

LLM & Databases. What's in it?

Rethinking Databases in the Age of LLMs

Invited Talk @ FGDB Meeting 2026

Carsten Binnig

TU Darmstadt & DFKI

Main Collaborator & PhD Students



SYSTEMS



TECHNISCHE
UNIVERSITÄT
DARMSTADT

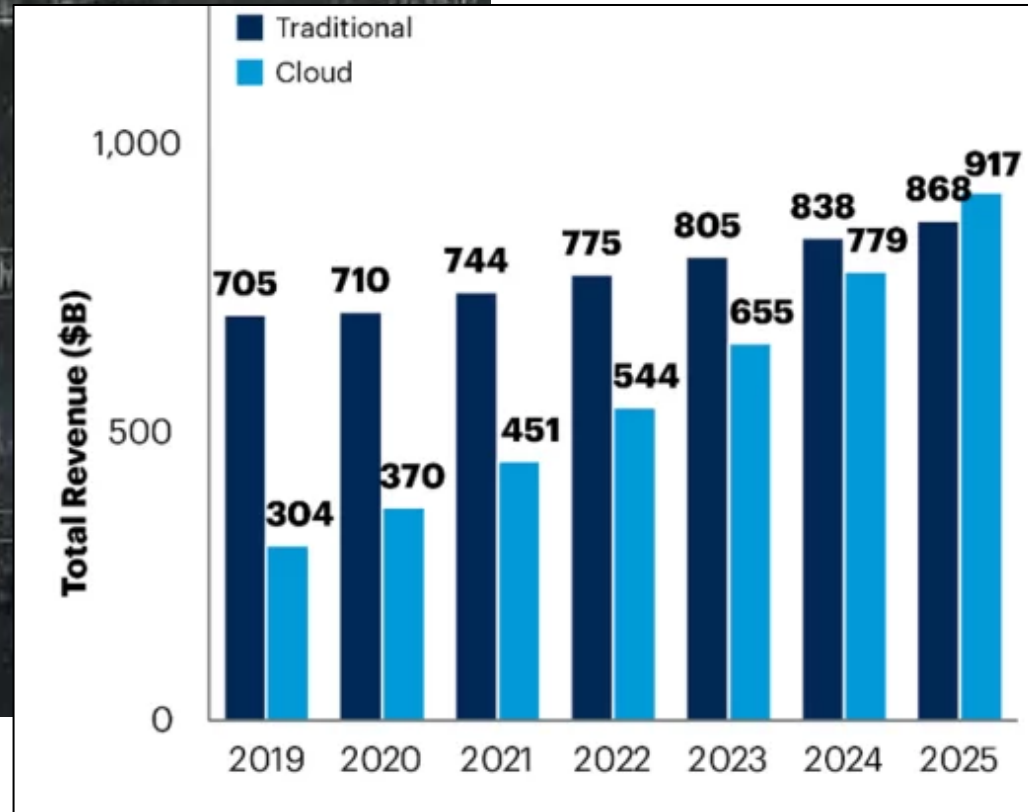
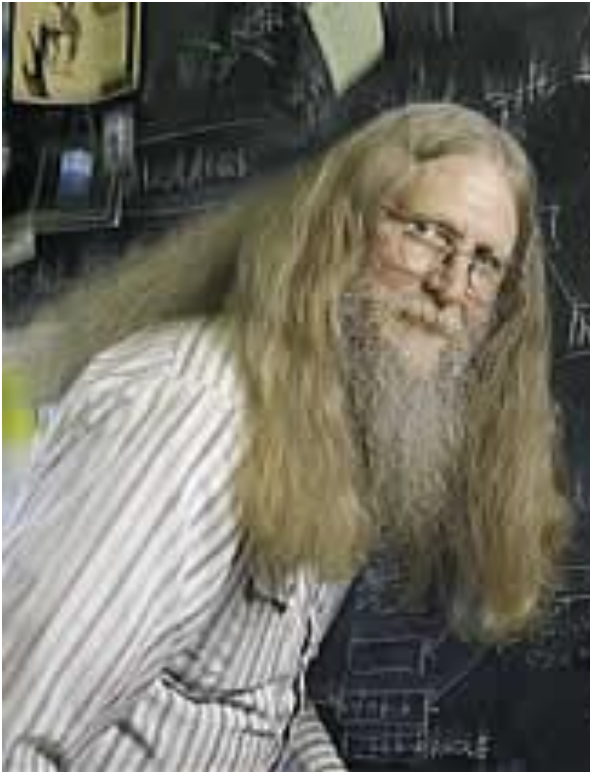
dfki
ai



hessian.AI



Relational Databases are a Success Story ...



*Relational Databases
Foundation of
Civilization”*

... and the Success continues in the Cloud!



How do users expect to interact with data in the age of AI?

But this is what databases are giving us ...

**There is a high overhead
to pay to use databases!**



**“Relational Tax”=
overheads deeply rooted in
the relational design!**

What were the original Promises?

```
SELECT p.age  
FROM patients WHERE name = 'Alice'
```

patients		
name	age	gender
Alice	42	f
Bob	23	m

Promises of Codd & Friends: (1) Easy-to-use Data Model & Queries, and (2) Database optimizes execution for you!

And? Did Relational Databases deliver? No!

Finds patients with flu whose age is above average & stayed longer than the average patient

And this is still a

Tax 1: The Query Tax
→ Query Authoring is Complex

Domain Expert
Data Scientist

```
SELECT MAX(stay) FROM Patients  
WHERE diagnosis = 'Flu A'  
AND age > (SELECT AVG(age) FROM AvgValues);
```

Tax 3: The Tuning Tax
→ Databases need Massive Tuning

Tax 2: The Data Tax
→ High Overheads for Extracting Data into Tables

=
Relational Taxes!

