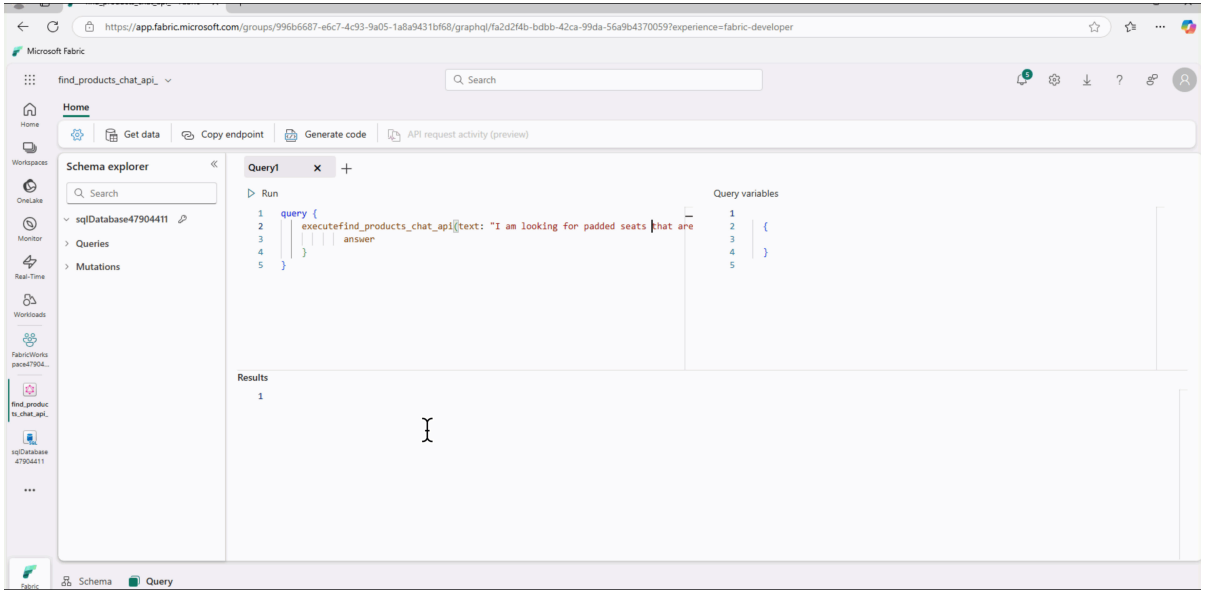
Build an AI App with SQL DB in Fabric

Today's lab will guide you through creating a set of GraphQL RAG application APIs that use relational data, Azure OpenAI, and SQL DB in Fabric.



Key Learning Objectives

- Query and work with data in SQL DB in Microsoft Fabric
- Build AI application design pattens/endpoints
- Use AI with relational data
- Understand how similarity searching benefits queries

Lab Overview

1. The Azure Fabric Portal and creating a SQL database in Fabric
2. Working with the SQL Database in Microsoft Fabric
3. REST in the database
4. Creating embeddings for relational data with Azure OpenAI
5. Create a set of GraphQL endpoints for AI Applications
6. Create a PowerBI report with Copilot
7. DevOps in Fabric (demo only)


The Azure Fabric Portal
and creating a SQL
database in Fabric

SQL Database in Fabric

Autonomous database built on the foundation of Azure SQL Database

Simple

Integrated with Visual Studio and GitHub
Developer optimized with GraphQL and CI/CD support
Unify OLAP + OLTP

Autonomous

Deploy in seconds
Auto-scaled compute and storage
Auto-managed HADR
Auto-optimized performance

Integrated

Integration with Azure AI
Integrated with Fabric services
Built-in vector and RAG support
AI-assisted

Secure by default

Powered by SQL

OneLake integrated

Beyond RDBMS

https://aka.ms/sqldev

- JSON and RegEx
- Spatial
- Graph
- Column store
- Memory-optimized
- Ledger
- Query Intelligence
- REST API integrated
- Data API Builder, GraphQL
- Azure Functions
- Hyperscale, Serverless, Pools
- Containers, DevOps, GitHub
- AI
- Fabric

