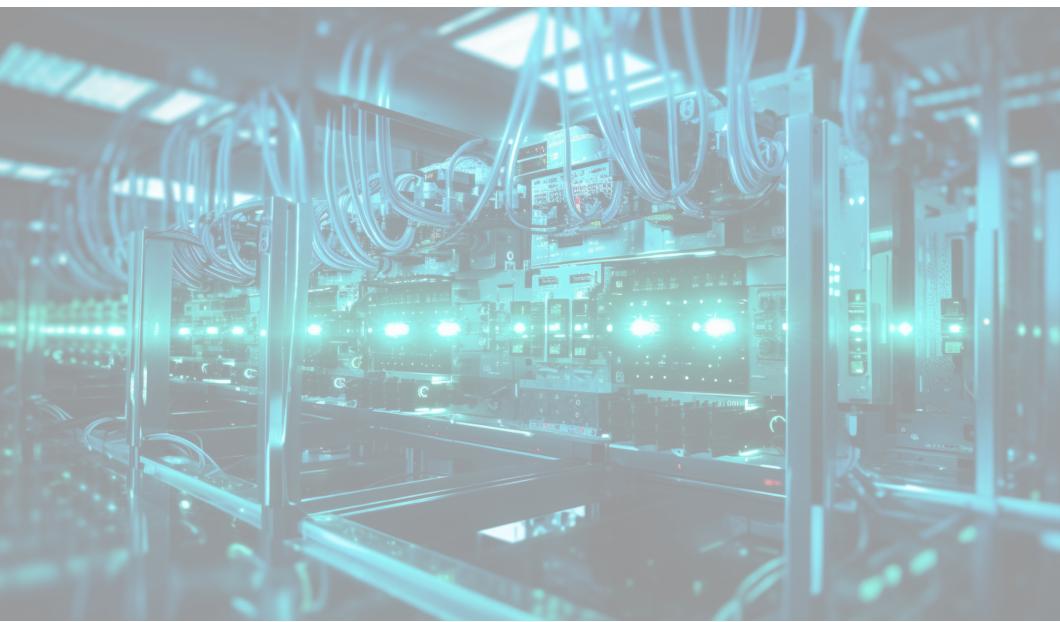
[https://github.com/RamiKrispin/  
osdc-2025-llm-workshop](https://github.com/RamiKrispin/osdc-2025-llm-workshop)