My Research: Cutting the Relational Taxes

Tax 1: The Query Tax

→ Query Authoring is Complex

Tax 3: The Tuning Tax

→ Databases need Massive Tuning

Tax 2: The Data Tax

→ High Overheads for Mapping Data to Tables

TAX

AI

My Research: Cutting the Relational Taxes

Tax 1: The Query Tax
→ Query Authoring is Complex

~~**Tax 3: The Tuning Tax**~~
~~→ Databases need Massive Tuning~~

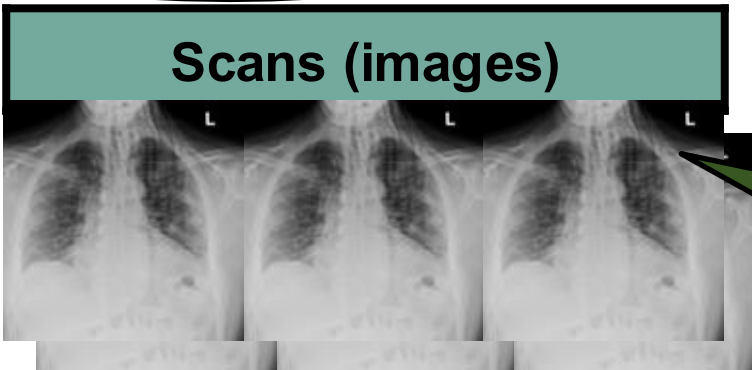
Tax 2: The Data Tax
→ High Overheads for Mapping Data to Tables

AI

How to cut the Data & Query Tax?

Patients (table)		
name	age	stay
Alice	42	2 weeks
Bob	23	1 week

Examinations (text)
Alice was diagnosed with Covid ...



No Query Tax:
Users do not need to know the schema
(Query can be „vague“)

„Sloppy“ SQL:
`SELECT
AVG (duration)
WHERE 'patient
has covid'`



No Data Tax:
No need to transform data into tables

Are LLMs the better Databases?

What is the average duration of the stay of patients with Covid?

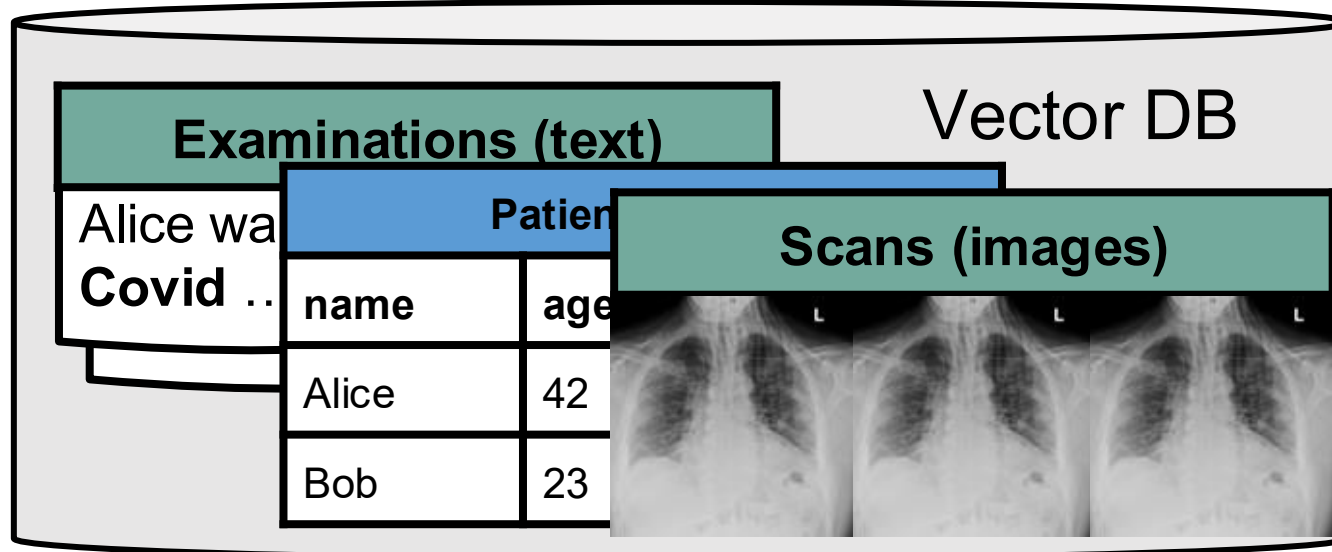
2. Construct Prompt
(with Data & Question)



LLM e.g. GPT-4o
(w reasoning)

1.4 weeks

1. Retrieve



Retrieval
Augmented
Generation (RAG)
& Reasoning?

Major Issues of “LLMs as Databases”

Issues regarding Queries & Data:

- Simple queries only (e.g., no aggregations)
- Limited data understanding: LLMs do not understand tables and are hard to extend toward new modalities

Issues regarding Execution:

- Black-box processing (i.e., not explainable) & Hallucinations
- LLMs are quite **slow and cause high cost \$\$\$ (Processing a 1TB database with a typical state-of-the-art LLM costs around 1.2M USD)**