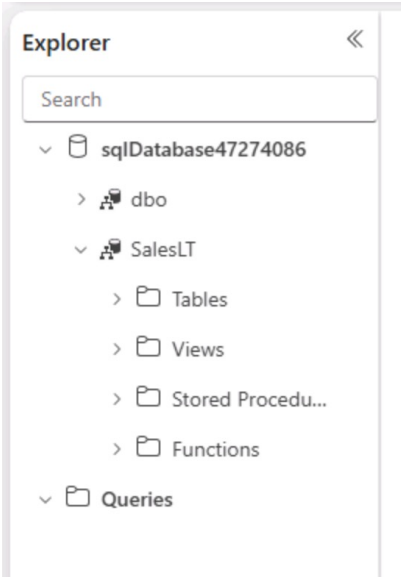
Chapter 1 Objectives

- Log into Microsoft Fabric
- Create a workspace
- Create a SQL DB
- Load the database with the Adventure Works sample data

Working with the SQL Database in Microsoft Fabric

SQL query editor in the Fabric portal

- View database schemas and objects via the database explorer
- Write and execute T-SQL queries
- Work with multiple editors/queries via tabs
- Use templates to jumpstart development
- Performance Monitor
- Save a query as a view
- Download results as Excel/JSON/CSV files
- Create GraphQL endpoints



Copilot for SQL database in Microsoft Fabric

- Natural Language to SQL
- Document-based Q&A
- Auto Code Generation
- Fix Query Errors
- Explain Query

Chapter 2 Objectives

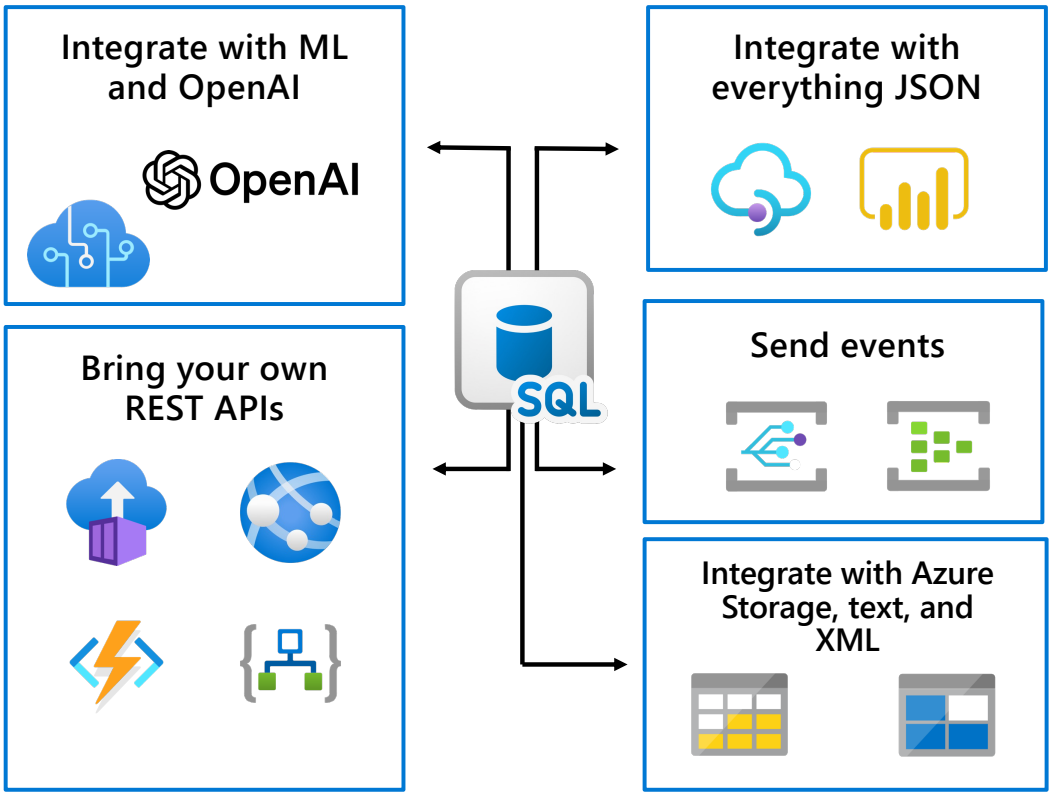
- Use the database explorer in Microsoft Fabric
- Run SQL statements in the SQL query editor
- Get answers with Copilot for SQL database in Microsoft Fabric

Note: Syntax highlighting may sometimes indicate errors, please ignore this.

