DATAlysis

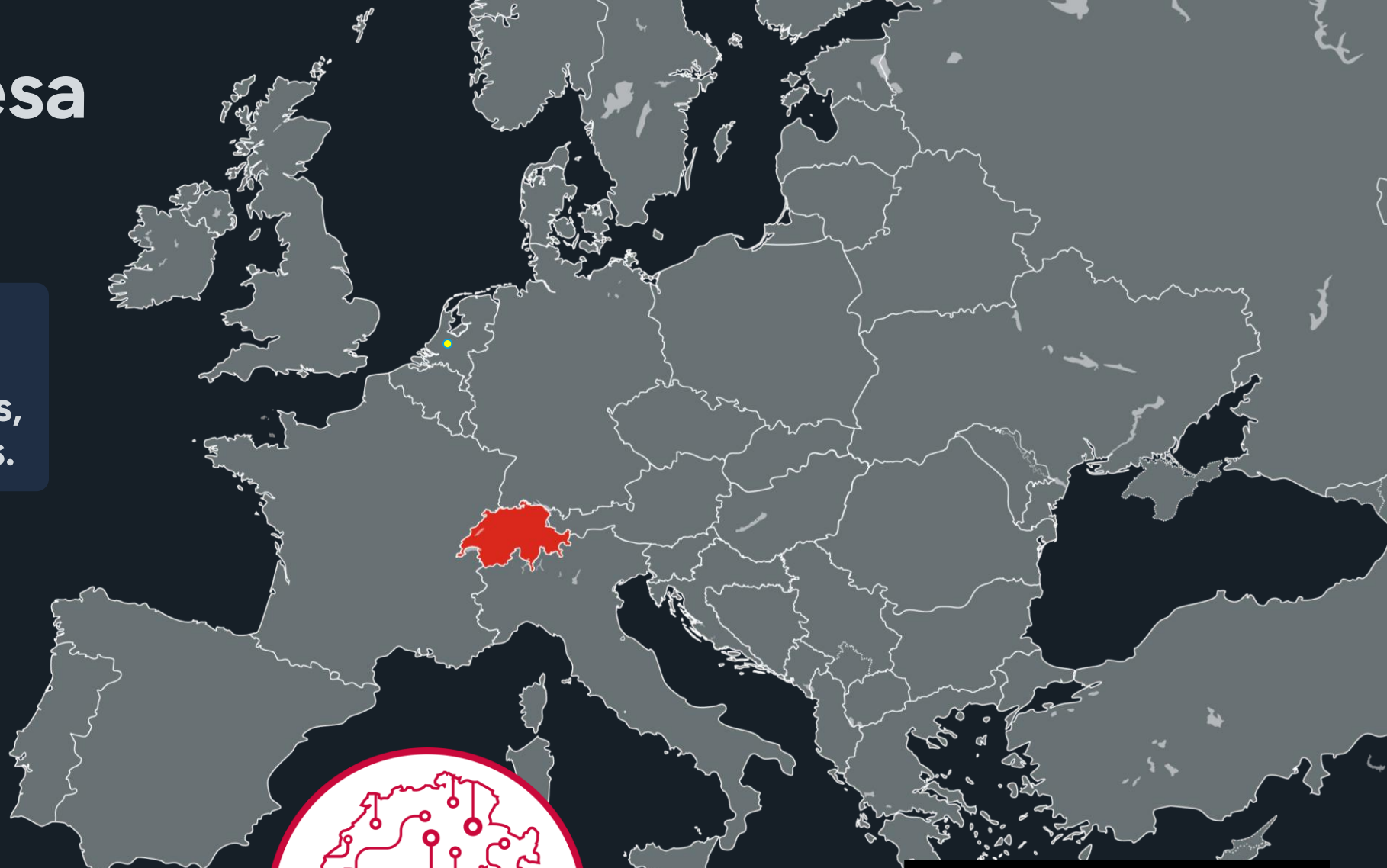
DB World 2026 · November 13, 2026

Can the modern Frankenstein pass the Turing test?

Gianni Ceresa · @G_Ceresa · @gianniceresa.bsky.social · datalysis.ch · info@datalysis.ch

Gianni Ceresa

Working with data of any shape and colour, and Business Analytics, for more than 15 years.



Oracle ACE Director

SWISS OUC

SYMOUC

DATAlysis

Oracle ACE

The Oracle ACE Program

1200+ technical experts helping peers globally



- The Oracle ACE Program recognizes and rewards community members for their technical and community contributions to the Oracle community
- 4 membership levels: Director, Pro, Associate, and Apprentice
- Nominate yourself or a colleague at ace.oracle.com/nominate
- Learn more at ace.oracle.com

aceprogram_ww@oracle.com
@oracleace.bsky.social
@oracleace
Oracle ACE Community

DISCLAIMER



Only 45 minutes?

Requires choices on what to ignore and cut corners.

It’s a “very old” topic in the AI-timescale.
Everything keeps changing, innovation doesn’t stop.
RAG is mostly automated nowadays.

(The first version of this session was prepared in July 2024.)

DATAlysis

DB World 2026 · November 13, 2026

Can the modern Frankenstein pass the Turing test?

Building your own custom Oracle Database ChatGPT-like solution

Gianni Ceresa · @G_Ceresa · @gianniceresa.bsky.social · datalysis.ch · info@datalysis.ch

LLMs

It always start with a LLM...

ChatGPT

ChatGPT is great for various tasks, you can ask it anything and it will (almost) never complain and will give you an answer.

How good is the answer?

Depends: how good was the question?

ChatGPT

Just like you and me can make up a story, so can ChatGPT with hallucinations:

- False or misleading information presented as facts
- Sources cited but incorrect information extracted

Hallucination rate heavily depends on the complexity of the task, and whether the model is given a reference document.

Help your LLM be right

LLMs supported by facts

LLMs based on facts your provide

How to “ground” LLMs to facts?

Where to start to build an Oracle Database ChatGPT-like solution?

The answer is RAG: retrieval-augmented generation

RAG

RAG is a technique for enhancing the accuracy and reliability of generative AI models with facts fetched from external sources.

Retrieval-augmented generation

RAG: one acronym, an almost infinite number of ways to achieve it...

- From the simplest retrieval fed to an LLM
- To more advanced processes with decisions made before feeding fact to an LLM, and evaluation of the quality of facts provided, and the quality of the answer generated (a sort of agentic RAG)

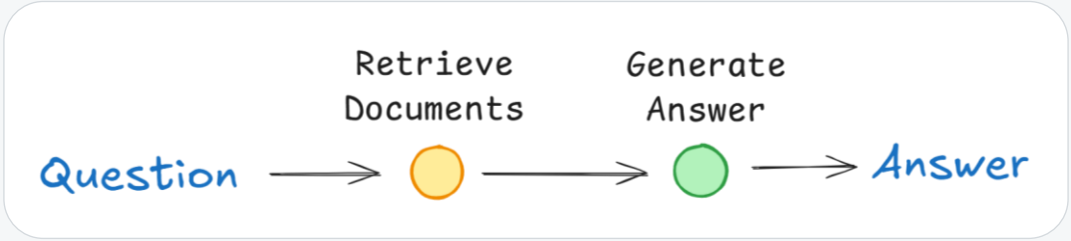
RAG

Various shapes of RAG

The very simple one

The bare minimum for RAG:

- **Retrieve documents**
  - Query a source, in Oracle AI Database 26ai using `VECTOR_DISTANCE`, to get the closest pieces of content by vector distance to the question asked.
    The question is embedded into a vector “on the fly”.
- **Generate answer**
  - Ask an LLM to generate an answer to the question using the provided documents.



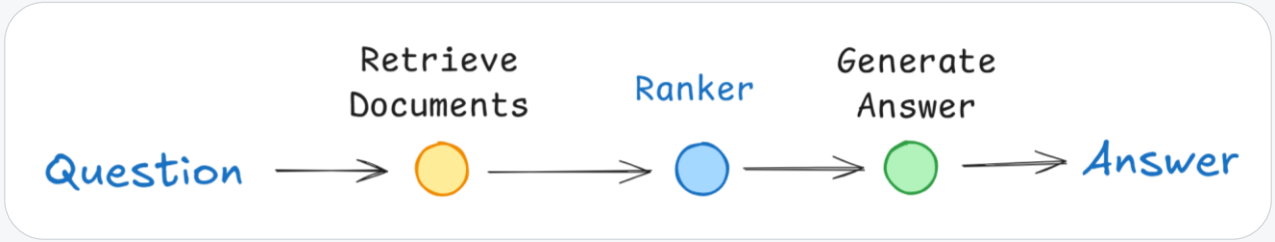
SQL

LLM

The improved simple one

Adding an extra step to the bare minimum to improve the result:

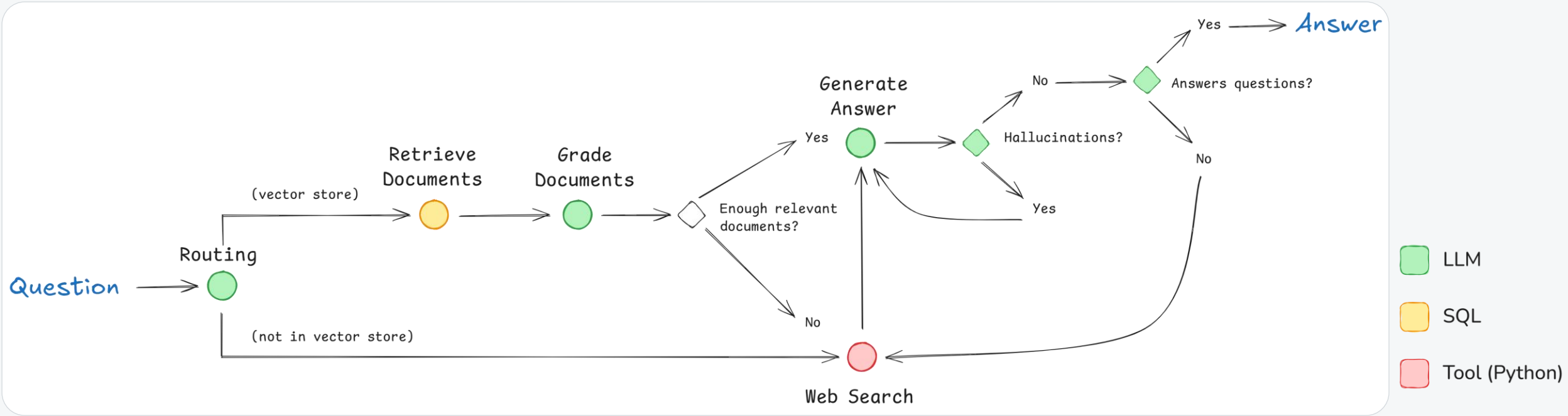
- **Ranker**
  - Retrieving documents is good, filtering the documents by the most relevant documents is better (based on the content, not the vector).
  - Retrieve more documents, rank them by relevance and take a shorter subset of the most relevant ones.
    For example, retrieve 10 documents and take the TOP5 from the Ranker to pass to the LLM.
  - Allows one more chance to remove “false positive” results of the retrieval step.



