



Microsoft Fabric

This presentation is the property of Microsoft and is intended for informational and educational purposes only. You may use, copy, and distribute this presentation for your personal, non-commercial purposes. You may not modify, alter, or create derivative works from this presentation without the prior written consent of Microsoft. You may not use this presentation to misrepresent, defame, or disparage Microsoft or its products, services, or affiliates. You may not use this presentation to endorse or promote any other products, services, or organizations without the prior written consent of Microsoft.

By using this presentation, you agree to abide by these terms. If you do not agree, you must not use this presentation. Microsoft reserves the right to change these terms and conditions at any time without notice. Microsoft disclaims any and all warranties, express or implied, relating to this presentation, including but not limited to the accuracy, completeness, timeliness, or suitability of the information contained herein. Microsoft is not liable for any damages, losses, or liabilities arising from your use of or reliance on this presentation.

Please review the terms of use posted in the content library.

#FABCONSQLCON2026

**FABCON**

**Microsoft Fabric**  
COMMUNITY CONFERENCE

**SQLCON**

**Microsoft SQL**  
COMMUNITY CONFERENCE

**ATLANTA** MARCH 16 - 20, 2026



# New in modern Developer Experiences in SQL

**Davide Mauri**

Principal PM Architect – Microsoft SQL

# Agenda

JSON

Regular Expression

String Functions

AI & Vectors

Change Event Stream

Query Optimizer Improvements

Drivers

Data API Builder: REST & GraphQL

# Let's start from a practical problem

- You want to build an AI-powered search solution for your database (vectors + full text)
- Semantic Search doesn't return you the high-quality result you would expect, therefore, you need to implement Semantic Re-Ranking
- Semantic Re-Ranking: “Rerank models sort text inputs by semantic relevance to a specified query. They are often used to sort search results returned from an existing search solution.”
  - <https://docs.cohere.com/v2/docs/rerank>

# Let's start from a practical problem

- You need to keep your data as secure as possible
  - So, the less you move your data through applications layers, the better
- You want to keep the solution as scalable as possible
  - Once again, the less you move your data through applications layers, the better
- You want to minimize the cost of this improvement
  - The simpler the solution the better, which means...
  - ...the less you move your data through applications layers, the better

# Let's start from a practical problem

- Can you do everything from SQL?
- Call the Cohere Semantic Reranking model
- Build the process the JSON Payload
  - Cohere asks for YAML data in JSON payload 🤔
- Extract the re-ranked result and correlate it to data in your database
  - To enrich the result set with additional details

# DEMO

Modern Problems, Modern Solutions

<https://github.com/Azure-Samples/azure-sql-db-vector-search/tree/main/DiskANN/Wikipedia>

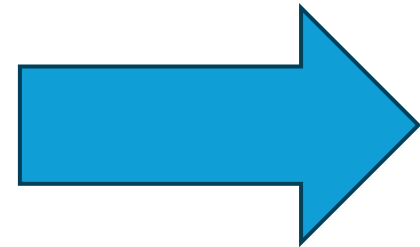
# A brief history of JSON in SQL Server

# JSON support - SQL Server 2016

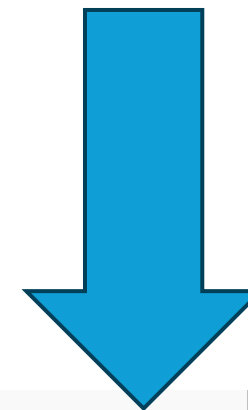
- JSON documents stored in character columns (NVARCHAR)
- JSON Functions
  - ISJSON
  - JSON\_VALUE
  - JSON\_QUERY
  - JSON\_MODIFY
- Relational Operators
  - OPENJSON -> Converts JSON document into a set of rows and columns
  - FOR JSON -> Make a query result set into a JSON document

# Relational Operators

```
[  
{  
  "Number": "S043659",  
  "Date": "2025-05-31T15:34:02",  
  "AccountNumner": "AW29825",  
  "Price": 59.99,  
  "Quantity": 1  
},  
{  
  "Number": "S043661",  
  "Date": "2025-06-01T08:13:47",  
  "AccountNumner": "AW73565",  
  "Price": 24.99,  
  "Quantity": 3  
}
```



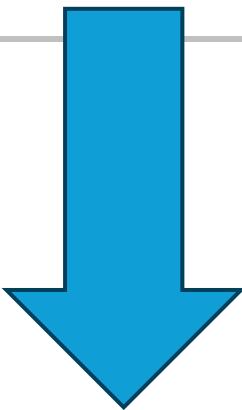
```
select * from openjson(@j) with (  
  [Number] varchar(10),  
  [Date] datetime2,  
  [AccountNumber] varchar(10),  
  [Price] decimal(9,4),  
  [Quantity] int  
)
```



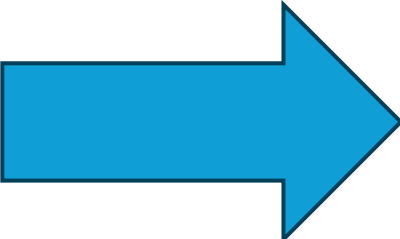
	Number	Date	AccountNumber	Price	Quantity
1	S043659	2025-05-31 15:34:02.00000000	AW29825	59.9900	1
2	S043661	2025-06-01 08:13:47.00000000	AW73565	24.9900	3