Getting started with REST in the database


Event Hub
Microsoft Dynamics
PowerBI
Azure Containers
Azure Queue Storage
IoT Central
Microsoft Graph
Cognitive Services
Azure Blob Storage
SQL
API Management
Analysis Services
Event Grid
Azure App Services
Power Apps
Azure Table Storage
Logic Apps
Static Web Apps
Dataverse
Azure Functions
OpenAI
Azure Files
Microsoft Confidential

SQL Database and External REST endpoints



sp_invoke_external_rest_endpoint

EXECUTE ANY EXTERNAL ENDPOINT

External REST Endpoint Invocation (EREI)

sp_invoke_external_rest_endpoint

```
declare @url nvarchar(4000) = N'https://chat.openai.azure.com/';
declare @payload nvarchar(max) = N'{"messages":[{"role":"system","content":"XYZ"},
{"role":"system","content":"ABC"}]}'
declare @ret int, @response nvarchar(max);

exec @ret = sp_invoke_external_rest_endpoint
    @url = @url,
    @method = 'POST',
    @payload = @payload,
    @credential = [https://chat.openai.azure.com/],
    @timeout = 230,
    @response = @response output;

select @response;
```

Chapter 3 Objectives

- Create database scoped credentials
- Test the connection to Azure OpenAI from in the database
  - Chat completion
  - Content safety
  - AI Translator

Creating embeddings for relational data with Azure OpenAI

Azure OpenAI Service

Azure OpenAI Service provides REST API access to OpenAI's powerful language models.

Models	Description
o1-preview and o1-mini	Designed to tackle reasoning and problem-solving tasks with increased focus and capability (In Preview).
GPT-4o & GPT-4o mini & GPT-4 Turbo	The latest most capable Azure OpenAI models with multimodal versions, which can accept both text and images as input.
GPT-4o audio	A GPT-4o model that supports low-latency, "speech in, speech out" conversational interactions.
GPT-4	A set of models that improve on GPT-3.5 and can understand and generate natural language and code.
Embeddings	A set of models that can convert text into numerical vector form to facilitate text similarity.
DALL-E	A series of models that can generate original images from natural language.

REST and Inferencing Endpoints

Embeddings

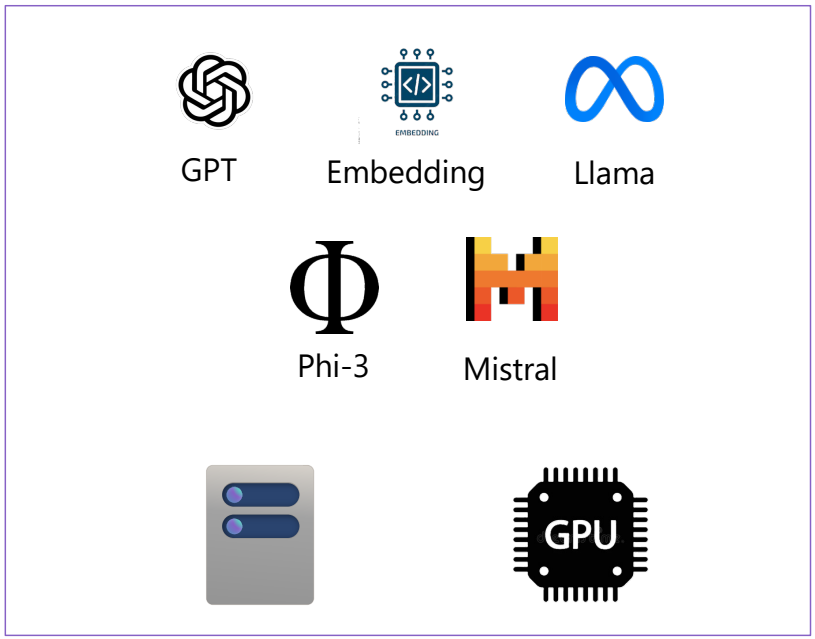
```
{ "input": [ "this is a test" ] }
```