# **LLM in a Nutshell**

# LLM in a Nutshell



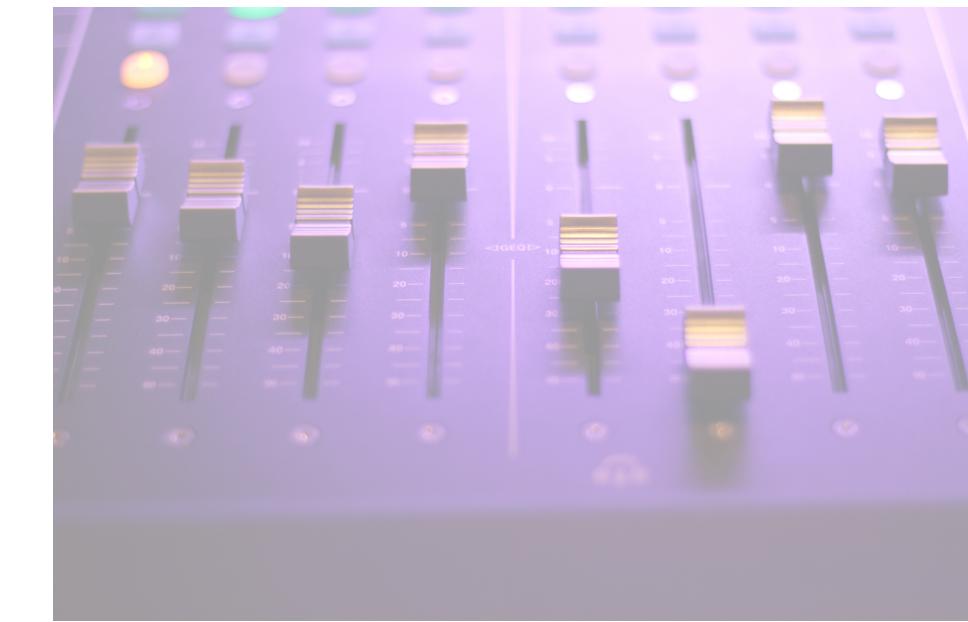
Data



Model Training



Inference



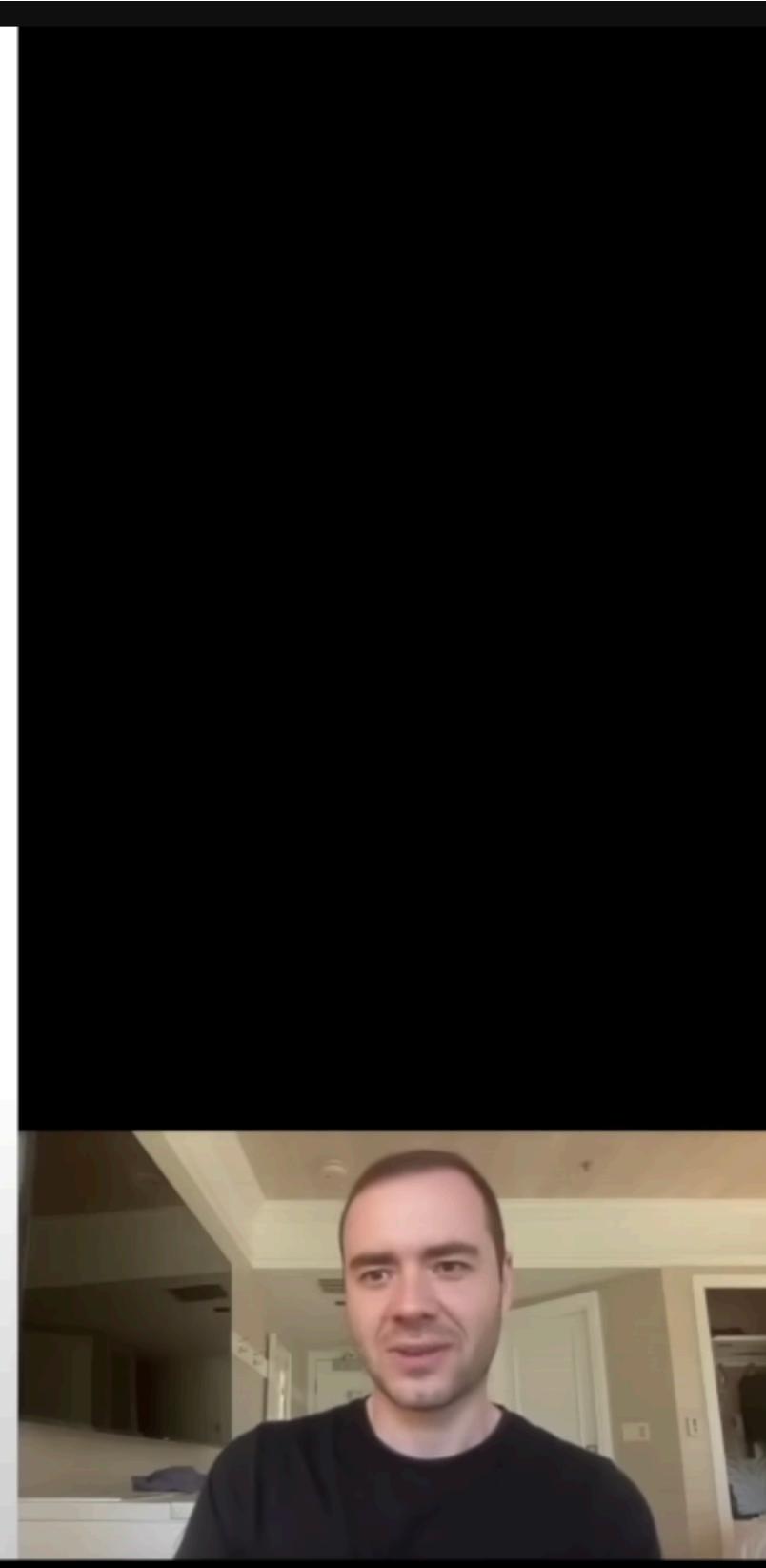