# Relational Operators

	Number	Date	AccountNumber	Price	Quantity
1	S043659	2025-05-31 15:34:02.0000000	AW29825	59.9900	1
2	S043661	2025-06-01 08:13:47.0000000	AW73565	24.9900	3



```
select
  *
from
  #t
for json auto
```



```
[
  {
    "Number": "S043659",
    "Date": "2025-05-31T15:34:02",
    "AccountNumber": "AW29825",
    "Price": 59.99,
    "Quantity": 1
  },
  {
    "Number": "S043661",
    "Date": "2025-06-01T08:13:47",
    "AccountNumber": "AW73565",
    "Price": 24.99,
    "Quantity": 3
  }
]
```

# JSON support - SQL Server 2022

- ISJSON enhancement – RFC 8259 support
- JSON functions
  - JSON\_PATH\_EXISTS
- ANSI SQL JSON constructors
  - JSON\_OBJECT
  - JSON\_ARRAY

# JSON\_OBJECT + JSON\_ARRAY

```
declare @emailValue nvarchar(100) = 'jd@acme.com';
declare @emailKey nvarchar(100) = 'email';

declare @j json = json_object(
    'firstName': 'John',
    'lastName': 'Doe',
    'roles': json_array('reader', 'writer'),
    'contacts': json_object(
        @emailKey: @emailValue,
        'phone': '123 456 789'
    )
)
```

# JSON\_OBJECT + JSON\_ARRAY

```
{  
  "firstName": "John",  
  "lastName": "Doe",  
  "roles": ["reader", "writer"],  
  "contacts": { "email": "jd@acme.com", "phone": "123 456 789" }  
}
```

# What's new in Azure SQL and SQL Server 2025

# JSON support - Azure SQL DB & MI

- **Native JSON type** 🥰
- ANSI SQL JSON aggregates
  - JSON\_OBJECTAGG,
  - JSON\_ARRAYAGG
- JSON Aggregates - GROUPING SETS, OVER support
- JSON\_VALUE enhancement
  - RETURNING data\_type – Convert JSON text in a SQL/JSON path to a SQL type (deployment in progress...)

# JSON support - Azure SQL DB & MI

- Path Expressions 🥰
  - Array wildcard searches with ARRAY WRAPPER
  - Supported in
    - OPENJSON, JSON\_VALUE, JSON\_QUERY, JSON\_MODIFY, JSON\_CONTAINS
  - Allows you to specify
    - all elements: `$.phoneNumbers[*]`
    - range of elements: `$.phoneNumbers[0 to 1]`
    - list of elements: `$.phoneNumbers[0,2]`
    - the "last" element: `$.phoneNumbers[last]`

# JSON support - SQL Server 2025

- SQL Server 2025 introduce the new JSON data type
  - JSON documents stored in optimized binary format
  - Parsed once upon initialization or insert
  - Uses existing LOB capabilities in the storage engine
  - Supports in-place modification of values or keys
  - Support for JSON INDEX 🥰 🥰 🥰
- Supports documents up to ~2 GB (binary format)
  - PostgreSQL – jsonb size is limited to 255MB
  - Oracle – json size is limited to 32MB

# JSON support - SQL Server 2025

- JSON\_CONTAINS function

- Searches for a SQL value in a path in a JSON document
- Supports JSON Path expression

👉 Best practice is to use JSON\_CONTAINS as provides better performance with index

- Assumes that \*all\* values of a property are of the same type
- This means we can do better optimization => faster execution

# JSON Support - Drivers

- Microsoft.Data.SqlClient  $\geq$  6.0.2
  - SqlUson Class
  - SqlDataReader.GetSqlUson
- JDBC 13.1.1
  - <https://github.com/microsoft/mssql-jdbc/pull/2558>
- As of today JSON streamed as an UTF-8 string and converted on the server side

# JSON Type vs Legacy JSON (n/var/char)

Native JSON Type	Legacy JSON
JSON document parsed into optimized binary format	JSON document is stored “as is” including whitespaces. Engine only sees strings.
JSON values (numbers, strings, boolean) are stored as SQL types	JSON types are string tokens always & interpreted at runtime
Strict parser: JSON document is validated upon initialization	Lazy Parser: JSON document is validated as it is parsed
Optimized for fast query processing i.e., lookup keys quickly, locate array elements	No optimization. JSON document requires processing/parsing for every operation
In-place modifications of JSON values or keys	Modification of single value requires rewriting the entire JSON document
Storage savings for larger document Ex: Repeated keys, spurious whitespaces, formatting characters	No storage savings unless compression is used or columnstore

# JSON index - SQL Server 2025

- SQL/JSON path(s) that point to scalar or arrays or objects
- Use for covering queries that touch many SQL/JSON paths
- Index will be bigger in size due to storing values at node level
- Arrays are always expanded
- Index maintenance can be expensive
- Azure SQL DB coming soon!

# DEMO

JSON data & JSON functions

JSON Index

# JSON – Question for the audience

JSON Schema Enforcement?

If yes, which format?