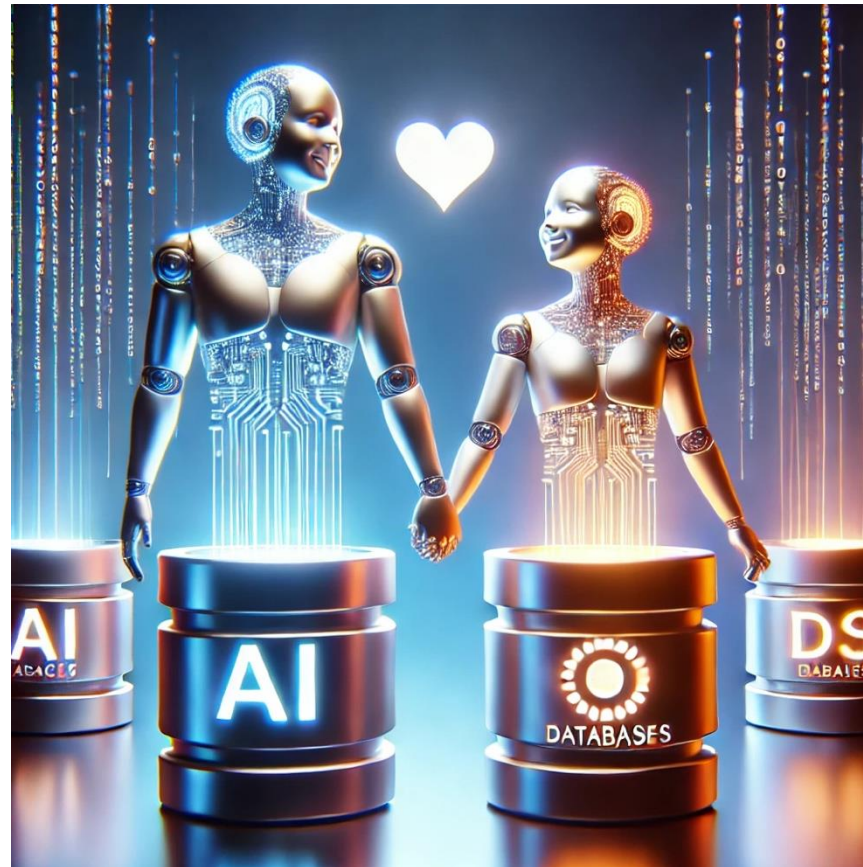
My Vision: “LLM-augmented” Databases
(Extend DB with LLMs as needed)

Vision: LLM-Augmented DBMSs

LLMs and DBs are a “perfect” couple:
Strength complement each other

LLM:

- Understanding of NL
- Multi-Modality (Images,)
- Reasoning
- Code Generation

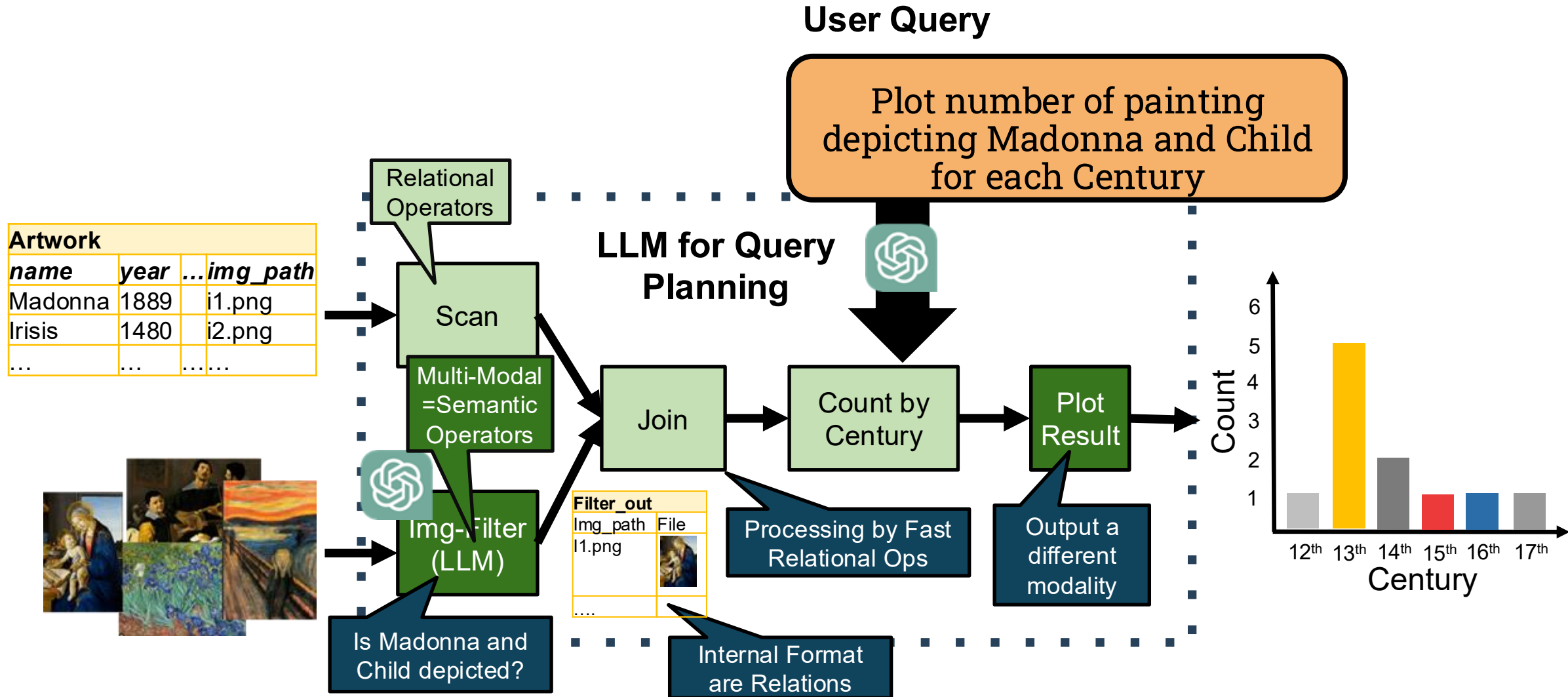


Relational DBMS:

- Fast & Scalable
- Complex queries with joins & aggregations
- Query Optimization
- ...