Fine Tuning

# LLM in a Nutshell



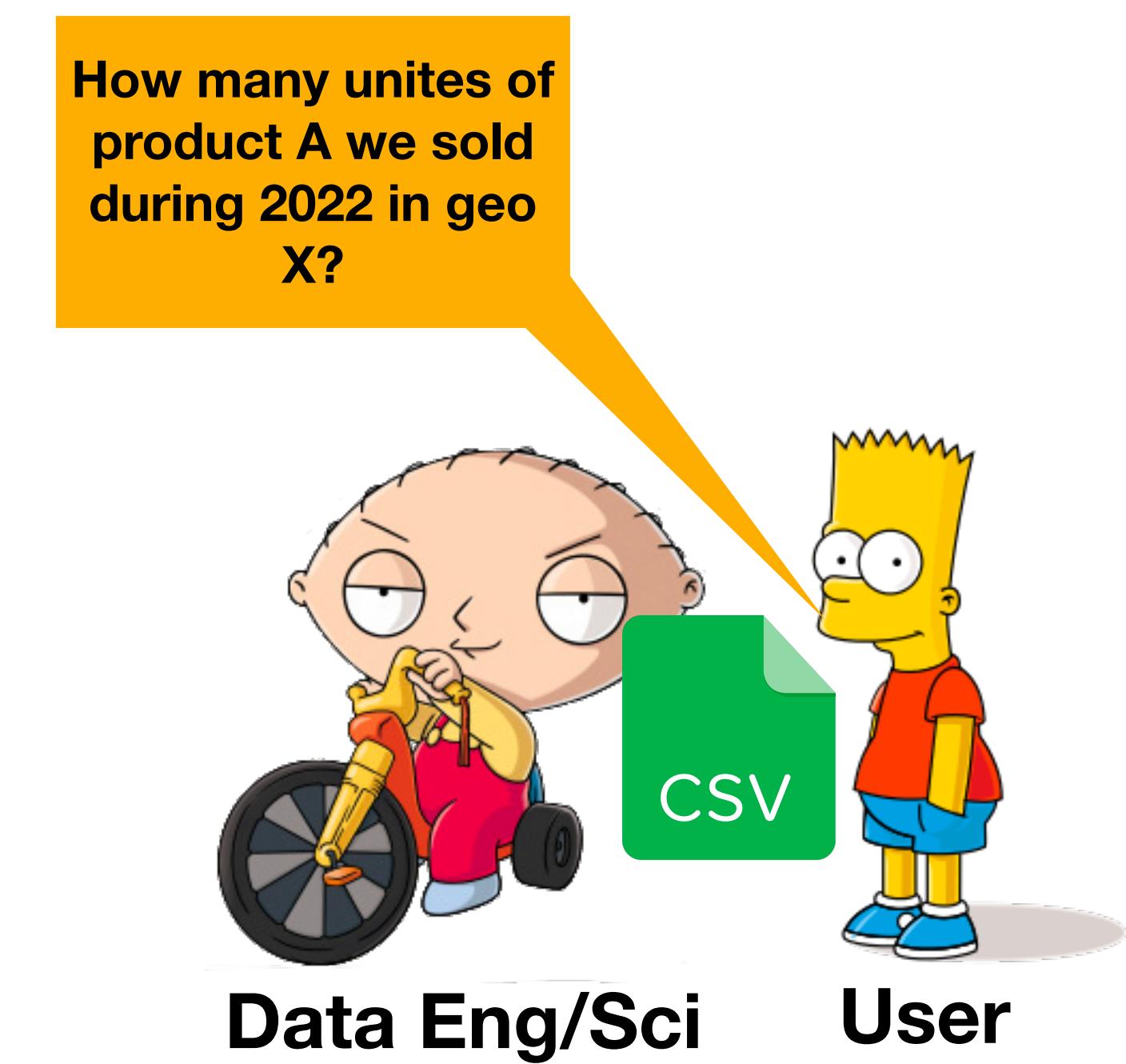
The busy person's  
intro to LLMs

*Andrey Karpathy*

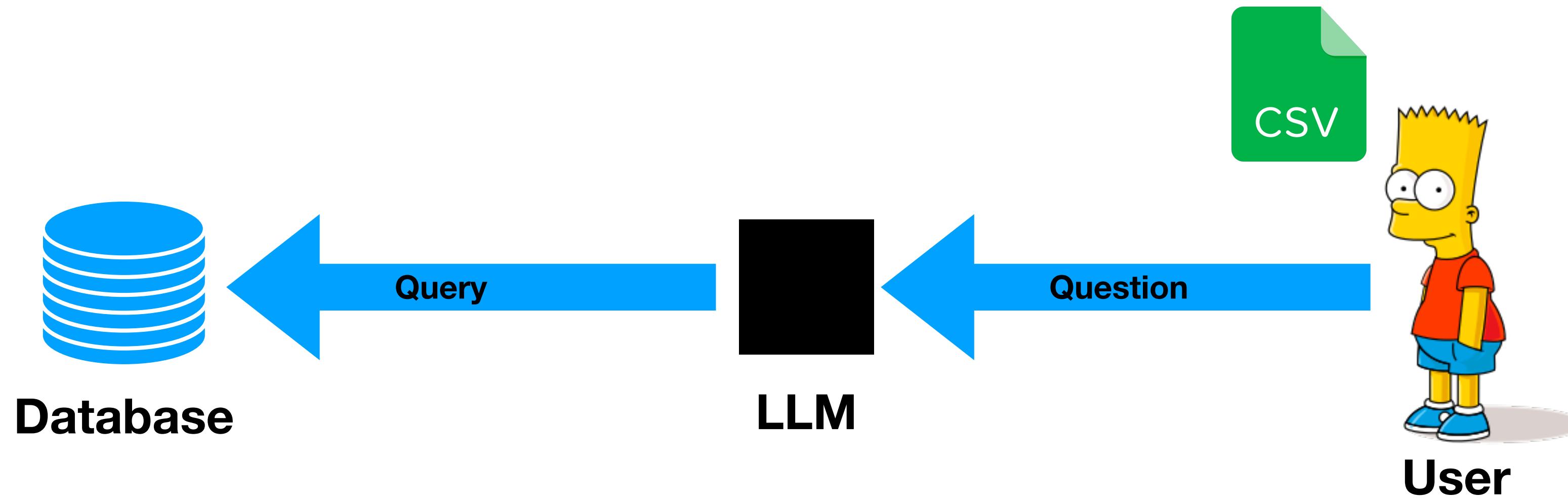


