Enhanced Enterprise Intelligence with your personal AI Data Copilot

by Marek Wiewiórka, Phd

getindata
Part of Xebia

Xebia

getindata
Part of Xebia

Xebia

Marek Wiewiórka

PhD | Chief Data Architect at GetInData

getindata Part of Xebia | Xebia

...how to turn best practices into AI coding assistant

1. Why do we need yet another (open-source) Copilot?
2. How can we build one?
3. Architecture and evaluation
4. DEMO

(Data) Context is king!

- *Explicit* and *precise* data context of your whole data platform
- Data transformation codebase
- Data models with comments and table relationships
- Other user queries
- Lineage and human curated dataset descriptions from data catalogs

Data Assistants landscape

- open-source tools, such as WrenAI, Venna.AI, Dataherald focus on Text-to-SQL to be embedded in web interfaces – i.e. chatbots or own SQL editors – meant for non-technical users.
- closed source AI-Powered Assistants to BigQuery (SQL+Dataform), Snowflake (SQL), Databricks (SQL+Python) web interfaces, more like a black-box not-meant for customizations.
- missing Analytics Engineer Copilot with a dbt/SQL support

getindata
Part of Xebia

Xebia

Customized and specialized models are the future.

databricks

Why Databricks Product Solutions Resources About DATA + AI SUMMIT

Build high-quality generative AI applications with DBRX customized for your unique data

by Jonathan Frankle, Ali Ghodsi, Naveen Rao, Hanlin Tang, Abhinav Venigalla and Matei Zaharia

March 27, 2024 in Company Blog

Share this post

Databricks' mission is to deliver data intelligence to every enterprise by allowing organizations to understand and use their unique data to build their own AI systems. Today, we are excited to advance our mission by open sourcing DBRX, a general purpose large language model (LLM) built by our Mosaic Research team that outperforms all established open source models on standard benchmarks. We believe that pushing the boundary of open source models enables generative AI for all enterprises that is customizable and transparent.

OpenAI

We believe that in the future, the vast majority of organizations will develop customized models that are personalized to their industry, business, or use case. With a variety of techniques available to build a custom model, organizations of all sizes can develop personalized models to realize more meaningful, specific impact from their AI implementations. The key is to clearly scope the use case, design and implement evaluation systems, choose the right techniques, and be prepared to iterate over time for the model to reach optimal performance.

snowflake
ARCTIC

Top-tier enterprise intelligence at incredibly low training cost

At Snowflake, we see a consistent pattern in AI needs and use cases from our enterprise customers. Enterprises want to use LLMs to build conversational SQL data copilots, code copilots and RAG chatbots. From a metrics perspective, this translates to LLMs that excel at SQL, code, complex instruction following and the ability to produce grounded answers. We capture these abilities into a single metric we call **enterprise intelligence** by taking an average of Coding (HumanEval+ and MBPP+), SQL Generation (Spider) and Instruction following (IFEval).

sqlcoder-7b and others

- Many other small (7-34b) models licensed for commercial use, e.g. :

