# SQL database in Microsoft Fabric

# Series Scenario – Conference session application

You need:

- to store and serve data for the conference webpage.
- to be able to run analysis over the data.
- an application that allows attendees to easily search for and find sessions of interest to them.
- to follow modern best practices for application lifecycle.
- to be able to monitor and troubleshoot database and query performance.

# Course Overview and agenda

Episode 1: Introduction and Overview; Getting started

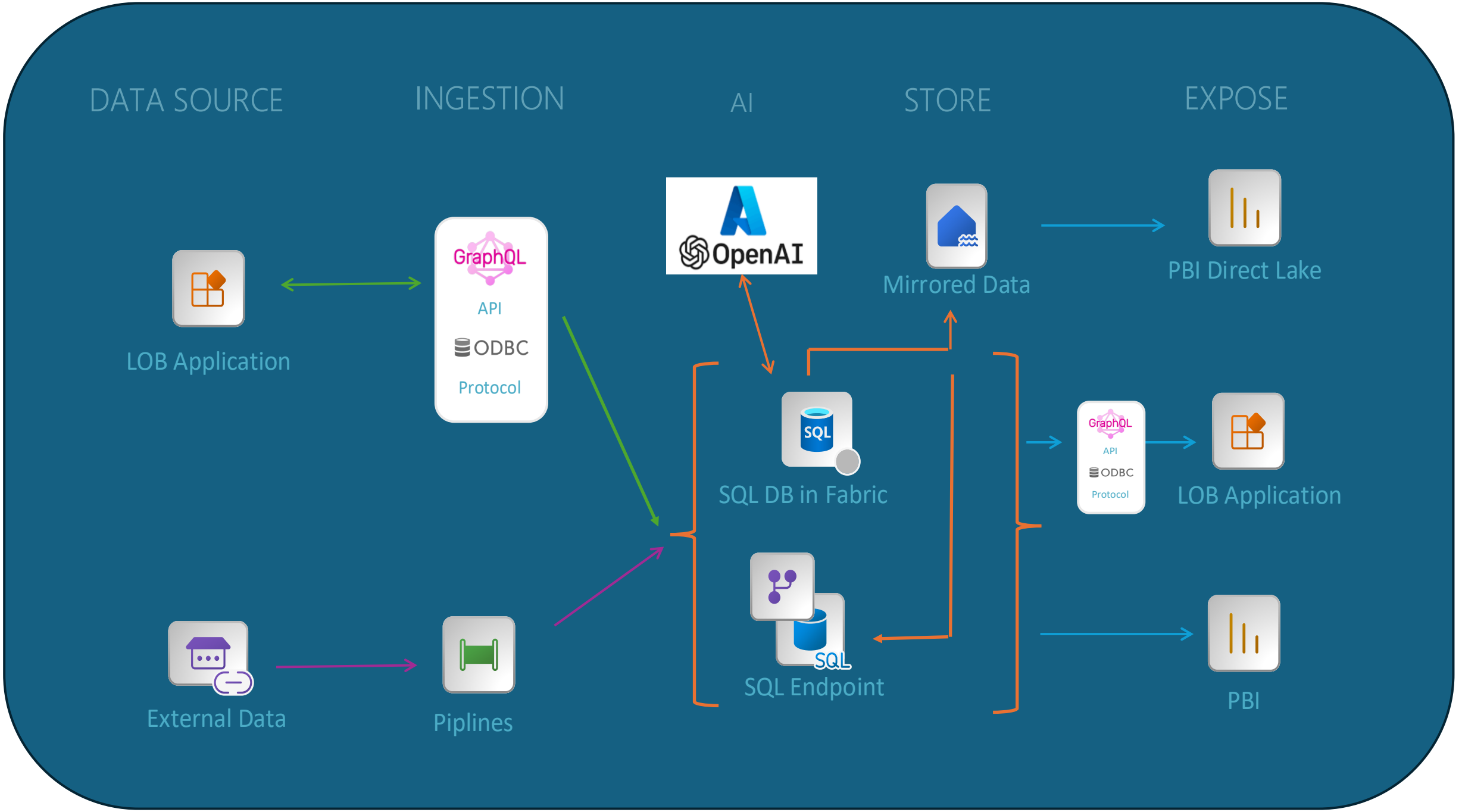
Episode 2: Dataflows, Notebooks, Reports

Episode 3: GenAI and vector databases

Episode 4: GraphQL and application development

Episode 5: Application lifecycle management

Episode 6: Performance Dashboard, Recap



# Episode 6: Performance Dashboard, Recap

# SQL db in Fabric comes with intelligence

How can I ensure that things are going smoothly?

