Microsoft

LAB360

Build an Agentic App with GraphRAG, Semantic Kernel, and the new VS Code Extension for PostgreSQL

Speakers:
Jonathon Frost
Joshua Johnson

Proctors:
Guy Bowerman
Ismael Mejía Useche

Agenda

- Lab Overview
- PostgreSQL AI Core Concepts
- Part 0 - Login to Azure
- Part 1 - Setup your PostgreSQL Database
- Part 2 - Use AI-driven features in PostgreSQL
- Part 3 - Build the Agentic App

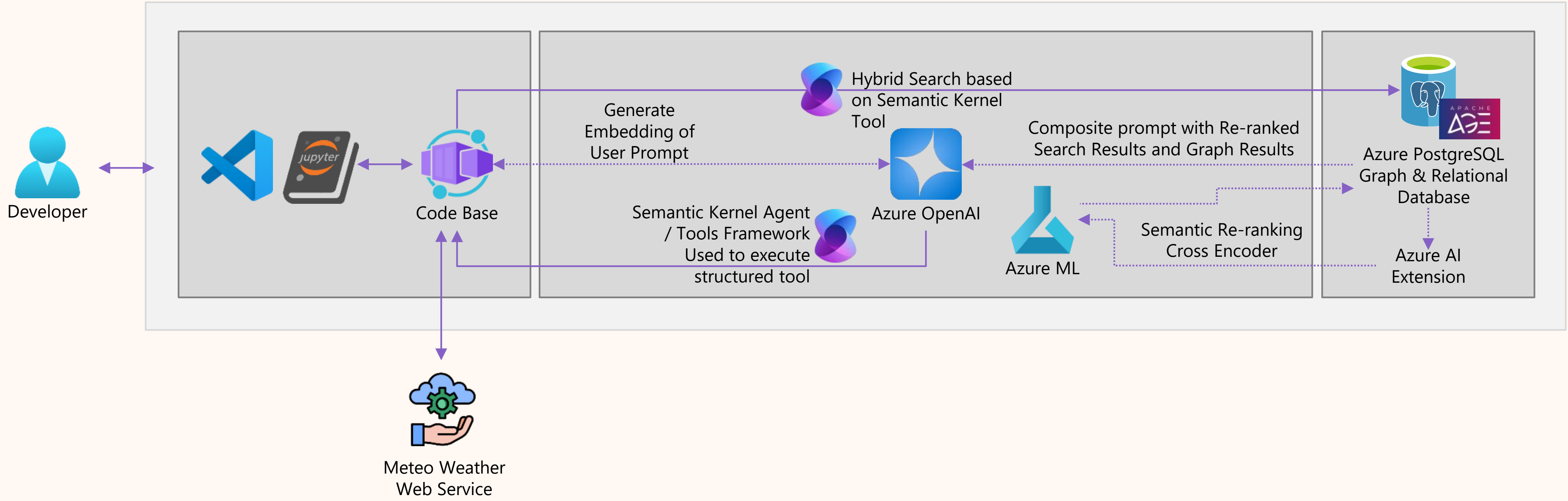
Lab Overview

What you will learn:

- How to use the new VS Code PostgreSQL Extension
- Understand how to use Vector and Vector Indexes with PostgreSQL
- Learn about Agentic App architectures and coding patterns
- Hands-on building an Agentic App with PostgreSQL

Agentic App Architecture

The App we are going to build today.



Dataset for the Lab

- Caselaw Dataset for Washington State
- Subset of 337 unique legal cases
- Columns include: id, name, opinion, etc.
- Located at /Dataset/cases.csv