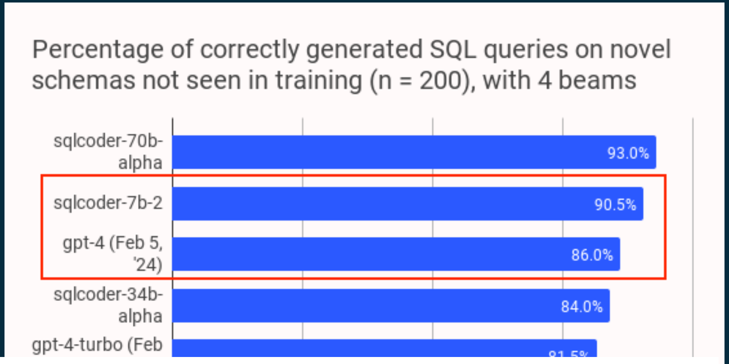
✓ starcoder2

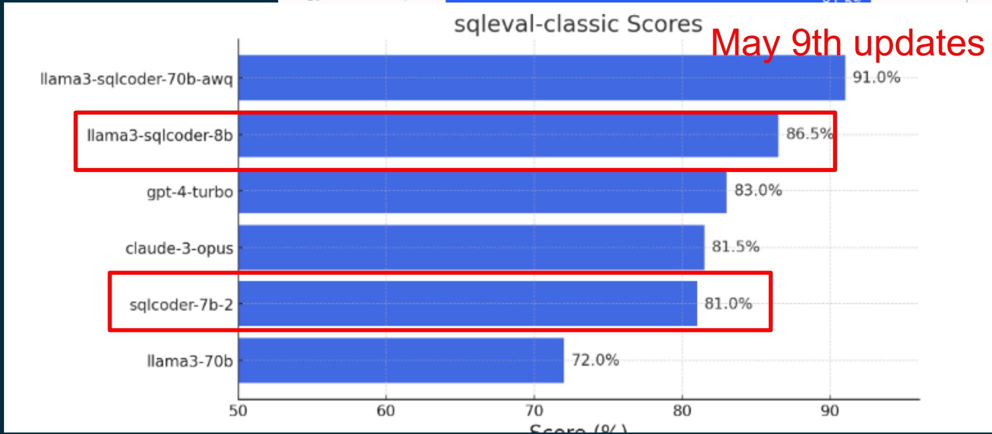
✓ dolphincoder

✓ deepseeek-coder

✓ Opencodeinterpreter

✓ Llama3





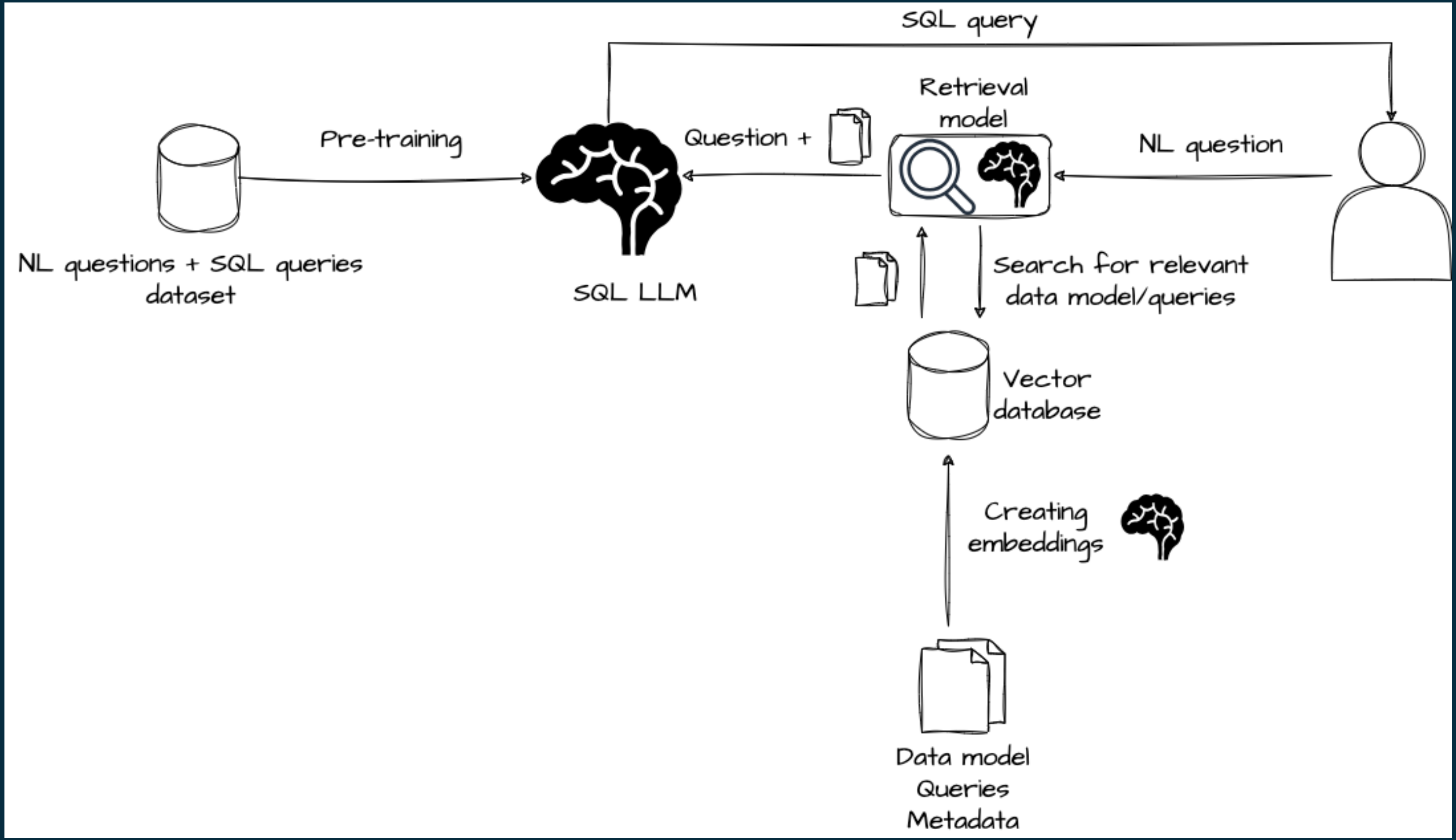
How turn your best practices into Copilots ?

- Vector database as a *<u>knowledge base </u>*- what ?
- Prompts as instructions following *<u>best practices </u>*- how ?
- LLM to *<u>combine</u>* both...

Retrieval-Augmented Generation(RAG)