# Motivation

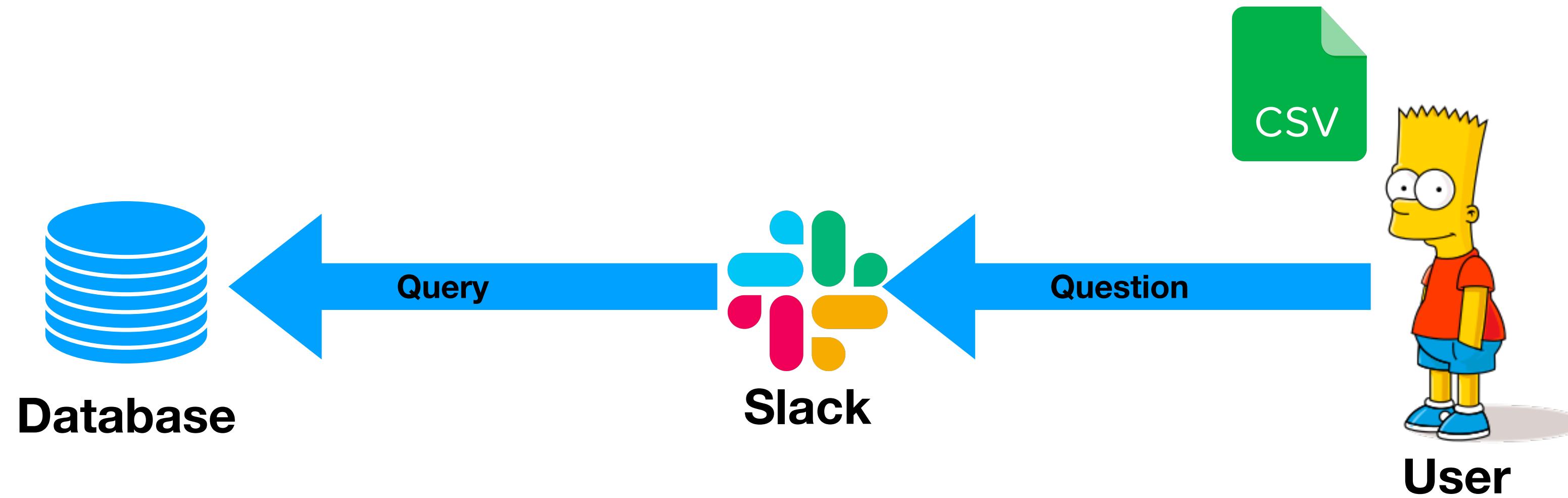
# Motivation



# Motivation



# Motivation

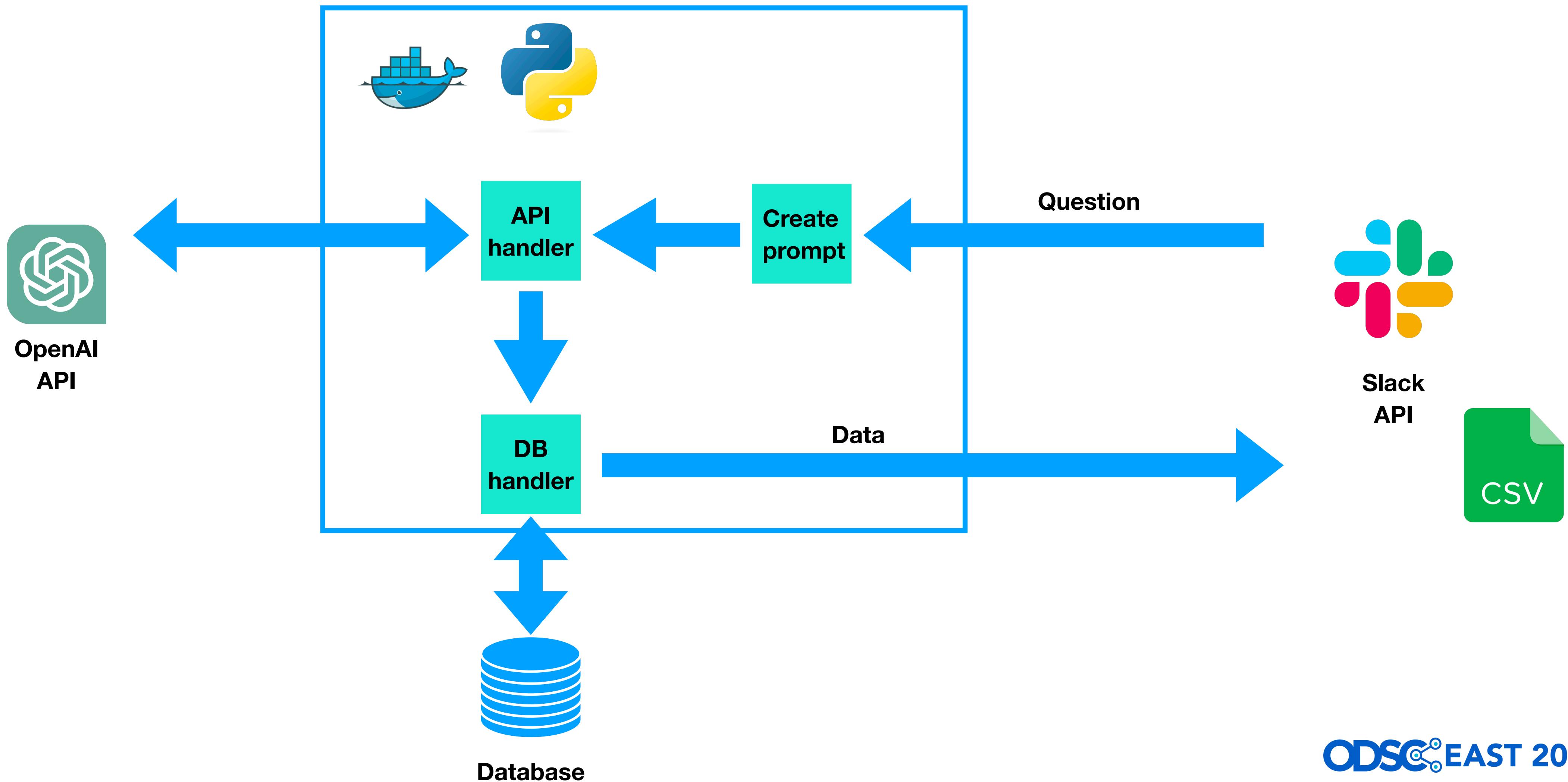