**Look at database size**

See auto scaling in action

**Alerts**

Ongoing

Retrospective

**Identify queries of Interest**

High IO queries

High CPU queries

**Observe auto-indexing activity**

# Developer-friendly tuning

**Instead of enabling or using knobs....**

| Compare | Monitor | Track | Troubleshoot | Observe |
|---|---|---|---|---|
| Query performance details include comparison to previous executions | Quickly identify if query execution count has increased unexpectedly. | Track specific queries by creating favorites | Look for high-concurrency issues by finding blocking queries | Auto index behavior<br>Alerts |

# Alerts

- Blocking queries – any blocking queries will trigger an alert
- CPU Usage – if CPU usage exceeds 80% of Vcores for 5 minutes or more
- Allocated Size – If size exceeds 80% of allocated size for 5 minutes or more

# Timing of alerts

Active alerts are ongoing in the database, alerts that happened in the past 24 hours will also appear.

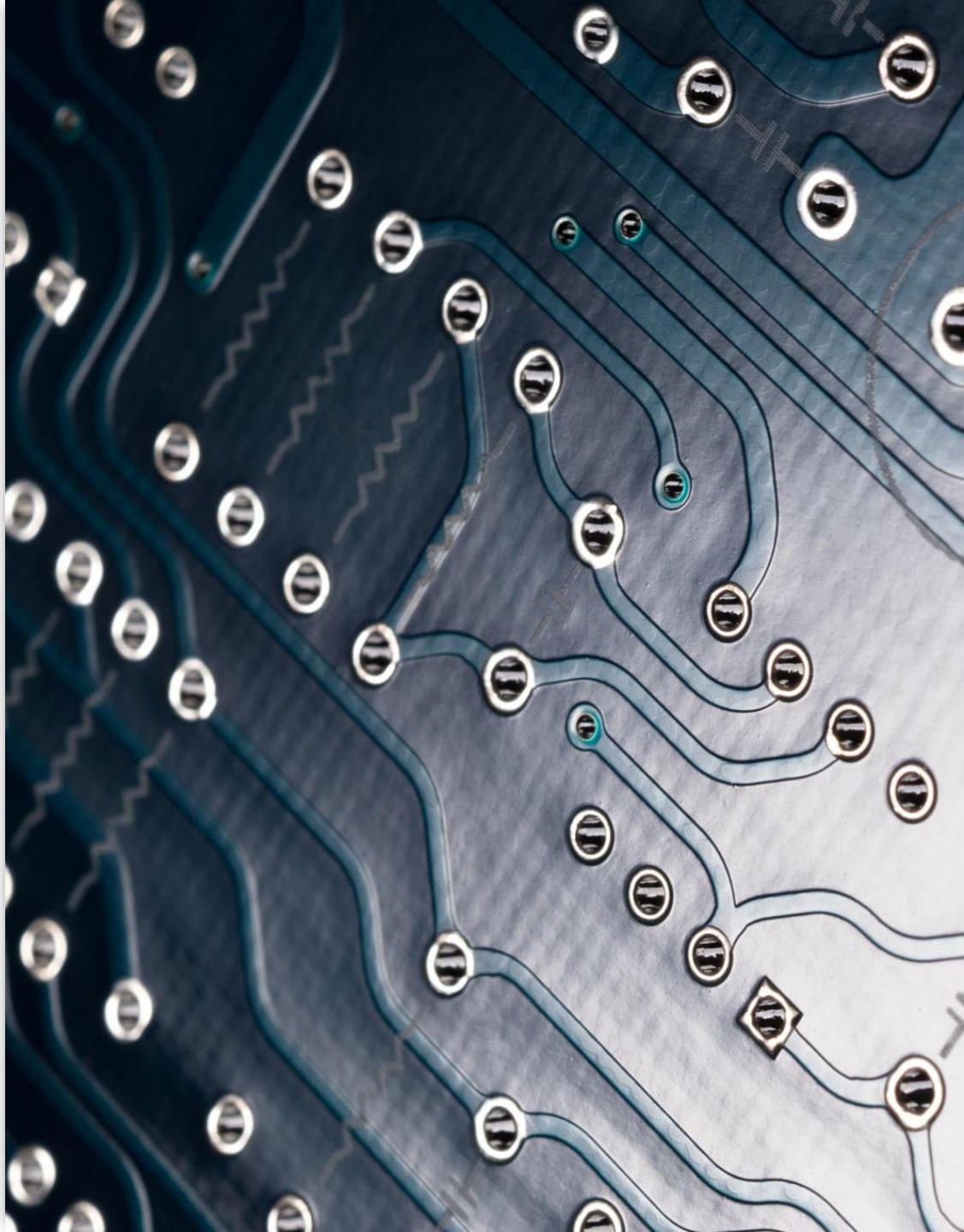
Alerts will display in performance summary tab for 24 hours, and if resolved, will go away.