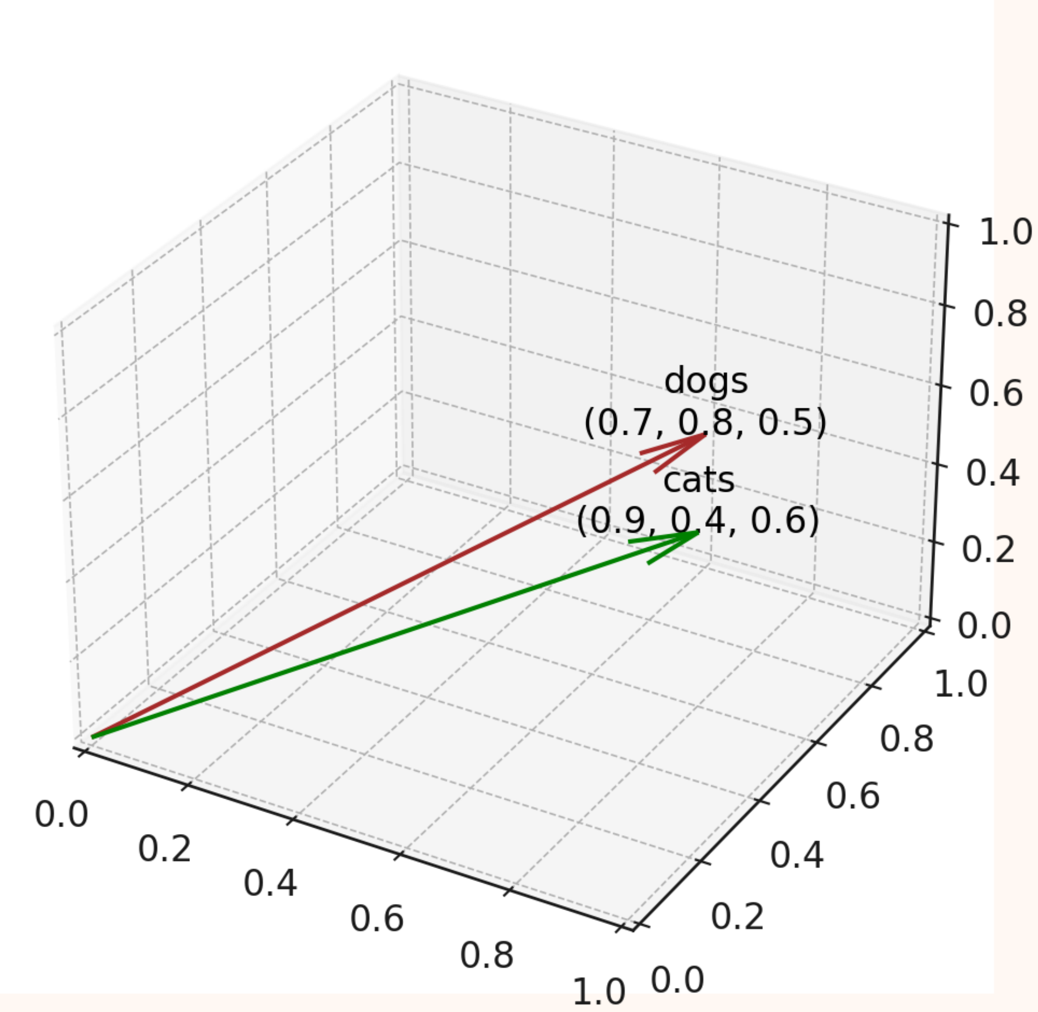
	id	name	decision_date	court_id	opinion
1	507122	Berschauer/Phillips Construction Co. v. Seattle Sc…	1994-10-06	9029	"Guy, J.\nWe granted review to decide whether a gen…
2	5041745	Frisken v. Art Strand Floor Coverings, Inc.	1955-10-13	9029	"Rosellini, J.\nThe respondent, Florence Frisken, i…
3	5008733	Pate v. General Electric Co.	1953-09-04	9029	"Weaver, J.\nPlaintiff was injured while engaged in…
4	5007905	Cambro Co. v. Snook	1953-11-05	9029	"Donworth, J.\nPlaintiff instituted this action to …
5	5008594	Buttnick v. Clothier	1953-11-16	9029	"Donworth, J.\nThis action was instituted by plaint…

PostgreSQL AI Core Concepts

- Vectors
- Vector Indexes
- Semantic Search

What is a Vector?

- Lists of numbers that represent items in a high-dimensional space.
- For example, a vector representing the string “**dogs**" might be [0.7, 0.8, 0.5].
- Each number in the vector is a dimension of the space.



How to generate a vector?

Use a model to generate vectors for items:

Input	→ Model	→ Vector
"**dog**"	text-embedding-3-small	[0.017198, -0.007493, -0.057982, ..]
"**cat**"	text-embedding-3-small	[0.004059, 0.06719, -0.093874, ...]

Model (bi-encoder)	Input types	Dimensions
OpenAI: text-embedding-3-small	Text	1536
OpenAI: text-embedding-3-large	Text	3072
Mistral: e5-mistral-7b-instruct	Text	4096

Popular models (find more on HuggingFace):

What should we care about vectors?