JSON Schema - <https://json-schema.org/>

Old? Last update of the schema only in 2020

JSON Type Definition

Being standardized: <https://www.ietf.org/rfc/rfc8927.html>

## Some T-SQL Love



**RegEx**  
functions

**PRODUCT()**

**BASE64**  
functions

**Fuzzy string match**  
functions

**CURRENT\_DATE**  
returns DATE

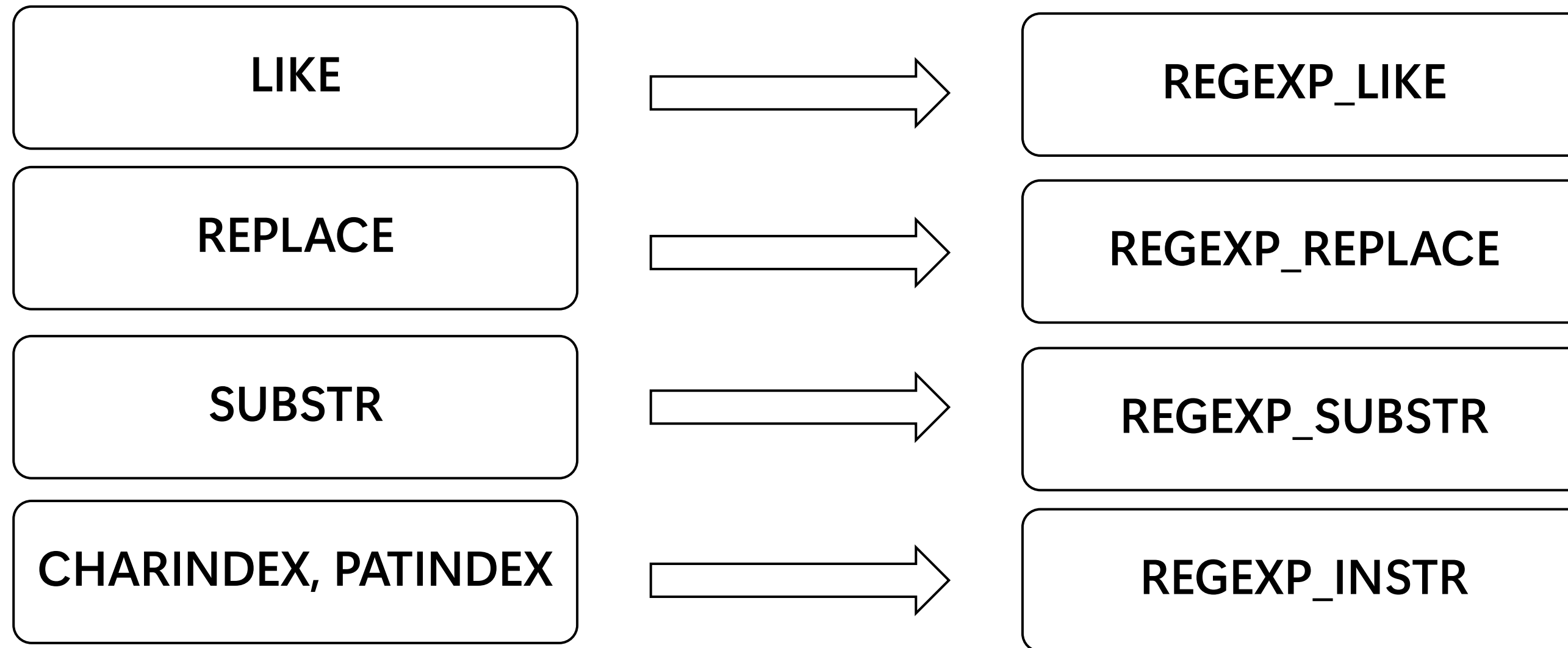
**SUBSTRING()**  
optional length

**UNISTR()**

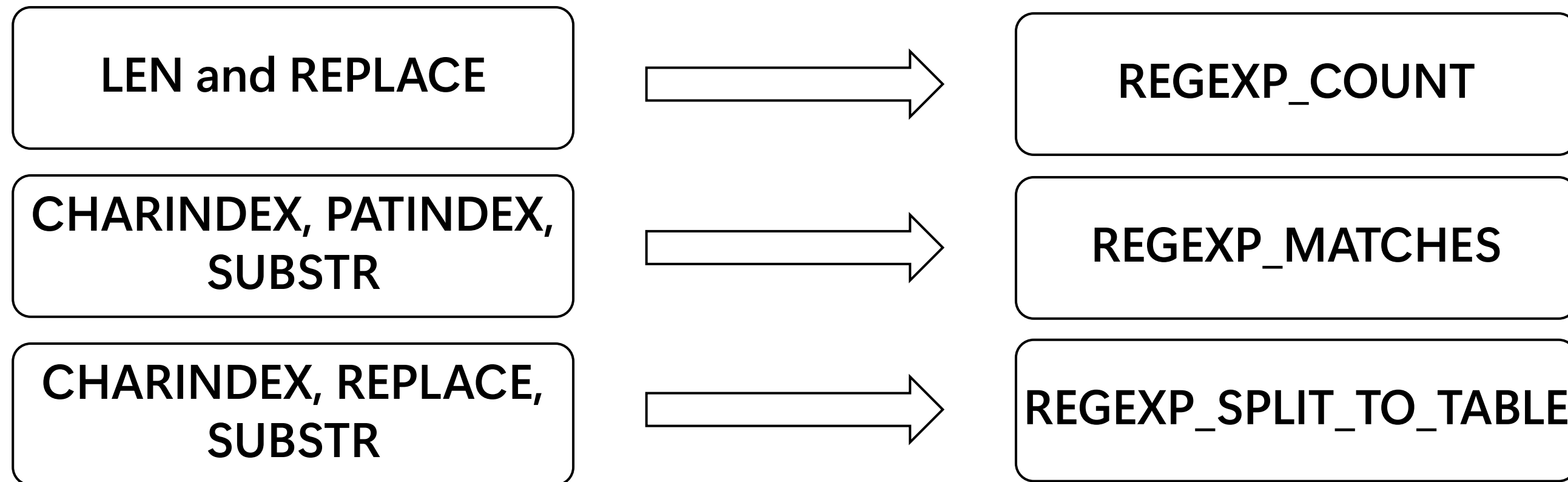
**||**  
string concat  
operator

**DATEADD()**  
supports bigint

# The T-SQL RegEx library



# The T-SQL RegEx library



# DEMO

RegEx

# String Functions

String Similarity with fuzzy, or approximate, string matching to check if two strings are (typographically) similar

EDIT\_DISTANCE

EDIT\_DISTANCE\_SIMILARITY

JARO\_WINKLER\_DISTANCE

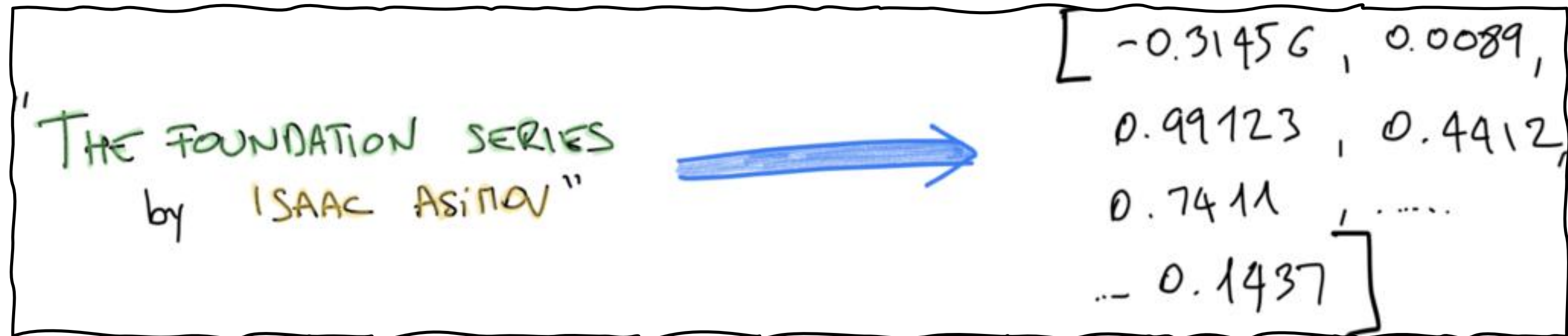
JARO\_WINKLER\_SIMILARITY

String Concatenation operator || 🥰

UNISTR to easily use Unicode literals via codepoints

```
SELECT UNISTR(N'Hello! \+01F603');
```

# Just enough concepts - Embeddings



# Vectors in Azure SQL

```
2
3 declare
4     @v1 vector(3) = '[1,1,1]',
5     @v2 vector(3) = '[2,2,2]'
6
7 select vector_distance('euclidean', @v1, @v2)
8
9
```

TERMINAL

PROBLEMS

OUTPUT

PORTS

DEBUG CONSOLE

QUERY RESULTS (PREVIEW)

SPELL CHE

Results

Messages

(No column name)



1 1.7320508075688772

# Generating Embedding (in T-SQL)

An embedding model is needed

You can use Azure OpenAI or any other embedding model