# Motivation

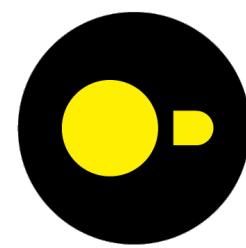


# Architecture

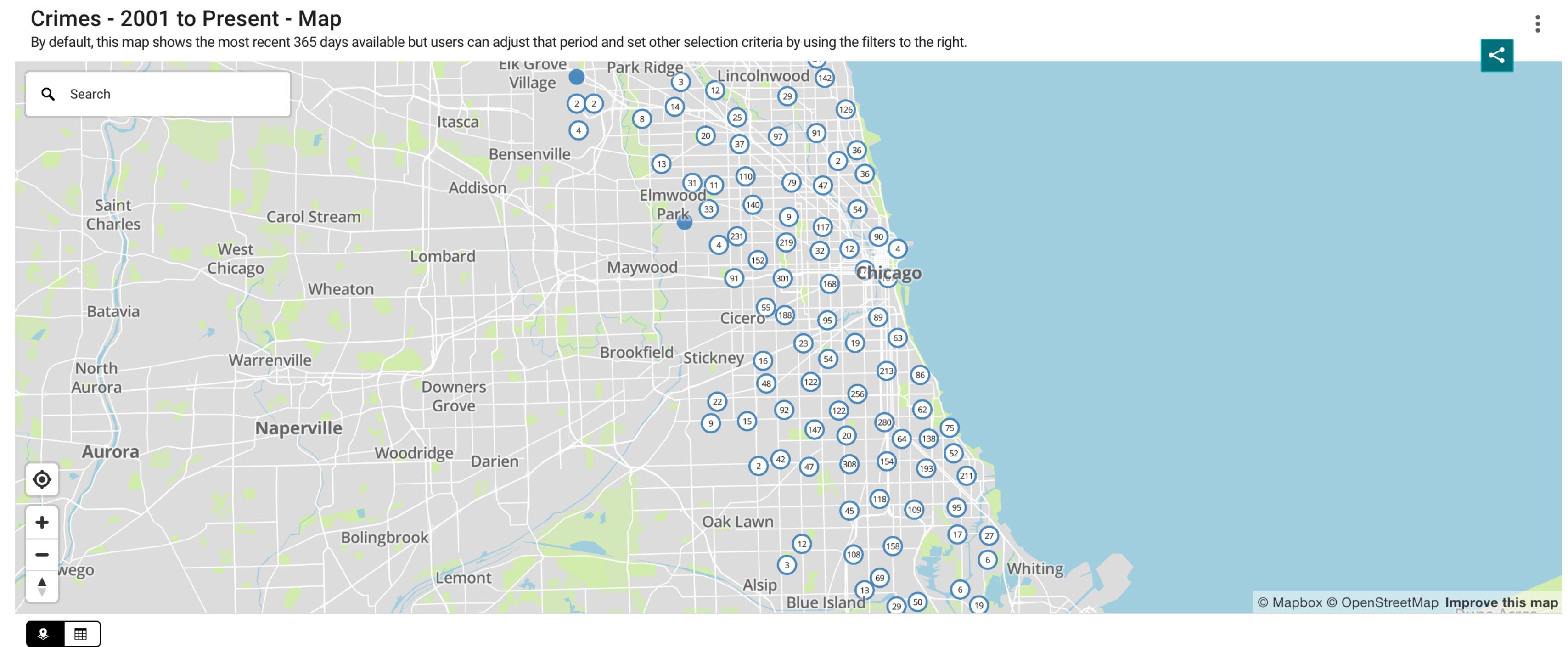
# Architecture



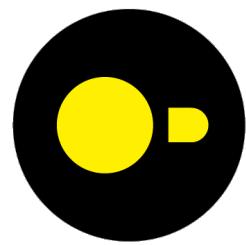
