EDB
Postgres for the AI Generation

EDB AI : Modern Analytics & AI

Franck Sidi - Senior DIrector EMEA & WW Field CTO Leader
franck.sidi@enterprisedb.com

Franck Sidi - Joined EDB, February 2024

Driving Innovation in Databases & Analytics

- **30 Years of Experience**: Leadership roles across top-tier US software companies, including Sybase, ATG, Microsoft, EMC/Pivotal, and VMware.
- **Proven Track Record**: Expertise in Sales, Pre-Sales, Post-Sales, and Customer Success Management, consistently delivering results and building high-performing teams.
- **Passion for Technology & Innovation**: Deeply involved in cutting-edge projects—POCs, query optimization, infrastructure sizing, and the application of LLMs and Generative AI.

Key Leadership Contributions

- **Team Building & Development**: Dedicated to fostering collaboration, growth, and success within teams.
- **Strategic Execution**: Driving technology and innovation while optimizing solutions for enterprise-scale success.

Personal Snapshot

- **Background**: 57 years old, French, living in Israel since 2008.
- **Family**: Proud father of four (ages 26, 25, 25, 14).
- **Hobbies**: Passionate about music, history, running, and supporting the Nice football team.

EDB

Agenda

Agenda

- EDB Overview
- EDB Analytics Platform
  - Demos
- EDB AI
  - Demos
- Q&A

EDB POSTGRES AI PLATFORM

UNIFIED WORKLOAD MANAGEMENT

- TRANSACTIONAL
- ANALYTICAL
- ARTIFICIAL INTELLIGENCE

SINGLE PANE OF GLASS ADMINISTRATION

- HYBRID DATA ESTATE
- INTELLIGENT OBSERVABILITY
- ENTERPRISE SECURITY

HYBRID AND MULTI-CLOUD DEPLOYMENT

- PUBLIC CLOUD (MANAGED)
- PRIVATE CLOUD (SOFTWARE)
- ON PREMISES (APPLIANCE)

PLATFORM TOOLS AND SERVICES

- MIGRATION PORTAL
- CONTINUOUS HIGH AVAILABILITY
- BACKUP AND RECOVERY

EXTENSIBILITY

- CSP INTEGRATIONS
- DEVOPS TOOLING
- KUBERNETES TOOLING
- GENAI & LLM INTEGRATIONS
- LAKEHOUSE INTEGRATIONS

Delivered with world-class strategic partners:

Red Hat

SUPERMICRO®

NUTANIX

EDB Analytics

EDB POSTGRES AI PLATFORM

UNIFIED WORKLOAD MANAGEMENT

- TRANSACTIONAL
- ANALYTICAL
- ARTIFICIAL INTELLIGENCE

SINGLE PANE OF GLASS ADMINISTRATION

- HYBRID DATA ESTATE
- INTELLIGENT OBSERVABILITY
- ENTERPRISE SECURITY

HYBRID AND MULTI-CLOUD DEPLOYMENT

- PUBLIC CLOUD (MANAGED)
- PRIVATE CLOUD (SOFTWARE)
- ON PREMISES (APPLIANCE)

PLATFORM TOOLS AND SERVICES

- MIGRATION PORTAL
- CONTINUOUS HIGH AVAILABILITY
- BACKUP AND RECOVERY

EXTENSIBILITY

- CSP INTEGRATIONS
- DEVOPS TOOLING
- KUBERNETES TOOLING
- GENAI & LLM INTEGRATIONS
- LAKEHOUSE INTEGRATIONS

Delivered with world-class strategic partners:

Red Hat

SUPERMICRO

NUTANIX

Key Business Challenges Addressed by EDB Analytics

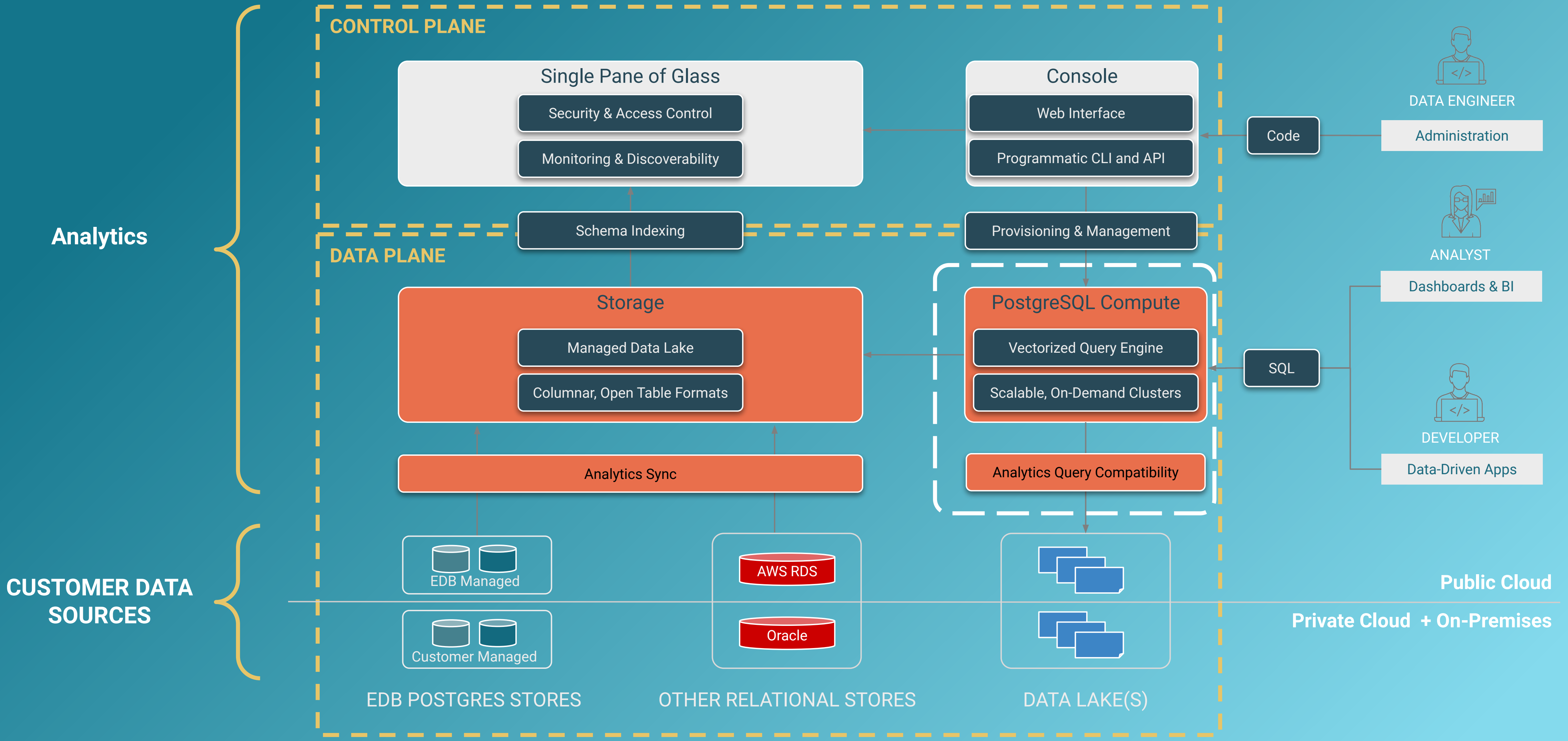
High-Performance Analytics: Extremely Fast Queries

Cost-Effective Solution: Lower Ownerships Costs & Ease of Use

Real-Time Data Insights: Efficient Data Processing & Open Data Formats

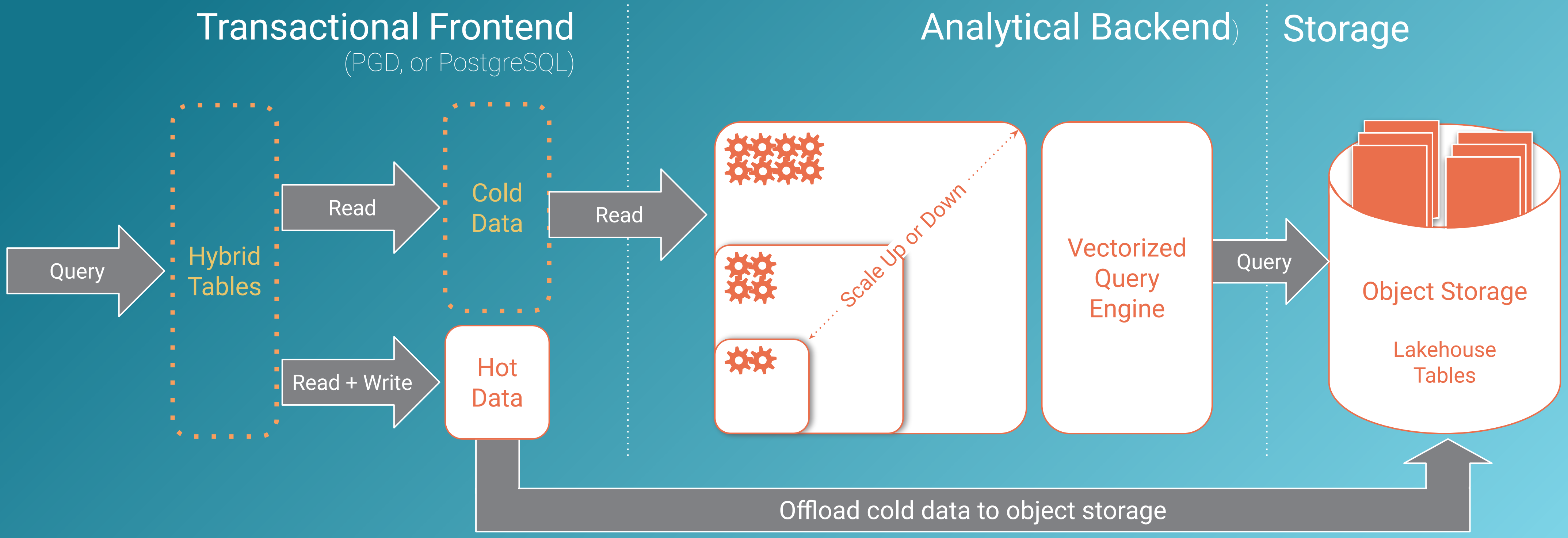
EDB Analytics: Hybrid-Cloud Solution, 100% Postgres

EDB Analytics is how you run analytical workloads on the EDB Postgres AI platform