[On Azure use Managed Identities to avoid using keys and secrets](#)

Embedding models are hosted in *inference servers* and exposed as REST endpoint

Option 1: Use CREATE EXTERNAL MODEL + AI\_GENERATE\_EMBEDDINGS

Take care of everything for you

Option 2: Use sp\_invoke\_external\_rest\_endpoint

Allows calls to any HTTP REST endpoint

Use it to invoke ***any*** model

# Generating Embedding (in T-SQL)

```
create database scoped credential [https://ms-open-ai-3.openai.azure.com/]  
with  
    identity = 'Managed Identity',  
    secret = '{"resourceid": "https://cognitiveservices.azure.com"}';  
go
```

# Generating Embedding (in T-SQL)

```
create external model Text3Small
with (
    location = 'https://ms-open-ai-3.openai.azure.com/openai/deployments/text-embedding-ada-002'
    credential = [https://ms-open-ai-3.openai.azure.com/],
    api_format = 'Azure OpenAI',
    model_type = embeddings,
    model = 'embeddings'
);
go
```

# Generating Embedding (in T-SQL)

```
insert into
    dbo.grant_microsoft_chunks (GrantID, ChunkID, TextChunk, EmbeddingText3Small)
select
    GrantID,
    chunk_order as ChunkID,
    chunk as TextChunk,
    AI_GENERATE_EMBEDDINGS(c.TextChunk use model Text3Small) as EmbeddingText3Small
from
    [dbo].[grant_microsoft] as g
cross apply
    AI_GENERATE_CHUNKS(
        source = [Title] || ' ' || [Description],
        chunk_type = fixed,
        chunk_size = 1000,
        overlap = 10
    ) as c
go
```