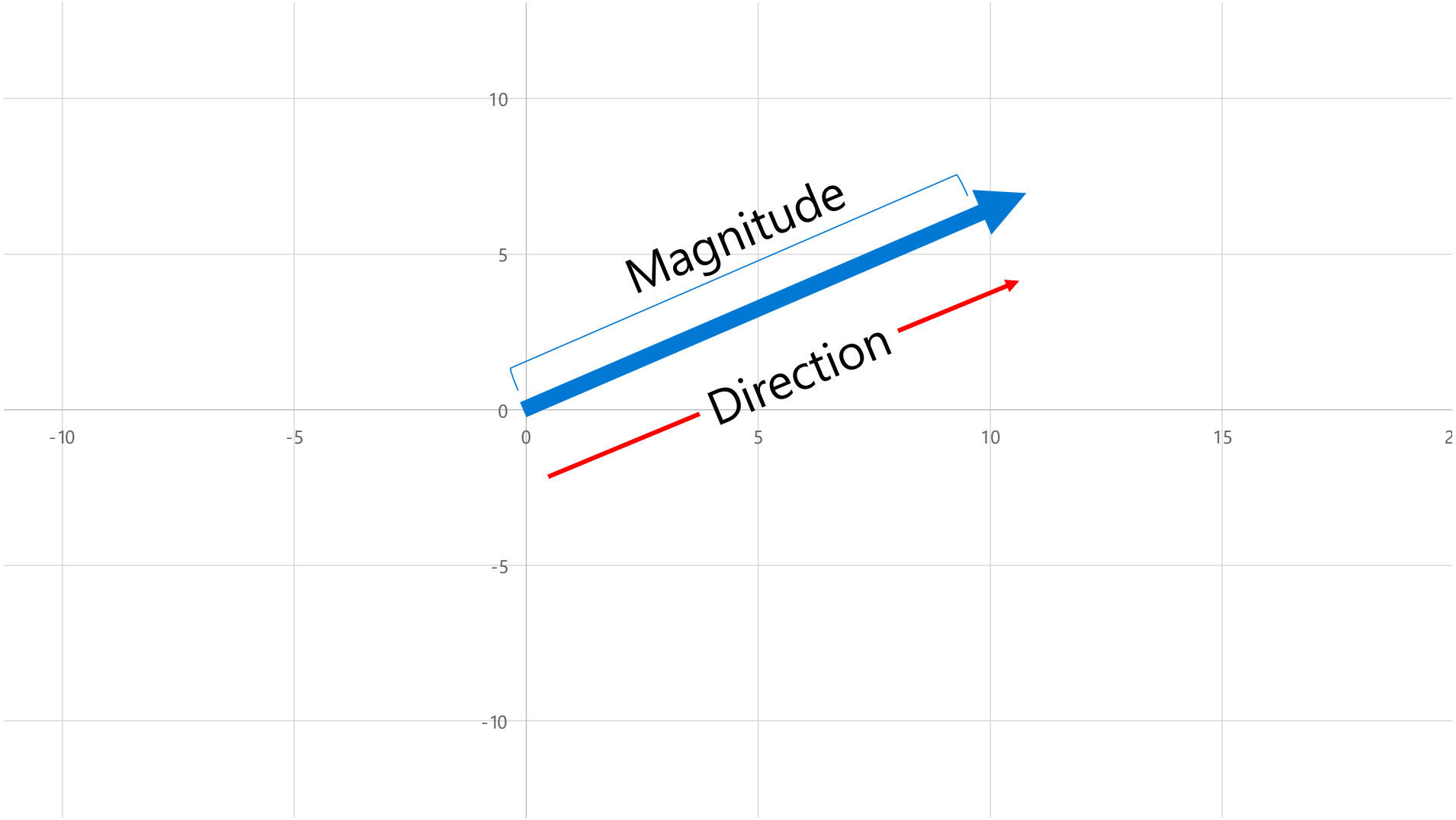
```
{ "body": { "data":
[ { "index": 0, "embedding": [ 0.13, 1.43, ... ] }}}-
```

Chat Completions

```
[
 {"role": "system", "content": "You are a helpful assistant."},
 {"role": "user", "content": "Who founded Microsoft?"}
]
```

```
[
 {"role": "assistant", "content": "Microsoft was founded by Bill Gates and Paul Allen."}
]
```


10
5
0
-5
-10
-10
-5
0
5
10
15
2
Magnitude
Direction

How many stars did each user give these 3 movies?

```
User 1: [5,3,3]
User 2: [1,3,4]
```

Vectors and Embeddings

What is vector embedding?

A **Vector Embedding** is a technique used in machine learning to transform data (text, images, etc.) into numerical representations (vectors) that capture the essential characteristics and relationships of the data.