Tiered Analytics on “hot” and “cold” data

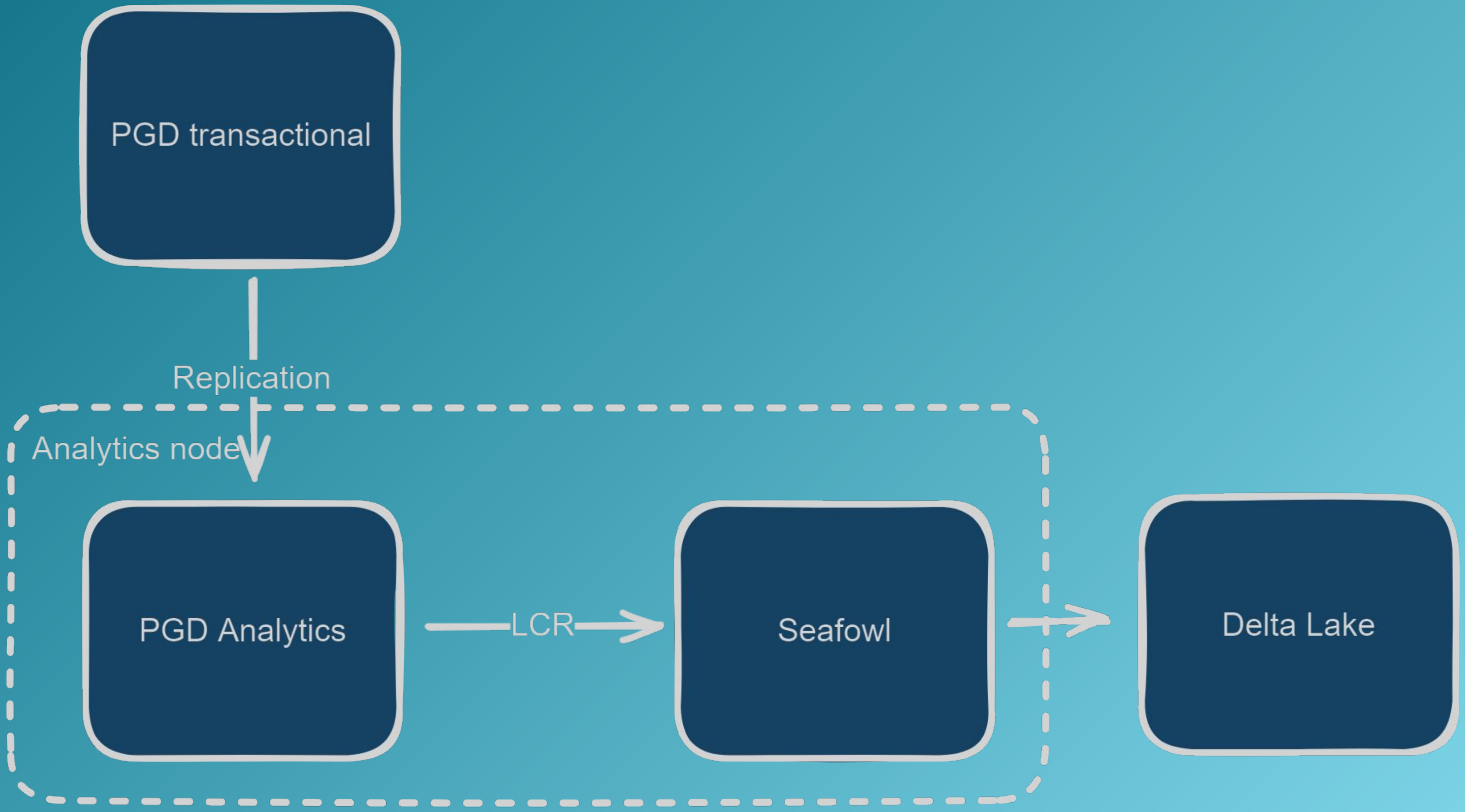
Solution: Hybrid tables merge hot data with cold data offloaded to analytics system for storage and processing



HTAP Scenario

H2 2024

Continuous replication with Postgres Distributed (PGD) → EDB Analytics



Tables Objects in EDB Analytics

4 kinds of tables on our EDB Analytics

1. **Transactional tables**
   a. The normal local PG tables
2. **Offloaded/Analytical tables**
   a. Tables that live on S3 or Storage Location
3. **HTAP tables**
   a. tables that have both local data and also have a copy of the data on S3 or Storage Location
4. **Volume tables**
   a. Analogue of Databricks' volume - for AIDB use with fixed column definitions (something like filename text, size int8, last_modify timestamptz, contents bytea) representing a list of files/objects in a storage location.

Tables Objects in EDB Analytics

4 kinds of tables on our EDB Analytics

- **1 - Transactional tables**
  - The normal local PG tables

```
CREATE TABLE cities (
city_id    bigserial NOT NULL PRIMARY KEY,
name       text NOT NULL,
population bigint
);
```

- **2 - Offloaded/Analytical tables**
  - Tables that live on S3 or Storage Location

```
create table tpch_sf_1000_lfs.lineitem ()
using PGAA WITH (pgaa.storage_location = 'local-fs-1000', pgaa.path = 'lineitem');
```

Tables Objects in EDB Analytics

4 kinds of tables on our EDB Analytics

- **3 - HTAP tables**
  - tables that have both local data and also have a copy of the data on S3 or Storage Location

```
CREATE TABLE cities (
  city_id    bigserial NOT NULL PRIMARY KEY,
  name       text NOT NULL,
  population bigint
) WITH (pgd.replicate_to_analytics = true, pgfs.server = 'my_ai_lakehouse', pgfs.path =
'parquet/table1/');
```

- **4 - Volume tables**
  - Analogue of Databricks' volume - for AIDB use with fixed column definitions (something like filename text, size int8, last_modify timestamptz, contents bytea) representing a list of files/objects in a storage location

```
CREATE FOREIGN TABLE my_pdf_volume () SERVER my_ai_lakehouse
OPTIONS (path 'pdf/', mime_type 'application/pdf');
```

How to Work with EDB Analytics

1. Add EDB Extension
2. Define Storage Location using S3 or Local Storage
3. Migrate Data to Delta Table Format
4. Create Table
5. Run Queries

No Index, No Tuning

```
postgres=# \dx
                    List of installed extensions
  Name   | Version |   Schema   |              Description
---------+---------+------------+-------------------------------------------------
 pgaa    | 0.1.0   | pgaa       | pgaa extension: (c) 2023 EnterpriseDB Corporation
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```

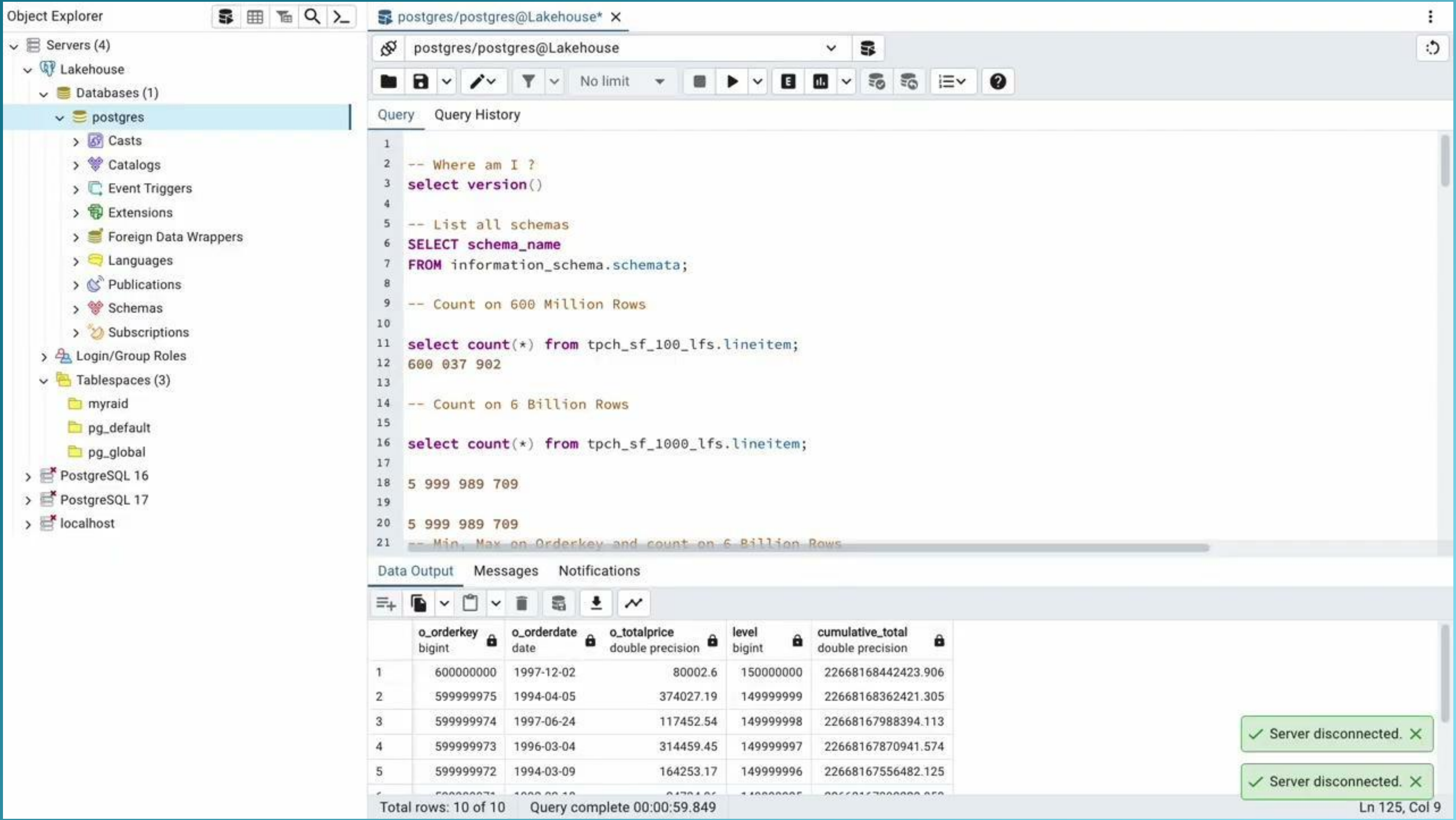
```
SELECT pgaa.create_storage_location('local-fs-100', 'file:///mnt/raid/parquet/100', '{}', NULL);
SELECT pgaa.create_storage_location('local-fs-1000', 'file:///mnt/raid/parquet/1000', '{}', NULL);
```

```
./lakehouse-loader pg-to-delta postgres://postgres@localhost:5432/postgres?sslmode=disable -q
"SELECT L_ORDERKEY, L_PARTKEY, L_SUPPKEY, L_LINENUMBER, L_QUANTITY, L_EXTENDEDPRICE,
L_DISCOUNT, L_TAX, L_RETURNFLAG::TEXT AS L_RETURNFLAG, L_LINESTATUS::TEXT AS
L_LINESTATUS, L_SHIPDATE, L_COMMITDATE, L_RECEIPTDATE, L_SHIPINSTRUCT::TEXT AS
L_SHIPINSTRUCT, L_SHIPMODE::TEXT AS L_SHIPMODE, L_COMMENT::TEXT AS L_COMMENT from
tpch_sf_1000.lineitem" file:///mnt/raid/parquet/1000/lineitem
```

```
create table tpch_sf_1000_lfs.lineitem () using PGAA WITH
(pgaa.storage_location = 'local-fs-1000', pgaa.path =
'lineitem');
```

```
postgres=# \timing
Timing is on.
postgres=# select count(*) from tpch_sf_1000_lfs.lineitem;
   count
------------
 5999989709
(1 row)

Time: 121.911 ms
postgres=#
```

Explore a 6 Billion Rows Dataset
in a matter of a few seconds with
EDB Analytics



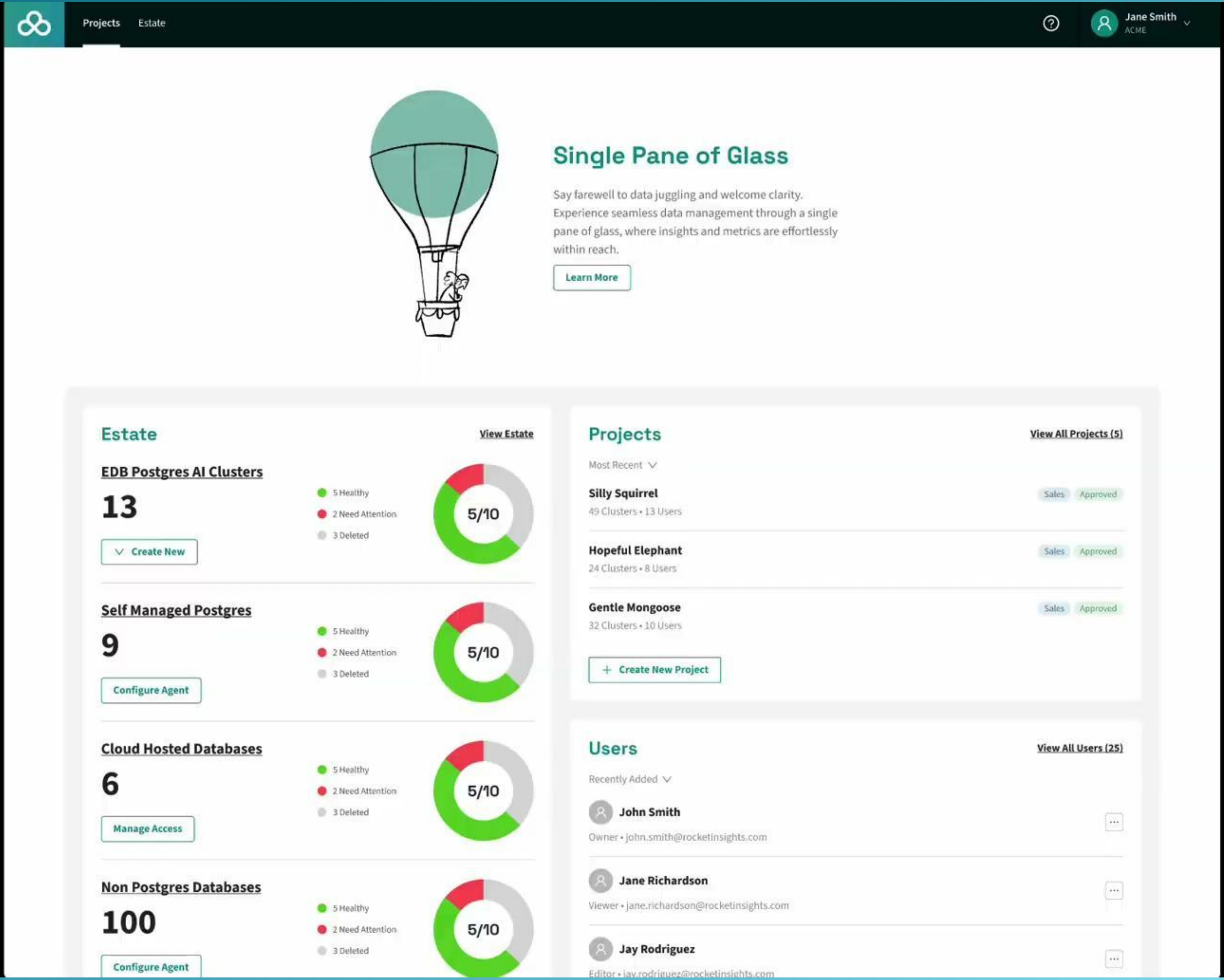
```