# (Exact) Semantic Similarity Search

```
declare @inputText nvarchar(max) = 'the foundation series by isaac asimov';
declare @embedding vector(1536);

set @embedding = ai_generate_embeddings(@inputText use model Ada2Embeddings)

select top(10)
|   a.id,
|   a.title,
|   a.url,
|   vector_distance('cosine', @embedding, title_vector_ada2) as cosine_distance
from
|   dbo.wikipedia_articles_embeddings a
order by
|   cosine_distance;
go
```

<https://github.com/Azure-Samples/azure-sql-db-openai>

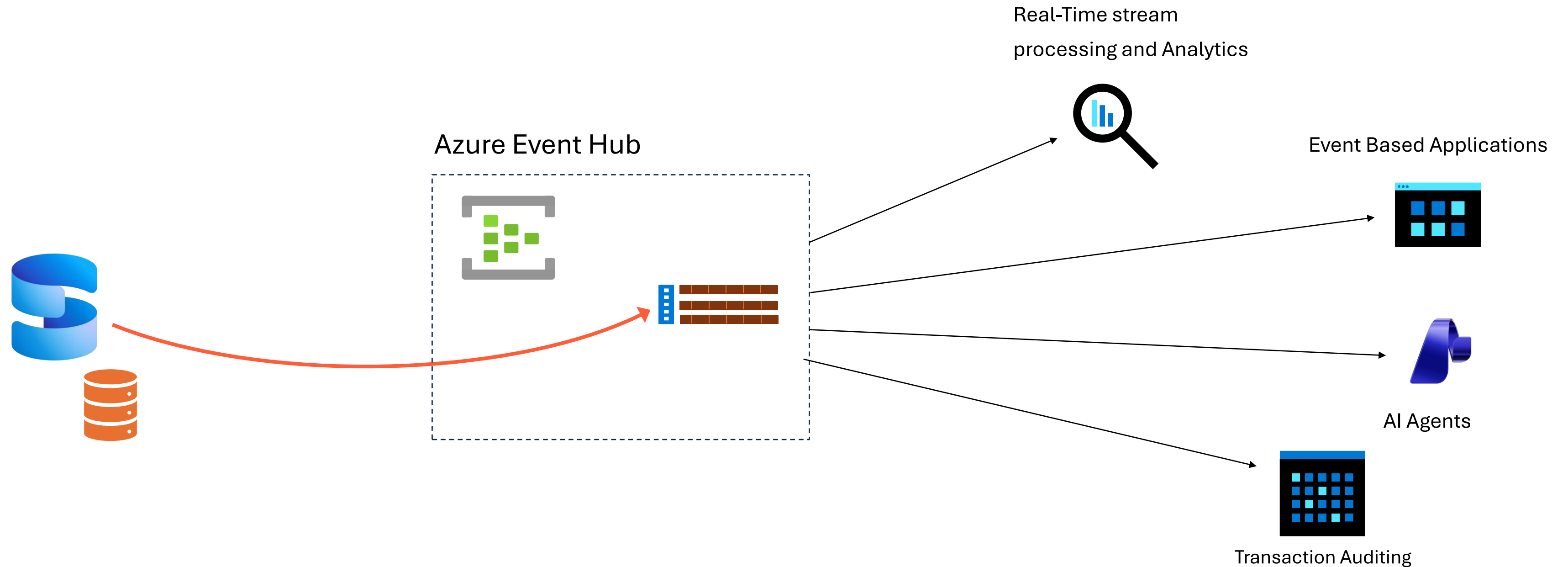
# Create vector index for approximate vectors search (ANN)

```
create vector index vec_idx2  
on [dbo].[wikipedia_articles_embeddings]([content_vector])  
with (metric = 'cosine', type = 'diskann');  
go
```

# Search for similar vectors using approximate index

```
select top(50) with approximate  
t.id, s.distance, t.title  
from  
vector_search(  
table = [dbo].[wikipedia_articles_embeddings] as t,  
column = [content_vector],  
similar_to = @qv,  
metric = 'cosine'  
) as s  
order by s.distance;  
go
```

# Change Event Streaming (CES)



CES enables you to stream your data changes in near real-time from SQL data directly into Azure Event Hubs.