The less simple but hopefully better one

Your imagination is the limit:

- **A LLM can do many things when instructed correctly**
  - Can decide if the vector store is able to answer the question.
  - Can decide if documents are relevant for the question.
  - Can decide if the answer sounds plausible or not.
  - Can decide if the answer is good enough.



Recipe

Back to the Oracle Database ChatGPT-like solution

How to “cook” a ChatGPT-like chatbot

Ingredients:

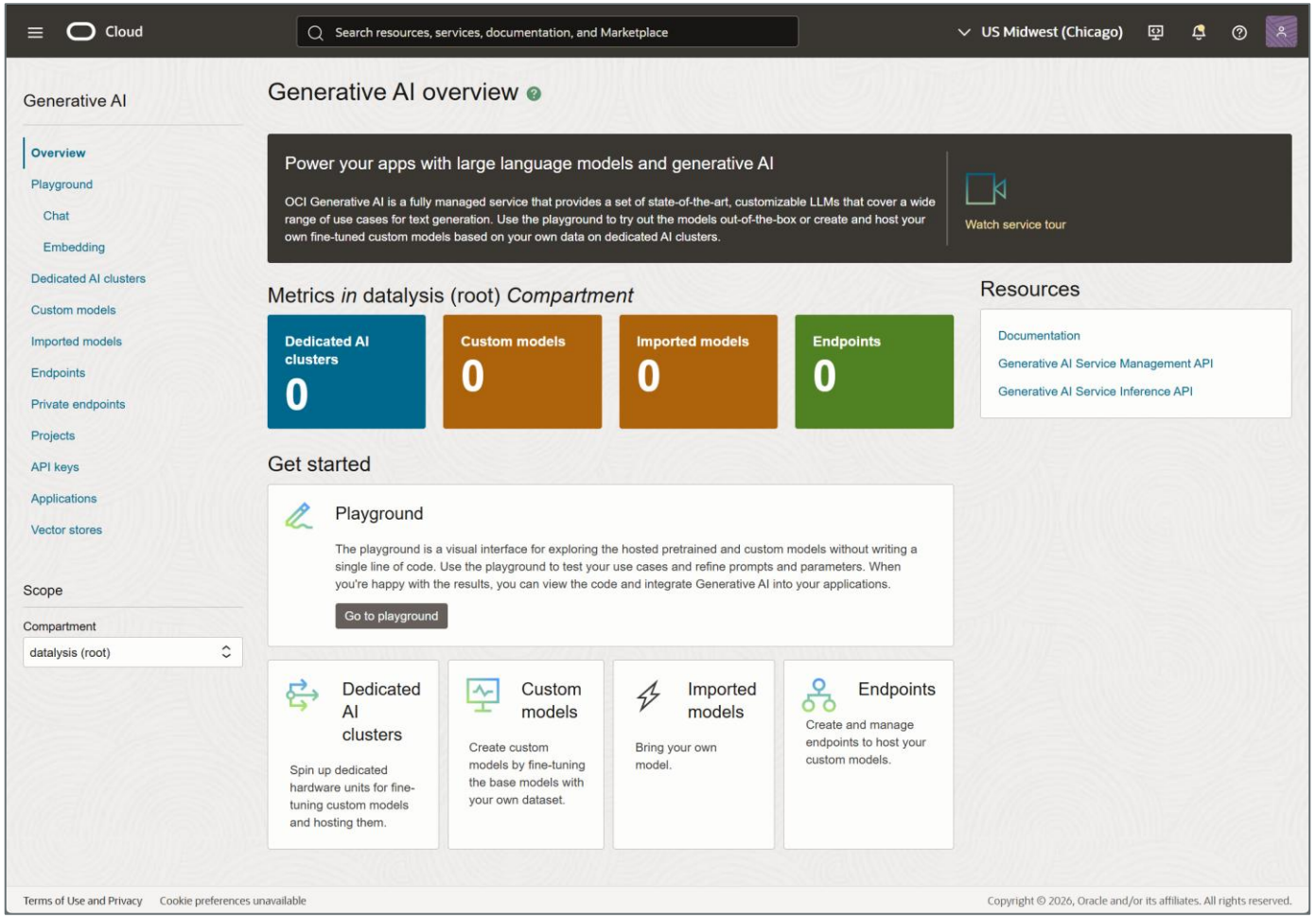
- **Data, documents, some source content**
- **An Oracle AI Database 26ai**
- **An embedding model**
  - In the database
  - A webservice called by the database
    - OCI GenAI, Cohere, GoogleAI, HuggingFace, OpenAI etc.
  - A solution outside the database
    - Python etc.
    - Oracle Private AI Service Container
- **An LLM service**
  - A service called by the database
    - OCI GenAI, Cohere, GoogleAI, HuggingFace, OpenAI etc.
  - Some solution outside the database
    - Python etc.
    - Ollama etc.
- **Some code glueing together the various pieces**

An LLM is not available inside the database, and should not be!

OCI GenAI

A quick look at Oracle OCI Generative AI services

OCI Generative AI services



The UI keeps changing every few months, the services stays mostly the same.

Many services in one place

One product, multiple services:

- On-demand, pre-trained, models for embedding and chat LLMs
- Dedicated AI clusters ($$$)
- Custom models: starts with a pre-trained model, fine-tune it with your own dataset and knowledge, store and use it

> OCI Generative AI is now part of the general OCI Enterprise AI service.

Many services in one place

OCI Generative AI is NOT available in all regions!

The list of available models depends on the region

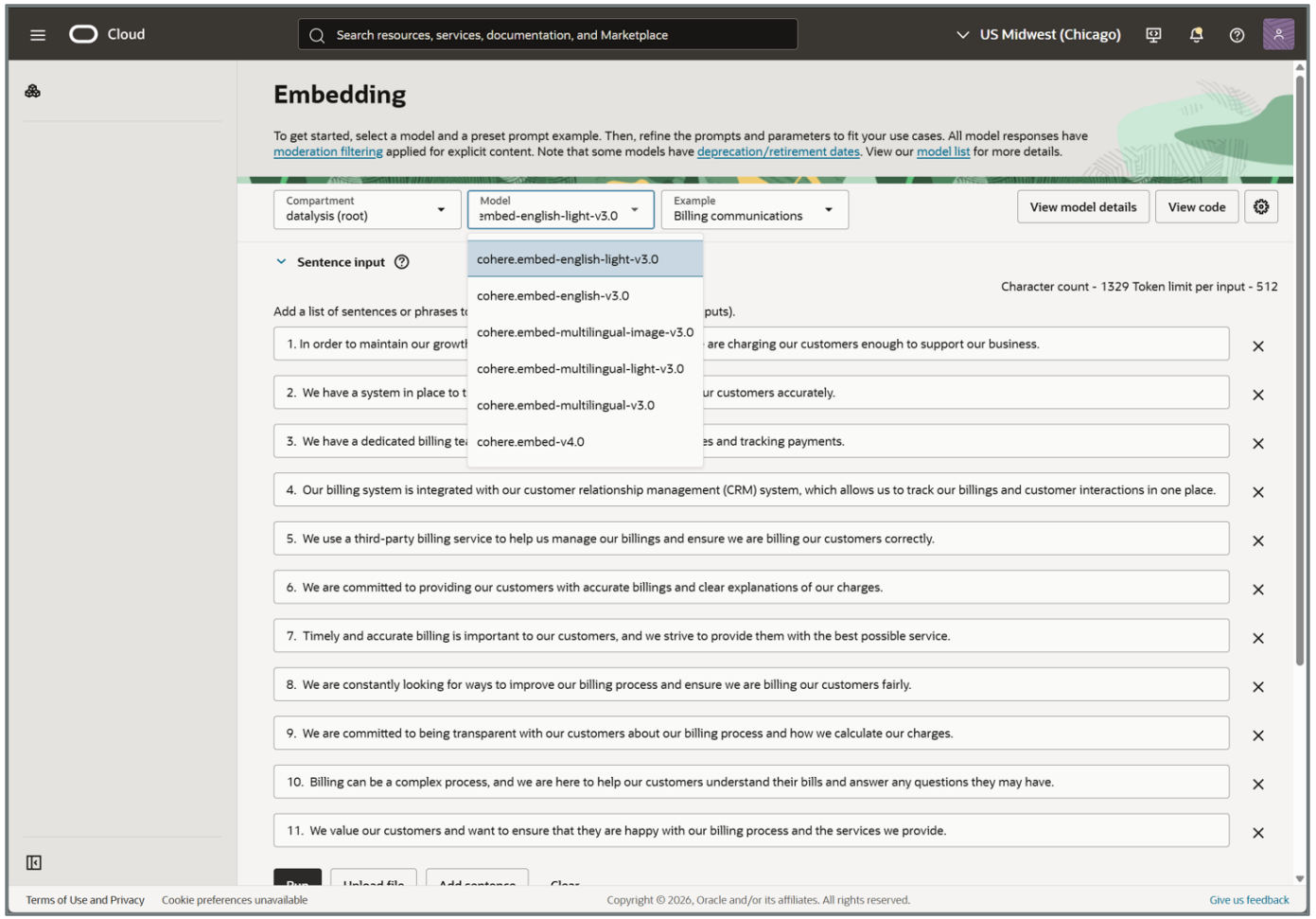
- Check the documentation to know what is available and where: https://docs.oracle.com/en-us/iaas/Content/generative-ai/model-endpoint-regions.htm

With Smart Model Router you can redirect requests to an available model in a different region.

Cheap and easily accessible for small tasks.