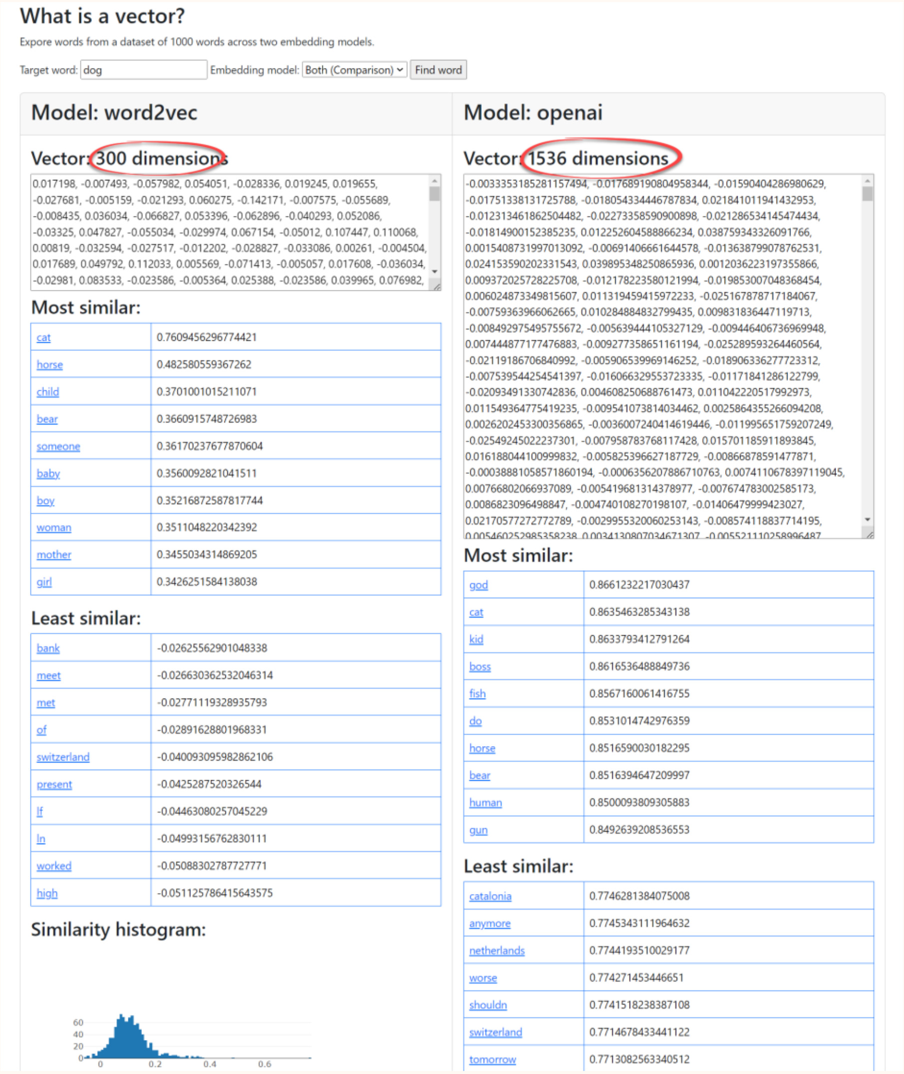
Search & Similarity

Search and retrieve items that are similar to what you’re querying.