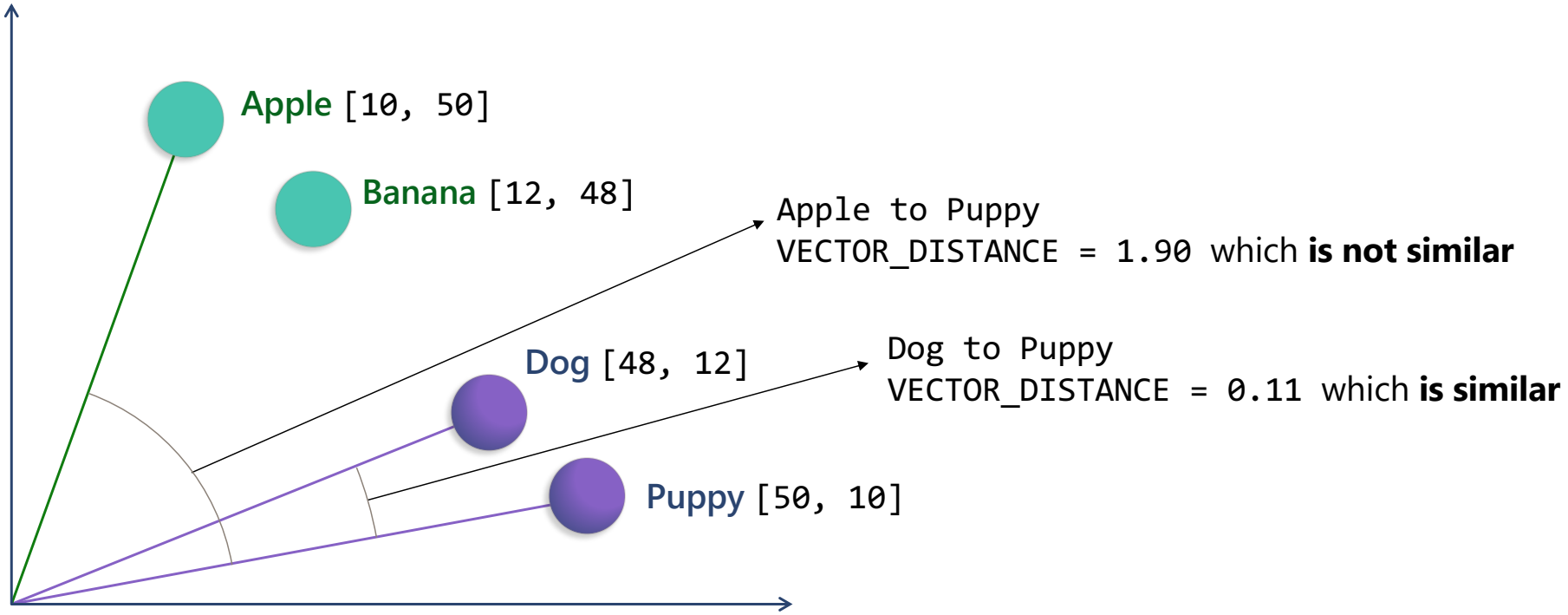
Each number is referred to as a dimension.

```
dog: [0.9, 0.3, 0.2, ...]

puppy: [0.88, 0.33, 0.21, ...]

Image of a dog: [0.9, 0.3, 0.2, ...]

"I took the dog for a walk": [1.5, -0.8, 2.1, ...]
```

New in SQL Database in Fabric

- Vector Datatype
  - `col_name VECTOR(int)`
    (int being the dimension count)
- Vector Functions
  - VECTOR_DISTANCE
  - VECTOR_NORM
  - VECTOR_NORMALIZE

Similarity Searching using VECTOR_DISTANCE (cosine)



Similarity Searches vs Text Only Searches

“I’m looking to buy a comfy red sofa”