-- Where am I ?
select version()

-- List all schemas
SELECT schema_name
FROM information_schema.schemata;

-- Count on 600 Million Rows

select count(*) from tpch_sf_100_lfs.lineitem;
600 037 902

-- Count on 6 Billion Rows

select count(*) from tpch_sf_1000_lfs.lineitem;

5 999 989 709

5 999 989 709
-- Min, Max on Orderkey and count on 6 Billion Rows
```

	o_orderkey bigint	o_orderdate date	o_totalprice double precision	level bigint	cumulative_total double precision
1	600000000	1997-12-02	80002.6	150000000	22668168442423.906
2	599999975	1994-04-05	374027.19	149999999	22668168362421.305
3	599999974	1997-06-24	117452.54	149999998	22668167988394.113
4	599999973	1996-03-04	314459.45	149999997	22668167870941.574
5	599999972	1994-03-09	164253.17	149999996	22668167556482.125

Total rows: 10 of 10   Query complete 00:00:59.849   Ln 125, Col 9

Server disconnected.

Server disconnected.



Projects Estate

Jane Smith
ACME

Single Pane of Glass

Say farewell to data juggling and welcome clarity. Experience seamless data management through a single pane of glass, where insights and metrics are effortlessly within reach.

Learn More

Estate

View Estate

EDB Postgres AI Clusters

13

● 5 Healthy
● 2 Need Attention
● 3 Deleted

5/10

Create New

Self Managed Postgres

9

● 5 Healthy
● 2 Need Attention
● 3 Deleted

5/10

Configure Agent

Cloud Hosted Databases

6

● 5 Healthy
● 2 Need Attention
● 3 Deleted

5/10

Manage Access

Non Postgres Databases

100

● 5 Healthy
● 2 Need Attention
● 3 Deleted

5/10

Configure Agent

Projects

View All Projects (5)

Most Recent

Silly Squirrel
49 Clusters • 13 Users
Sales Approved

Hopeful Elephant
24 Clusters • 8 Users
Sales Approved

Gentle Mongoose
32 Clusters • 10 Users
Sales Approved

+ Create New Project

Users

View All Users (25)

Recently Added

John Smith
Owner • john.smith@rocketinsights.com

Jane Richardson
Viewer • jane.richardson@rocketinsights.com

Jay Rodriguez
Editor • jay.rodriguez@rocketinsights.com

Reduced total cost of ownership by decoupling compute and storage.

30X faster on *average* for analytical queries compared to Postgres

5X smaller on disk Lakehouse tables vs. Postgres tables and indexes

18X cost-effective Object storage vs. solid state drives (SSDs)

EDB AI

EDB POSTGRES AI PLATFORM

UNIFIED WORKLOAD MANAGEMENT

- TRANSACTIONAL
- ANALYTICAL
- ARTIFICIAL INTELLIGENCE

SINGLE PANE OF GLASS ADMINISTRATION

- HYBRID DATA ESTATE
- INTELLIGENT OBSERVABILITY
- ENTERPRISE SECURITY

HYBRID AND MULTI-CLOUD DEPLOYMENT

- PUBLIC CLOUD (MANAGED)
- PRIVATE CLOUD (SOFTWARE)
- ON PREMISES (APPLIANCE)

PLATFORM TOOLS AND SERVICES

- MIGRATION PORTAL
- CONTINUOUS HIGH AVAILABILITY
- BACKUP AND RECOVERY

EXTENSIBILITY

- CSP INTEGRATIONS
- DEVOPS TOOLING
- KUBERNETES TOOLING
- GENAI & LLM INTEGRATIONS