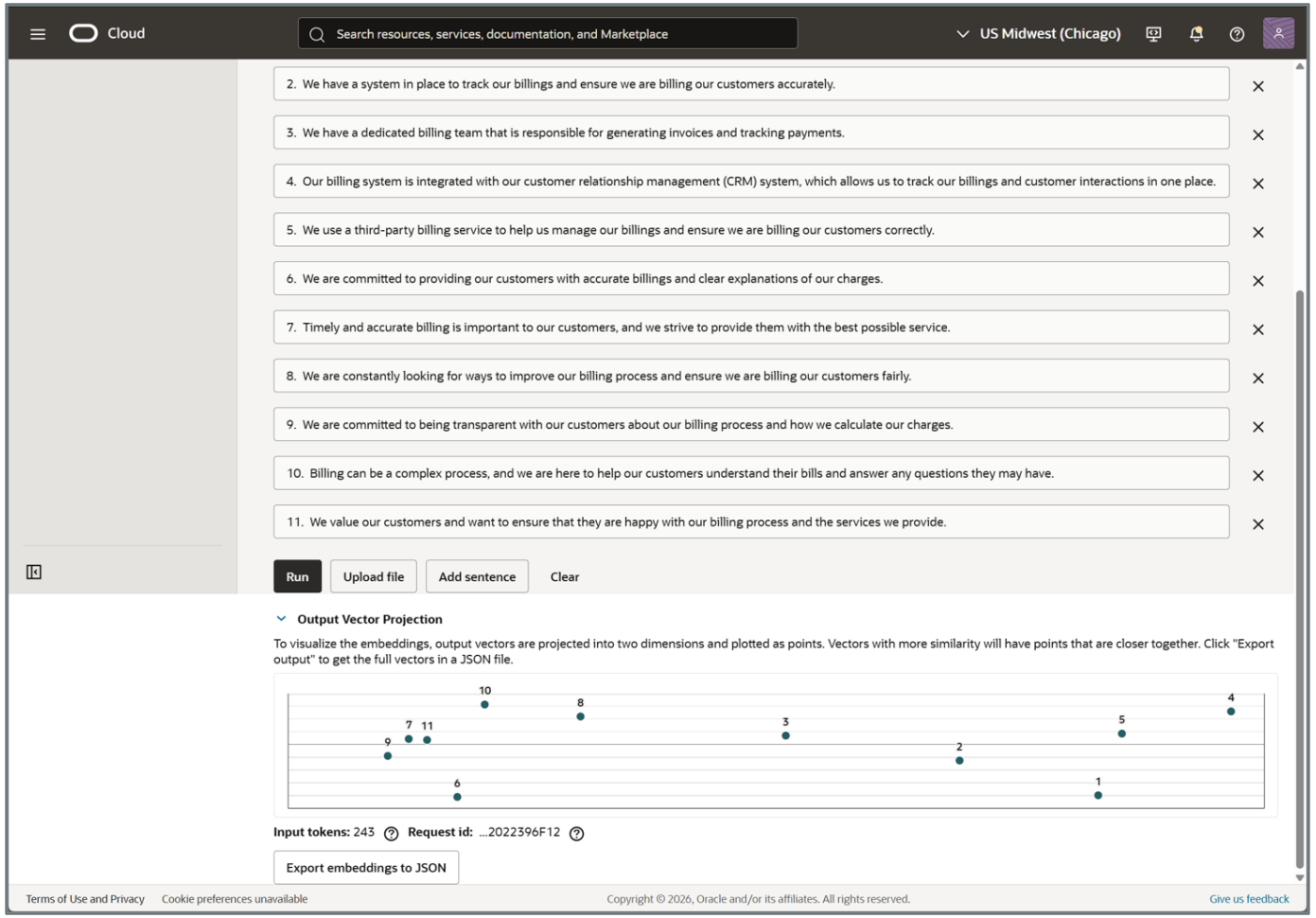
Calculate your potential costs for large activities.

Embedding



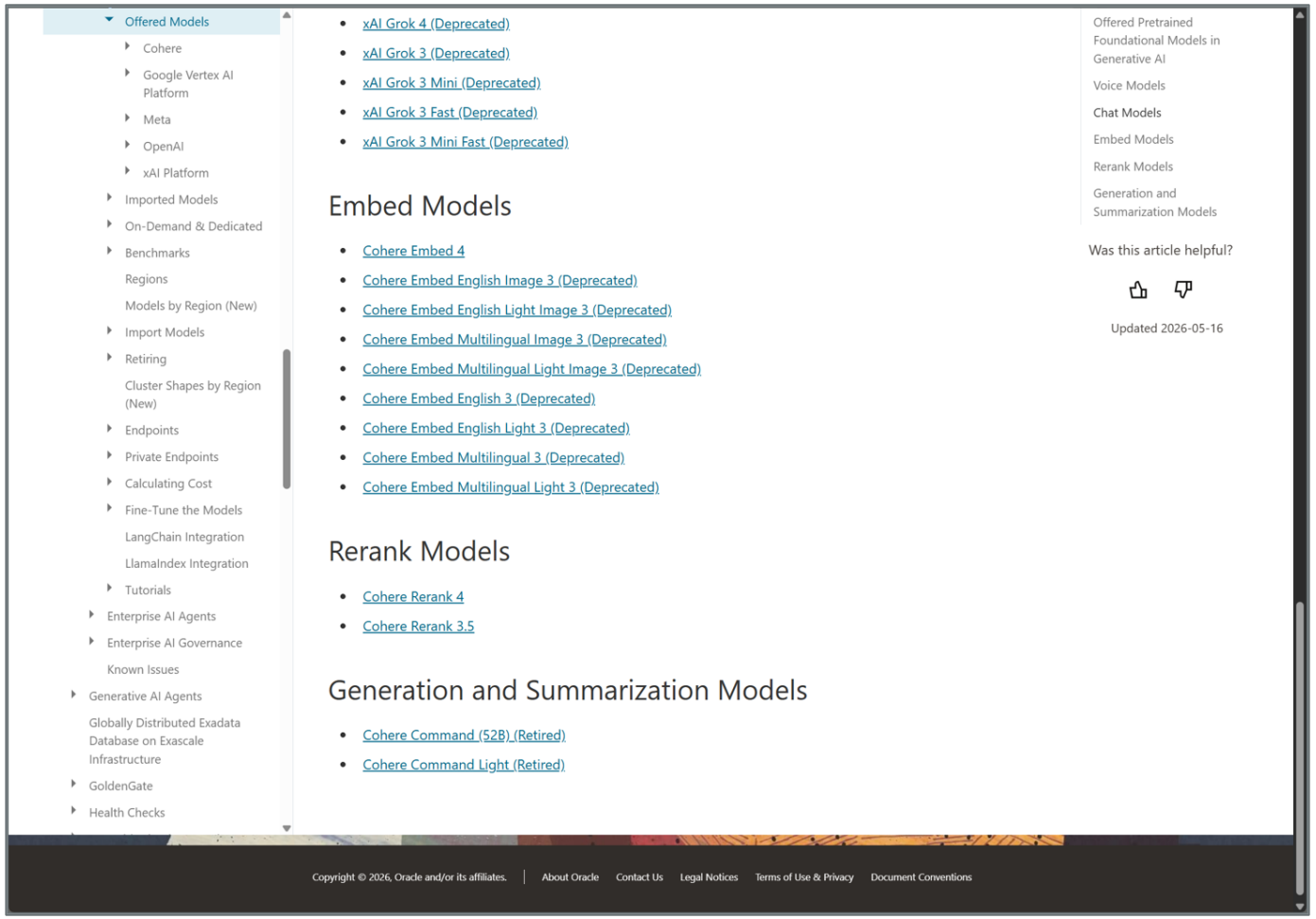
Pick a model and enter the content you want to embed.

Embedding



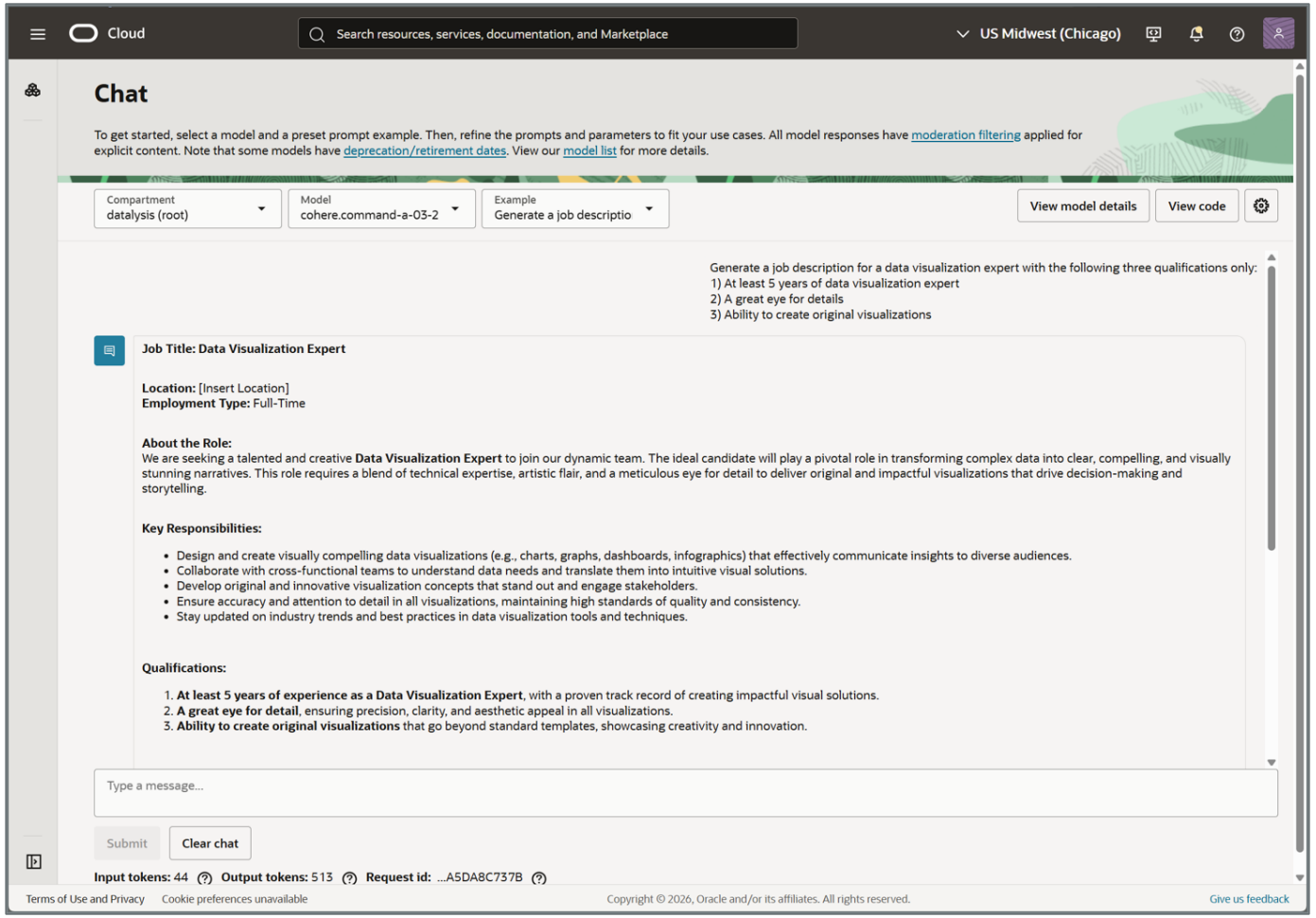
The 2D visualization of the resulting vectors is very pointless and misleading (reducing a vector with a dimensionality between 256 and 1536 to 2, how correct can it be?)