Text Search Results

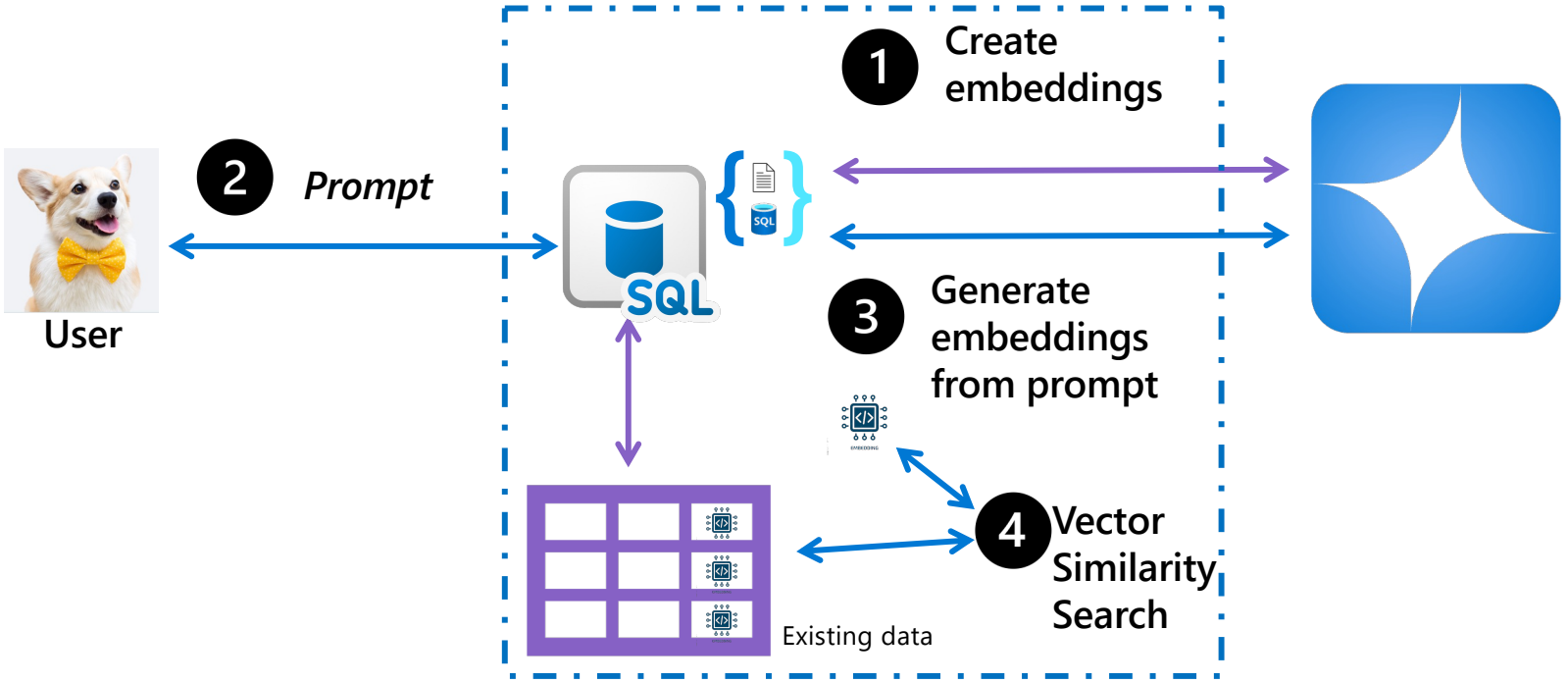
Product Name
Red Sofa
Comfy Sofa - Red

Similarity Search Results

Product Name
Red Sofa
Comfy Sofa - Red
Soft maroon couch
Sleepy time crimson daybed