- LAKEHOUSE INTEGRATIONS

Delivered with world-class strategic partners:

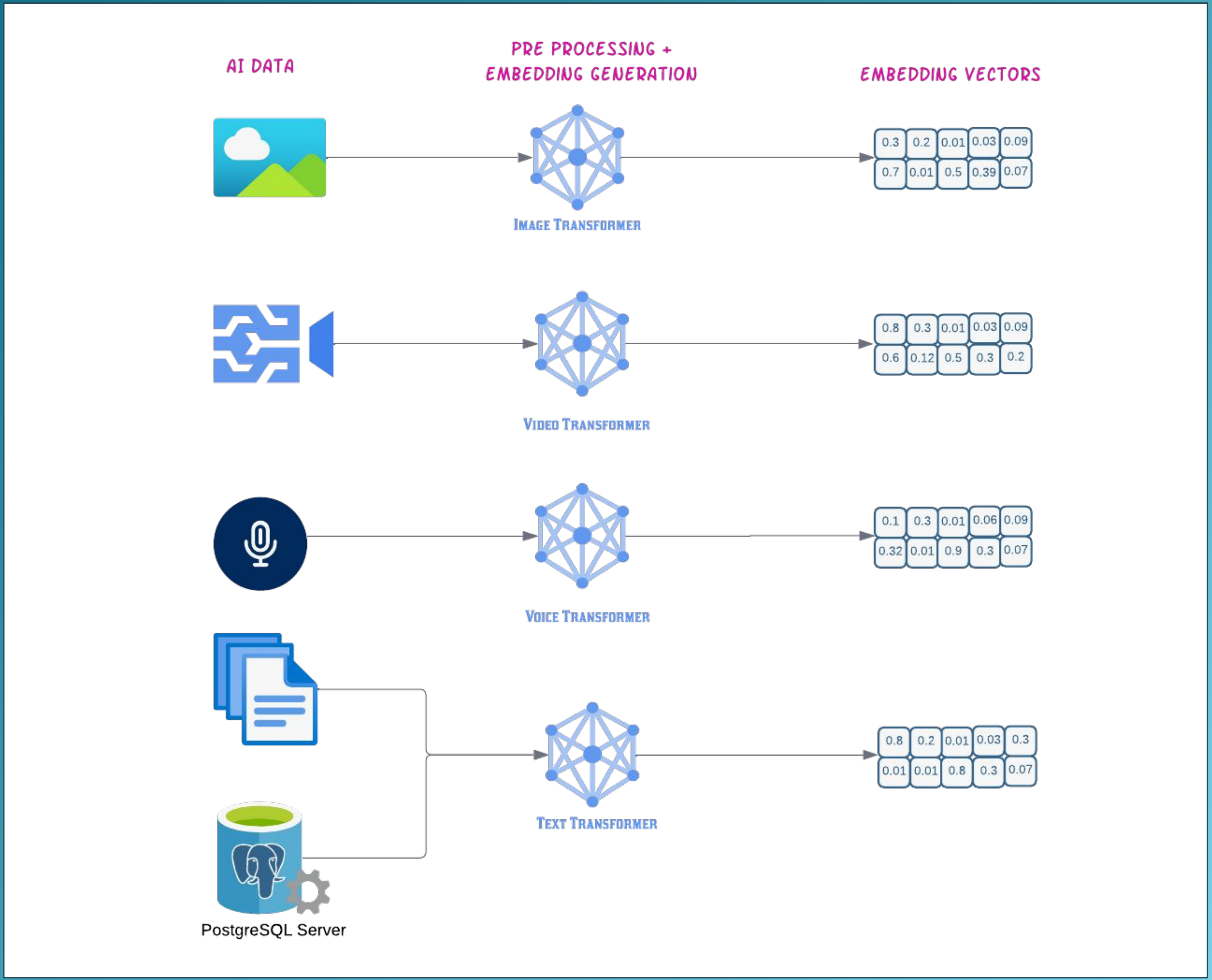
Red Hat

SUPERMICRO®

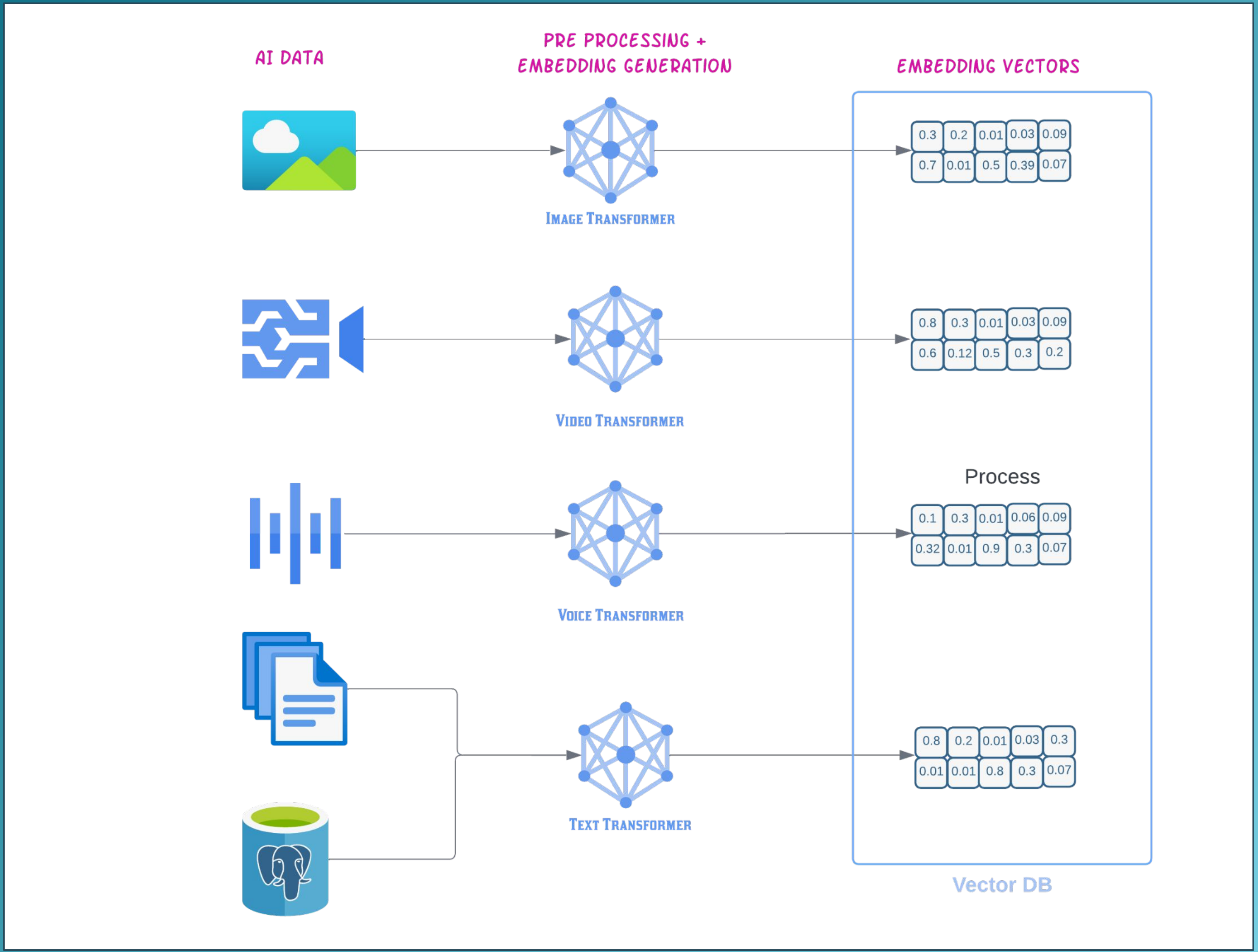
NUTANIX

What is PostgreSQL after pg vector extension for a DBA vs AI Scientist?

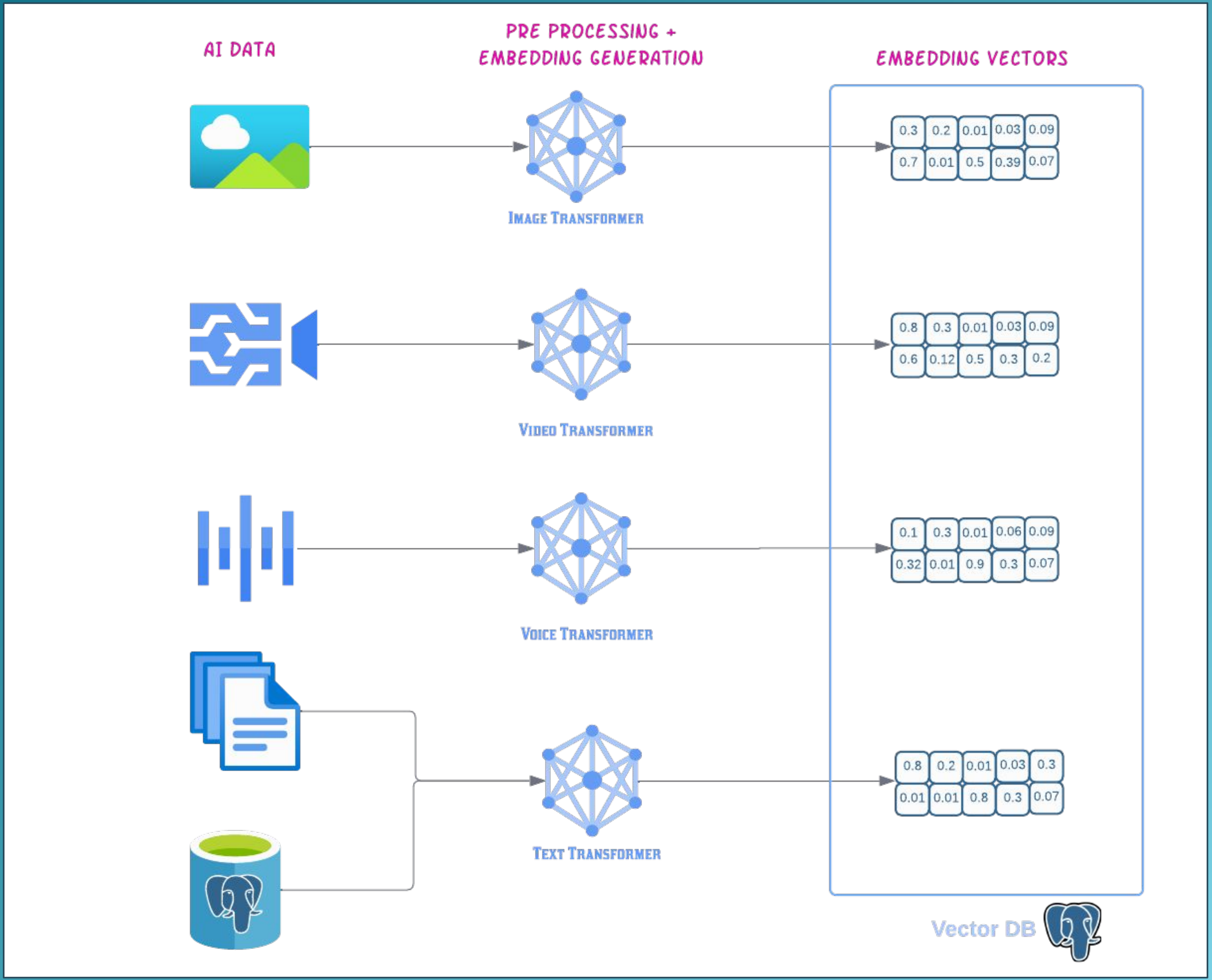
How are data & vectors connected to each other?



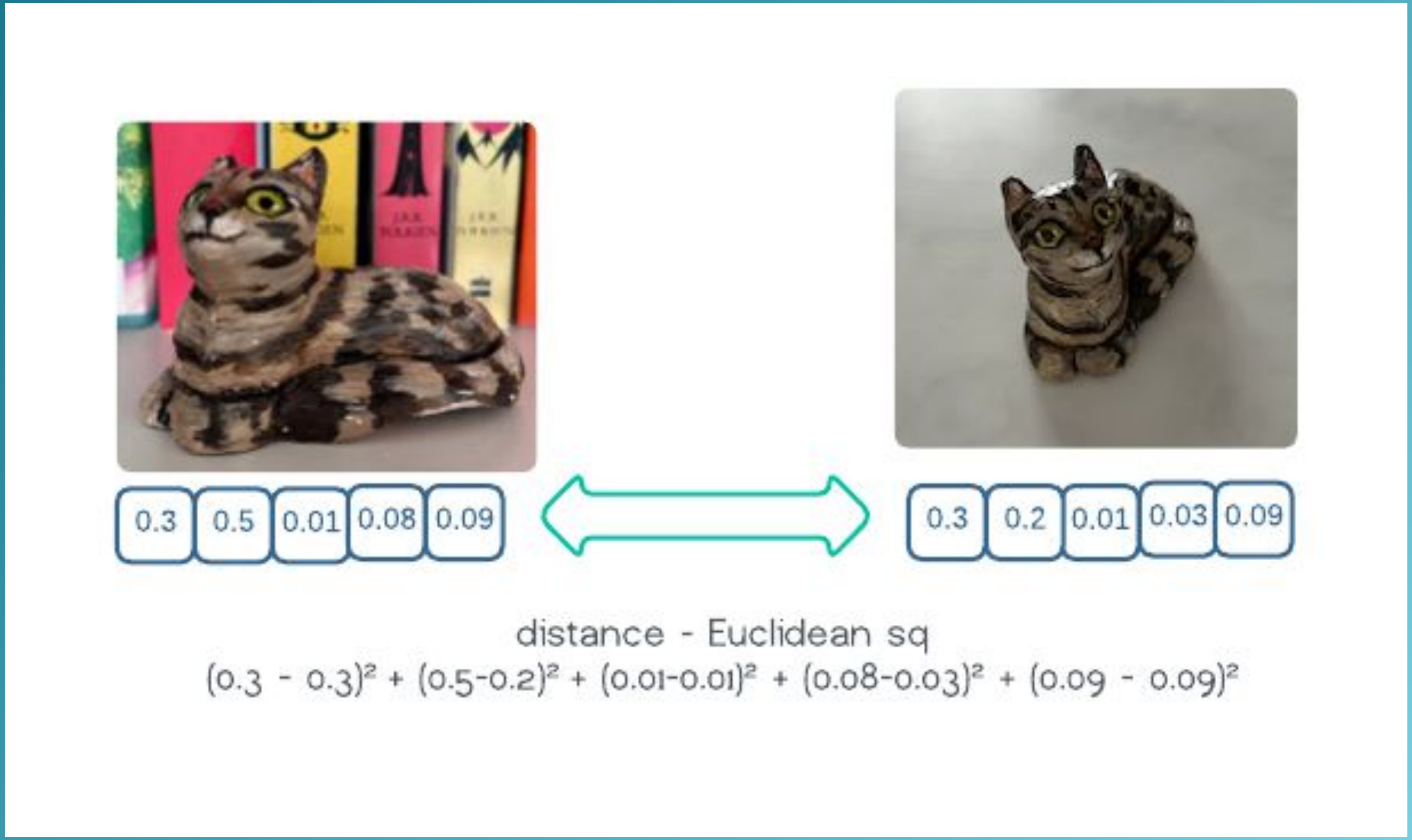
How are data & vectors connected to each other?



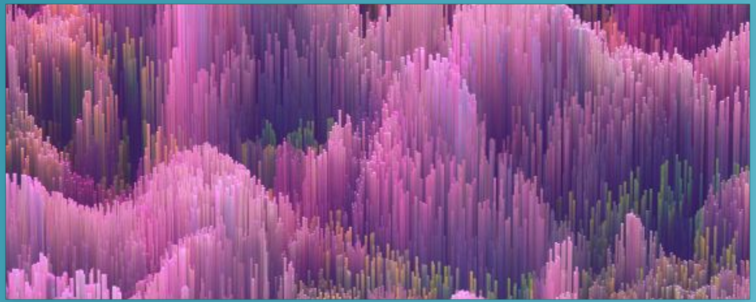
How are data & vectors connected to each other?



Similarity Search



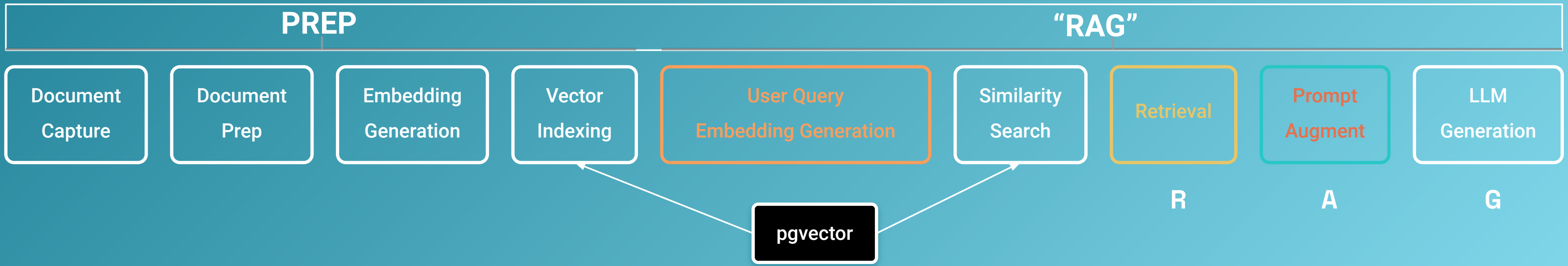
Pgvector



💸 **GenAI Applications** are the dominant focus of investment across the IT Industry today.

😵‍💫 **GenAI Applications** are *data centric*, *complex*, and the solutions are *nascent and largely piecemeal*.

"There is not a defacto enterprise grade standard"



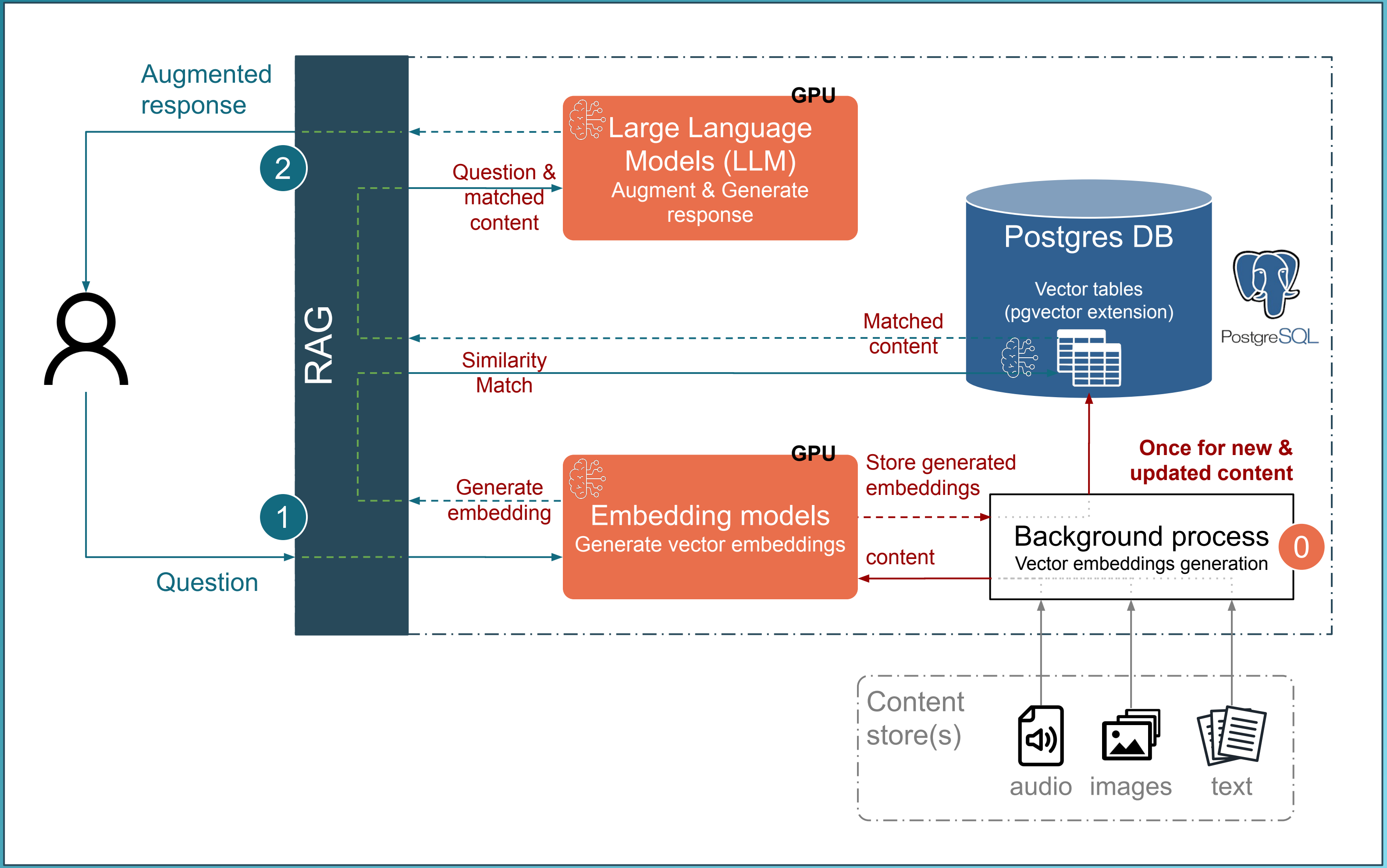
RAG (**R**etrieval **A**ugmented **G**eneration) Overview

Use Cases

- Conversational Assistance
- Chatbot
- Semantic search
- Visual and Semantic search for images
- Speech recognition and audio search (e.g. call center audio logs)
- Real-time and personalized search experience
- …

Content examples

- Documents (contracts, user guides, product/technical documentation, policies, CVs…)
- Emails
- Intranet documents (e.g. sharepoint, …)
- ….



Augmented response
2
RAG
1
Question
GPU
Large Language Models (LLM)
Augment & Generate response
Question & matched content
Postgres DB
Vector tables (pgvector extension)
PostgreSQL
Matched content
Similarity Match
GPU
Generate embedding
Embedding models
Generate vector embeddings
Store generated embeddings
Once for new & updated content
content
Background process
Vector embeddings generation
0
Content store(s)
audio
images
text

Models and Vector DB benefits

- Core for vector similarity search that provides fast and scalable (with improved accuracy) experience

RAG Interactions - detailed flow

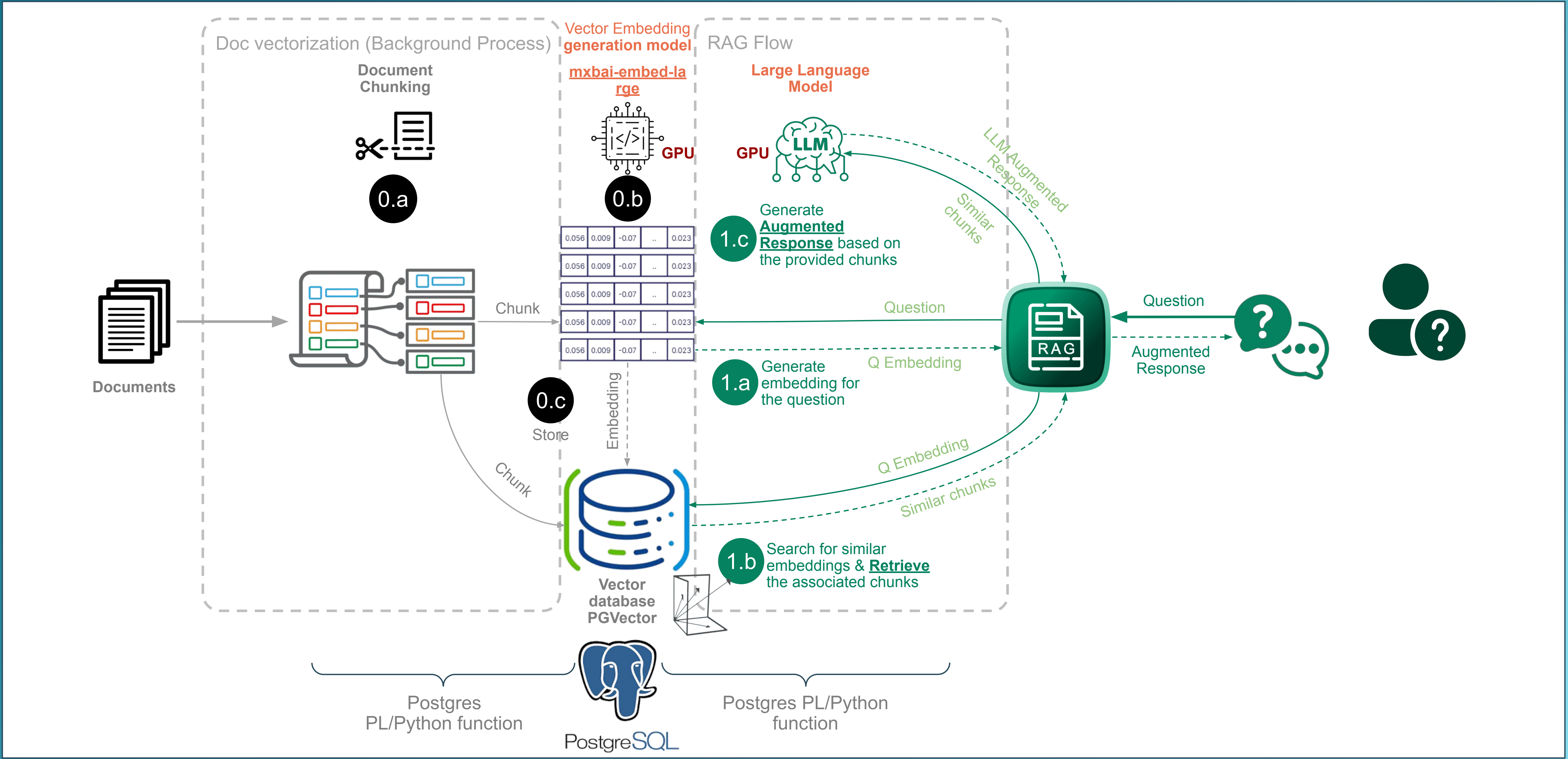


Image Search Interactions - detailed flow

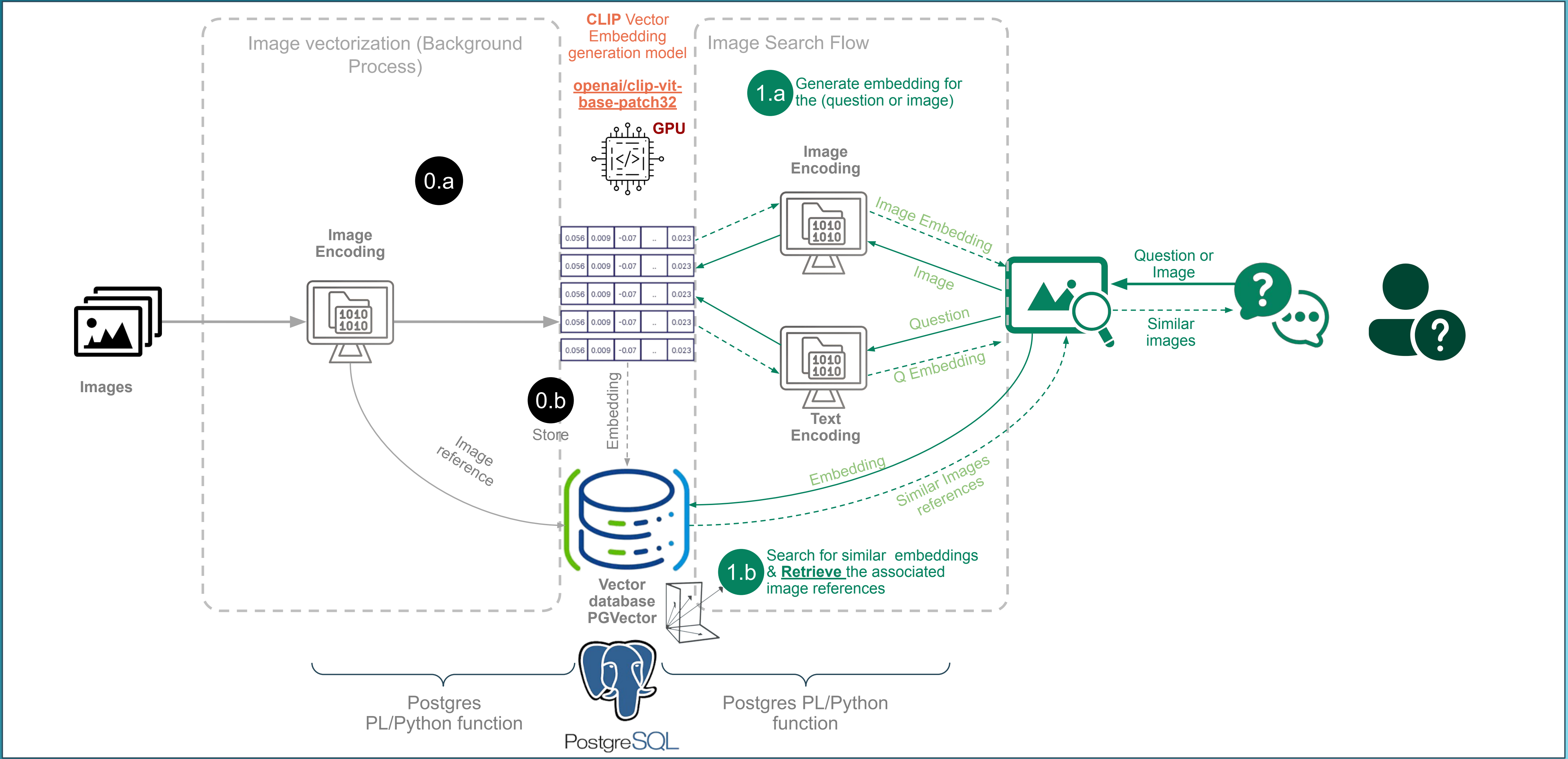
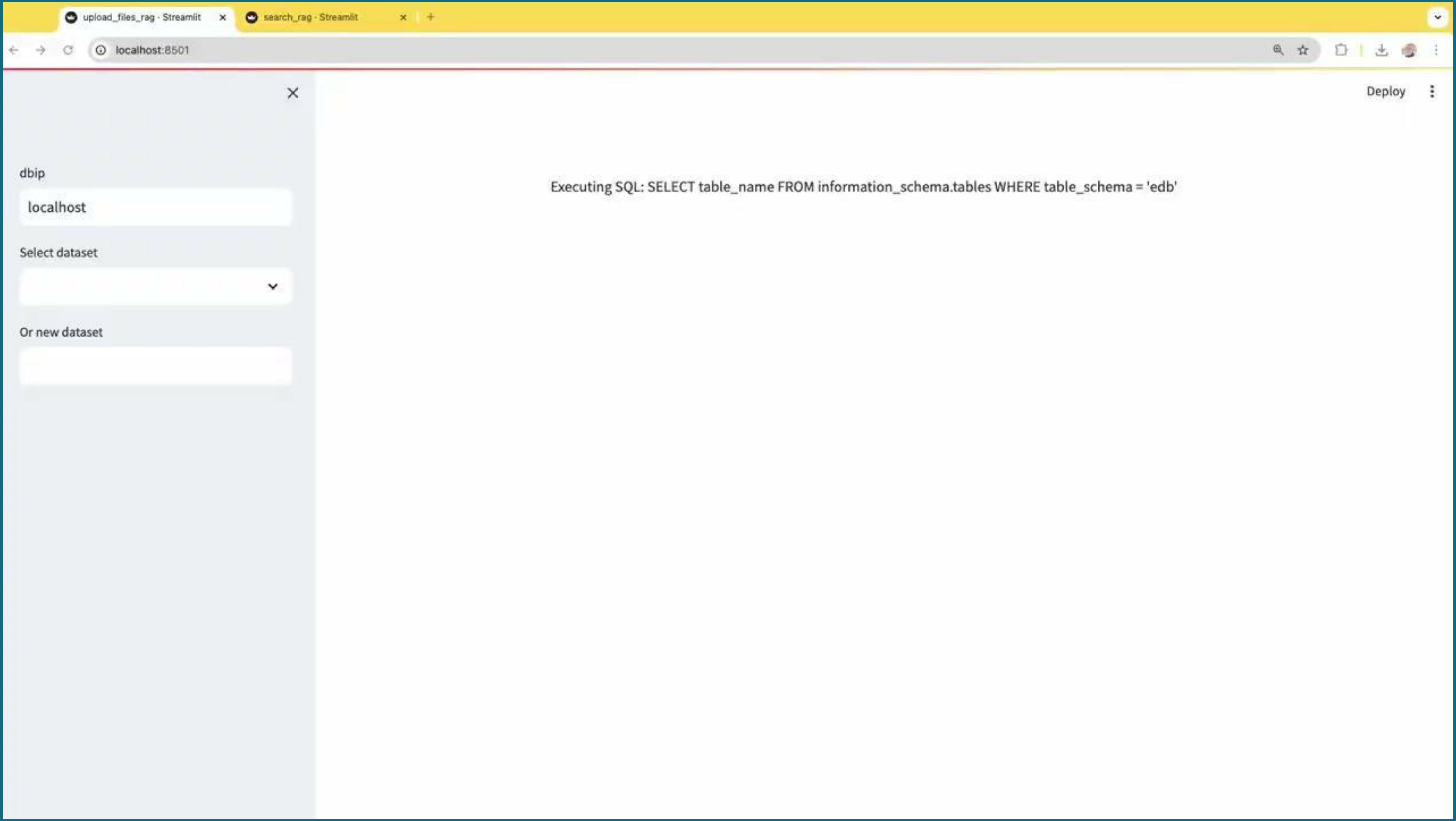


Image vectorization (Background Process)
CLIP Vector Embedding generation model
openai/clip-vit-base-patch32
GPU
Image Search Flow
1.a
Generate embedding for the (question or image)
0.a
Image Encoding
Images
Image Encoding
Image Embedding
Image
Question
Q Embedding
Text Encoding
Question or Image
Similar images
0.056 0.009 -0.07 .. 0.023
0.b
Store
Embedding
Image reference
Embedding
Similar Images references
Vector database PGVector
1.b
Search for similar embeddings & Retrieve the associated image references
Postgres PL/Python function
Postgres PL/Python function
PostgreSQL

Demo RAG



dbip

localhost

Select dataset

Or new dataset