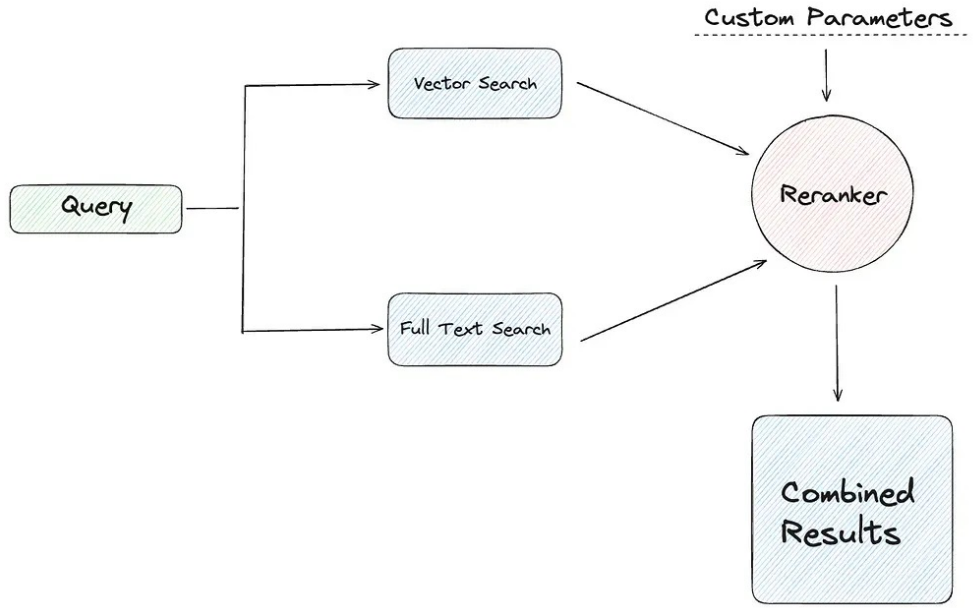
RAG for Text-to-SQL

getindata
Part of Xebia
Xebia



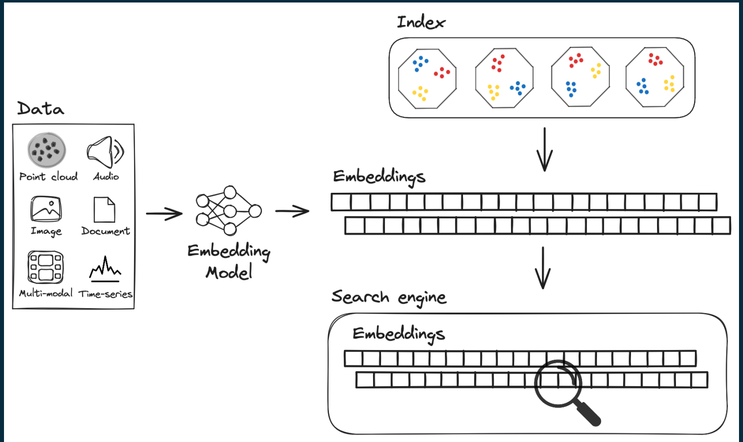
Hybrid search

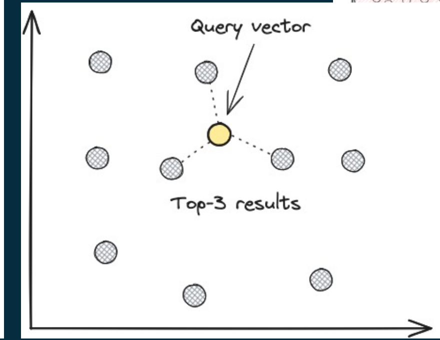
- combination of keyword and vector search

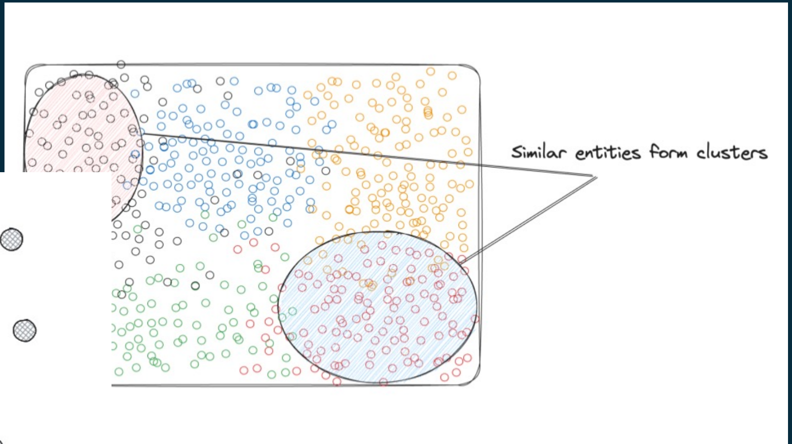


Vector search

- a technique used to search for similar items based on their vector representations, called embeddings
- Approximate Nearest Neighbours algorithms