# Tools & Data



DuckDB



# Tools & Data



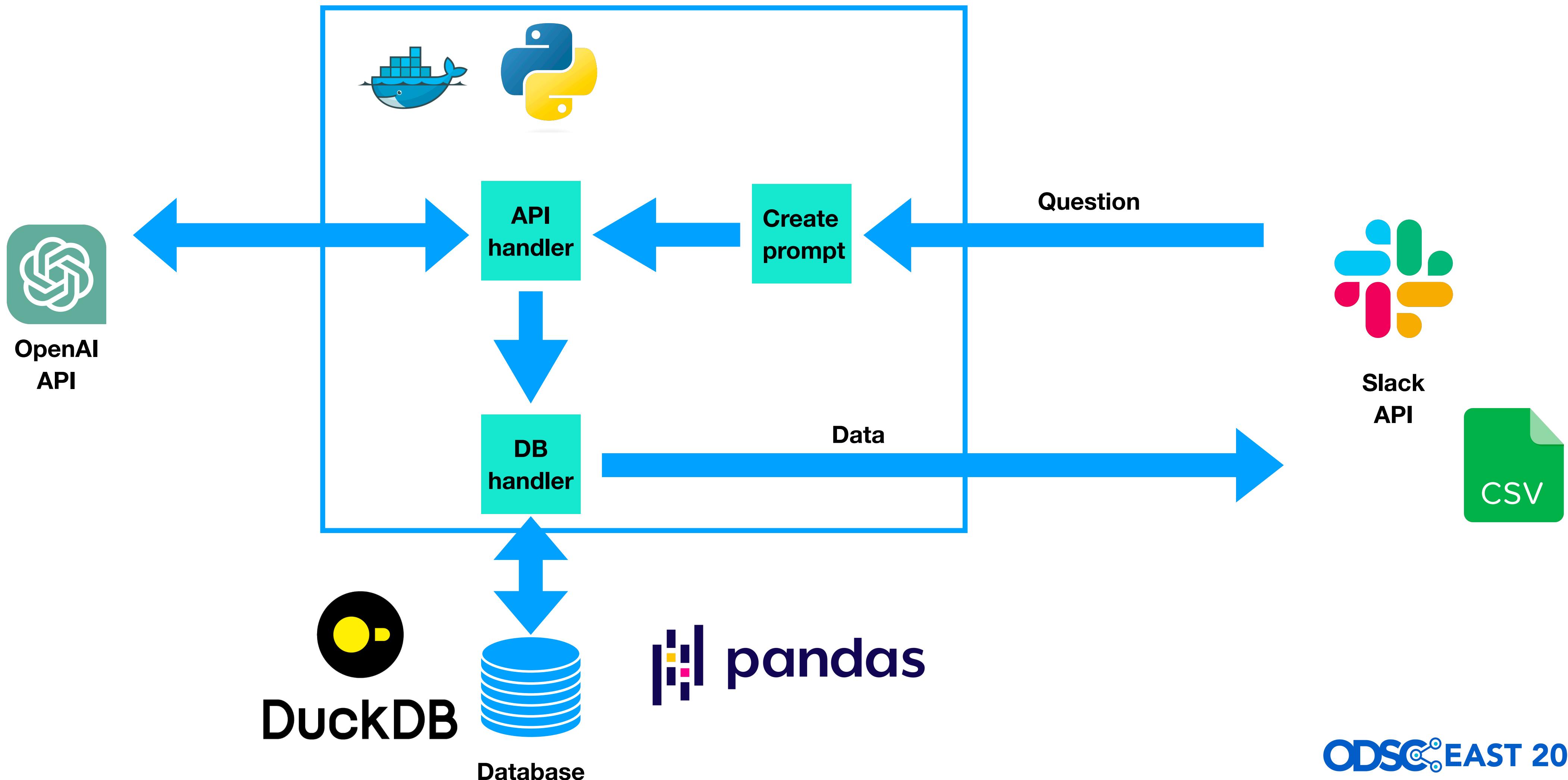
DuckDB



```
duckdb.sql("DESCRIBE SELECT * FROM chicago_crime;")
```