(Optional) Exploring Vectors

Generate Example Vectors

https://pamelafox.github.io/vectors-comparison/



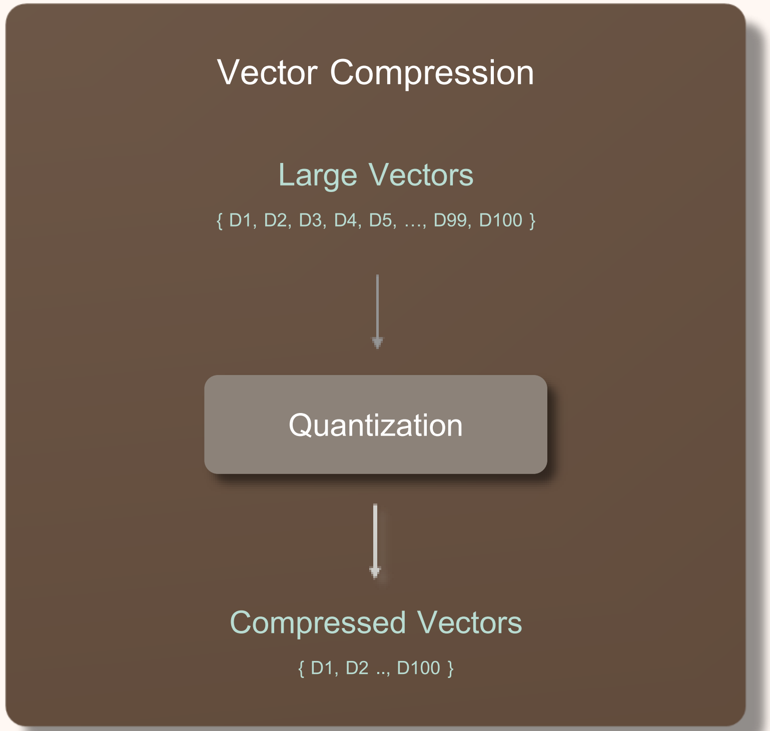
Storing vectors in the PostgreSQL Table

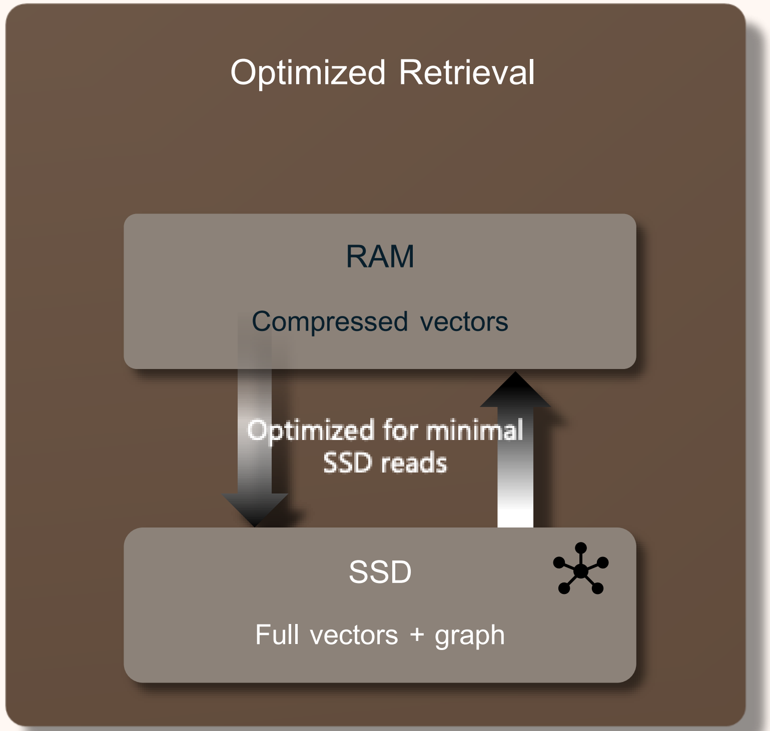
Results Messages

	id	name	opinions_vector
1	507122	Berschauer/Phillips Construction Co. v. Seattle Sc…	[-0.0077604363,0.034168452,0.022548927,0.058252566,0.0027358707,0.013302599,-0.04104158,-0.0011557909,-0.02792912,-0.00568652…
2	5041745	Frisken v. Art Strand Floor Coverings, Inc.	[0.008968134,0.04363906,-0.0017264026,0.0380413,0.006953235,0.0002628528,-0.022229837,-0.028633554,-0.011818302,0.0461009,0.0…
3	5008733	Pate v. General Electric Co.	[-0.009503542,0.052598044,-0.00058293104,0.051410984,0.013446276,0.017848289,-0.013997411,-0.02381185,-0.020533305,0.03219192…
4	5007905	Cambro Co. v. Snook	[0.02875072,0.033727877,0.00932174,0.004737335,0.037787456,0.01634954,-0.045406118,-0.019574959,-0.010670299,0.017281018,0.03…
5	5008594	Buttnick v. Clothier	[0.0077795624,0.035135385,0.029488107,0.02745043,-0.017844236,0.013717937,-0.023156751,-0.028396495,-0.03015763,-0.03202065,0…

Vector Indexing - DiskANN

- Highly performant, scalable, and accurate index for vectors
- Superior to IVFLAT and HNSW
- Reduced memory footprint by storing vectors on SSD
- Compression and quantization improve speed and accuracy of vector search
- Accuracy retained as data changed





Lab Part 0 – Login to Azure

In this part of the lab…

1. Log into Azure Portal
2. Verify Azure Services are provisioned

Lab Part 0 – Login to Azure

Azure Services:

ResourceGroup1:

- Azure OpenAI
- Azure PostgreSQL Database

Lab Part 1 – Setup your Azure PostgreSQL Database

(Work from the Lab Manual)

In this part of the lab…

1. Open VS Code
2. Use Connection Dialog to Setup Database Connection
3. Launch PSQL Command Line Shell in VS Code
4. Populate the Database with Sample Data
5. Install and configure the azure_ai extension
6. Explore the azure_ai extension schema
7. Review the Azure OpenAI Schema

Lab Part 2 – Using AI-driven Features in PostgreSQL

(Work from the Lab Manual)

In this part of the lab…

1. Open New Query Editor in VS Code PosgreSQL Extension
2. Using Pattern matching for queries
3. Using Semantic Vector Search and DiskANN Index
   - Create, Store, and Index Embedding Vectors
   - Perform a Semantic Search Query