Embedding



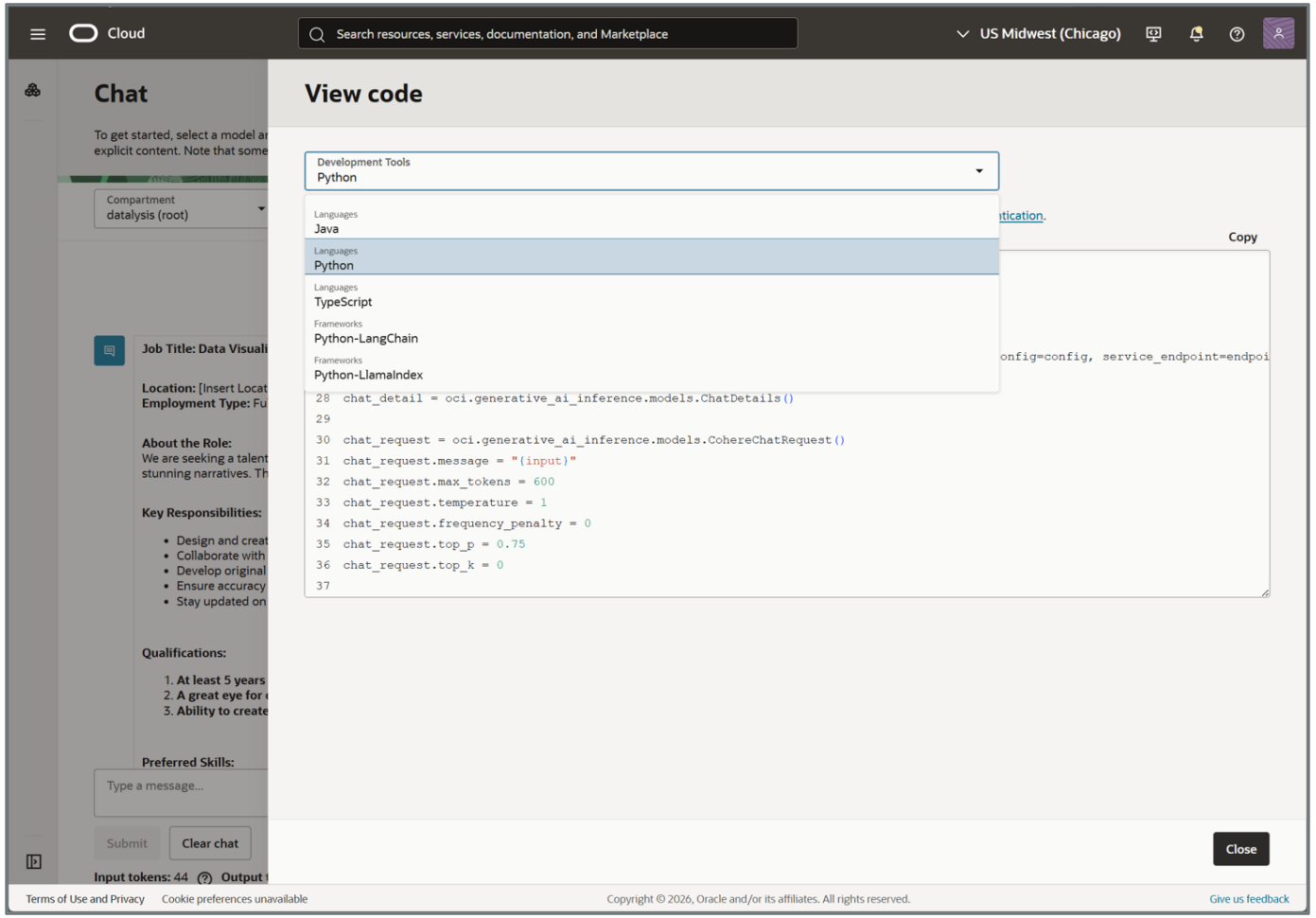
Models are deprecated when new versions are available

Chat



Select a model and start chatting with the LLM

Chat



You can generate working Python or Java code to call the LLm in your application

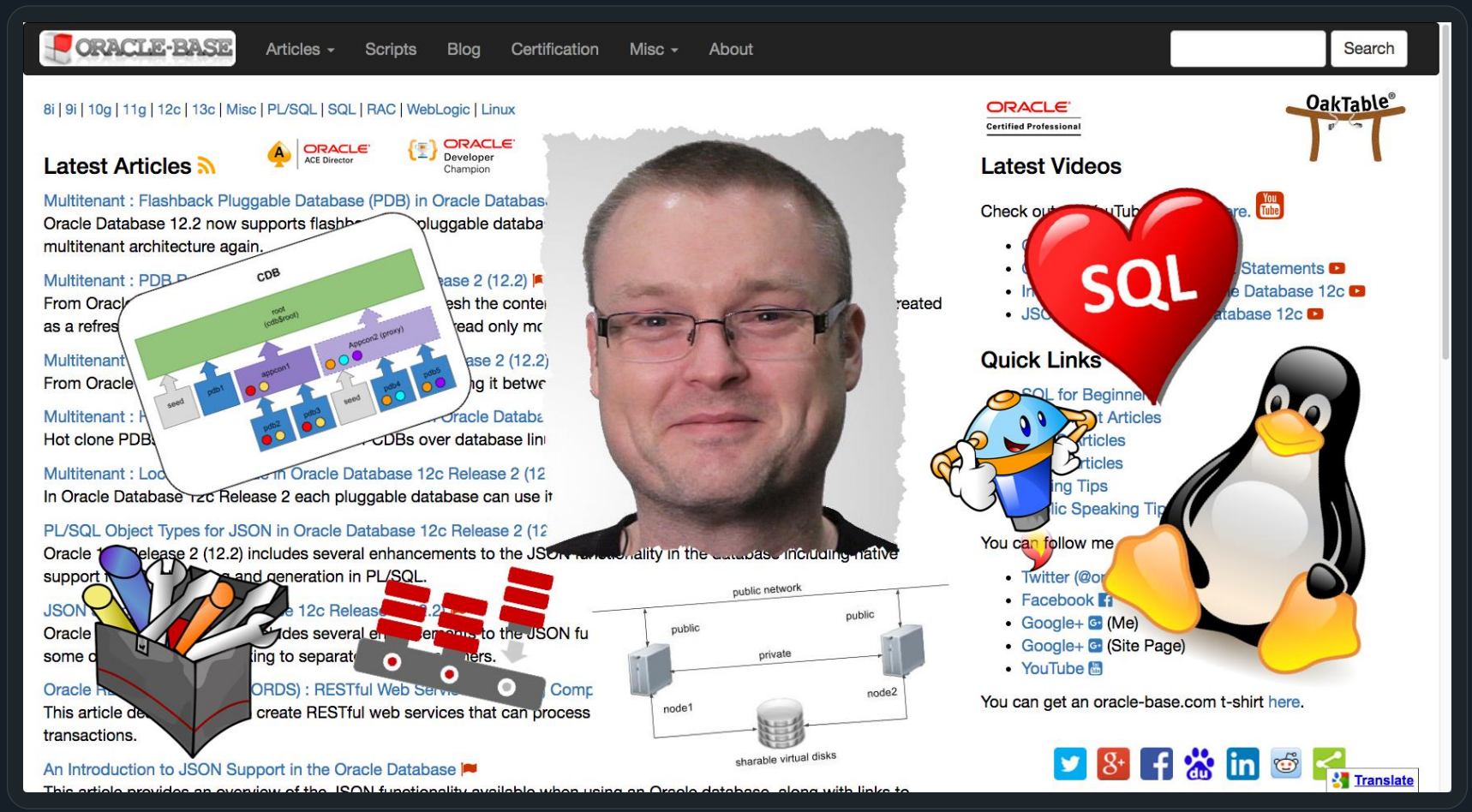
Data

Everything begins with data...

OracleBaseGPT

When googling something about an Oracle Database you often find oracle-base.com in the first results.

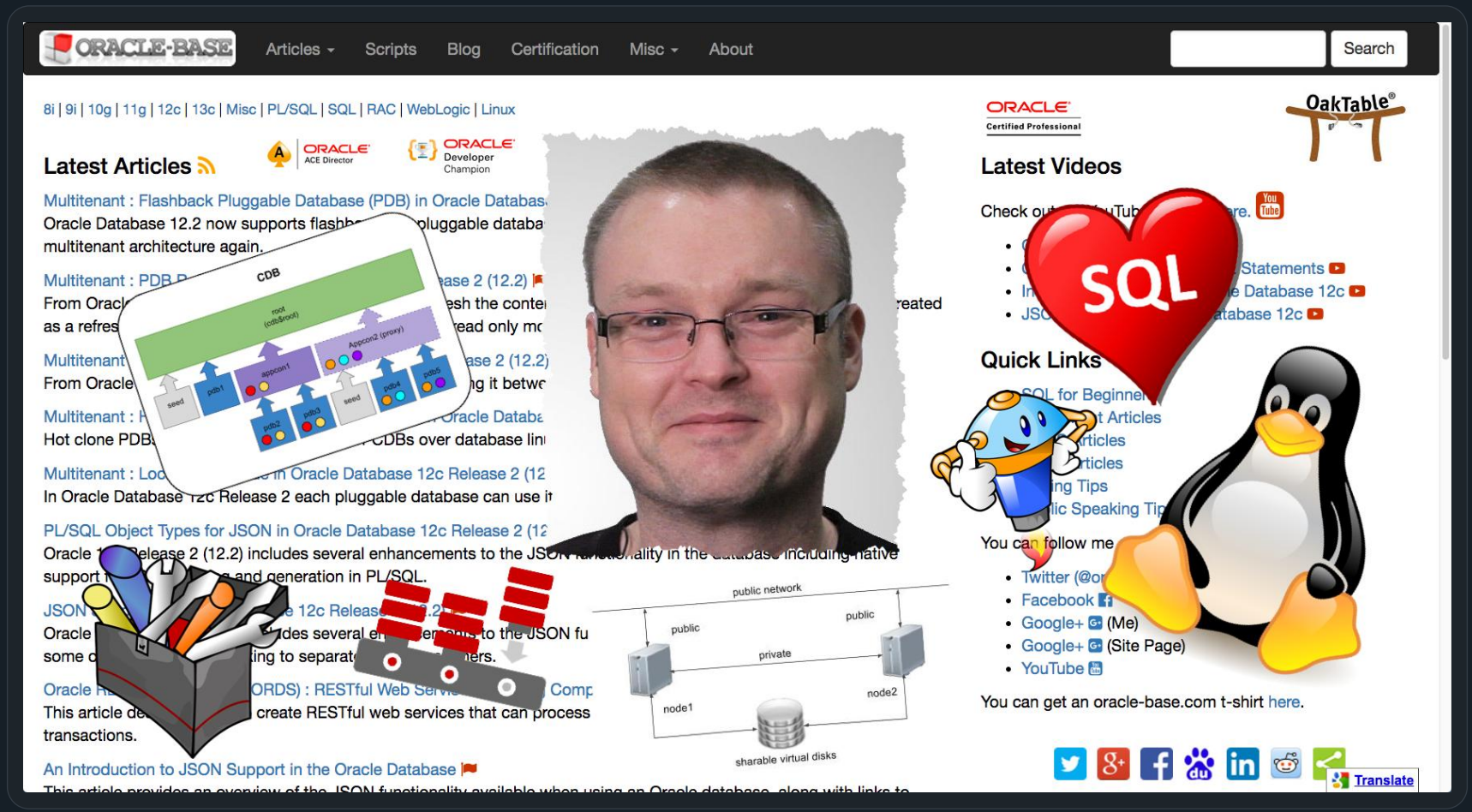
You maybe never met Tim Hall, but you know his website.



OracleBaseGPT

What happen if we take a chatbot powered by an LLM and feed it with oracle-base.com articles?

Can we get a personal OracleBaseGPT?



Before...

Do not randomly steal content online from somebody's website without being allowed to do so: you can maybe navigate the website, but you don't own the content, you can't always do whatever you want with it.

Do not kill websites by sending tons of requests just because you are playing with some embedding task.

Be respectful, ask permission.