# Performance Dashboard

Entry Point, Alerts

# CPU Consumption

- This shows the developer what the current CPU consumption is, and a historical chart of CPU consumption over the selected time interval.
- There is list at the bottom of the pane showing queries with the top CPU consuming queries at the top.
- **Developer insight:**
  - Should I write or re-write queries to be less CPU intensive?
  - If I am using more CPU than expected, how might I need to change my application?



# Query details

Number of executions of the query over time

Ability to compare query performance over different time intervals

Has it changed recently in frequency or performance?

Query text

- Can copy or open in different editors with a click of a button

# Query details

All data is obtained from Query Store

**Developer Insight:**

- Is my query performing as expected?
- Has it changed recently in frequency or performance?

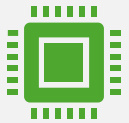
# Queries

- Lists queries across multiple interesting dimesons:
  - High CPU queries
  - Longest Running queries
  - Most Frequent queries
  - High Read queries

# Queries

Can all be sorted by total or average.

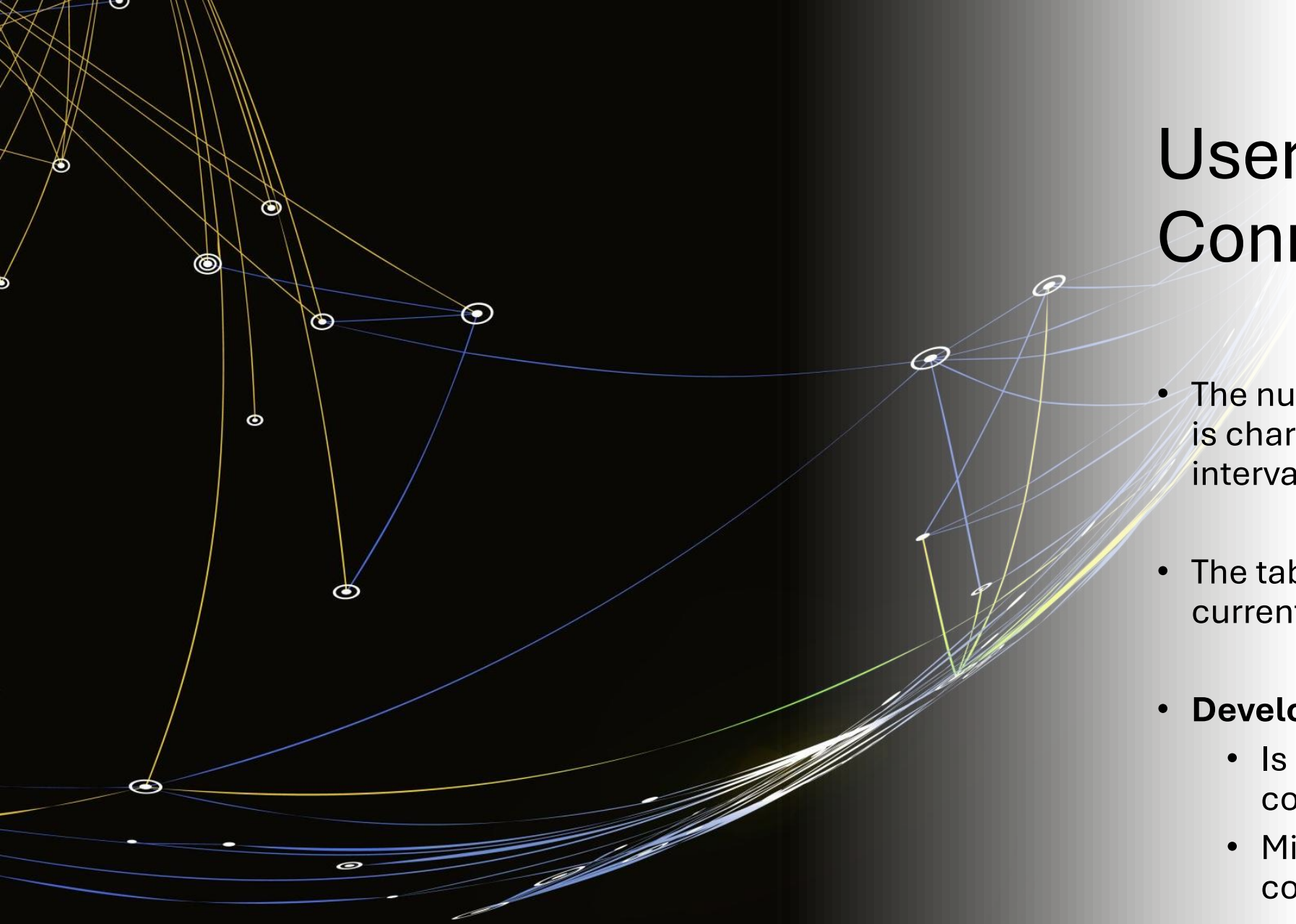
Can filter to only starred queries – allowing you to track specific queries.