Lab Part 3 – Build an Agentic App

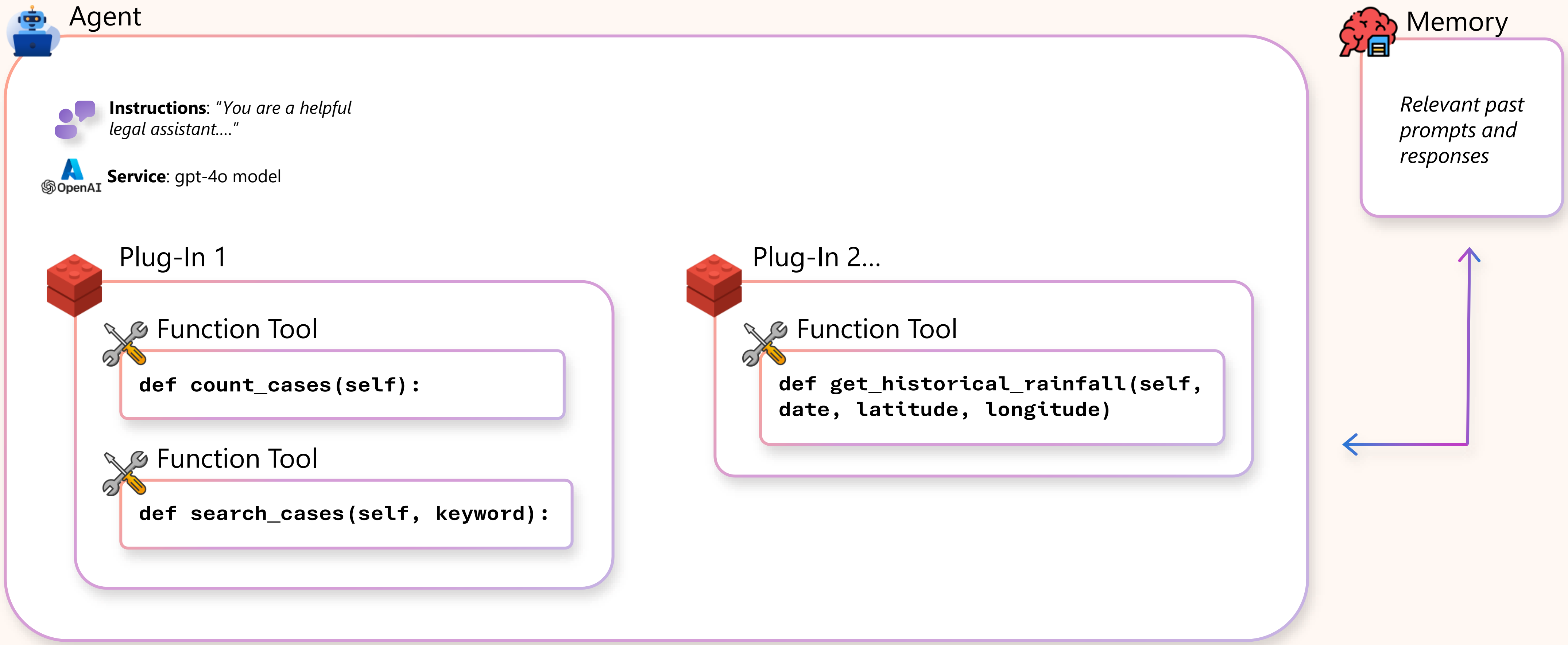
(Work from the VS Code Notebook)

In this part of the lab…

1. Setup Python Imports
2. Setup environmental connection variables
3. Create Semantic Kernel Plugin for Basic Database Queries
4. Test Run our New Agent
5. Improve Agent Accuracy with Semantic Re-ranking
6. Add GraphRAG Plug-In to Agent
7. Re-assemble our Agent with new GraphRAG Plug-In
8. Add a Weather Service Plug-In
9. Re-Test our Agent with all Plug-Ins Together
10. Add memory into the Agent