This presentation is done with the permission of Tim Hall to use his articles.

Know your source

Any ML, AI, LLM usage will follow the same rule:

- “garbage in, garbage out”

You must know your source, its format, and type.

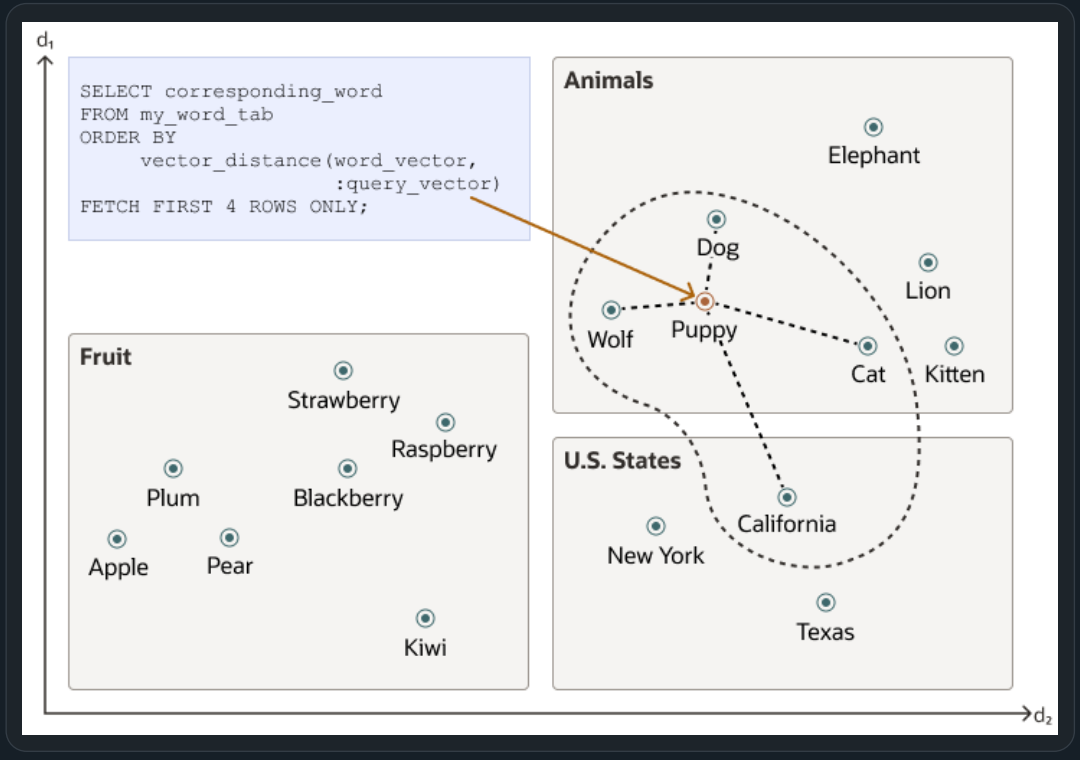
Know your source

This is even more important for RAG activities:

- Generate an embedding of garbage content and you will get a fairly pointless embedding.

Reality isn't as simple as the images in the documentation...

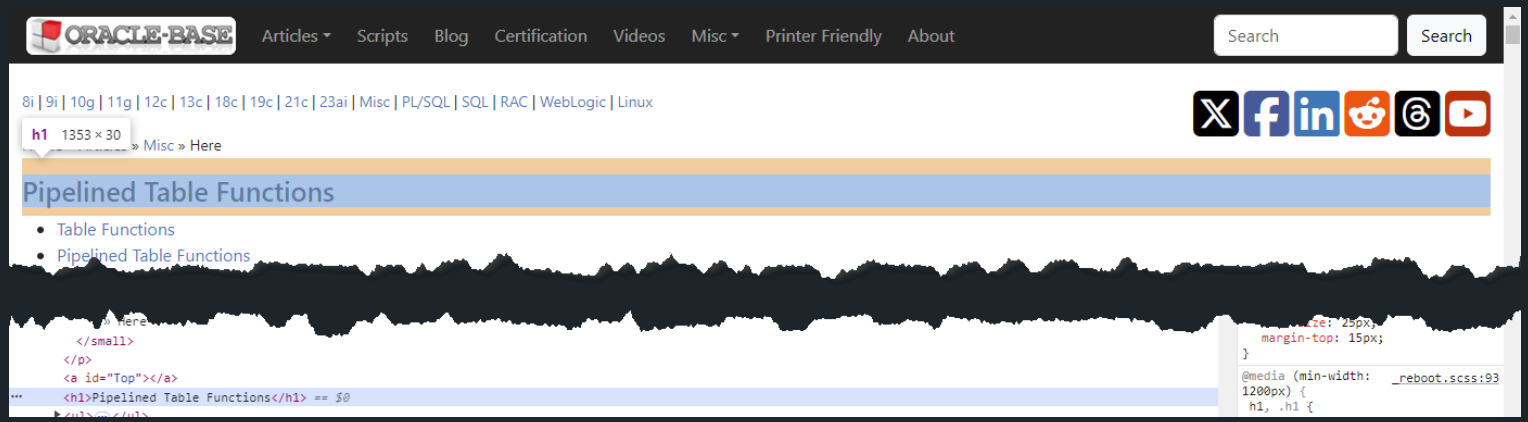
Fruits, Animals or US States nicely grouped together is an illustration of an idealistic view of how vectors look like.



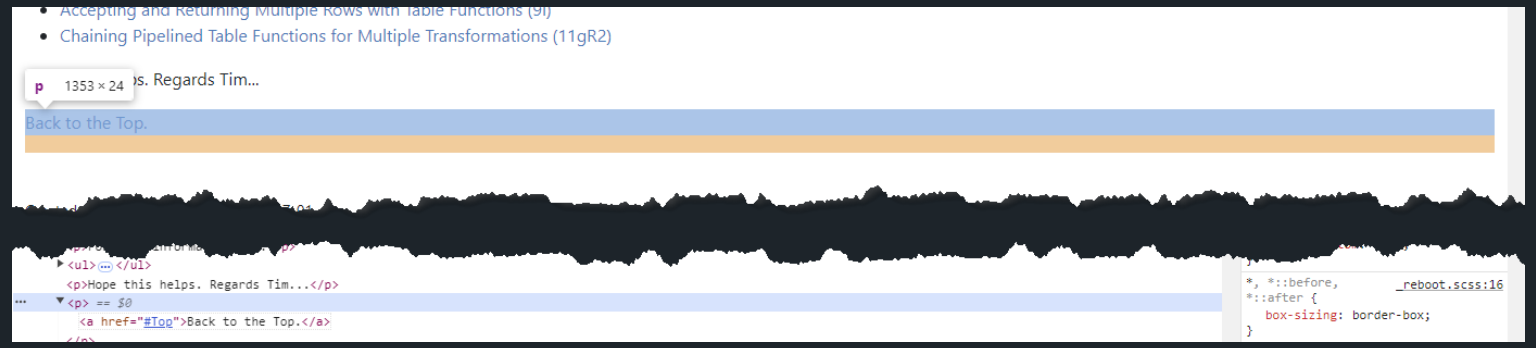
Know your source

Are header and footer useful? No!

Identify the real content and how to get rid of the “noise” around it.



All articles starts after the anchor.



All articles ends before a link back to the top anchor.

Chunking

Big or small? Your choice!

Chunking: splitting your source documents into smaller pieces for embedding.

Finding the best way to chunk your documents is usually the most challenging task.

The most import step for a RAG initiative

Embedding generate a vector for each input you provide.

Should you provide a whole document as one piece?

You will get a single vector for the whole document.

How precisely could a single vector represent a whole document and every concept contained in it?

The most import step for a RAG initiative

Should you provide every single word of a document as input for embedding?

You will get a vector for each word, but no context.

Each word will be an individual, separate, piece without any relation one with the other.

An unlimited number of strategies

If you search online “chunking” you will find about as many strategies as people talking about it.

Depends heavily on your content.

Test various strategies till you find the sweet spot.

> **“The best is the enemy of the good” is true for chunking.**

Embedding

Give me vectors!

The simplest task

Embedding doesn't require much efforts:

- **Select your embedding model and stick to it!**
- **Text in → vector out**
- **Store the vector and the piece of text it comes from**

You can't freely replace your embedding model.
Everything need to be embedded again!

Generation

An LLM doing what LLMs do

Retrieval before generation

Before to ask an LLM to generate the answer, we need to find the supporting documents on which the answer will be based.

01

Embed question

Produce a vector for the question asked.

02

Metric function

Decide the metric function to calculate vectors' distances.

03

SQL Query

Use ORDER BY VECTOR_DISTANCE(...) to get the nearest chunks.

04

Get the "content"

Get the chunks content to support the LLM in generating the answer.

Supported metric functions in 26 ai are:
COSINE, DOT, EUCLIDEAN, EUCLIDEAN_SQUARED, HAMMING, MANHATTAN, JACCARD

LLM generating an answer

All the pieces are now available.

Build the prompt:

- Question
- Retrieved content
- Any other instruction you want to provide to the LLM

Call the LLM

The generated text is always a surprise: can be good, or very bad!

RAG

RAG is “just” that: help the LLM generate better answers.

An example of RAG

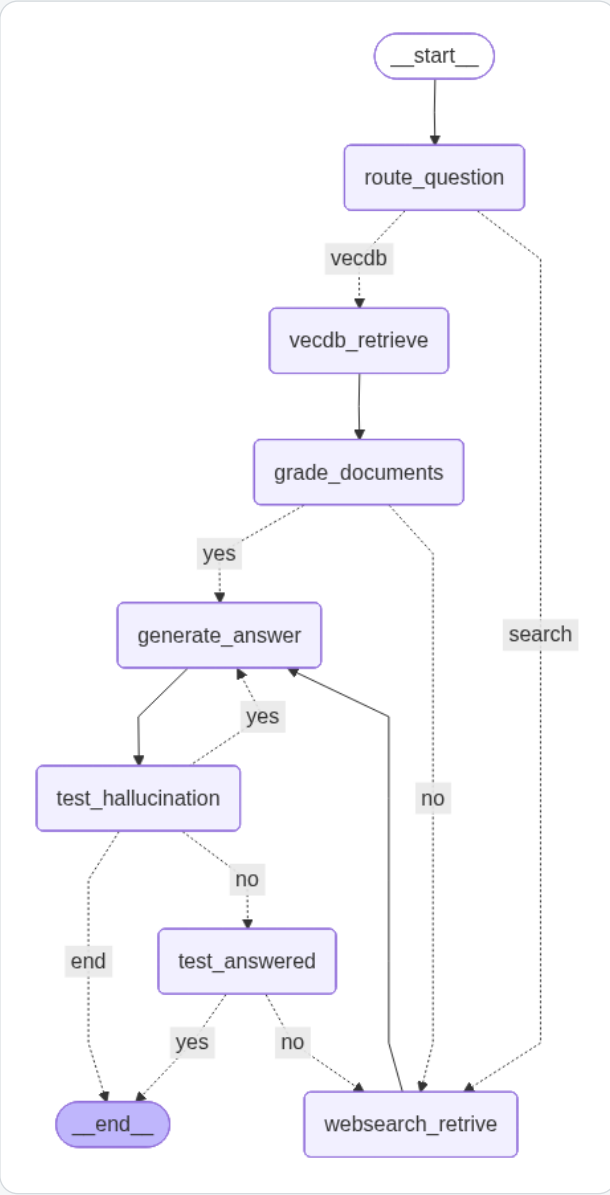
OracleBaseGPT

- **1327 articles**
- **9608 chunks**

The code was built 2 years ago...