```
Executing SQL: SELECT table_name FROM information_schema.tables WHERE table_schema = 'edb'
```

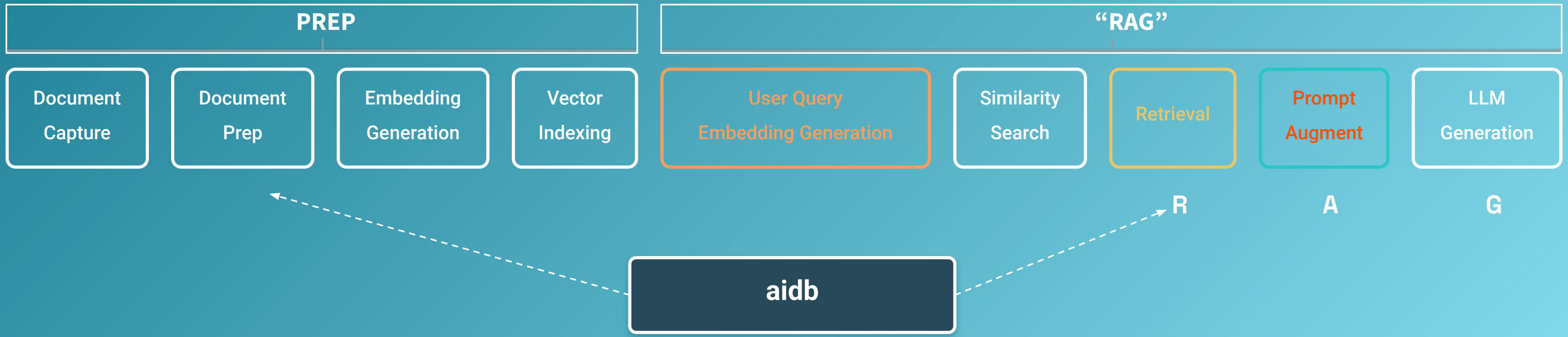
What about aidb?

aidb Extension

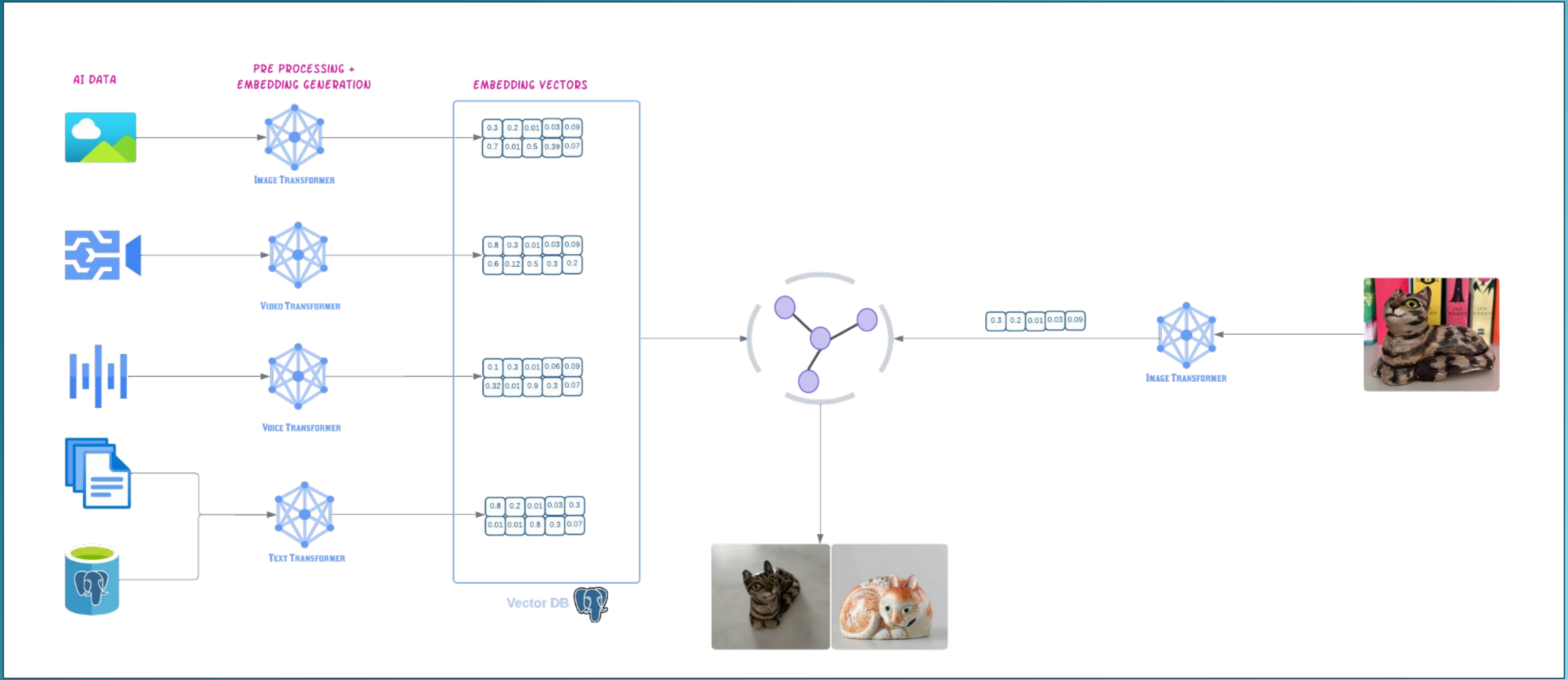
Our opportunity is necessitated by two dominant factors:

🥇 For businesses to run enterprise-grade, mission-critical GenAI apps, they need an *enterprise-grade data management platform*.

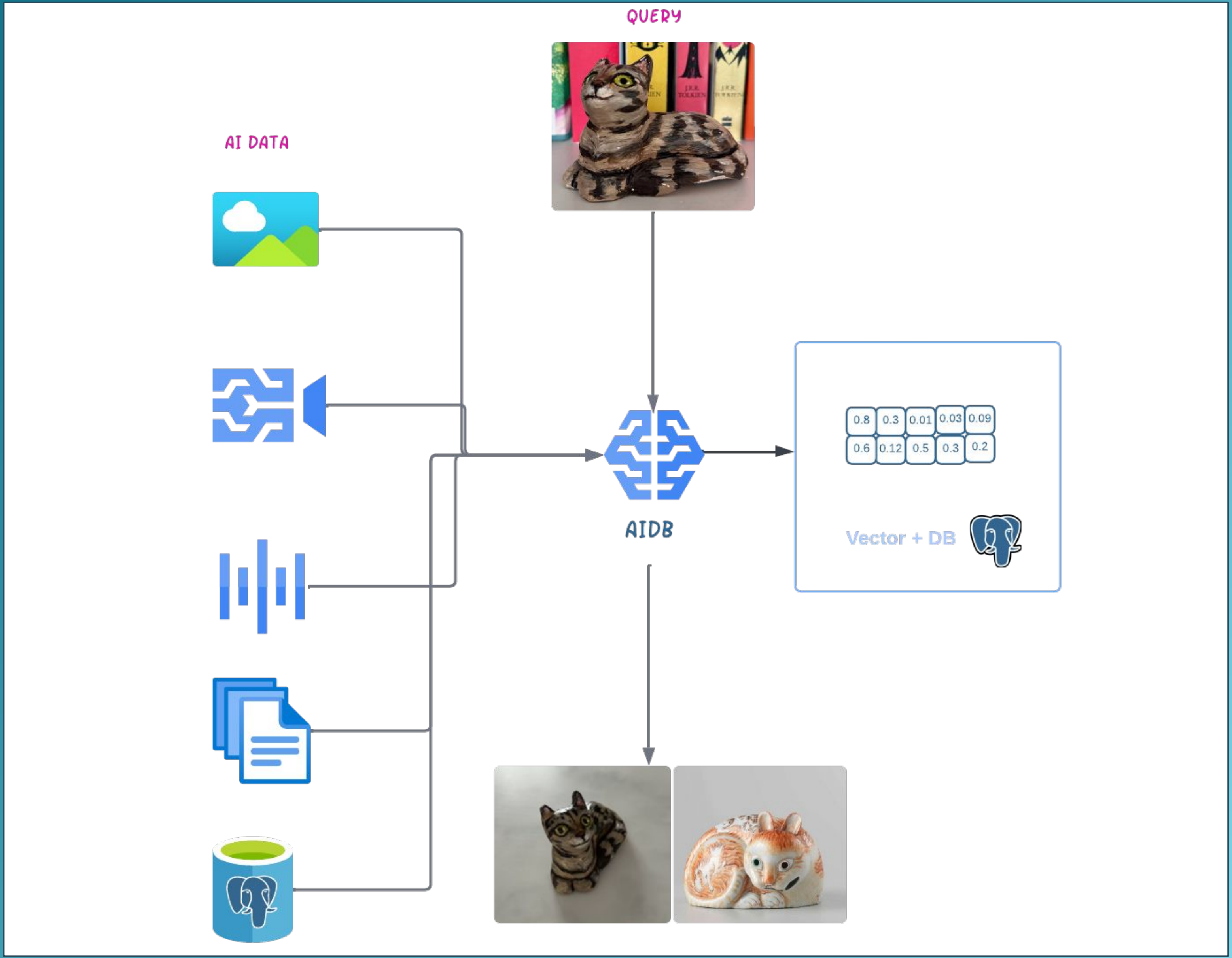
🥈 Our *partner strategy* is anchored around enabling the development of GenAI applications on Postgres *within those partners' ecosystems*.



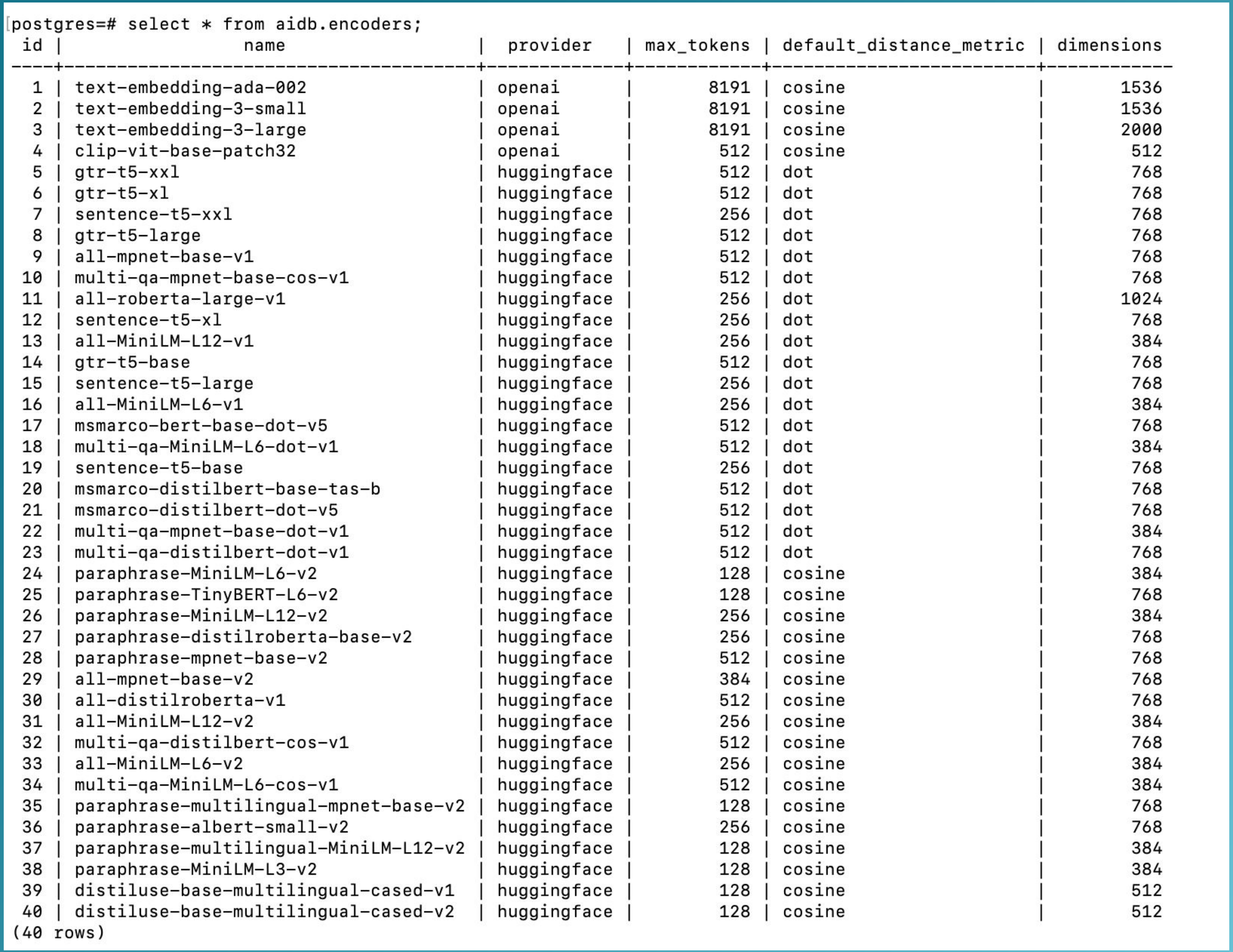
A recommendation engine with pgvector



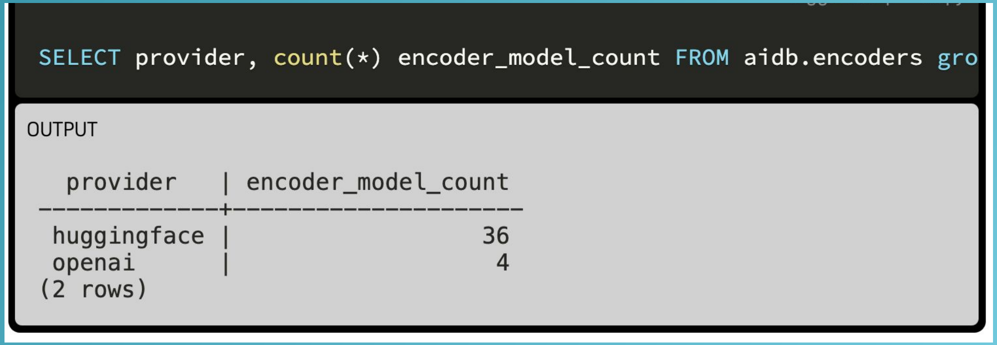
A recommendation engine with AIDB



AIDB - Encoders