Agents

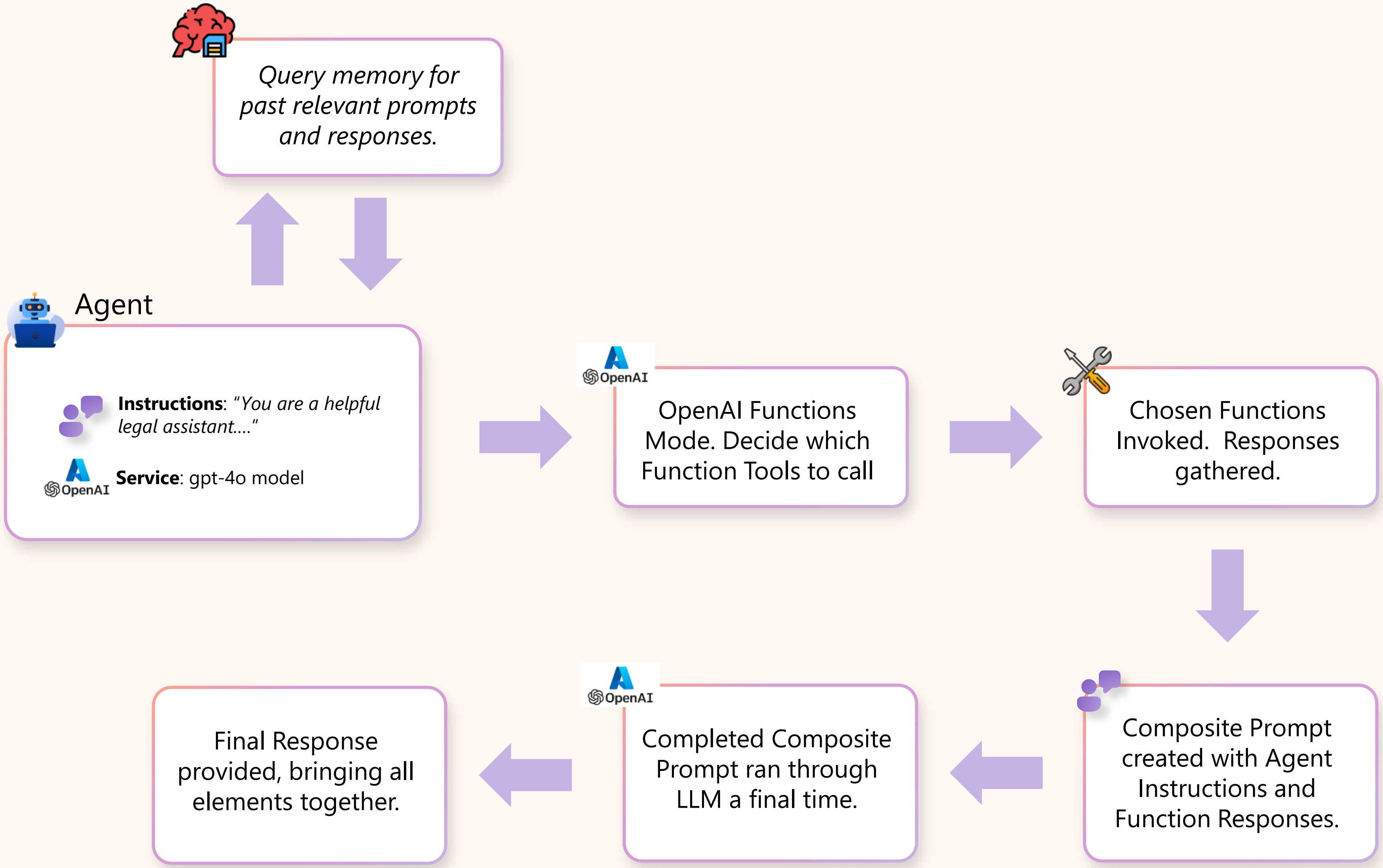
Semantic Kernel



Agents
Logical Flow

Prompt:

"How many cases are there, and high accuracy is important, help me find 10 highly relevant cases related to water leaking in client's apartment."



Semantic Re-Ranking

Process

1. Takes top 100 vector search results
2. Re-ranks them using cross-encoder model
3. Return top 10 most relevant items

Pros/cons

- Cross-encoder model performs deeper comparison at text level
- Better relevance on good models
- Requires GPU hardware to run the model