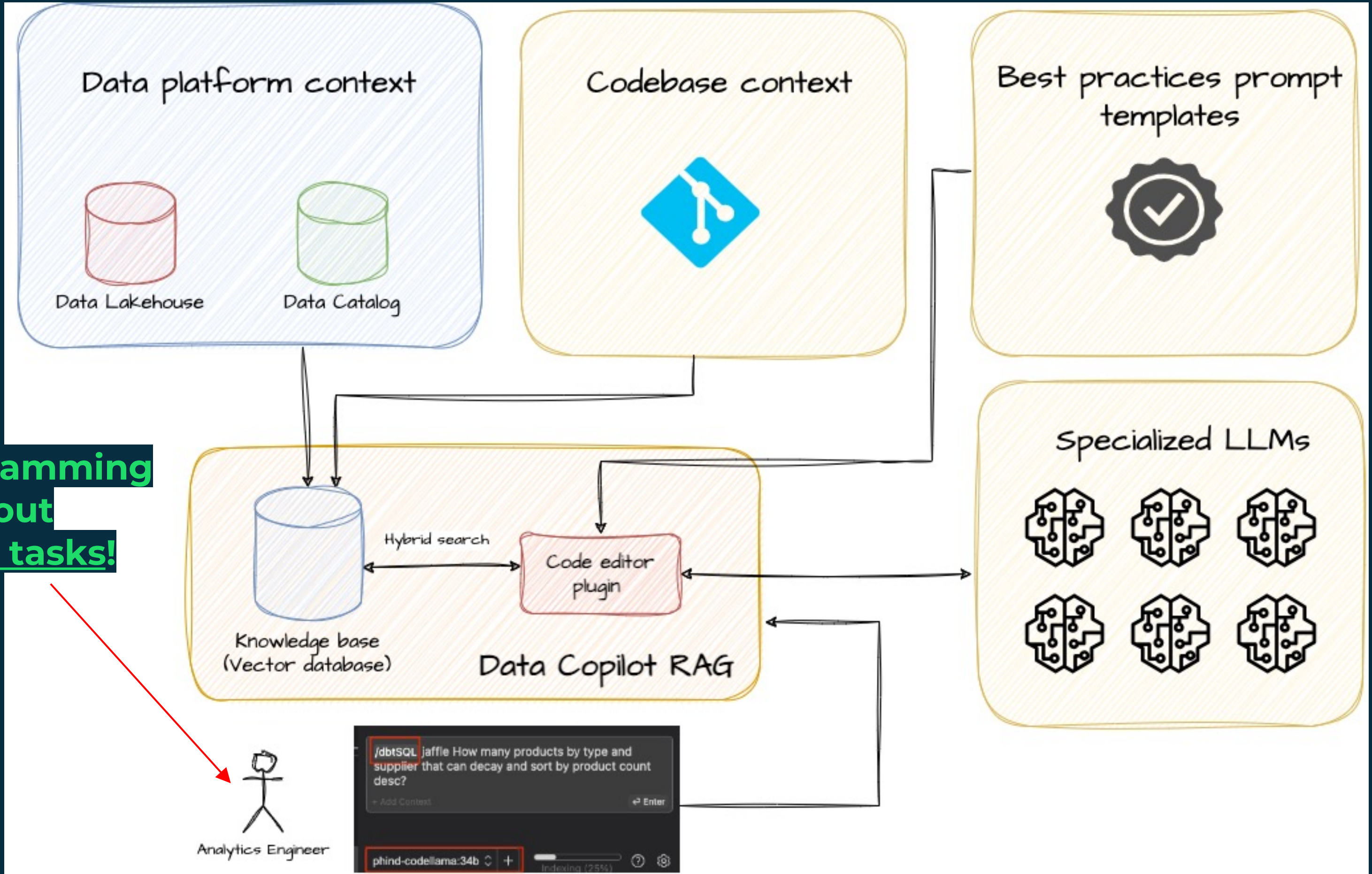




Data Copilot RAG architecture

getindata
Part of Xebia

Xebia

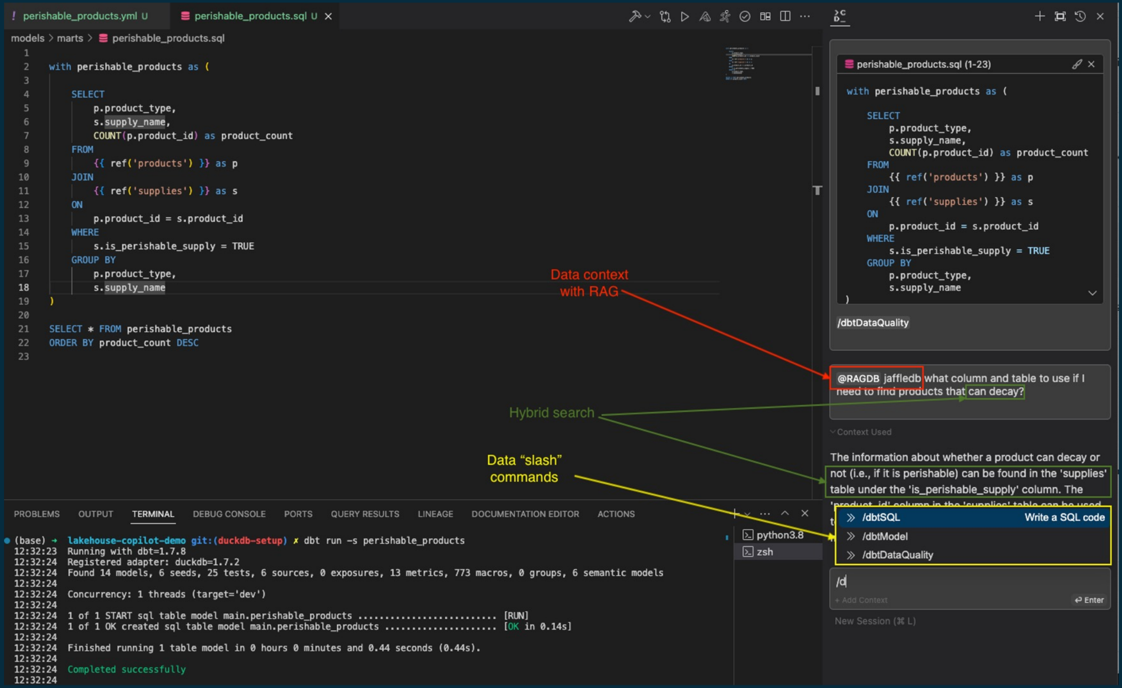


Data programming is more about repeatable tasks!

GID Data Copilot (GDC)

getindata
Part of Xebia

- An extensible AI **programming assistant** for **SQL** and **dbt** code
- Powered by:
  - **Large Language Models** (SOTA LLMs)
  - Robust **Retrieval Augmented Generation** (RAG) architecture
  - **Hybrid search** techniques
  - Fast **Vector Database**
  - Curated **Prompts**
  - Builtin **Data commands**



Continue - an open-source copilot

- support for both tasks and tab autocompletion
- highly extensible
  - use any LLM model you wish - also ***multiple, specialized models*** for different languages or tasks
  - support for many ***model providers***, such as Ollama, vLLM, LM Studio
  - custom ***context providers*** for more control over LLMs augmentation
  - custom ***slash commands*** that can combine own <u>prompts</u>, <u>contexts</u> and <u>models</u> for specialized, reusable tasks
- support for VSCode and Jetbrains
- secure (i.e. can be run locally, on-premise or cloud VPC)
- translate <u>your best practices</u> into ”slash data commands”