CAESURA: A very first LLM-augmented DBMS

[CIDR'24, SIGMOD'24 (Best Demo Award), VLDB'25]



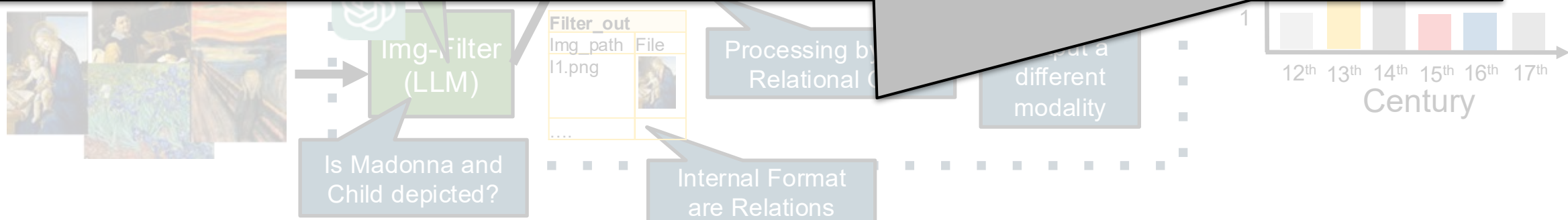
CAESURA: A very first LLM-augmented DBMS

[CIDR'24, SIGMOD'24 (Best Demo Award), VLDB'25]

User Query

Plot number of painting

- 👍 **Complex queries** on multi-modal data w joins, aggregates
- 👍 **Query plan** enables optimizations (e.g., join reordering)
- 👍 **Fast execution** outside LLM (if possible), building on relational techniques (efficient/scalable joins)
- 👍 **Users can inspect** the query plan (no black box)



CAESURA: Query Planning with LLMs

Plot number of painting depicting Madonna and Child for each Century

metadata.csv
name, year, ..., img_path
Madonna, 1889, ..., i1.png
Irises, 1480, ..., i2.png
...
Scream, 1893



Use LLM to reason over data & logical operations

Logical Op
Read, Sc

Planning Phase



Logical Plan

Step 1: Read century, img from metadata CSV

Step 2: Scan images, which depicts Madonna & Child?

Step 3: Join images with m

Use LLM to select a tool & generate code per step

Mapping Phase



Physical Plan

```
df["century"] = Python  
df.apply(  
lambda x: int(x[:2])  
+ 1)
```

```
madonna_img = VisualQA  
filterByBlib(text="Is  
Madonna and Child  
depicted?", imgs)
```

```
sqldf("SELECT *  
FROM metadata_df m  
JOIN madonna_img i ON  
m.img_path =  
i.img_path")
```

CAESURA: Query Planning with LLMs

Plot number of painting depicting Madonna and Child for each Century

metadata.csv
name, year, ..., img_path
Madonna, 1889, ..., i1.png
Irises, 1480, ..., i2.png
...
Scream, 1893



Use LLM to reason over data & logical operations

Logical Op
Read, Sc

Planning Phase



Logical Plan

Step 1: Read centu
img from metadat
CSV

Step 2: Scan
images, which depicts
Madonna & Child?

Step 3: Join imag
with m

Use
a tool & generate
code per step

Opportunities:

- Optimizations on logical plans (e.g., join ordering)
- Optimizations on physical plan (e.g., small vs. large model or even regex?)

Challenges:

- Error Recovery
- No answer possible?


CAESURA: Overall Results (Accuracy)

Dataset 1: Artwork (Tables & Images)

metadata.csv
name, year, ..., img_path
Madonna, 1889, ..., i1.png
Irises, 1480, ..., i2.png
...
Scream, 1893, ...



Dataset 2: Rotowire (Wiseman et al, 2017) (Tables & Textual Reports)

teams.csv	reports
name, region	
Heat, Northwest	
Huskies, Northeast	

Accuracy for 48 Queries (24 per dataset)

Models	GPT-4	
	logical	physical
Plan type		
Results per dataset		
Varying Query Difficulty		
Overall results		

Code: <https://github.com/DataManagementLab/caesura>
Demo video: https://link.tuda.systems/caesura_demo

CAESURA: Overall Results (Accuracy)

Dataset 1: Artwork (Tables & Images)

metadata.csv
name, year, ..., img_path
Madonna, 1889, ..., i1.png
Irises, 1480, ..., i2.png
...
Scream, 1893, ...



Multi-modal queries are more difficult

Dataset 2: Rotowire (Wisconsin) (Tables & Textual)

teams.csv
name, region
Heat, Northwest
Huskies, Northeast

The San Antonio Spurs defeated the Miami Heat...

Queries for simple results are easier

GPT-4 as backend model

Up to 100% correct plans

Accuracy for 48 Queries (24 per dataset)

Models Plan type	GPT-4	
	logical	physical
Artwork overall	100%	100%
Rotowire overall	87.5%	75.0%
Single modality	100%	92.7%
Multiple modalities	87.5%	83.3%
Single value	100%	93.8%
Table	87.5%	81.3%
Plot	93.8%	87.5%
All	93.8%	87.5%

Code: <https://github.com/DataManagementLab/caesura>

Demo video: https://link.tuda.systems/caesura_demo

LLM-Augmented DBMSs are happening!

In the recent two years, several **LLM-augmented data systems** have been **proposed by academia (aka semantic data systems)**

- **2024:** CAESURA (Darmstadt), ThalamusDB (Cornell)
- **2025:** Palimpzest (MIT), LOTUS (Stanford), docETL (Berkley) ... many more

Recently also many **commercial database systems** have added LLM support

- **Snowflake:** AI Functions / Cortex-AI
- **BigQuery:** BigQueryML ... many more