- Embedding: cohere.embed-english-v3.0
- LLM: Meta Llama 3.3 70B

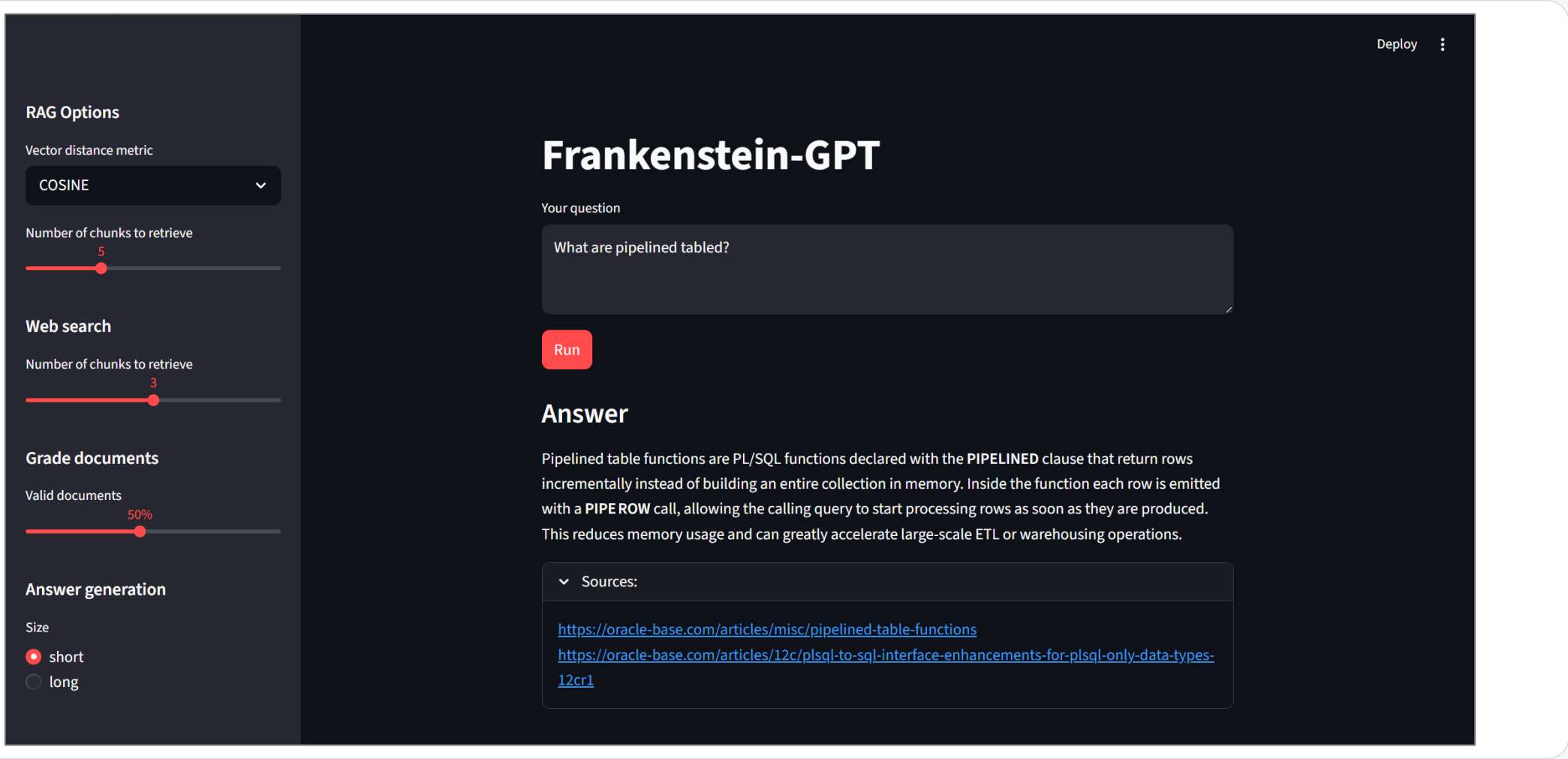
Both are fairly old models, there are better alternatives these days...



Built with LangGraph and
OCI Generative AI for embedding and LLM

An example of RAG

OracleBaseGPT



UI built with Streamlit

Automated RAG

In 2026 RAG is fully automated

No manual action anymore

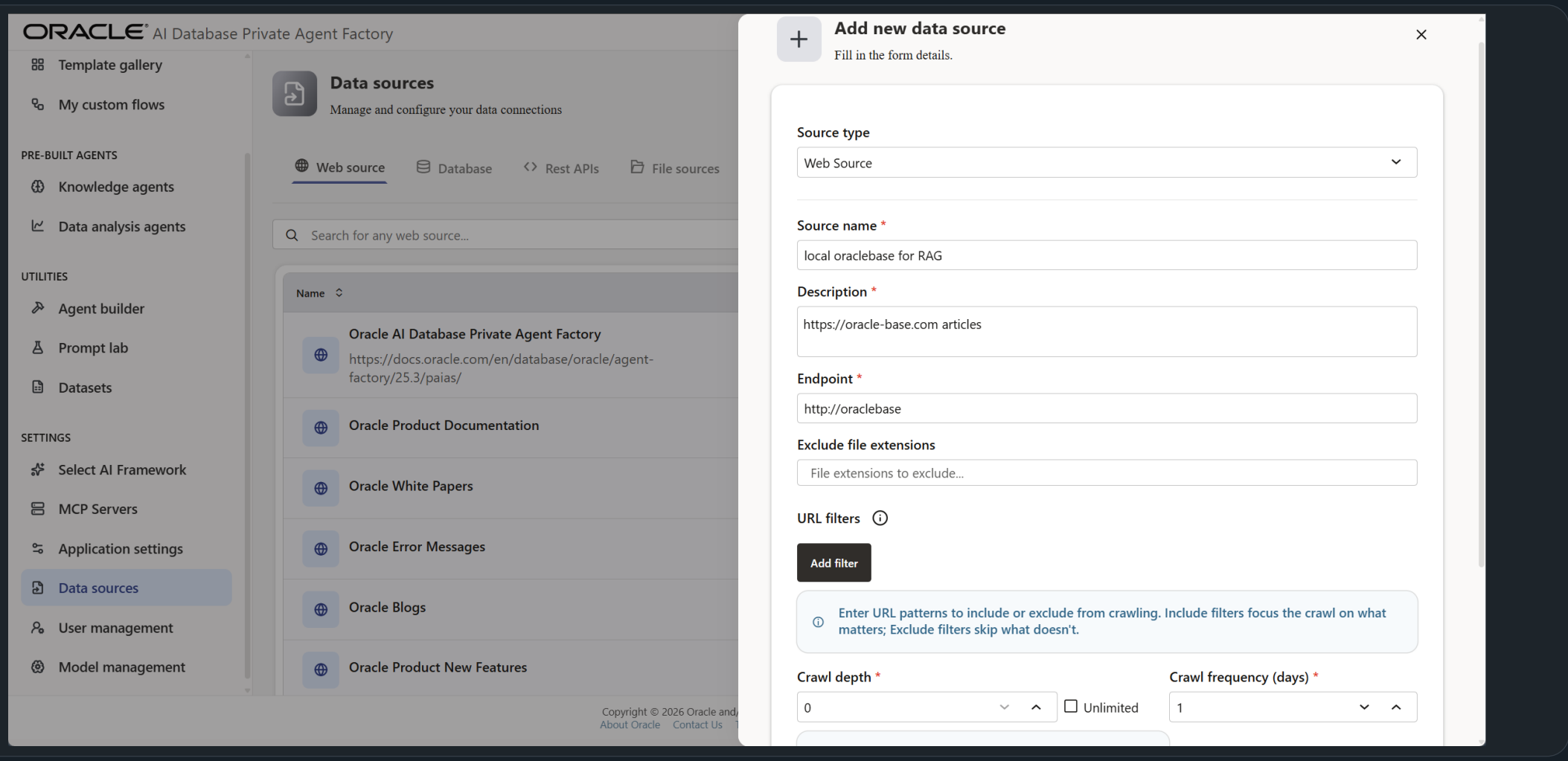
Oracle AI Database Private Agent Factory is one of many products doing RAG in a fully automated way:

- You provide the data source
- Wait some minutes
- You start asking questions

You don’t do chunking, select vector sizes, use a ranker or evaluate the retrieved documents. You just use it!