Continue - a custom context provider

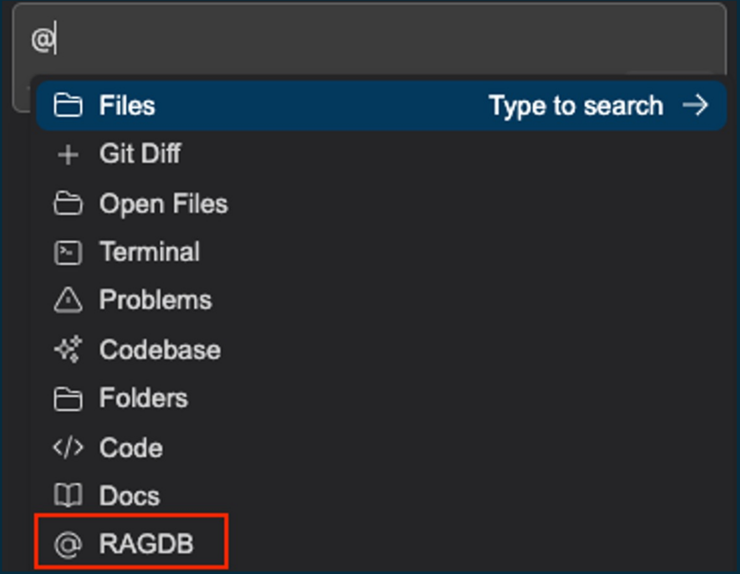
getindata
Part of Xebia

```
const RagContextProvider: CustomContextProvider = {
  title: "ragdb",
  displayTitle: "RAGDB",
  description:
    "Retrieve DB schema from our vector database of internal documents",
  type: "normal",

  getContextItems: async (
    query: string,
    extras: ContextProviderExtras
  ): Promise<ContextItem[]> => {
    console.info(extras.fullInput)
    const inputArray = extras.fullInput.split(' ');
    const db = inputArray[0];
    const userQuestion = inputArray.slice(2).join(' ');
    const response = await fetch("http://localhost:8000/retrieve", {
      method: "POST",
      headers: {
        'content-type': 'application/json;charset=UTF-8',
      },
      body: JSON.stringify({ query: userQuestion }),
    });

    const results = await response.json();

    return results.map((result: { title: any; contents: any; }) => ({
      name: result.title,
      description: result.title,
      content: result.contents,
    }));
  },
};
```

generate context

call retriever

user question



dbtSQL task = custom(context + prompt + model)

getindata
Part of Xebia

Xebia

```
export function modifyConfig(config: Config): Config {
  config.slashCommands?.push({
    name: "dbtSQL",
    description: "Write a SQL code",
    run: async function* (sdk) {

      const inputArray = sdk.input.split(' ');
      const db = inputArray[0];
      const userQuestion = inputArray.slice(2).join(' ');
      const response = await fetch("http://localhost:8000/retrieve", {
      method: "POST",
      headers: {
        'content-type': 'application/json;charset=UTF-8',
      },
      body: JSON.stringify({ query: userQuestion }),
    });

    const results = await response.json();

    const ragResponse = results.map((result: { title: any; contents: any; }) => ({
      name: result.title,
      description: result.title,
      content: result.contents,
    }))[0];

      const PROMPT = `
      ### Task
      Generate a SQL query to answer: ${userQuestion}

      ### Instructions
      - If you cannot answer the question with the available database schema, return 'I
      - Format answer as a code using markdown in the chat.

      ### Database Schema
      The query will run on a database with the following schema:
      ${ragResponse.content}
```

/dbtSQL jaffle How many products by type and supplier that can decay and sort by product count desc?

+ Add Context

↩ Enter

task = context + prompt + model

- GPT-4 (Free Trial)
- GPT-4 Vision (Free Trial)
- Gemini Pro (Free Trial)
- Codellama 70b (Free Trial)
- GPT-3.5-Turbo
- gpt-4-turbo-preview
- Ollama-llama2
- sqlcoder-7b
- pxlksr/defog_sqlcoder-7b-2:Q4_K
- pxlksr/defog_sqlcoder-7b-2:F16
- sqlcoder-15b
- starcoder2-15b
- deepseek-coder-6.7b

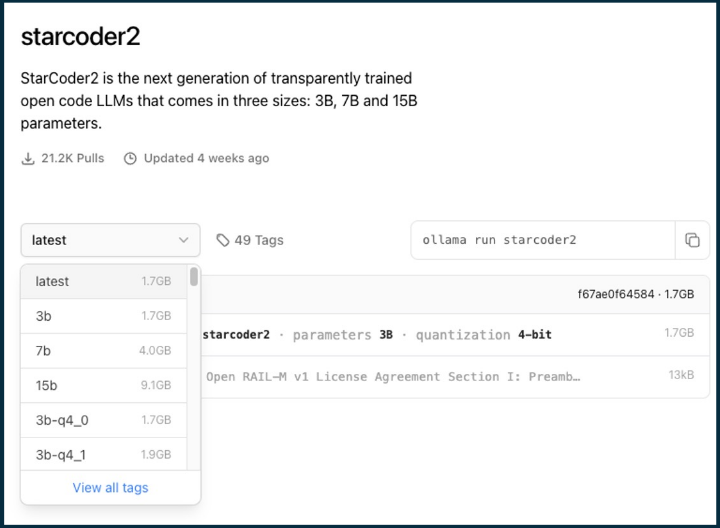
Ollama

- fast and easy self-hosting of LLMs almost *everywhere*
- hybrid CPU+GPU inference
- powered by llama.cpp
- rich library of existing LLMs in different flavours
- GGUF - fast and memory efficient quantization for GPU
- Serve model with one-liner:

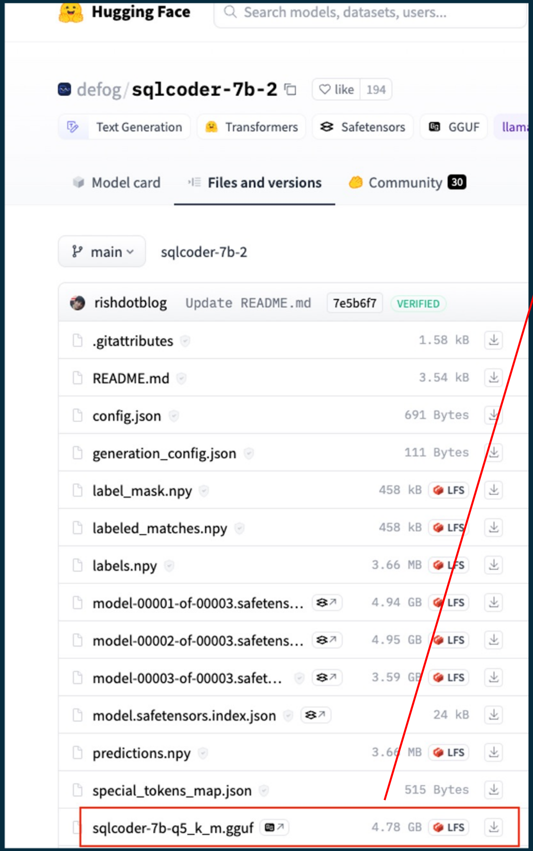
```
ollama run starcoder2:7b
```

- vLLM for production deployments

(Our video tutorial)



Ollama - custom model in 4 steps



1. Download a model in the GGUF format
2. Create a Modelfile, e.g.:

```
FROM ./sqlcoder-7b-q5_k_m.gguf
TEMPLATE """{{ .Prompt }}"""
PARAMETER stop "<|endoftext|>"
```

3. Create a model with Ollama

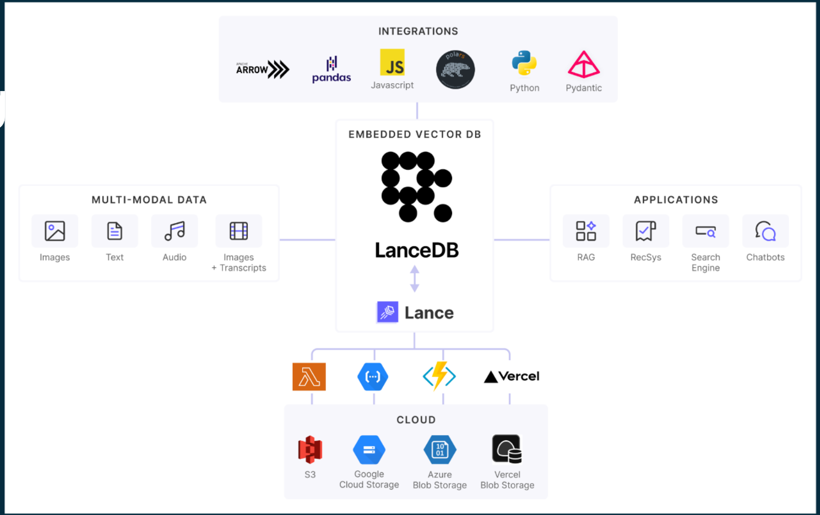
```
ollama create sqlcoder-7b-2 -f Modefile
```