Use T-SQL for Vector and Hybrid Search

Connected to AI



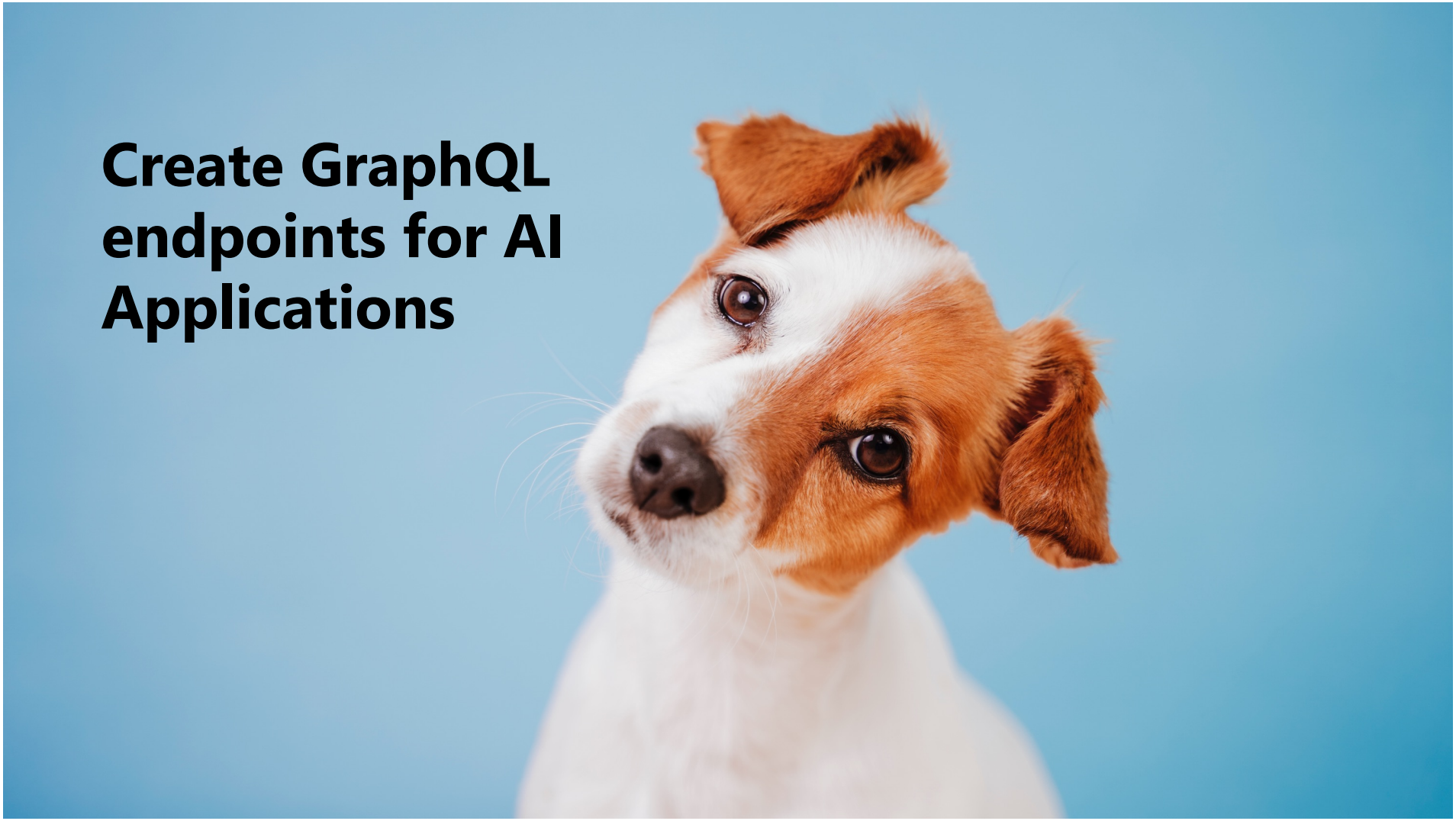
How do I handle changes to the data?

- Database Change Tracking/Change Data Capture
- SQL Trigger Binding for Azure Functions
- Batch Jobs
- Event Architectures

Chapter 4 Objectives

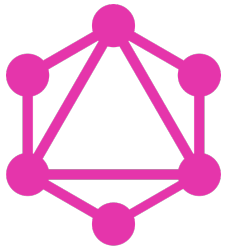
- Add columns to the Products table for embeddings
- Create a stored procedure to embed table data
- Create embeddings for the Products table
- Use the new VECTOR_DISTANCE function for similarity searches

Create GraphQL endpoints for AI Applications



What is GraphQL

Ask for what you need, get exactly that



Clients specify exactly what data they need, and the server returns only that data.

```
{
  user(id: "1") {
    name
    email
  }
}
```

```
{
  "data": {
    "user": {
      "name": "Frank",
      "email": "frank@frank.com"
    }
  }
}
```