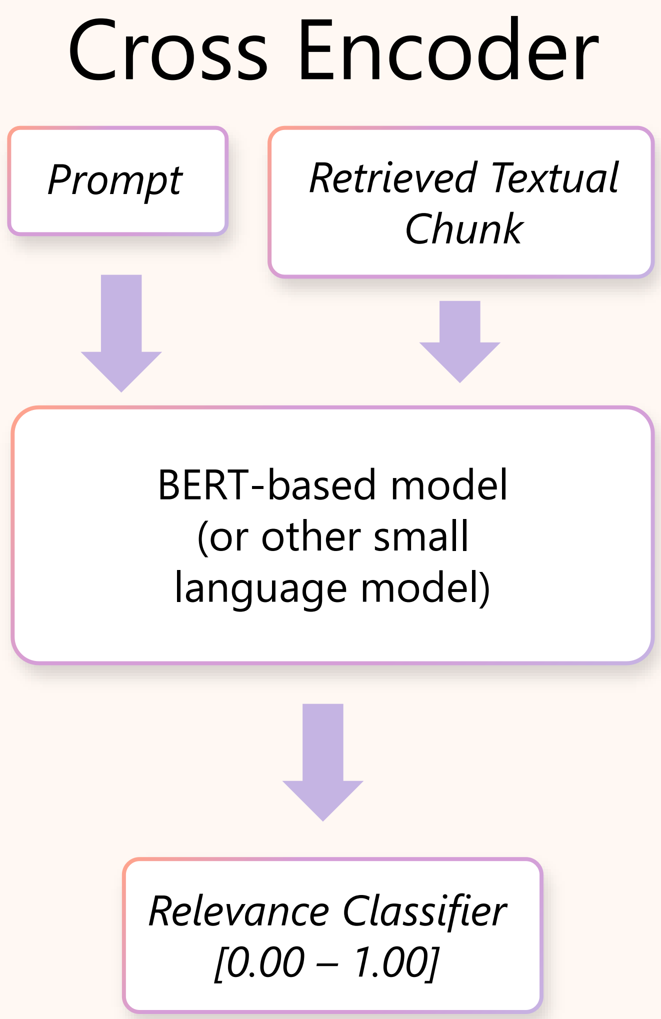
Semantic Re-Ranking

Cross Encoders

- **Process**:
  - A cross-encoder model (e.g., BERT, T5, or Cohere Rerank) compares each retrieved document with the query jointly, considering context from both before ranking.
- **Efficiency**:
  - Higher computational cost, as every document-query pair is encoded dynamically.
- **Example Models**:
  - BGE-reranker-v2-m3, MS MARCO-trained BERT Cross-Encoders, Cohere Rerank Models, T5-based Rerankers

2021 was a major year for efficiency improvements, making them more viable at scale.

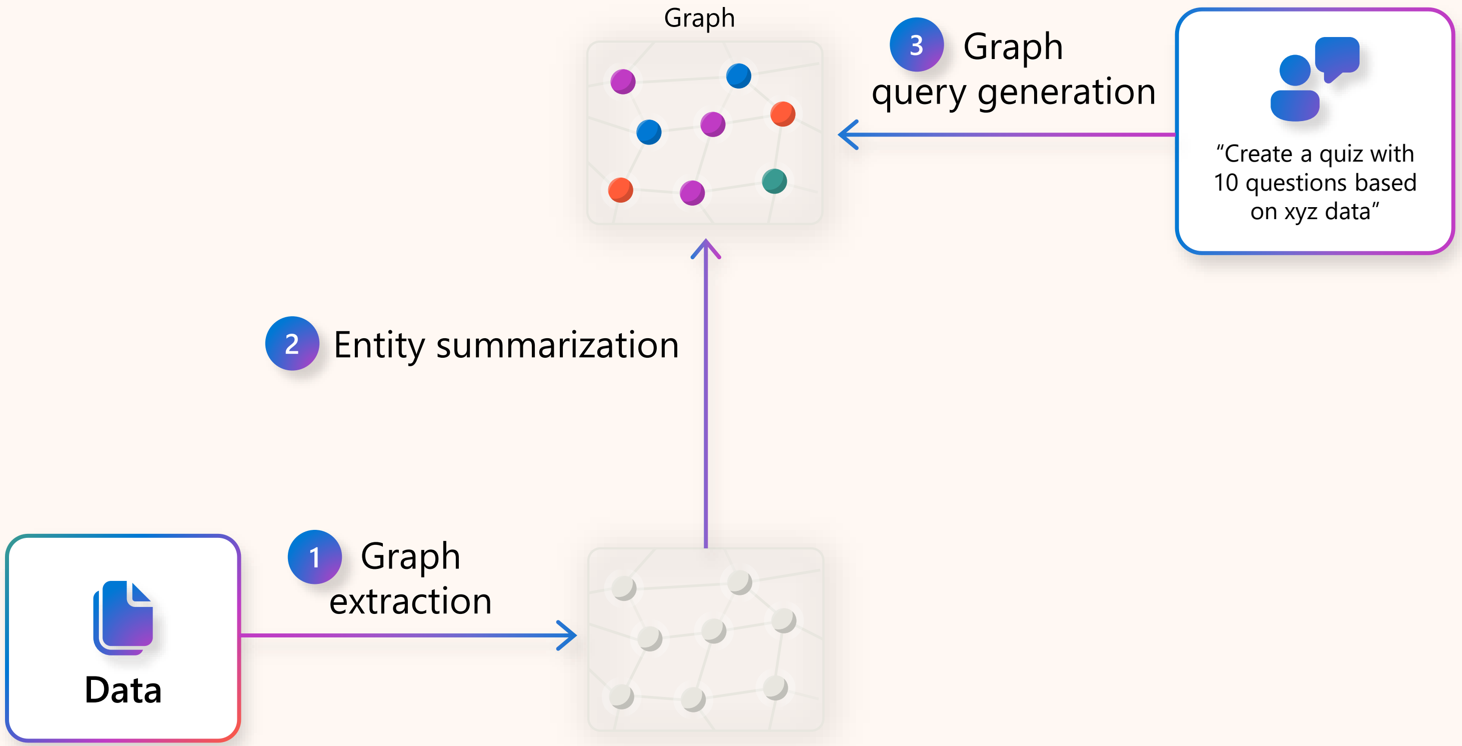
Key papers:
ColBERT (2020) – Khattab & Zaharia, MonoBERT & DuoBERT (2020) – MacAvaney et al., TAS-B (2021) – Hofstätter et al., ColBERTv2 (2021) – Santhanam et al.