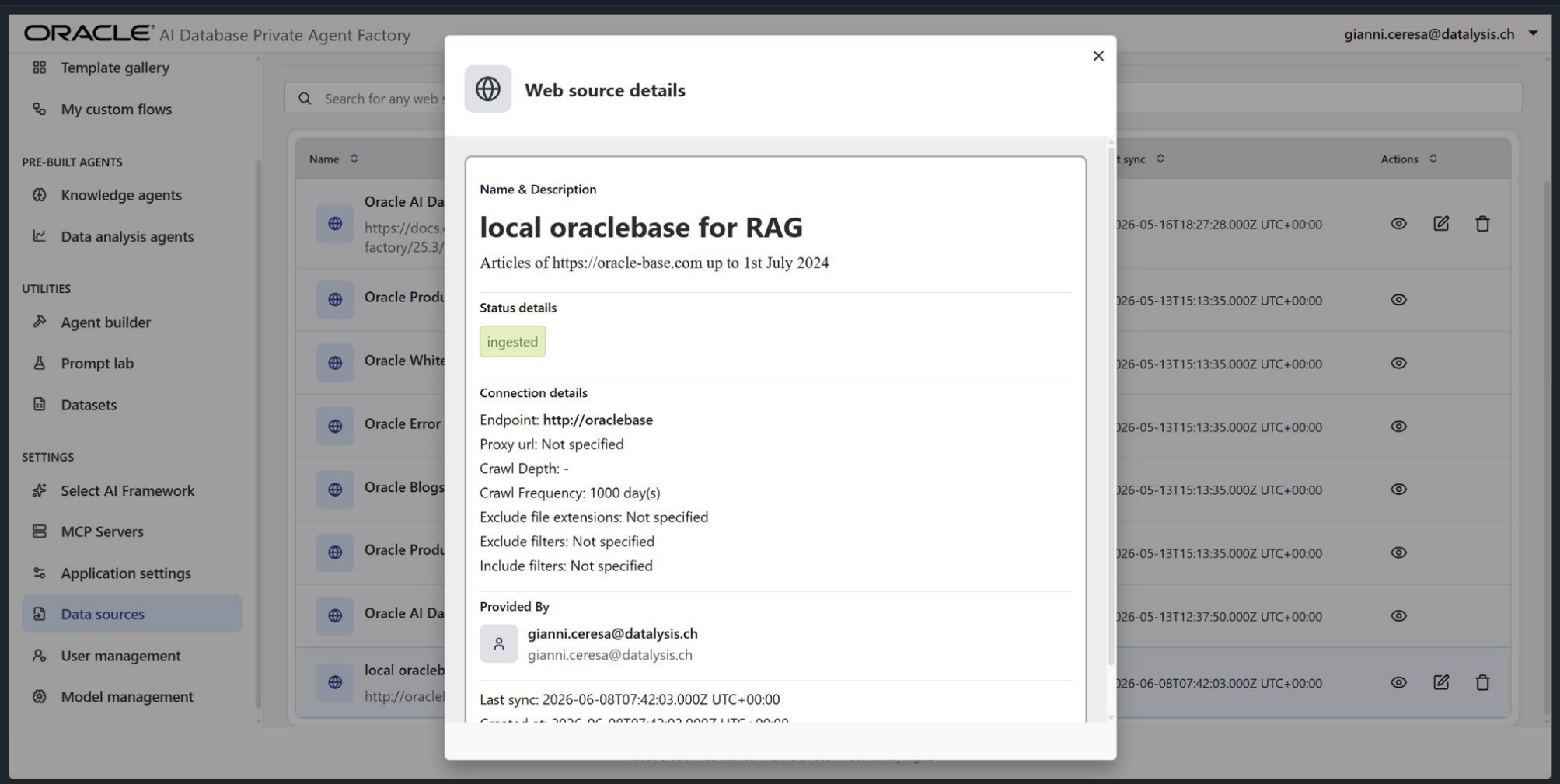
Automated RAG

Oracle Private Agent Factory



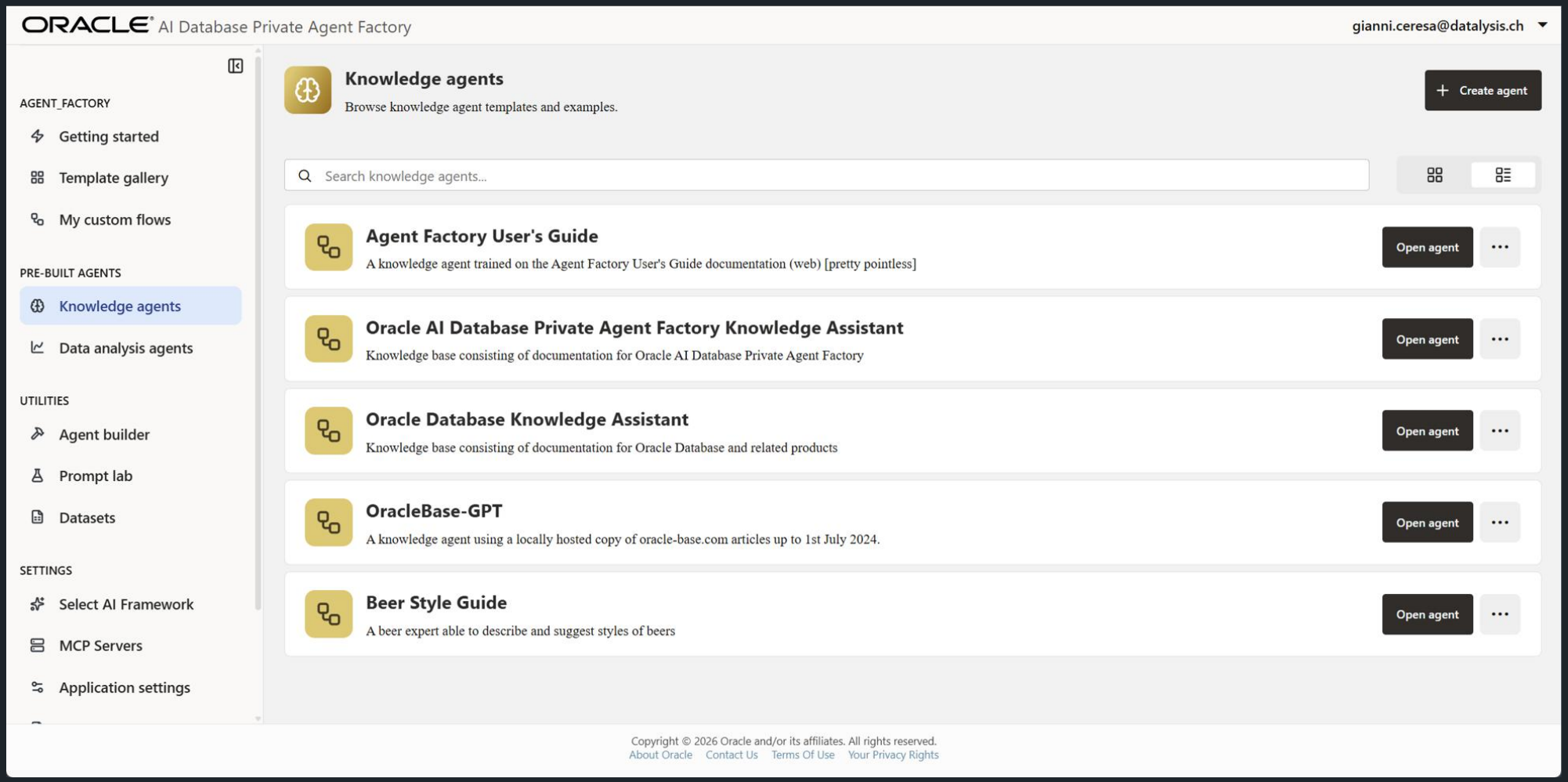
Define a data source for the website to crawl.

Oracle Private Agent Factory



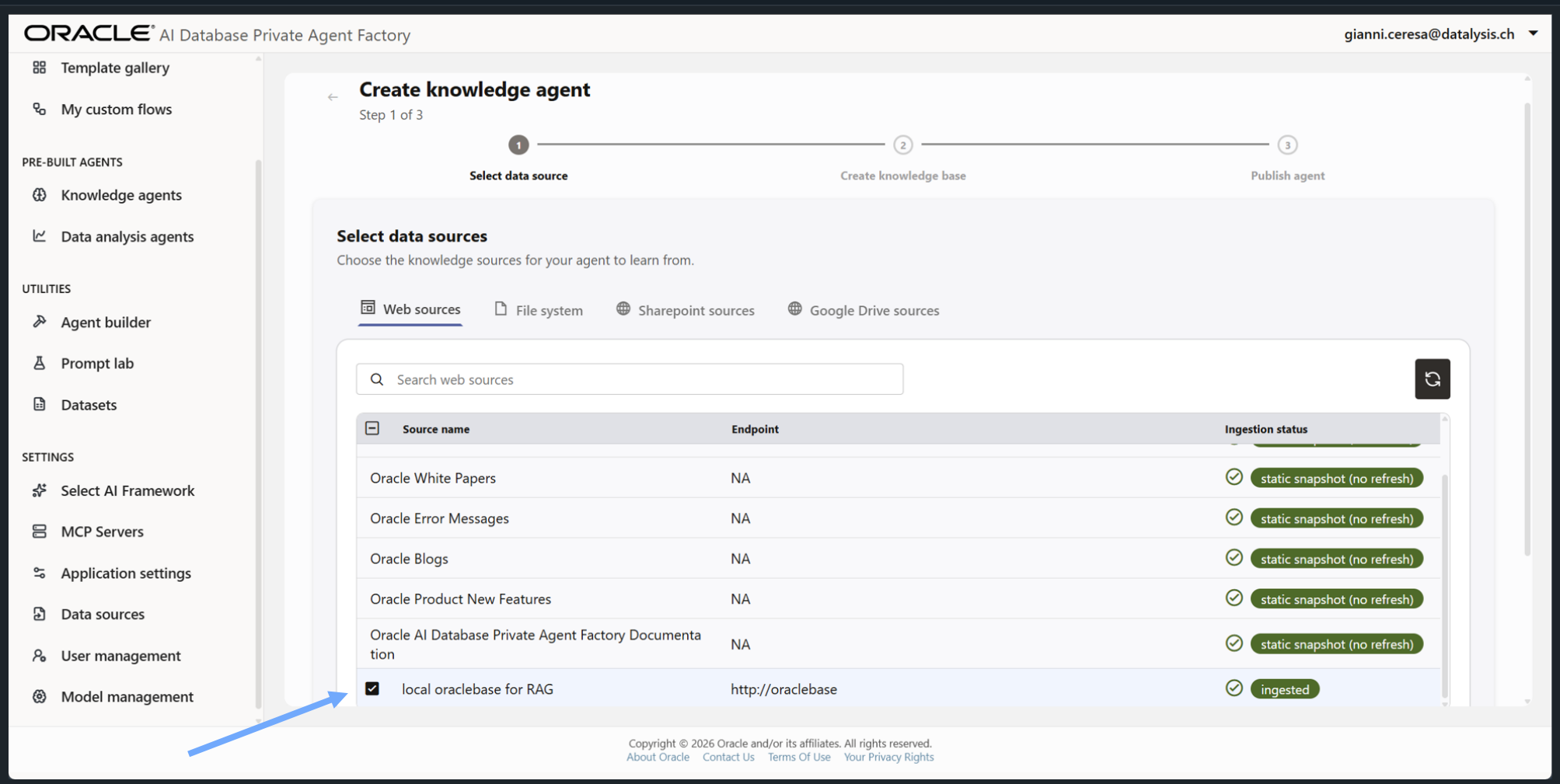
The content of the data source is ingested.