4. Serve it

```
ollama run sqlcoder-7b-2
```

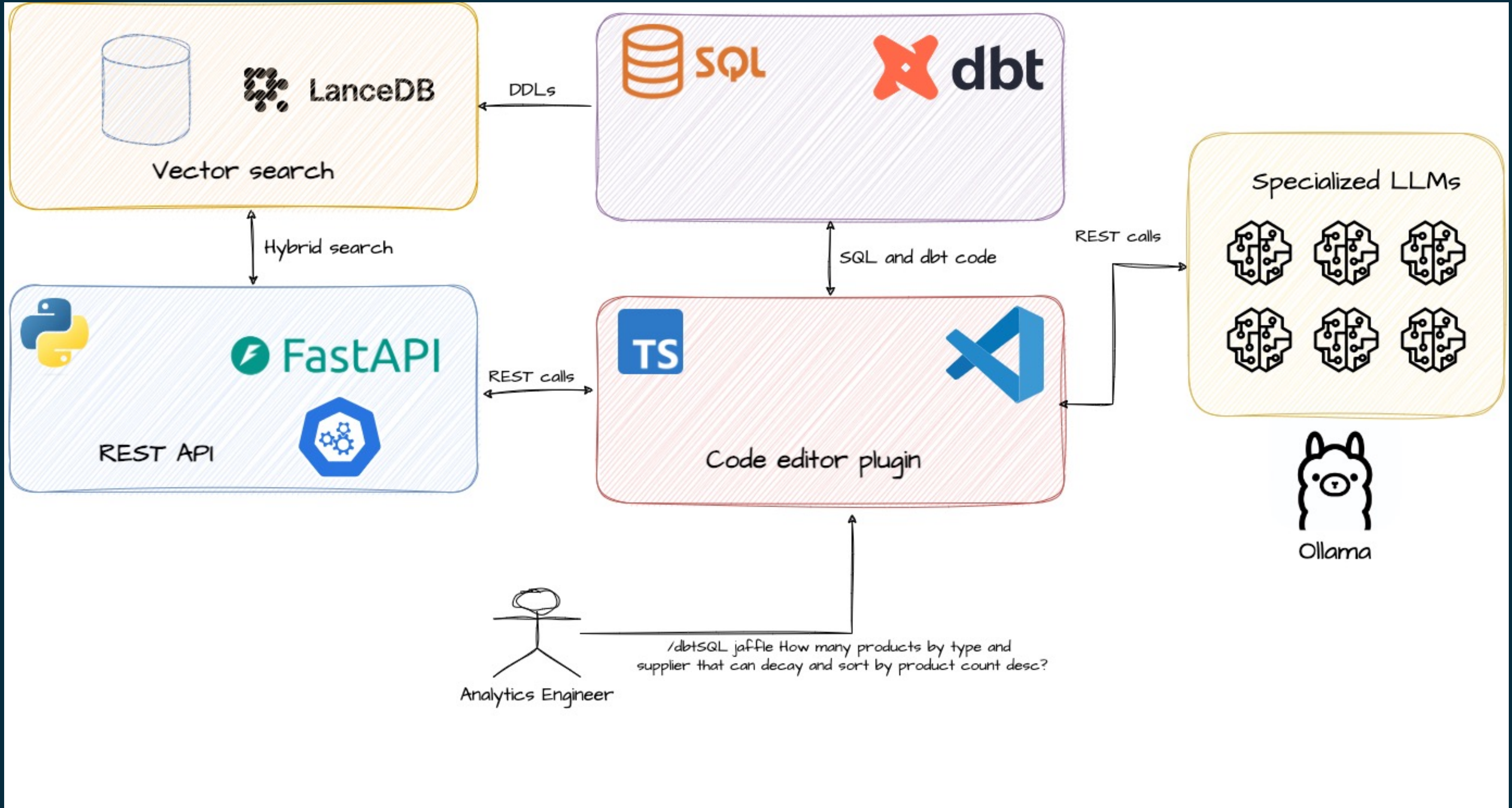
LanceDB

- fast (Rust❤️), serverless and embeddable - DuckDB for ML
- powered by Lance file format for ML (versioning, zero-copy)
- multi-modal
- support for hybrid (semantic + keyword) search
- Llamaindex integration
- Python API and fast data exchange with polars and Arrow



Technical architecture

getindata
Part of Xebia
Xebia



Question representation[1]

```
Table continents, columns = [ContId, Continent]
Table countries, columns = [CountryId, CountryName,
↳ Continent]
Q: How many continents are there?
A: SELECT
```

Listing 1: Example of Basic Prompt

```
Given the following database schema:
continents: ContId, Continent
countries: CountryId, CountryName, Continent

Answer the following: How many continents are there?
SELECT
```

Listing 2: Example of Text Representation Prompt

```
### Complete sqlite SQL query only and with no
↳ explanation
### SQLite SQL tables, with their properties:
#
# continents(ContId, Continent)
# countries(CountryId, CountryName, Continent)
#
### How many continents are there?
SELECT
```

Listing 3: Example of OpenAI Demostration Prompt

```
/* Given the following database schema: */
CREATE TABLE continents(
    ContId int primary key,
    Continent text,
    foreign key(ContId) references countries(Continent)
);

CREATE TABLE countries(
    CountryId int primary key,
    CountryName text,
    Continent int,
    foreign key(Continent) references continents(ContId)
);

/* Answer the following: How many continents are there?
↳ */
SELECT
```

Listing 4: Example of Code Representation Prompt

[1]Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation

LLMs evaluation 1/2

- Not meant to be yet another benchmark, such as: Spider sql-eval or Bird-SQL for jus SQL generation
- Jaffle Shop example - simple but not trivial
- Zero-shot – Agentic Workflow with Reflection TBD
- 4 typical data tasks
  - Data model exploration/discovery
  - SQL: an easy one (single table) and more complex (joins with sorting and aggregations)
  - dbt model generation
  - dbt tests generation based on rules

LLMs evaluation 2/2

getindata
Part of Xebia

Xebia

Model	Licence	size [b]	Data discovery	SQL - simple	SQL - complex	dbt - model	dbt - tests
deepseek-coder	deepseek	33	+	+	+/-	+/-	-
deepseek-coder	deepseek	6.7	+/-	+/-	+/-	-/+	-
codellama	Llama2	70	+	+	-/+	-	-
starcoder2	bigcode-openrail-m	15	-	-	-	-	-
sqlcoder	CC BY-SA 4.0	7	-	+/-	+/-	N/A	N/A
phind-codellama	Llama2	34	+	+	+	+	+/-
wizardcoder	Llama2	33	+	+	+/-	+/-	-/+
gpt-3.5-turbo	Commercial	N/A	+	+	+	+	+/-
gpt-4	Commercial	N/A	+	+	+	+	+
gpt-4-turbo-preview	Commercial	N/A	+	+	+	+	+/-
Gemini Pro	Commercial	N/A	+	+/-	+	+/-	-
OpenCodeInterpreter	Apache-2.0	33	+	+	-/+	-	-