GraphRAG – Option 1

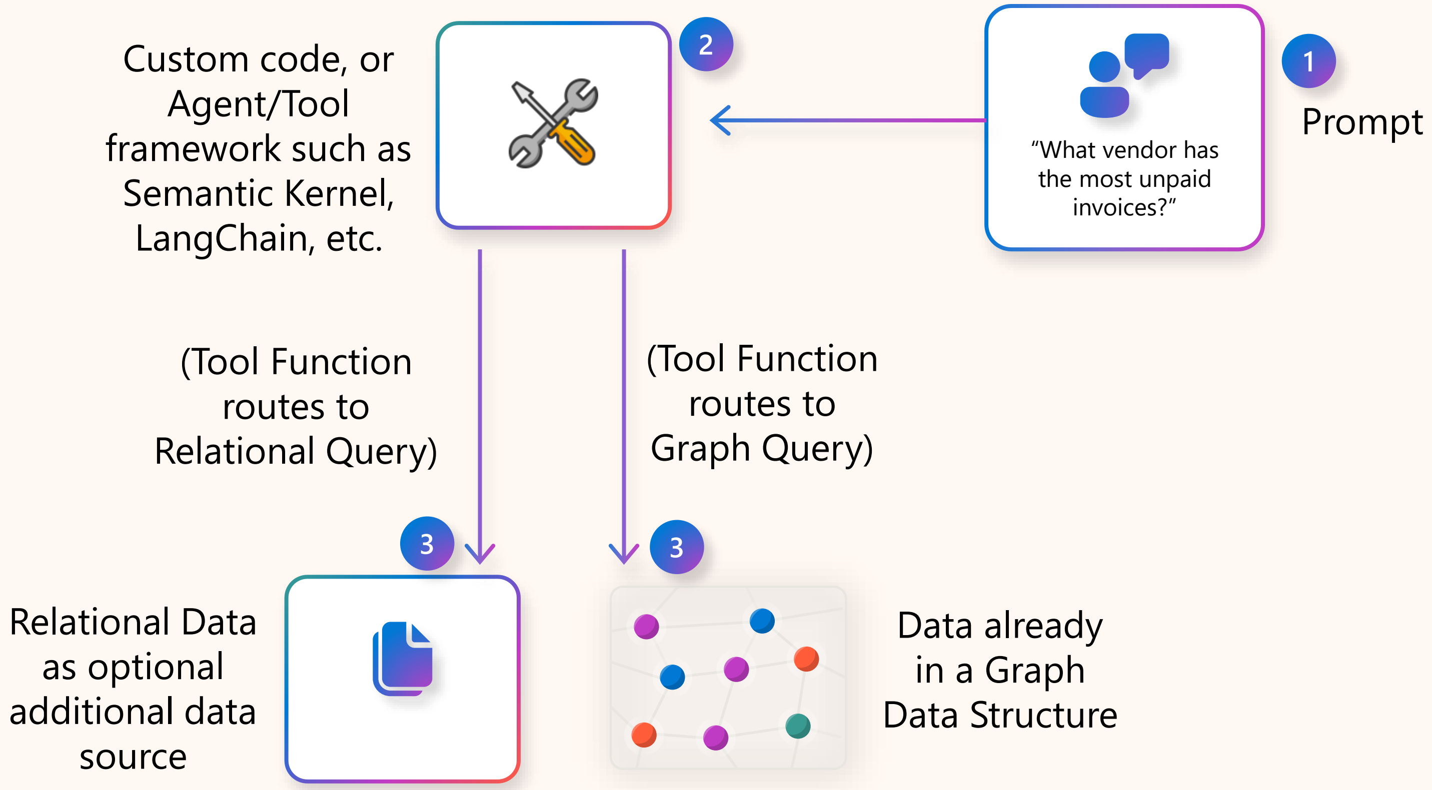
GraphRAG via Post-Processing Graph Construction

(Knowledge Graph Generation)



GraphRAG – Option 2

GraphRAG via Native Graph Data Querying



Related Sessions

BRK211	Develop Smarter Agentic Apps with PostgreSQL, GraphRAG & LangChain	**Date**: Tuesday, May 20 **Time**: 11:45 AM - 12:45 PM Pacific Daylight Time **Location**: Arch, 705 Pike, Level 4, Room 4C-4

GitHub Repo

https://github.com/jjfrost/pg-sk-agents-lab

Evals

https://aka.ms/build/evals

Thank you!