Main Notion: LLMs as User-defined Functions

My Research: Cutting the Relational Taxes

Tax 1: The Query Tax

→ Query Authoring is Complex

Tax 3: The Tuning Tax

→ Databases need Massive Tuning

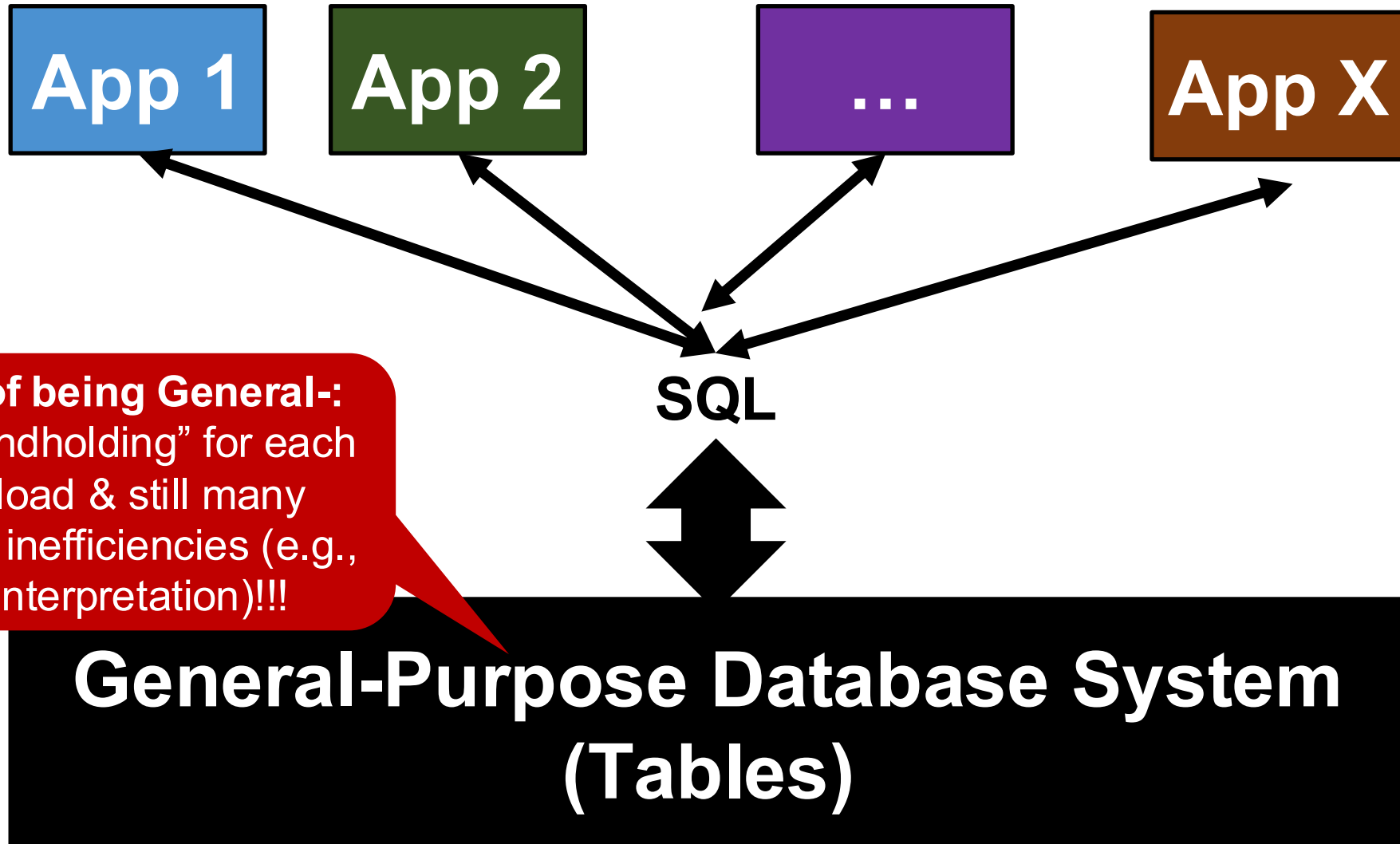
Tax 2: The Data Tax

→ High Overheads for Mapping Data to Tables

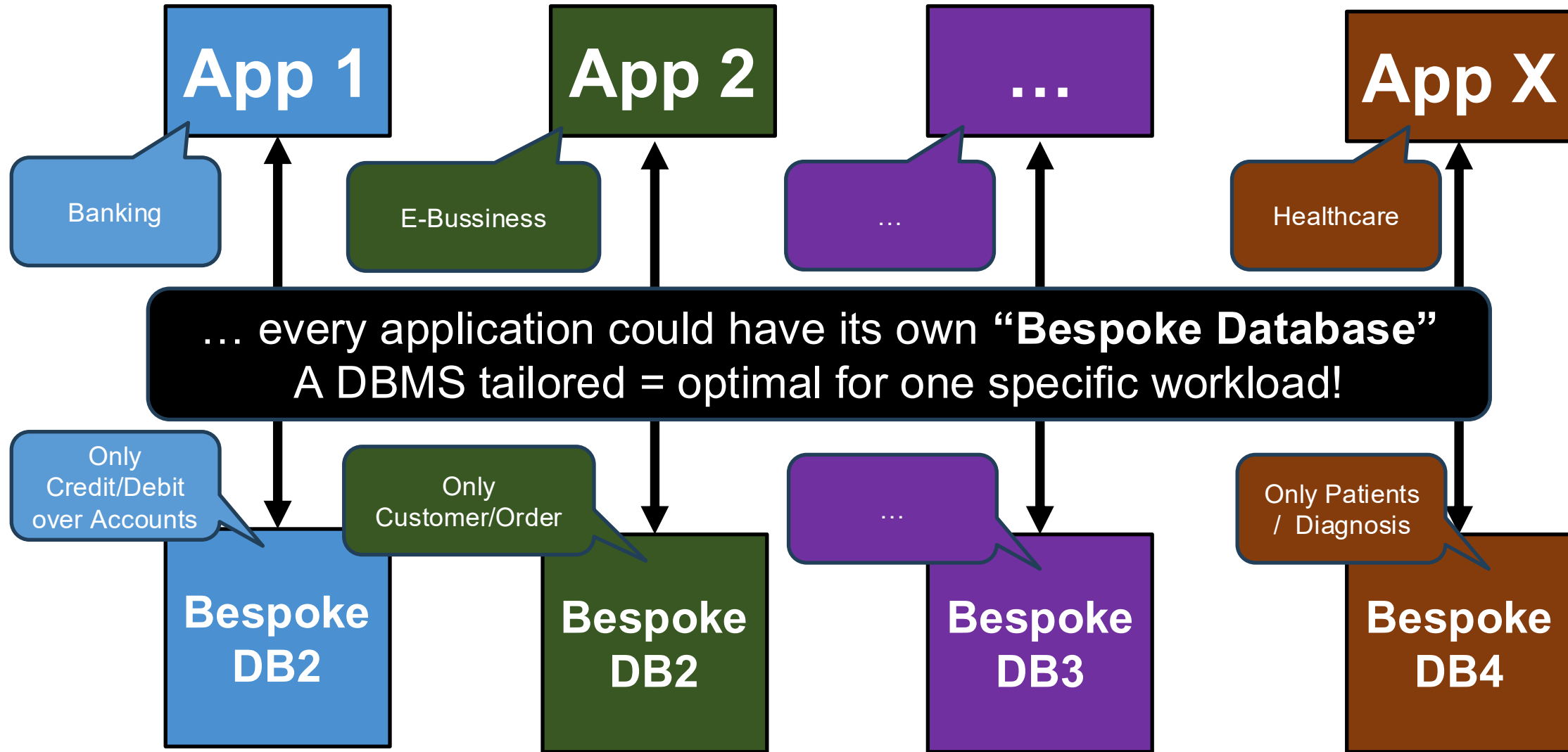
TAX

AI

DBMS Today: Being General-Purpose



Our Vision: Bespoke Databases!



Bespoke has already happened!