# Change Event Streaming (CES)

Enable

```
EXEC sys.sp_enable_event_stream
```

Register EventHub connection

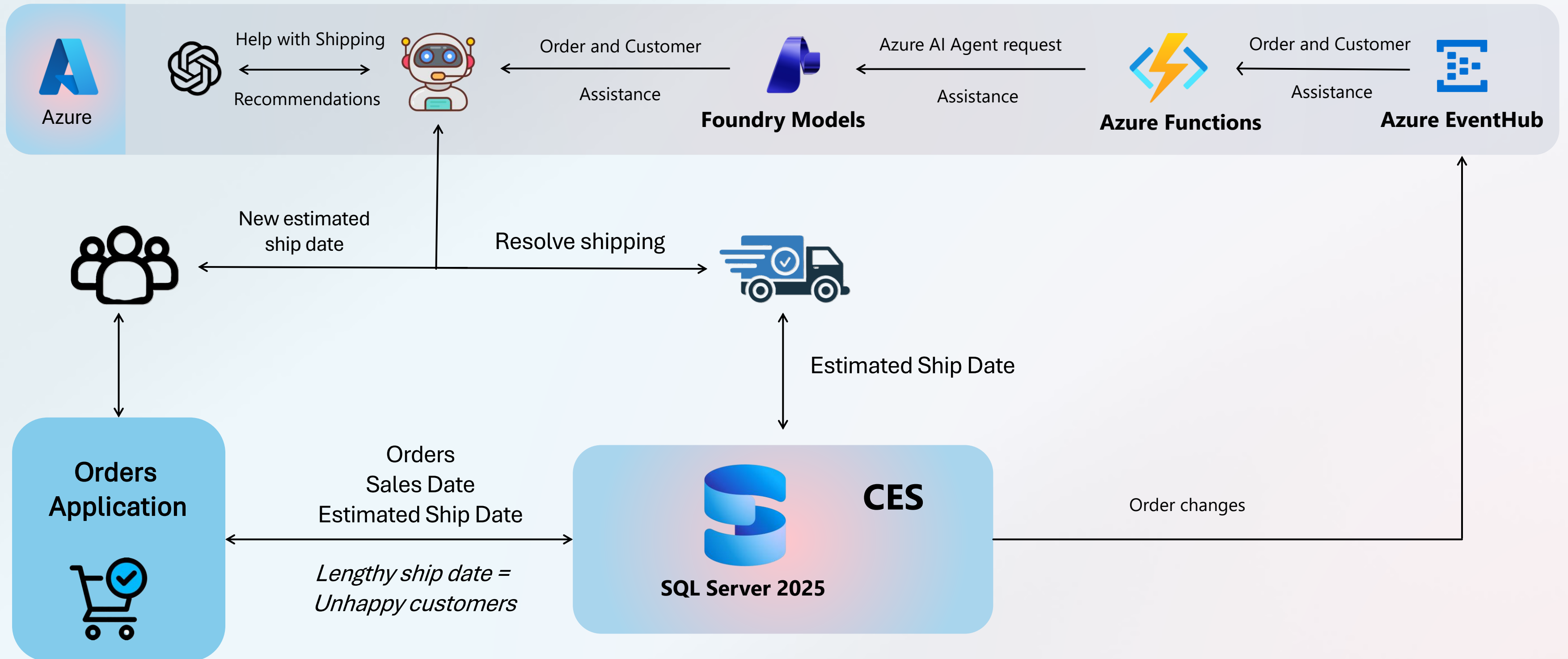
```
EXEC sys.sp_create_event_stream_group
```

Add Table

```
EXEC sys.sp_add_object_to_event_stream_group
```

# Contoso shipping uses CES

Push vs Pull  
"IOLess"  
Built into the engine



Object Explorer

Connect - (SQL Server 17.0.600.9 - sqladmin)

- Databases
  - System Databases
  - Database Snapshots
  - ContosoOrders
- Security
- Server Objects
- Replication
- Always On High Availability
- Management
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler

```

-- Enable CES
EXEC sys.sp_enable_event_stream;
GO

CREATE TABLE Orders (
  OrderID INT PRIMARY KEY CLUSTERED IDENTITY,
  CustomerFirstName NVARCHAR(50),
  CustomerLastName NVARCHAR(50),
  Company NVARCHAR(100),
  SalesDate DATE,
  EstimatedShipDate DATE,
  ShippingID INT,
  ShippingLocation NVARCHAR(100),
  Product NVARCHAR(100),
  Quantity INT,
  Price DECIMAL(10, 2)
);

```

Results

table_group_id	table_group_name	destination_location	destination_credential	destination_type
1	OrdersCESGroup	sqleventhubspace.servicebus.windows.net/sqlevent...	eventhubscred	1

table_group_id	table_group_name	schema_name	table_name	table_id	destination_locatic
1	OrdersCESGroup	dbo	Orders	E1CE6830-1E00-49AB-97C9-09DC82B4BF80	sqleventhubspace

Query executed successfully.

(local) (17.0 CTP1.5) sqladmin (54) ContosoOrders 00:00:00 2 rows

Object Explorer

Connect - [Server Icon] [Server Name] [Refresh] [Disconnect]

- . (SQL Server 17.0.600.9 - sqladmin)
  - Databases
    - System Databases
    - Database Snapshots
    - ContosoOrders
  - Security
  - Server Objects
  - Replication
  - Always On High Availability
  - Management
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler

```

enablece...in (54)  testorde...in (51)
USE [ContosoOrders];
GO

-- Insert a test record
INSERT INTO Orders (CustomerFirstName, CustomerLastName, Company, SalesDate, EstimatedShipDate, ShippingID, ShippingLocation
VALUES
('Test', 'User', 'Test Company', '2025-04-20', '2025-04-25', 1, 'Test Location', 'Test Product', 1, 100.00);

-- Check our changes
SELECT * FROM sys.dm_change_feed_log_scan_sessions
ORDER by start_time DESC;
GO

-- Delete test from Orders
DELETE FROM Orders WHERE Company = 'Test Company';
GO
  
```

100% | Ln: 9 Ch: 21 TABS CRLF

Connected. (1/1) | (local) (17.0 CTP1.5) sqladmin (51) ContosoOrders 00:00:00 0 rows