```
postgres=# select * from aidb.encoders;
 id |                 name                   |  provider   | max_tokens | default_distance_metric | dimensions
----+----------------------------------------+-------------+------------+-------------------------+------------
  1 | text-embedding-ada-002                 | openai      |       8191 | cosine                  |       1536
  2 | text-embedding-3-small                 | openai      |       8191 | cosine                  |       1536
  3 | text-embedding-3-large                 | openai      |       8191 | cosine                  |       2000
  4 | clip-vit-base-patch32                  | openai      |        512 | cosine                  |        512
  5 | gtr-t5-xxl                             | huggingface |        512 | dot                     |        768
  6 | gtr-t5-xl                              | huggingface |        512 | dot                     |        768
  7 | sentence-t5-xxl                        | huggingface |        256 | dot                     |        768
  8 | gtr-t5-large                           | huggingface |        512 | dot                     |        768
  9 | all-mpnet-base-v1                      | huggingface |        512 | dot                     |        768
 10 | multi-qa-mpnet-base-cos-v1             | huggingface |        512 | dot                     |        768
 11 | all-roberta-large-v1                   | huggingface |        256 | dot                     |       1024
 12 | sentence-t5-xl                         | huggingface |        256 | dot                     |        768
 13 | all-MiniLM-L12-v1                      | huggingface |        256 | dot                     |        384
 14 | gtr-t5-base                            | huggingface |        512 | dot                     |        768
 15 | sentence-t5-large                      | huggingface |        256 | dot                     |        768
 16 | all-MiniLM-L6-v1                       | huggingface |        256 | dot                     |        384
 17 | msmarco-bert-base-dot-v5               | huggingface |        512 | dot                     |        768
 18 | multi-qa-MiniLM-L6-dot-v1              | huggingface |        512 | dot                     |        384
 19 | sentence-t5-base                       | huggingface |        256 | dot                     |        768
 20 | msmarco-distilbert-base-tas-b          | huggingface |        512 | dot                     |        768
 21 | msmarco-distilbert-dot-v5              | huggingface |        512 | dot                     |        768
 22 | multi-qa-mpnet-base-dot-v1             | huggingface |        512 | dot                     |        384
 23 | multi-qa-distilbert-dot-v1             | huggingface |        512 | dot                     |        768
 24 | paraphrase-MiniLM-L6-v2                | huggingface |        128 | cosine                  |        384
 25 | paraphrase-TinyBERT-L6-v2              | huggingface |        128 | cosine                  |        768
 26 | paraphrase-MiniLM-L12-v2               | huggingface |        256 | cosine                  |        384
 27 | paraphrase-distilroberta-base-v2       | huggingface |        256 | cosine                  |        768
 28 | paraphrase-mpnet-base-v2               | huggingface |        512 | cosine                  |        768
 29 | all-mpnet-base-v2                      | huggingface |        384 | cosine                  |        768
 30 | all-distilroberta-v1                   | huggingface |        512 | cosine                  |        768
 31 | all-MiniLM-L12-v2                      | huggingface |        256 | cosine                  |        384
 32 | multi-qa-distilbert-cos-v1             | huggingface |        512 | cosine                  |        768
 33 | all-MiniLM-L6-v2                       | huggingface |        256 | cosine                  |        384
 34 | multi-qa-MiniLM-L6-cos-v1              | huggingface |        512 | cosine                  |        384
 35 | paraphrase-multilingual-mpnet-base-v2  | huggingface |        128 | cosine                  |        768
 36 | paraphrase-albert-small-v2             | huggingface |        256 | cosine                  |        768
 37 | paraphrase-multilingual-MiniLM-L12-v2  | huggingface |        128 | cosine                  |        384
 38 | paraphrase-MiniLM-L3-v2                | huggingface |        128 | cosine                  |        384
 39 | distiluse-base-multilingual-cased-v1   | huggingface |        128 | cosine                  |        512
 40 | distiluse-base-multilingual-cased-v2   | huggingface |        128 | cosine                  |        512
(40 rows)
```

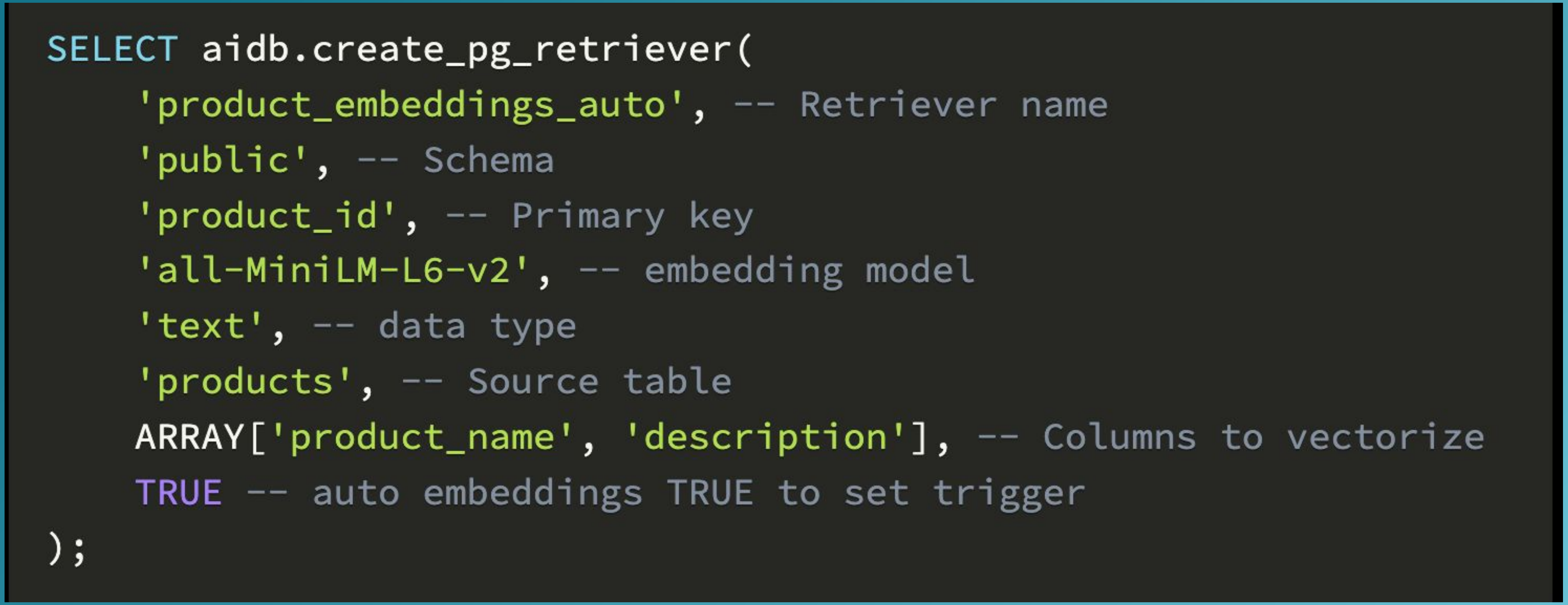