+- perfect or almost perfect
+/- - not optimal or some minor tweaks needed
-/+ - not very helpful, serious hallucinations
- - totally useless

gpt4 vs dbrx vs sqlcoder-7b-2 vs llama-3-sqlcoder-8b

getindata
Part of Xebia

Xebia

```sql
--gpt 4
SELECT
    p.product_type,
    s.supply_name,
    COUNT(p.product_id) AS product_count
FROM
    products p
JOIN
    supplies s ON p.product_id = s.product_id
WHERE
    s.is_perishable_supply = TRUE
GROUP BY
    p.product_type, s.supply_name
ORDER BY
    product_count DESC;
```

```sql
-- dbrx
SELECT
    p.product_type,
    s.supply_name,
    COUNT(DISTINCT p.product_id) as product_count
FROM
    products p
JOIN
    supplies s ON p.product_id = s.product_id
WHERE
    s.is_perishable_supply = TRUE
GROUP BY
    p.product_type, s.supply_name
ORDER BY
    product_count DESC;
```

```sql
--llama-3-sqlcoder-8b
SELECT
    p.product_type,
    s.supply_name,
    COUNT(p.product_id) AS product_count
FROM
    products p
JOIN
    supplies s ON p.product_id = s.product_id
WHERE
    p.is_perishable_product = TRUE
GROUP BY
    p.product_type, s.supply_name
ORDER BY
    product_count DESC NULLS LAST;
```

```sql
-- sqlcoder-7b-2 (codellama-7b)
SELECT p.product_type,
       s.supply_name,
       COUNT(DISTINCT p.product_id) AS product_count
FROM
    supplies s
JOIN
    order_items oi ON s.supply_id = oi.product_id
AND
    s.is_perishable_supply = TRUE
JOIN
    products p ON oi.product_id = p.product_id
GROUP BY
    p.product_type, s.supply_name
ORDER BY
    product_count DESC
```

fine-tuning impact: llama3-8b vs llama-3-sqlcoder-8b

```sql
--llama-3
SELECT
    s.supply_name,
    p.product_type,
    COUNT(*) AS product_count
FROM
    supplies s
JOIN
    order_items oi ON s.product_id = oi.product_id
JOIN
    products p ON oi.product_id = p.product_id
WHERE
    s.is_perishable_supply = TRUE AND
    (oi.is_food_item = TRUE OR oi.is_drink_item = TRUE)
GROUP BY
    s.supply_name, p.product_type
ORDER BY
    product_count DESC;
```

```sql
--llama-3-sqlcoder-8b
SELECT
    p.product_type,
    s.supply_name,
    COUNT(p.product_id) AS product_count
FROM
    products p
JOIN
    supplies s ON p.product_id = s.product_id
WHERE
    p.is_perishable_product = TRUE
GROUP BY
    p.product_type, s.supply_name
ORDER BY
    product_count DESC NULLS LAST;
```

Quantization effects: dbrx 8/4/2 bit

getindata
Part of Xebia

Xebia

```sql
-- dbrx
SELECT
    p.product_type,
    s.supply_name,
    COUNT(DISTINCT p.product_id) as product_count
FROM
    products p
JOIN
    supplies s ON p.product_id = s.product_id
WHERE
    s.is_perishable_supply = TRUE
GROUP BY
    p.product_type, s.supply_name
ORDER BY
    product_count DESC;
```

```sql
--dbrx:132b-instruct-q8_0-cpu (144GB)
SELECT
    p.product_type,
    s.supply_name,
    COUNT(*) AS product_count
FROM
    supplies s
JOIN
    products p ON s.product_id = p.product_id
WHERE
    s.is_perishable_supply = TRUE
AND (p.is_food_item = TRUE OR p.is_drink_item = TRUE)
GROUP BY
    p.product_type, s.supply_name
ORDER BY
    product_count DESC;
```

```sql
--132b-instruct-q4_0 (74GB)
SELECT
    p.product_type,
    s.supply_name,
    COUNT(p.product_id) AS product_count
FROM
    supplies s
JOIN
    order_items oi ON s.product_ id = oi.product_id
JOIN
    products p ON oi.product_id = p.product_id
WHERE
    is_perishable_supply = TRUE
AND
    (is_food_item = TRUE OR is_drink_item = TRUE)
GROUP BY
    1,2
ORDER BY
    product_count DESC;
```

```
--132b-instruct-q2_K (48GB)
is.;.

is or order in files are orders' will. _product. orders.
`ot order to not order objects. (ut products, order
```

A handful of conclusions...with a grain of salt

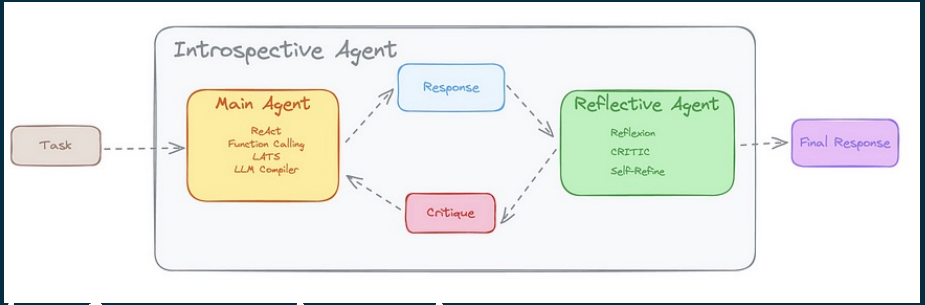
getindata
Part of Xebia

Xebia

- NL-to-SQL and dbt code generation are ***challenging***
- ***commercial*** models are in most cases still better... but
- there are very promising ***open-source 7-30b alternatives***
- ***smaller*** models perform better than larger after **quantization**
- ***SQL-dedicated*** and fine-tuned models can turn out a bit a disappointing (beam search?), e.g. :
    - unnecessary joins elimination
    - wrong data types inference
    - occasional hallucinations

Future directions

- Implementation of in-context learning, such as Query Similarity Selection (few-shot strategy) and *Agentic RAG with Reflection Strategy*
- *Model fine-tuning (dbt)*
- *Data Modeling (DV 2.0)*
- *Various SQL dialects and platform migrations*
- *Prompt optimizations, e.g. with DSPy*



Welcome to the GID data copilot

DEMO

getindata
Part of Xebia

Xebia

Want to talk about DATA & AI with US?

contact@getindata.com

SCHEDULE A CONSULTATION