# sqlleventsfunctionapp

Function App

Search

Browse Refresh Stop Restart Swap Get publish profile Reset publish profile Download app content Delete Send us your feedback

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Microsoft Defender for Cloud
- Events (preview)
- Recommended services (preview)
- Resource visualizer
- Functions
  - App keys
  - Proxies
- Deployment
  - Deployment Center
- Settings
  - Environment variables
  - Configuration
  - Deployment settings
  - Authentication
  - Identity
  - Custom domains
  - Certificates
  - Networking

## Essentials

View Cost JSON View

Resource group (move)	: <a href="#">sqlleventhurg</a>	Default domain	: <a href="#">sqlleventsfunctionapp.azurewebsites.net</a>
Status	: Running	Operating System	: Linux
Location (move)	: <a href="#">South Central US</a>	Plan type	: Flex Consumption
Subscription (move)	: <a href="#">AzureSQL_bobward</a>	Instance Memory	: 2048 MB
Subscription ID	: 0efc44aa-c965-420f-aac4-fff305dbcc97		
Tags (edit)	: <a href="#">Add tags</a>		

Functions Properties Notifications (0)

## Create functions in your preferred environment



### Create with Visual Studio

- Best optimized for:
- Local development within Visual Studio
  - Using a full .NET IDE

Create with Visual Studio



### VS Code Desktop

- Best optimized for:
- Local development within VS Code
  - Custom development tool requirements

Create with VS Code Desktop



### Other editors or CLI

- Best optimized for:
- Using preferred editor for development
  - Visual Studio, IntelliJ, command line

Set up your editor

```

C# SQLEventHubTrigger.cs 9+, U x
C# SQLEventHubTrigger.cs > ...
15     public class SQLEventHubTrigger
212     {
213         Messages =
214         {
215             new ChatMessage(ChatRole.System, "You are an AI agent to help analyze shipment for orders for Contoso")
216             new ChatMessage(ChatRole.User, prompt)
        }
    }

```

OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

Workload updates are available. Run `dotnet workload list` for more information.  
\* Terminal will be reused by tasks, press any key to close it.

\* Executing task: func host start --no-build --port 7212

Azure Functions Core Tools

Core Tools Version: 4.0.7030 Commit hash: N/A +bb4c949899cd5659d6bfe8b92cc923453a2e8f88 (64-bit)  
Function Runtime Version: 4.1037.0.23568

[2025-04-21T14:23:05.947Z] Found C:\demos\SQL2025\ces\azurefunction\azurefunction.csproj. Using for user secrets file configuration.

[2025-04-21T14:23:07.329Z] Worker process started and initialized.

Functions:

SQLEventHubTrigger: eventHubTrigger

For detailed output, run func with --verbose flag.

[2025-04-21T14:23:12.617Z] Host lock lease acquired by instance ID '00000000000000000000000000783C830A'.

pwsh  
func: h...

# Drivers

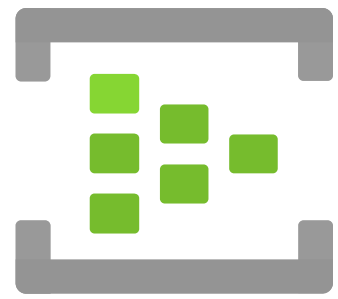
- Official MS Python driver: <https://github.com/microsoft/mssql-python>
  - It is just a great driver! Entra ID, BCP Support, etc...
- What's next?
  - Node?
  - Rust
- Or something completely different like ADBC?
  - Apache Arrow Flight SQL Driver
    - [https://arrow.apache.org/adbc/current/driver/flight\\_sql.html](https://arrow.apache.org/adbc/current/driver/flight_sql.html)
    - <https://github.com/adbc-drivers>

# Services Integration

I really want to use other Azure products and services from within my Azure SQL Database without having to write complex code, functions, or scheduled events. Is this even possible?

# External REST Endpoint Invocation

Using a single stored procedure, call REST/GraphQL service endpoints in Azure directly from within an Azure SQL Database.