Oracle Private Agent Factory



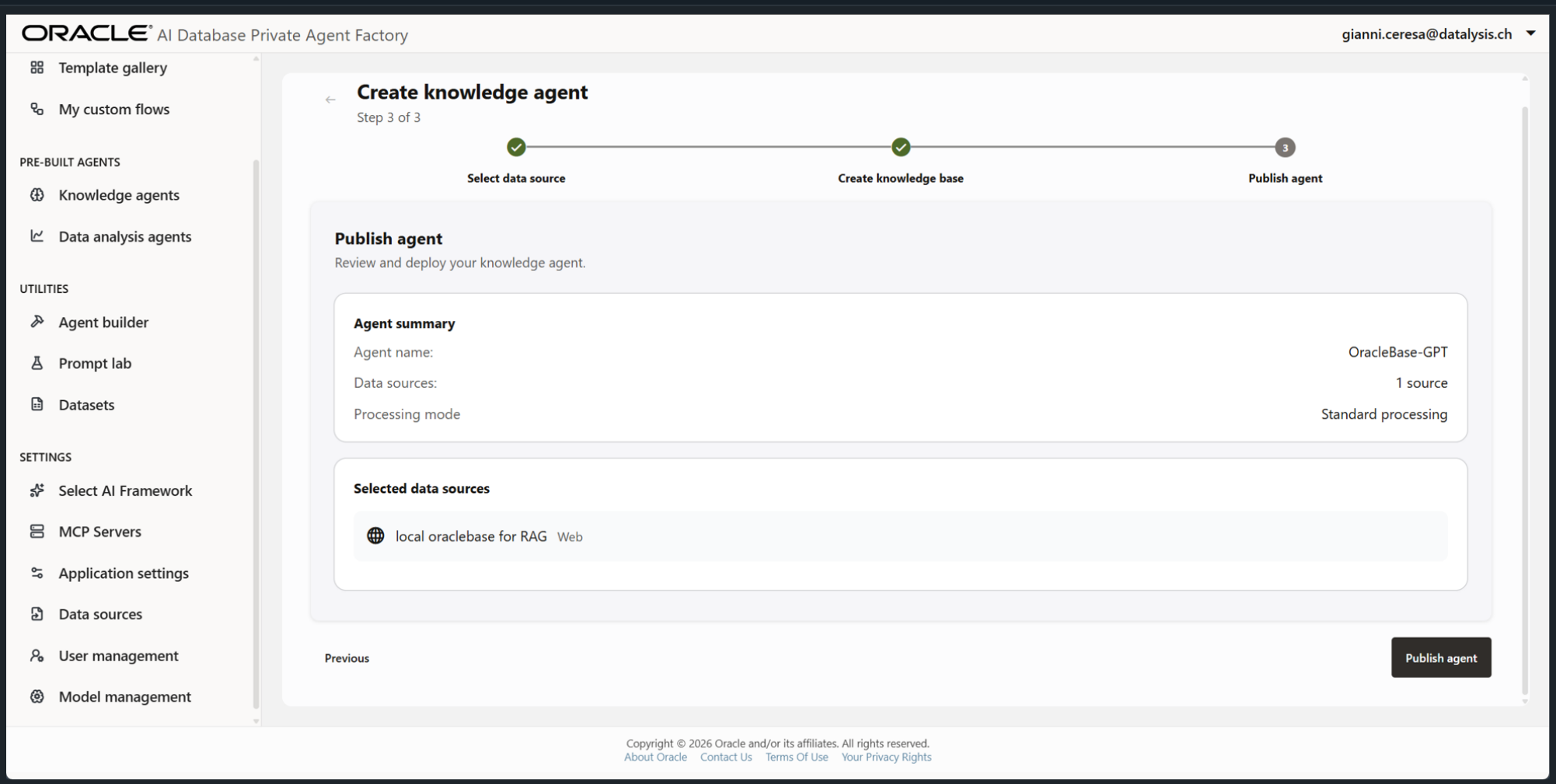
Create a Knowledge agent on the data source.

Oracle Private Agent Factory

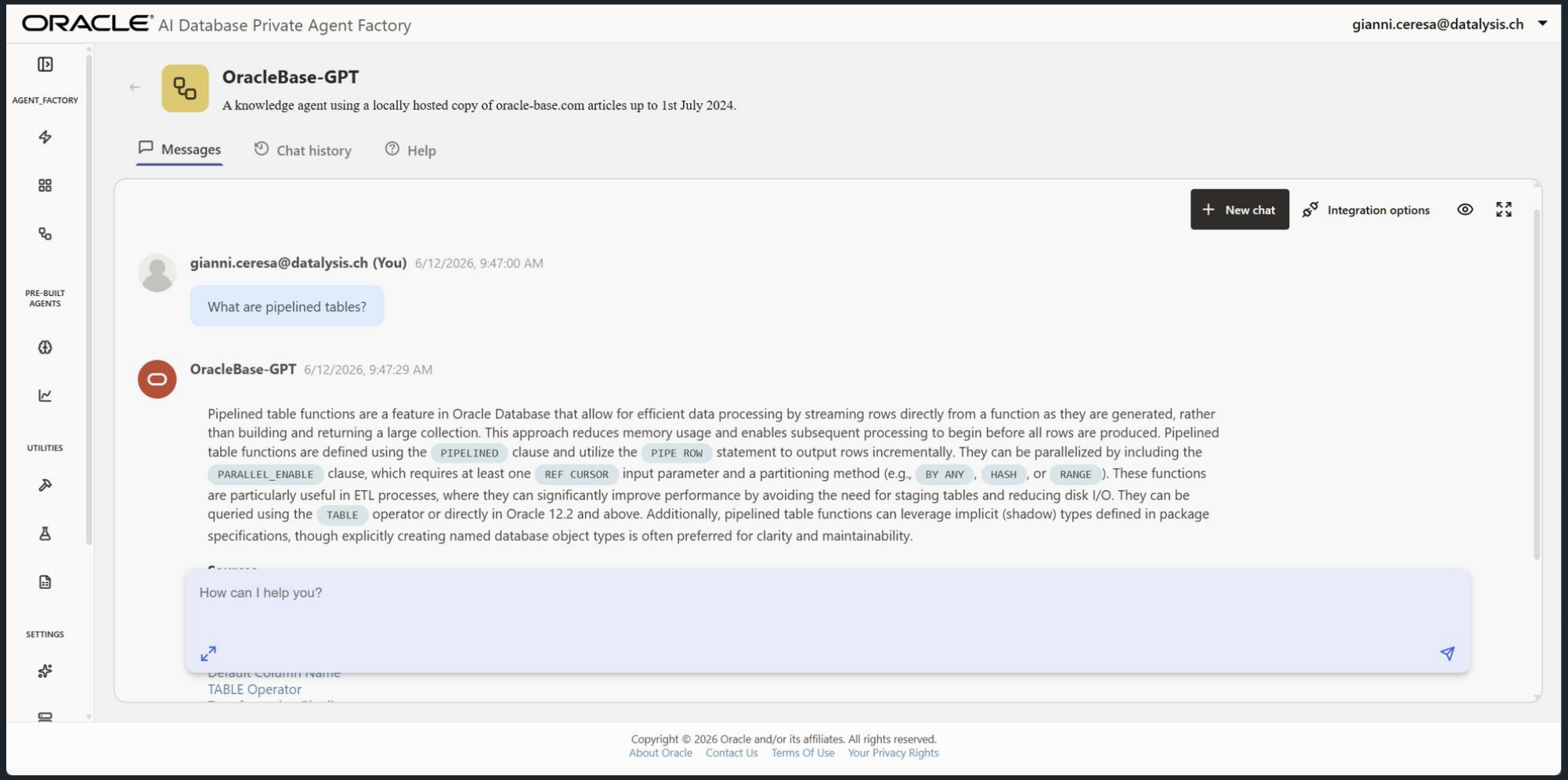


Automated RAG

Oracle Private Agent Factory

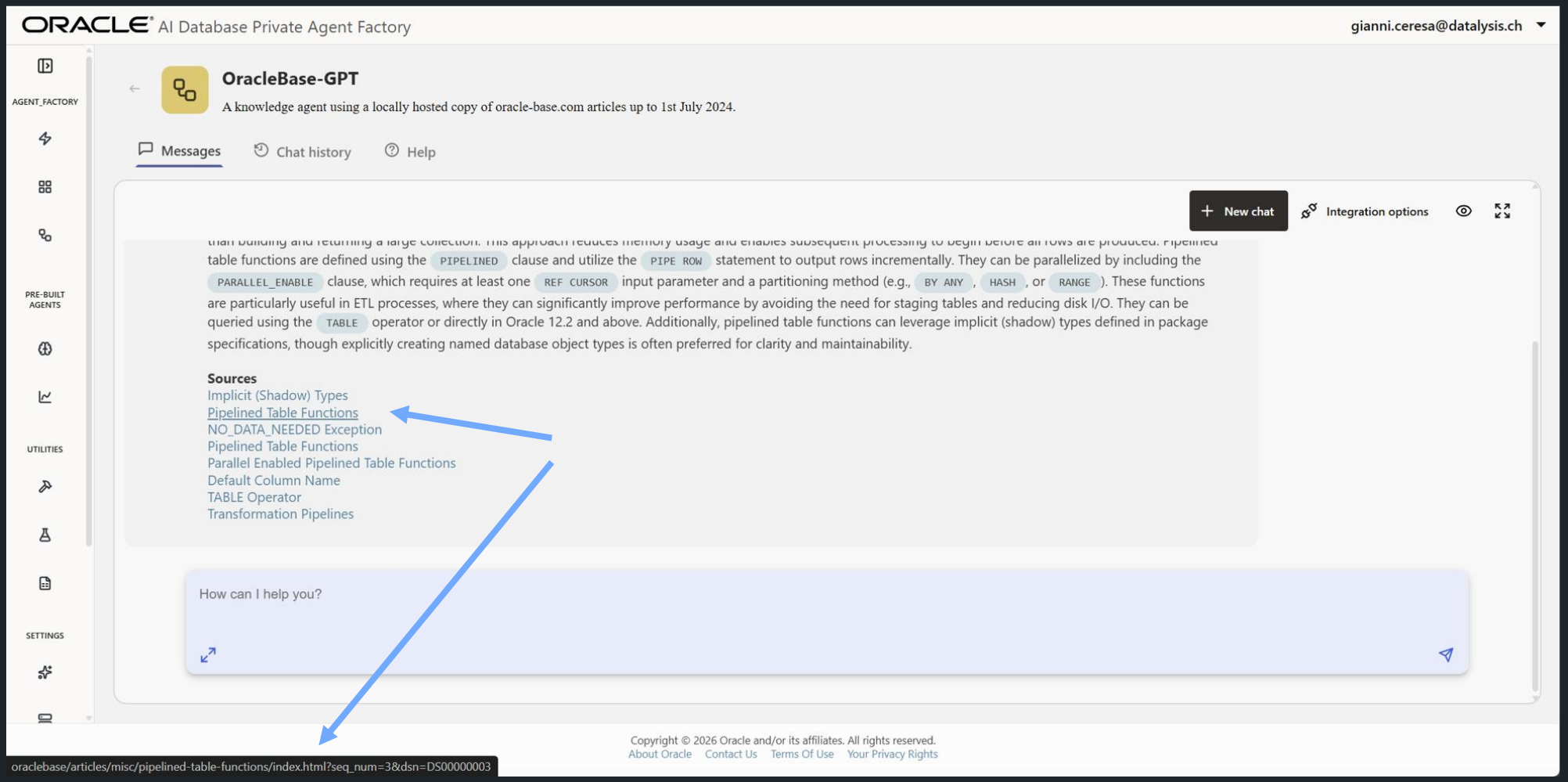


Oracle Private Agent Factory



Ask a question to the Knowledge agent, the answer is only based on content of the data source.

Oracle Private Agent Factory



The knowledge agent provide links to the sources used to generate the answer.

The bottom line

Doesn’t matter how you do RAG as long as you use RAG:

- Fully manual
- Automated by a tool
- Memory (state of the agent)

Grounding LLMs generation to factual data you trust allows to heavily limit hallucinations or “wrong context” answers.

RAG in 4 steps

01

Data & Chunking

Clean source data and chunk in smaller pieces.

02

Embedding

Embed each chunk into a vector.

03

Retrieval

Find closest chunks to the question by vector distance.

04

Generation

Generate an answer based on the content of the chunks.

Manual or automated, RAG is more or less always the same process.

DATAlysis

Thank you

Join me at 13:15 to learn more about Oracle AI Database Private Agent Factory.

Gianni Ceresa · @G_Ceresa · @gianniceresa.bsky.social · datalysis.ch · info@datalysis.ch