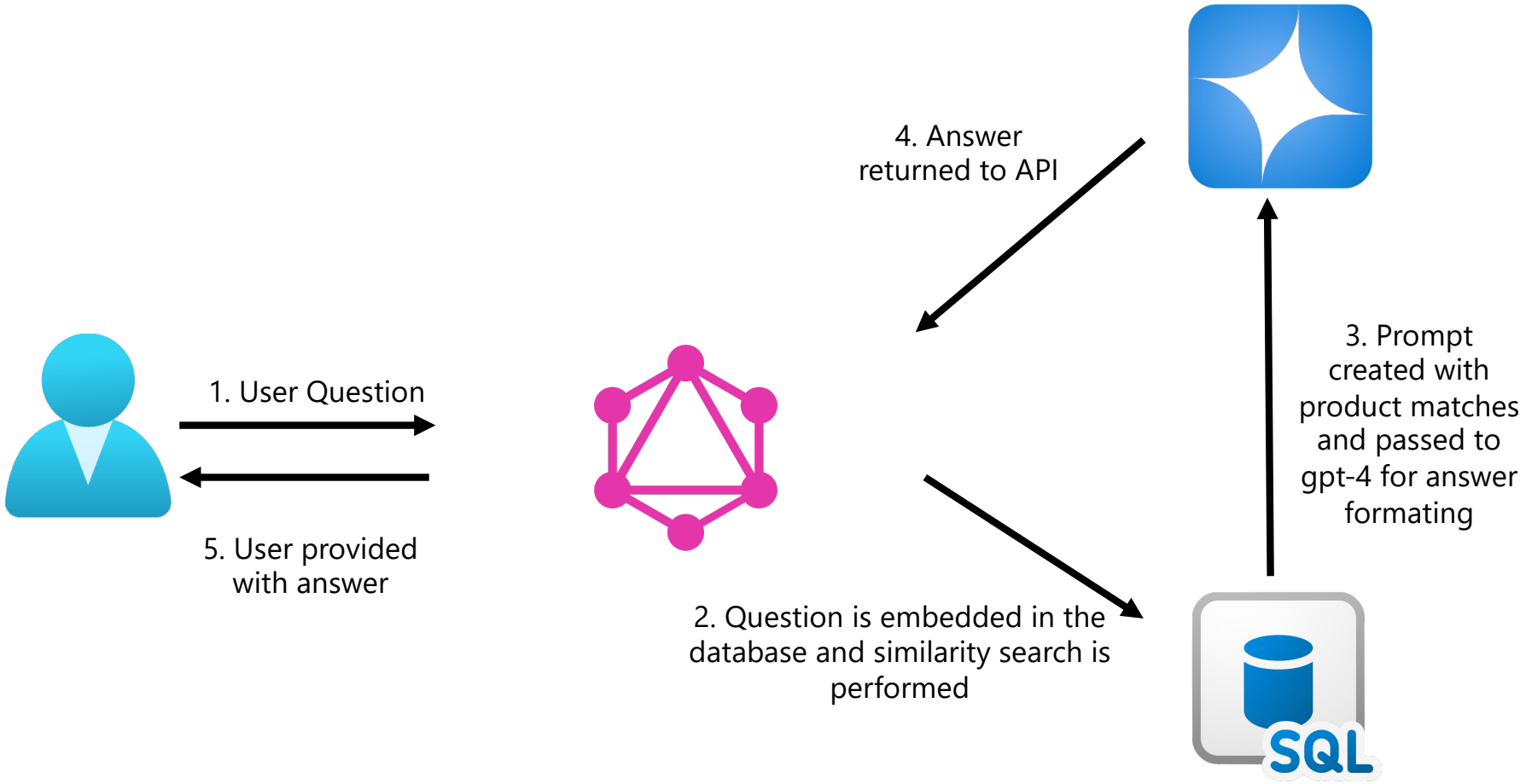
Retrieval-Augmented Generation and AI

Retrieval-Augmented Generation (RAG) is a significant advancement in AI because it enhances the capabilities of large language models (LLMs) by integrating external knowledge sources such as relational databases.

Key points:

- Improved Accuracy and Relevance
- Cost-Effective
- Greater Control for Developers

A GraphQL API using RAG and SQL



Chapter 5 Objectives

- Create a stored procedure for a RAG application
- Create a GraphQL API in Fabric and test it
- Add chat completion to our RAG stored procedure
- Create a GraphQL API in Fabric to expose the new chat function

Creating a PowerBI Report from a SQL Database in Microsoft Fabric with Copilot

Copilot for Power BI

Copilot for Power BI can help you:

- Get started on a new report by suggesting topics based on your data.
- Use the measures and columns in your semantic model to help you explore your data
- Builds a visual for you if the answer can’t be found in the report

Chapter 6 Objectives

- Add the SQL Database to a semantic model
- Ask Copilot for report suggestions
- Explore the PowerBI dashboard Copilot created from a SQL database

Azure DevOps and the SQL Database in Microsoft Fabric

Microsoft Fabric Git integration

Azure DevOps or GitHub

- Backup and version work
- Revert to previous stages as needed
- Collaborate with others or work alone using Git branches
- Apply the capabilities of familiar source control tools to manage Fabric items

Supported Services:

Data pipelines (preview)

Dataflows gen2 (preview)

Eventhouse and KQL database (preview)

EventStream (preview)

Lakehouse (preview)

Mirrored database (preview)

Notebooks

Paginated reports (preview)

Reflex (preview)

Reports Semantic models (preview)

Spark Job Definitions (preview)

Spark environment (preview)

SQL database (preview)

Warehouses (preview)