**Developer insight:** Provides and overall view of various queries, and how they are working within the system.

# Performance Dashboard

CPU, Query details, Queries



# User Connections

- The number of total connections is charted over the given time interval.
- The table at the bottom list the current connections.
- **Developer Insight:**
  - Is application creating more connections than expected
  - Might application be leaking connections.

Tracks the cumulative number of times a query has been executed over a period.

Table contains the most frequently executed queries.

**Developer insight**: If my query is executing more than expected, perhaps I need to update my code.

# Requests per second

# Blocked queries per second

Tracks queries that are experiencing blocks due to locking.

The chart shows the number of blocked queries per second.

The table shows which query is blocking – and all queries blocked by that query.

A blocked query *always* triggers an alert.

**Developer insight:** How might I rewrite or break up my query to prevent blocking?

- NOLOCK is not the answer.

# Allocated size

- Shows history of database size over selected time interval.
- Table shows largest database tables (current).
- **Developer insight:**
  - Do I have tables that are growing larger than expected?
  - What tables could I change up to make queries across them more performant?

# Automatic Indexes

Using SQL's built-in automatic index management.

Identifies possible indexes

- Will create indexes, and monitor performance over time.
- Will  revert an index if it doesn't help in the short term.
- May drop those indexes if they stop improving performance of the workload.

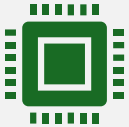
Never drops user created indexes

# Automatic Indexes

When considering source control integration:

Created indexes can become user created indexes automatically

They become seamlessly integrated in deployment pipelines.



**Developer insight:**

Which indexes might I wish to create or experiment in future deployments?

# Performance Dashboard

Remaining Tabs

# Recap

# Recap: Series in review

Episode 1: Introduction and Overview; Getting started

Episode 2: Dataflows, Notebooks, Reports

Episode 3: GenAI and vector databases

Episode 4: GraphQL and application development

Episode 5: Application lifecycle management

Episode 6: Performance Dashboard, Recap

# Episode 1

## About SQL database in Microsoft Fabric

## Getting started

- Creating initial objects and your first SQL database in Microsoft Fabric

# Episode 2

## Integrations within Fabric ecosystem

## Learning about Pipelines and Dataflows

- Importing data

## Learning about notebooks

- Creating a sample notebook connected to the analytics endpoint

## Learning about reports

- Creating a sample report

# Episode 3

## Why AI should integrate with SQL database

## About Open AI

- Deploying and connecting to an Open AI instance

## About vector databases

- Using vector support in SQL to generate embeddings

# Episode 4

Create stored procedure to run application

About GraphQL

- Create wrapper for stored procedure
- Using GraphQL API builder to create a simple application

# Episode 5

Overview of application lifecycle management

Fabric support for databases in source control

- Connect your database to source control.  Update and push changes.