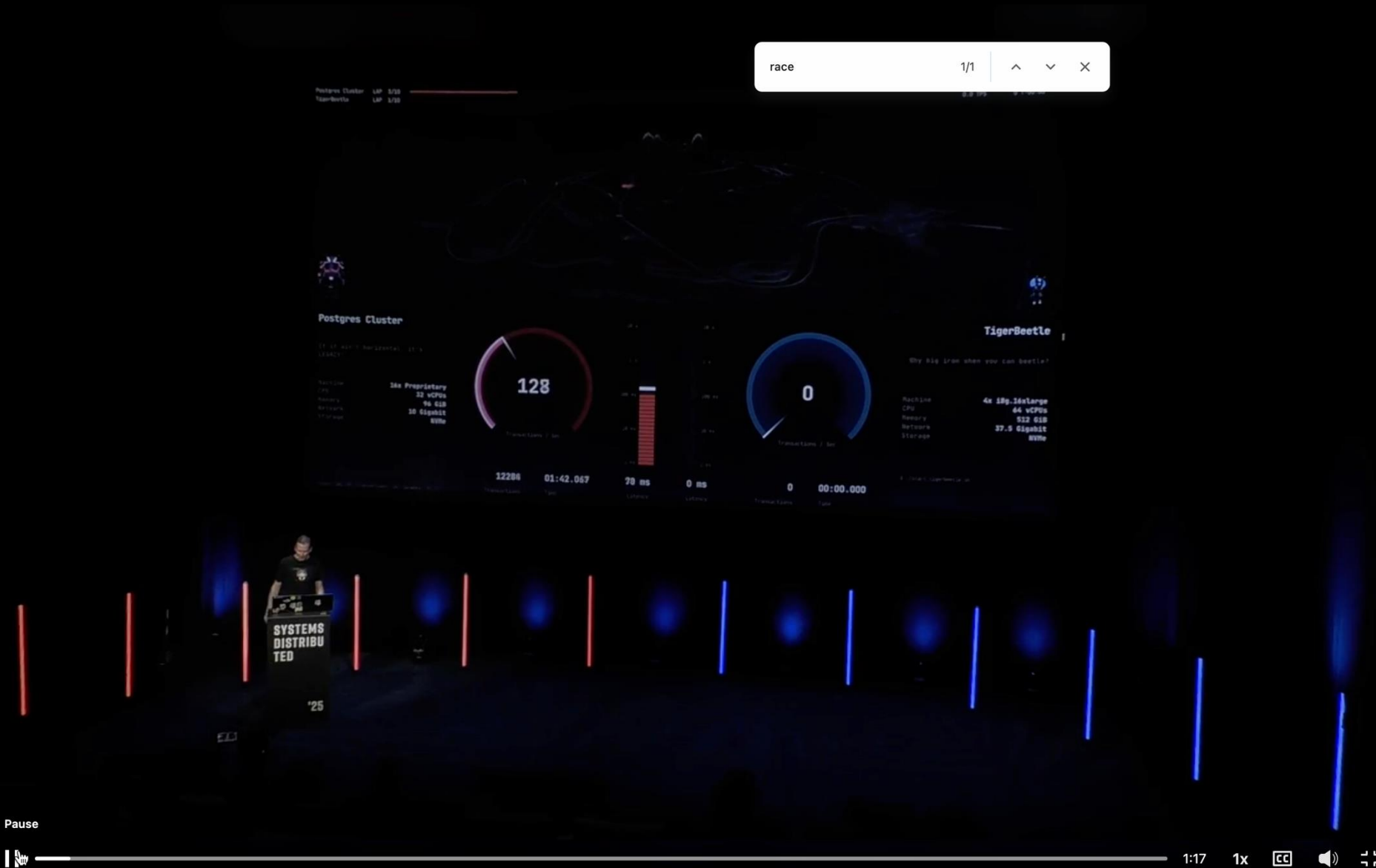
T *TigerBeetle*

The Financial Transactions Database

for Credit/Debit Transactions



Bespoke has already happened!



3000x
Faster
Than
Postgres

Bespoke is a Luxury! (Today)

Only
Credit/Debit
over Accounts

Bespoke
DB for
Finance

TigerBeetle is targeting a Big Market \$\$\$

Clone
Thomas?

BUT Bespoke for All? Too high Costs & ~~Missing Experts!~~

Bespoke
DB 1



Bespoke
DB 2



Bespoke
DB 3



...