```
SELECT provider, count(*) encoder_model_count FROM aidb.encoders gro
```

OUTPUT

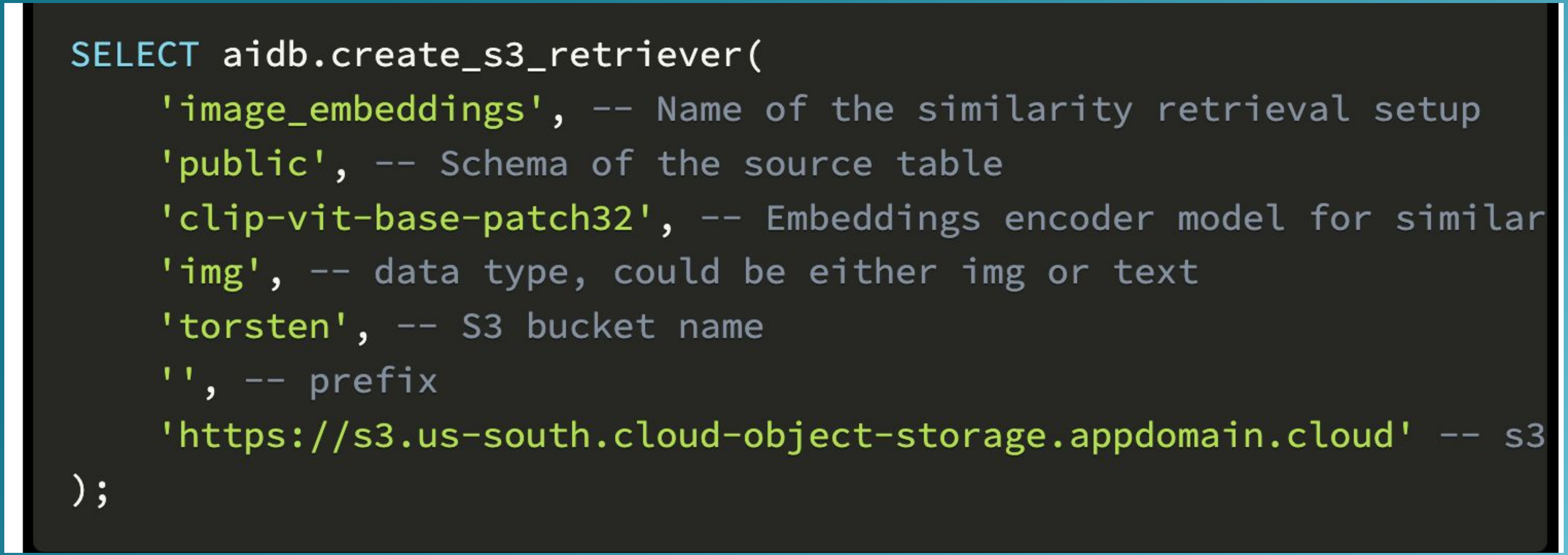
```
  provider   | encoder_model_count
-------------+---------------------
 huggingface |                  36
 openai      |                   4
(2 rows)
```

AIDB - Create Retriever - Postgres as a Source

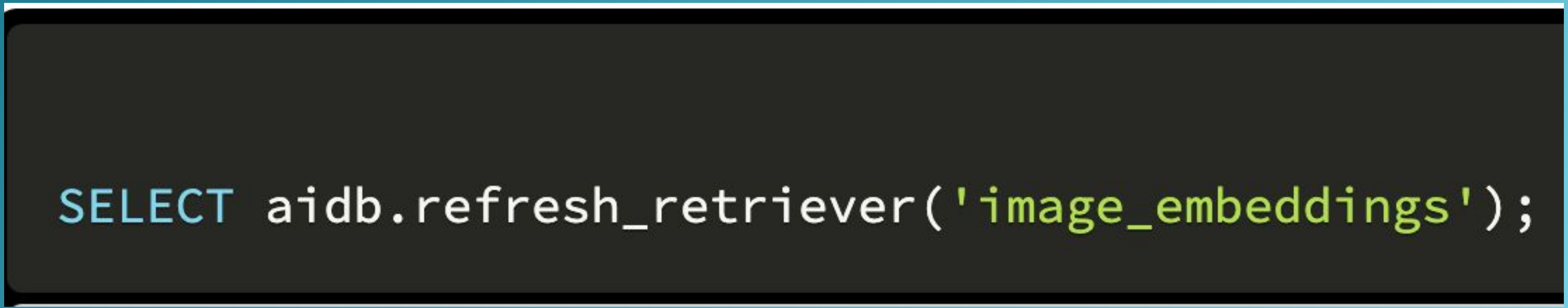