About deployment pipelines

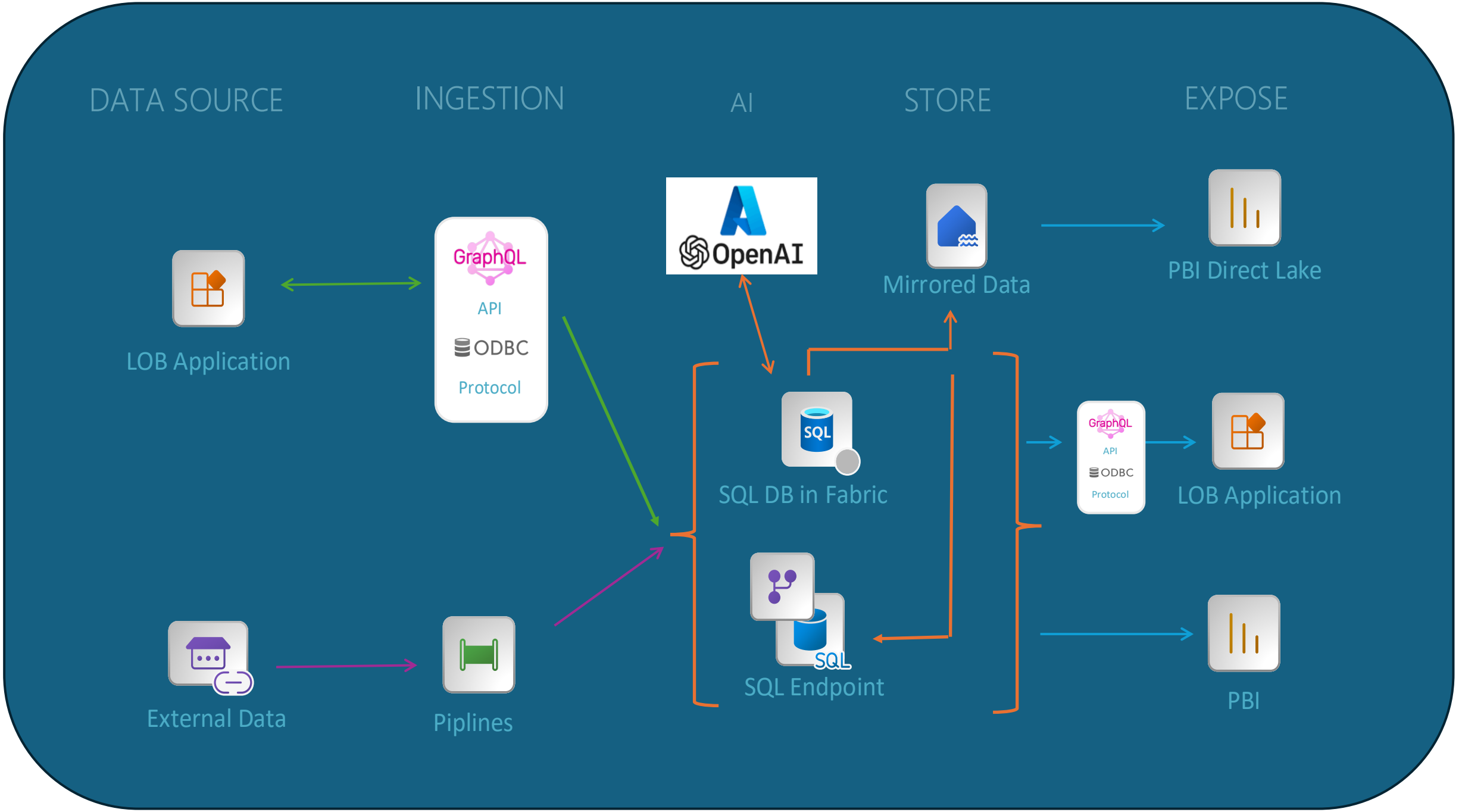
- Create an initial deployment pipeline

# Episode 6

## Overview of Performance Dashboard

- Developer insights for each tab

## Recap of series content



# End Episode 6

Performance Dashboard, Recap