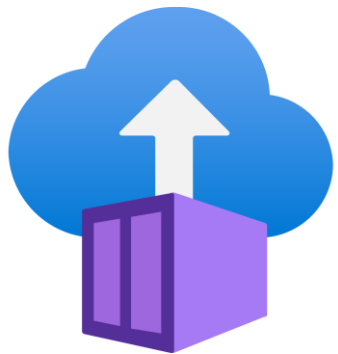
Event Hub



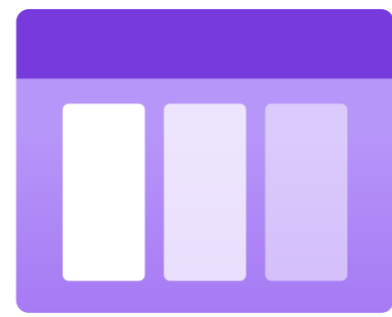
Microsoft Dynamics



PowerBI



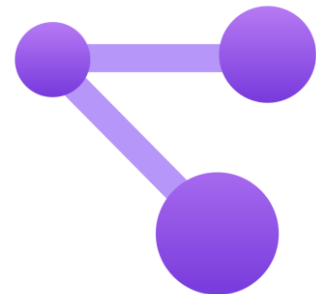
Azure Containers



Azure Queue Storage



IoT Central



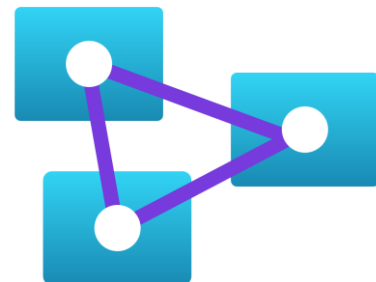
Microsoft Graph



SQL



API Management



Analysis Services



Cognitive Services



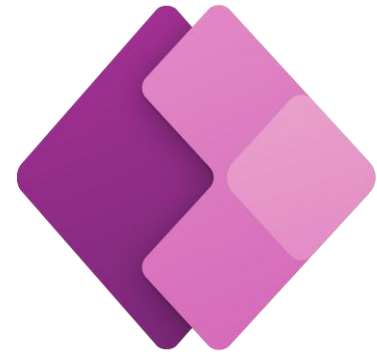
Event Grid



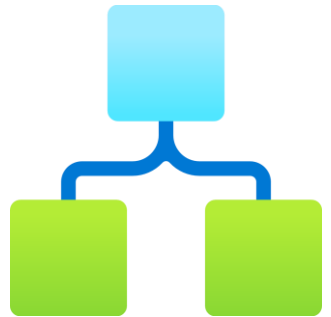
Azure Blob Storage



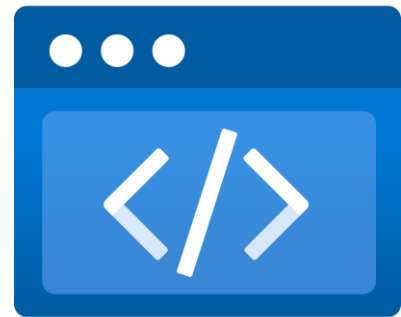
Azure App Services



Power Apps



Logic Apps



Static Web Apps



Dataverse



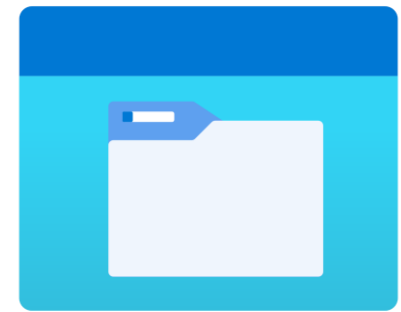
Azure Functions



Azure Table Storage



OpenAI



Azure Files

# Use Cases

Activate workflows

Data enrichment

Cache invalidation / update

Augment business/process logic

Update websites

Integrate with event-based architectures

Data streams

# Security Included

## In the Database

Requires EXECUTE ANY EXTERNAL ENDPOINT  
database permission.

## Endpoint Authentication

Header

Query String/URL

Managed Identity

# Easy to use

One simple and easy call to a REST endpoint directly from the Database

```
EXEC @returnValue = sp_invoke_external_rest_endpoint
  [ @url = ] N'url'
  [ , [ @payload = ] N'request_payload' ]
  [ , [ @headers = ] N'http_headers_as_json_array' ]
  [ , [ @method = ] 'GET' | 'POST' | 'PUT' | 'PATCH' | 'DELETE' | 'HEAD' ]
  [ , [ @timeout = ] seconds ]
  [ , [ @credential = ] credential ]
  [ , @response OUTPUT ]
```

# Data API Builder

Make the database available as a REST and/or GraphQL endpoint

- “CRUD as a Service” / “Backend as a Service”
  - Open Source, Stateless, Horizontally Scalable
  - .NET application, easy to containerize (pairs perfectly with Azure Container Apps)
  - Integrated with Aspire already
- 
- Features
    - Project, Filter, Paginate, Sort, Aggregate
    - Authentication via JWT
    - Authorization via policy
    - Caching (integrated with Redis)
    - REST (OpenAPI), GraphQL (HotChocolate), MCP!



<https://aka.ms/dab>

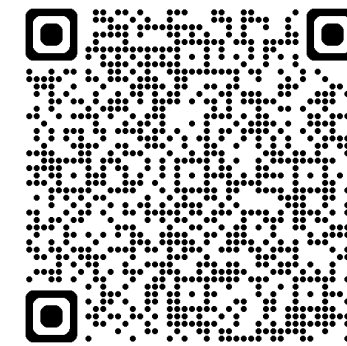
# DEMO

Data API Builder

# Thank you!

Coming Spring 2026

# SQL AI Database Developer Certification



Level up and prove your skills

Building AI-powered,  
LLM-integrated apps

AI-augmented SQL  
development

Advanced SQL  
Programming

Security, governance,  
and responsible AI

[aka.ms/learnSQLAI](https://aka.ms/learnSQLAI)

Sound off.  
The mic is all yours.  
Influence the product roadmap.

Join the Fabric User Panel



Share your feedback directly with our Fabric product group and researchers.

<https://aka.ms/JoinFabricUserPanel>

Join the SQL User Panel



Influence our SQL roadmap and ensure it meets your real-life needs

<https://aka.ms/JoinSQLUserPanel>

# How was the session?



Complete Session Surveys in  
*Whova* for your chance to WIN  
PRIZES!