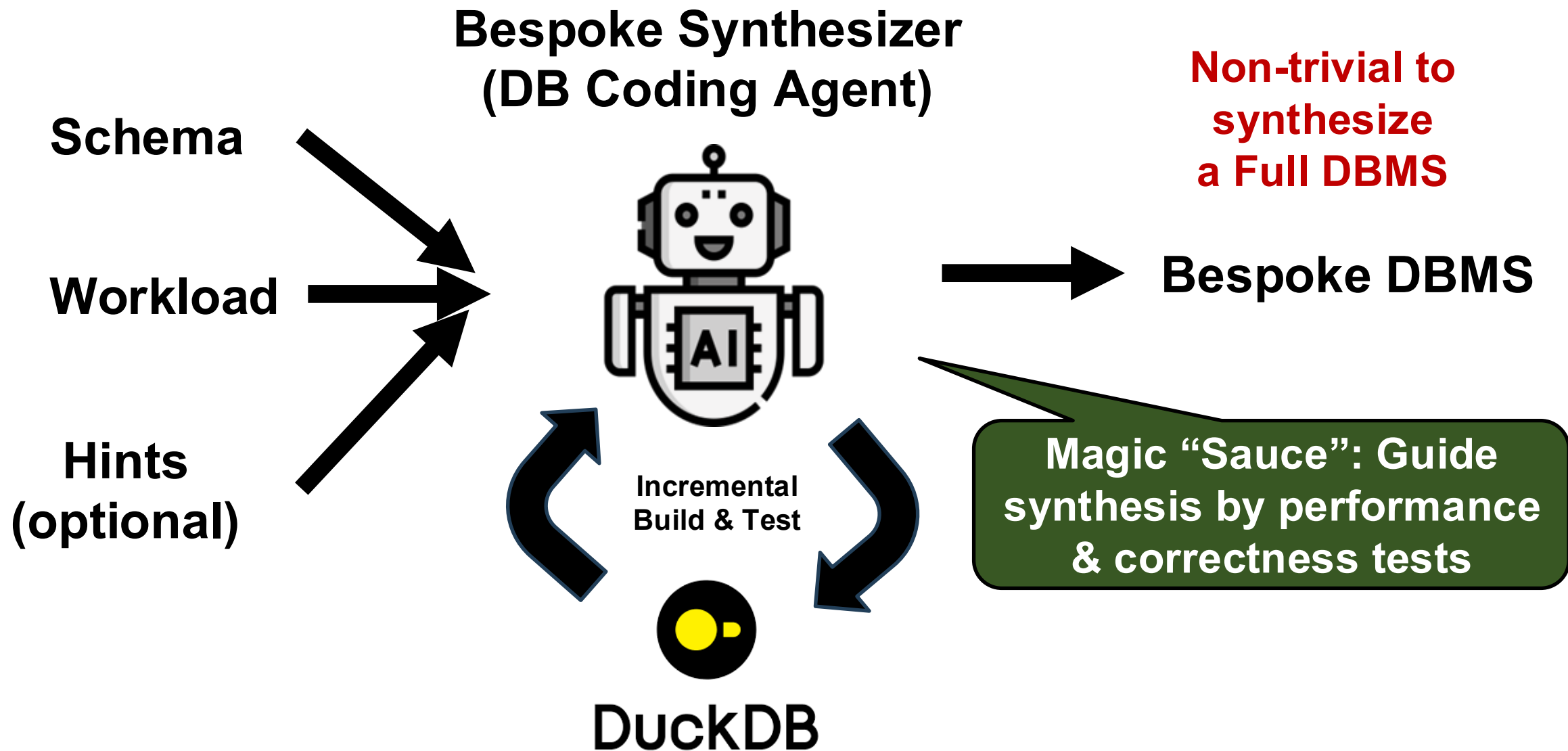


Bespoke
DB X

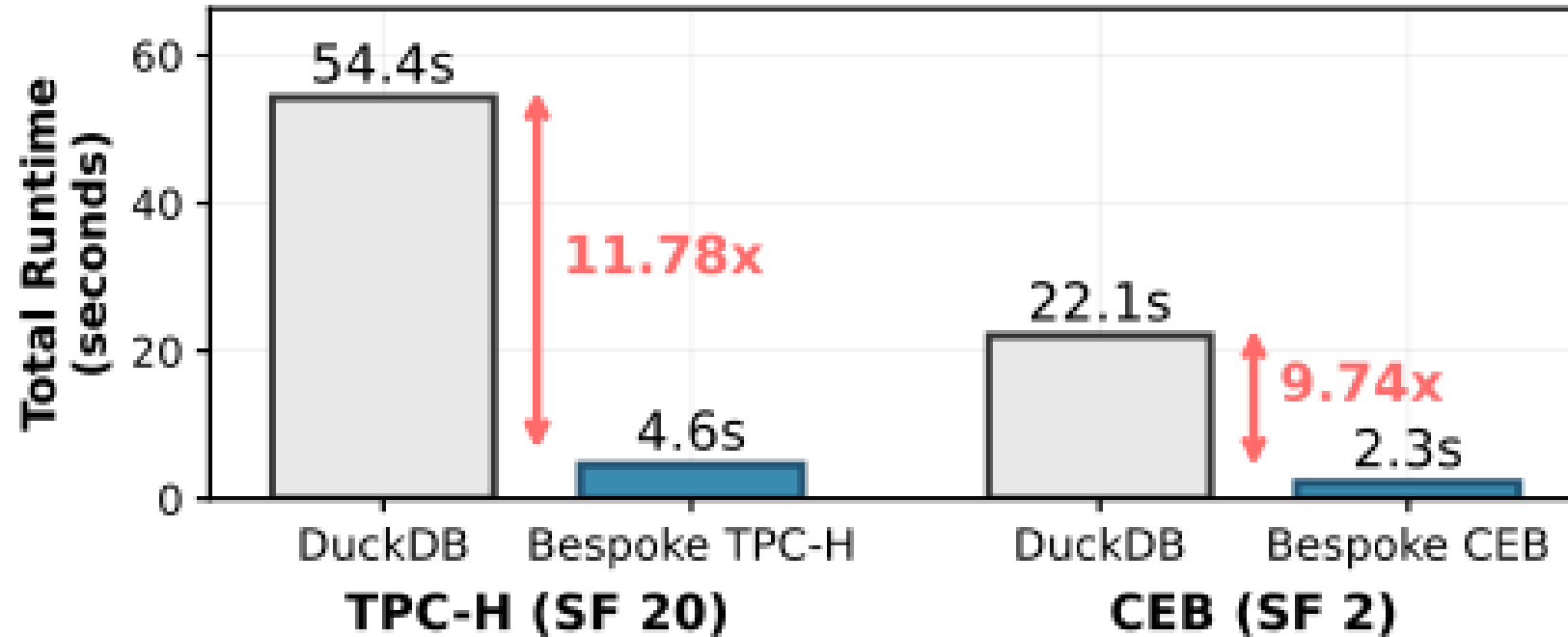


... wait? Can we not just synthesize a DBMS?

[Submitted to VLDB'26, on arXiv]



Results: Bespoke vs. General-Purpose OLAP

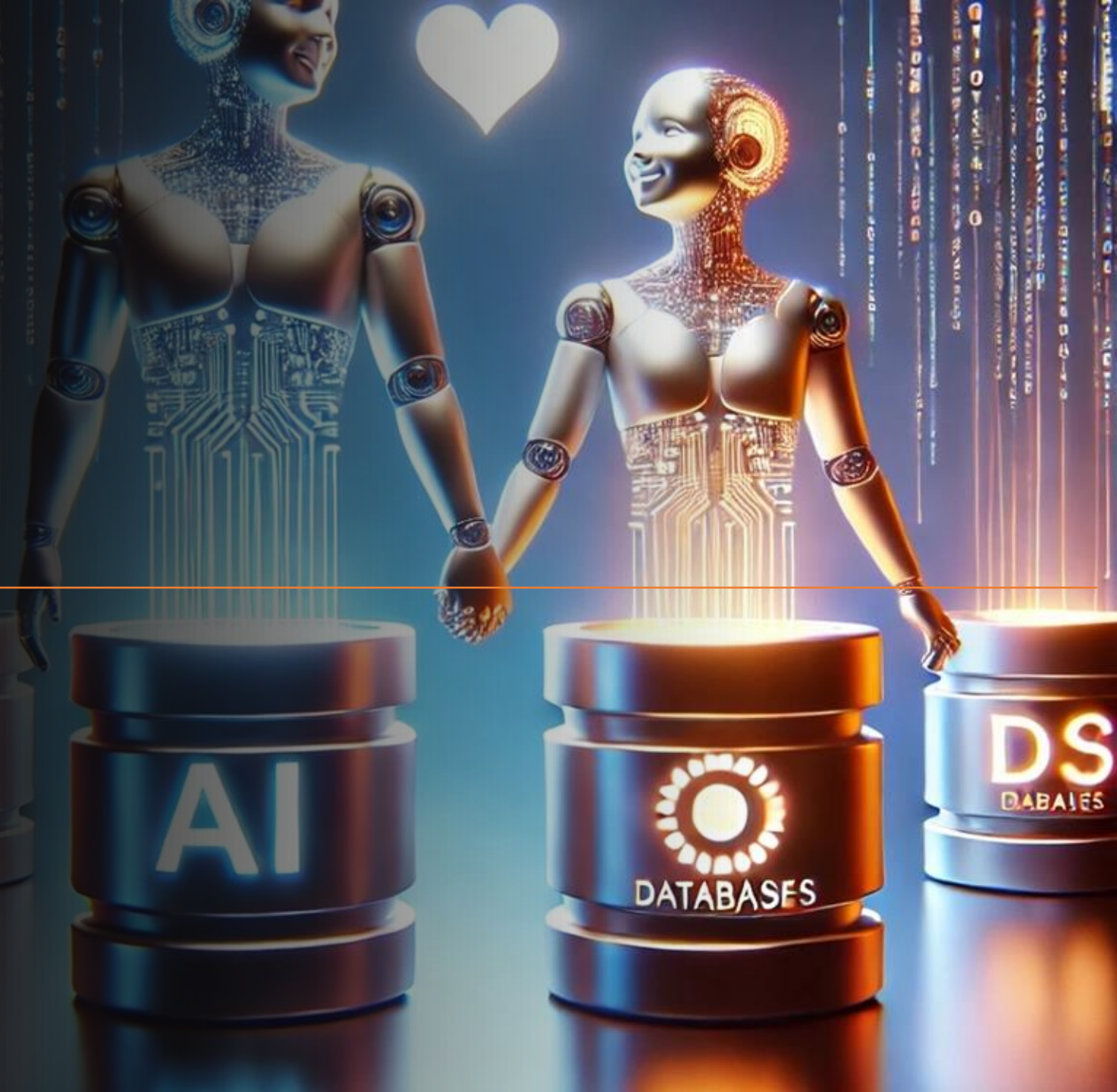


Bespoke is Fast & Cheap:

- Costs: ~100\$ for synthesizing query engine
- Generation Time: ~6 Hours

**We are still at
the beginning**

Join this effort!



Call to Action: Possible Directions

Multi-Modal Data Systems: Query Optimization (First Papers: Abacus from MIT, Stretto w Paolo Papotti – **submitted to VLDB, on arXiv**), Benchmarks, ...

LLM-based DBMS Synthesis: Only OLAP so far. Still many open opportunities in OLAP (e.g., Distributed OLAP)! How for OLTP? Other workloads than SQL...?

“Think out of the box”: NL-Querying data does not need to be SQL-like. Instead, NL-queries can be about problem solving with data (Example: SQL Murder Hunt → **Who is the murderer** given some facts)

(many more ...)

This is a Team Effort!



Thanks for your Attention