```sql
SELECT aidb.create_pg_retriever(
    'product_embeddings_auto', -- Retriever name
    'public', -- Schema
    'product_id', -- Primary key
    'all-MiniLM-L6-v2', -- embedding model
    'text', -- data type
    'products', -- Source table
    ARRAY['product_name', 'description'], -- Columns to vectorize
    TRUE -- auto embeddings TRUE to set trigger
);
```

AIDB - Create Retriever - S3 as a Source

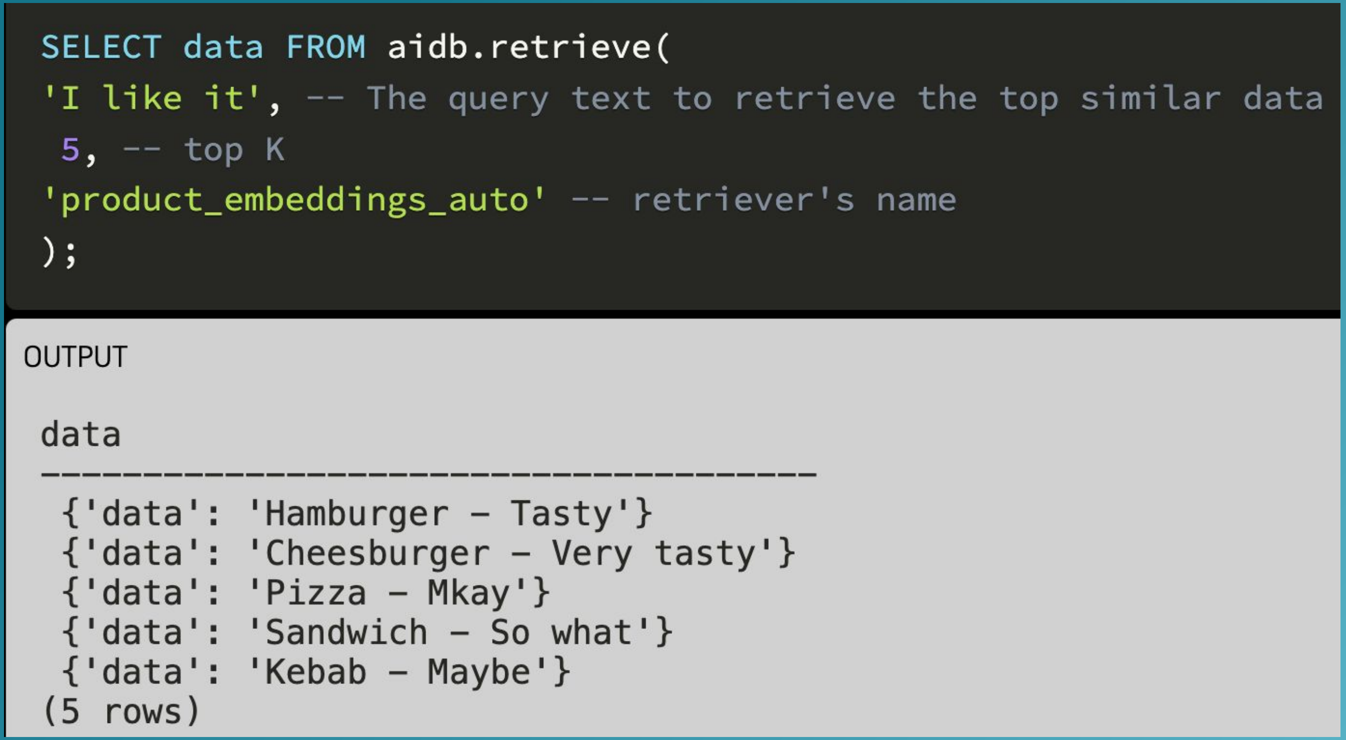


```
SELECT aidb.create_s3_retriever(
    'image_embeddings', -- Name of the similarity retrieval setup
    'public', -- Schema of the source table
    'clip-vit-base-patch32', -- Embeddings encoder model for similar
    'img', -- data type, could be either img or text
    'torsten', -- S3 bucket name
    '', -- prefix
    'https://s3.us-south.cloud-object-storage.appdomain.cloud' -- s3
);
```



```
SELECT aidb.refresh_retriever('image_embeddings');
```

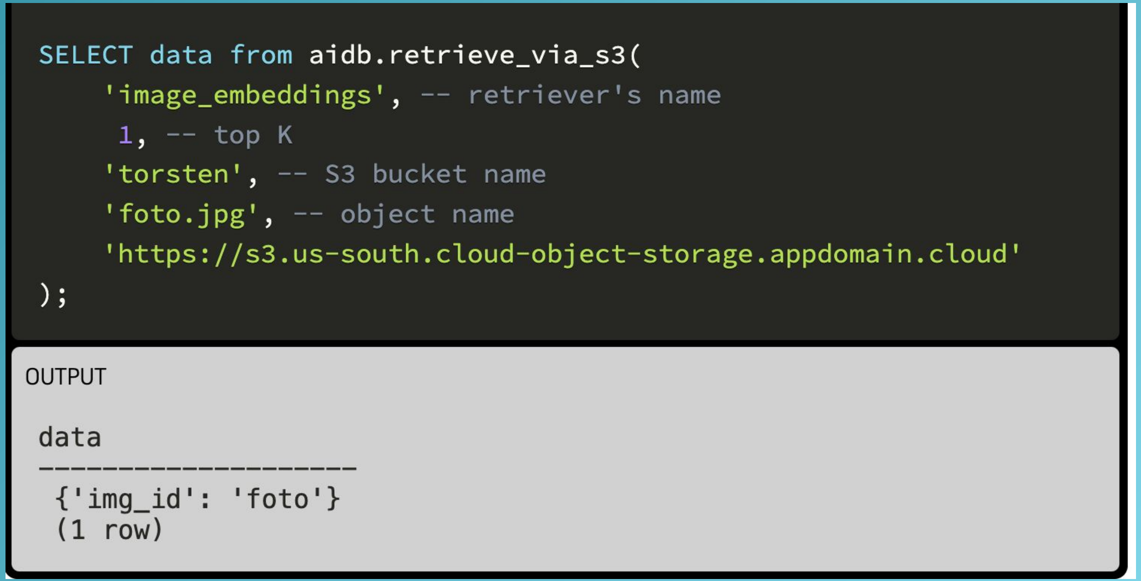
AIDB - Retrieve Data from Retriever



```sql
SELECT data FROM aidb.retrieve(
'I like it', -- The query text to retrieve the top similar data
 5, -- top K
'product_embeddings_auto' -- retriever's name
);
```

OUTPUT

```
data
---------------------------------------
 {'data': 'Hamburger - Tasty'}
 {'data': 'Cheesburger - Very tasty'}
 {'data': 'Pizza - Mkay'}
 {'data': 'Sandwich - So what'}
 {'data': 'Kebab - Maybe'}
(5 rows)
```



```sql
SELECT data from aidb.retrieve_via_s3(
    'image_embeddings', -- retriever's name
     1, -- top K
    'torsten', -- S3 bucket name
    'foto.jpg', -- object name
    'https://s3.us-south.cloud-object-storage.appdomain.cloud'
);
```

OUTPUT

```
data
----------------
 {'img_id': 'foto'}
 (1 row)
```

What's Next: Release Horizons

Where are we headed?

	Transactional Services	Analytical Services	AI Services	Platform
2024	• Distributed Postgres for k8s • Oracle Migration Copilot • Migration Service (preview) • Query Optimization • Enhance DR	• Postgres Lakehouse Clusters (On Prem) • Postgres Lakehouse Clusters (Cloud) • Tiered Analytics	• AIDB (Preview) • Automated Vectorization • Vector Storage	• Turnkey Physical Appliance (Beta) • Single Pane of Glass Management for Hybrid Deployments • Observability (Phase 1)
H1 2025	• Migration Services	• Hybrid Transactional and Analytical Processing (HTAP)	• AIDB (GA)	• Turnkey Physical Appliance (GA) • Observability (Phase 2)

EDB AI: Innovative Platform 100% Based on Postgres

New Apps and Workloads: Analytics, HTAP, RAG, Recommender

Deploy Anywhere and leverage All Infrastructure: True Hybrid Cloud Solution. Leverage GPUs, Object Storage

Support Any Data Format: Efficient Data Processing & Open Data Formats including Columnar & Compression