column_name varchar	column_type varchar	null varchar	key varchar	default varchar	extra varchar
id	BIGINT	YES	NULL	NULL	NULL
case_number	VARCHAR	YES	NULL	NULL	NULL
datetime	VARCHAR	YES	NULL	NULL	NULL
block	VARCHAR	YES	NULL	NULL	NULL
iucr	VARCHAR	YES	NULL	NULL	NULL
primary_type	VARCHAR	YES	NULL	NULL	NULL
description	VARCHAR	YES	NULL	NULL	NULL
location_description	VARCHAR	YES	NULL	NULL	NULL
arrest	BOOLEAN	YES	NULL	NULL	NULL
domestic	BOOLEAN	YES	NULL	NULL	NULL
beat	BIGINT	YES	NULL	NULL	NULL
district	BIGINT	YES	NULL	NULL	NULL
ward	DOUBLE	YES	NULL	NULL	NULL
community_area	BIGINT	YES	NULL	NULL	NULL
fbi_code	VARCHAR	YES	NULL	NULL	NULL
x_coordinate	DOUBLE	YES	NULL	NULL	NULL
y_coordinate	DOUBLE	YES	NULL	NULL	NULL
year	BIGINT	YES	NULL	NULL	NULL
updated_on	VARCHAR	YES	NULL	NULL	NULL
latitude	DOUBLE	YES	NULL	NULL	NULL
longitude	DOUBLE	YES	NULL	NULL	NULL
21 rows		6 columns			

# Architecture



# Prompt Engineering 101

**It is All About the Context**

**How many cases ended up with  
arrest?**

# Prompt Engineering 101

 You  
Create an SQL query that counts the number of records that ended up with an arrest

 ChatGPT  
To create an SQL query that counts the number of records where an arrest occurred, you need to have a column in your database that indicates whether an arrest happened or not. For the sake of this example, let's assume you have a table called `crime\_records` with a column named `arrested` that is a boolean column indicating whether an arrest occurred (1 for true, 0 for false).  
  
Here's a sample SQL query:

```
sql
SELECT COUNT(*) AS arrest_count
FROM crime_records
WHERE arrested = 1;
```

In this query:

- `SELECT COUNT(\*)`: This selects the count of records.
- `AS arrest\_count`: This aliases the result column as `arrest\_count` for easier reference.
- `FROM crime\_records`: Specifies the table from which to retrieve the data.
- `WHERE arrested = 1`: Filters the records to include only those where the `arrested` column is equal to 1 (indicating an arrest).

Please adjust the table and column names based on your actual database schema. If your schema is different, provide more details, and I can help you tailor the query accordingly.

# Prompt Engineering 101

 You  
Create an SQL query that counts the number of records that ended up with an arrest

 ChatGPT  
To create an SQL query that counts the number of records where an arrest occurred, you need to have a column in your database that indicates whether an arrest happened or not. For the sake of this example, let's assume you have a table called `crime\_records` with a column named `arrested` that is a boolean column indicating whether an arrest occurred (1 for true, 0 for false).  
Here's a sample SQL query:

```
sql
SELECT COUNT(*) AS arrest_count
FROM crime_records
WHERE arrested = 1;
```

In this query:

- `SELECT COUNT(\*)`: This selects the count of records.
- `AS arrest\_count`: This aliases the result column as `arrest\_count` for easier reference.
- `FROM crime\_records`: Specifies the table from which to retrieve the data.
- `WHERE arrested = 1`: Filters the records to include only those where the `arrested` column is equal to 1 (indicating an arrest).

Please adjust the table and column names based on your actual database schema. If your schema is different, provide more details, and I can help you tailor the query accordingly.

# Prompt Engineering 101

 You  
I have a table named chicago\_crime with the crime records in Chicago City since 2021. The Arrest field defines if the case ended up with arrest or not, and it is a boolean (true or false).  
  
I want to create an SQL query that counts the number of records that ended up with an arrest.

 ChatGPT  
Certainly! You can use the `COUNT` function along with a `WHERE` clause to filter records where the Arrest field is true. Here's an example SQL query:

sql

```
SELECT COUNT(*) AS ArrestedCases
FROM chicago_crime
WHERE Arrest = true;
```

 Copy code

This query selects the count of records from the `chicago\_crime` table where the `Arrest` field is true and aliases the result as `ArrestedCases`.

# Prompt Engineering 101

A screenshot of a ChatGPT conversation. The user (RA) asks about creating an SQL query to count records where the 'Arrest' field is true. ChatGPT responds with an example SQL query:

```
sql
SELECT COUNT(*) AS ArrestedCases
FROM chicago_crime
WHERE Arrest = true;
```

The code block is highlighted with a blue border. Below the code, ChatGPT explains its purpose: "This query selects the count of records from the `chicago\_crime` table where the `Arrest` field is true and aliases the result as `ArrestedCases`." There are also small icons for copy, share, and other actions.

# Prompt Engineering 101

**SYSTEM** Given the following SQL tables, your job is to write queries given a user's request.

```
CREATE TABLE Orders (
    OrderID int,
    CustomerID int,
    OrderDate datetime,
    OrderTime varchar(8),
    PRIMARY KEY (OrderID)
);
```

```
CREATE TABLE OrderDetails (
    OrderDetailID int,
    OrderID int,
    ProductID int,
    Quantity int,
    PRIMARY KEY (OrderDetailID)
);
```

```
CREATE TABLE Products (
    ProductID int,
    ProductName varchar(50),
    Category varchar(50),
    UnitPrice decimal(10, 2),
    Stock int,
    PRIMARY KEY (ProductID)
);
```

```
CREATE TABLE Customers (
    CustomerID int,
    FirstName varchar(50),
    LastName varchar(50),
    Email varchar(100),
    Phone varchar(20),
    PRIMARY KEY (CustomerID)
);
```

**USER** Write a SQL query which computes the average total order value for all orders on 2023-04-01.

# Prompt Engineering 101

SYSTEM

Given the following SQL tables, your job is to write queries given a user's request.

```
CREATE TABLE Orders (
    OrderID int,
    CustomerID int,
    OrderDate datetime,
    OrderTime varchar(8),
    PRIMARY KEY (OrderID)
);
```

```
CREATE TABLE OrderDetails (
    OrderDetailID int,
    OrderID int,
    ProductID int,
    Quantity int,
    PRIMARY KEY (OrderDetailID)
);
```

```
CREATE TABLE Products (
    ProductID int,
    ProductName varchar(50),
    Category varchar(50),
    UnitPrice decimal(10, 2),
    Stock int,
    PRIMARY KEY (ProductID)
);
```

```
CREATE TABLE Customers (
    CustomerID int,
    FirstName varchar(50),
    LastName varchar(50),
    Email varchar(100),
    Phone varchar(20),
    PRIMARY KEY (CustomerID)
);
```

USER

Write a SQL query which computes the average total order value for all orders on 2023-04-01.

# Prompt Engineering 101

**SYSTEM** Given the following SQL tables, your job is to write queries given a user's request.

```
CREATE TABLE Orders (
    OrderID int,
    CustomerID int,
    OrderDate datetime,
    OrderTime varchar(8),
    PRIMARY KEY (OrderID)
);
```

```
CREATE TABLE OrderDetails (
    OrderDetailID int,
    OrderID int,
    ProductID int,
    Quantity int,
    PRIMARY KEY (OrderDetailID)
);
```

```
CREATE TABLE Products (
    ProductID int,
    ProductName varchar(50),
    Category varchar(50),
    UnitPrice decimal(10, 2),
    Stock int,
    PRIMARY KEY (ProductID)
);
```

```
CREATE TABLE Customers (
    CustomerID int,
    FirstName varchar(50),
    LastName varchar(50),
    Email varchar(100),
    Phone varchar(20),
    PRIMARY KEY (CustomerID)
);
```

**USER**

Write a SQL query which computes the average total order value for all orders on 2023-04-01.

**Let's Build it!**

# Questions?

# **Thank You!**