

Quickstart: Create an Azure Cosmos DB for NoSQL account using the Azure portal

In this quickstart, you create a new Azure Cosmos DB for NoSQL account in the Azure portal. You then use the Data Explorer experience within the Azure portal to create a database and container configuring all required settings. Finally, you add sample data to the container and issue a basic query.

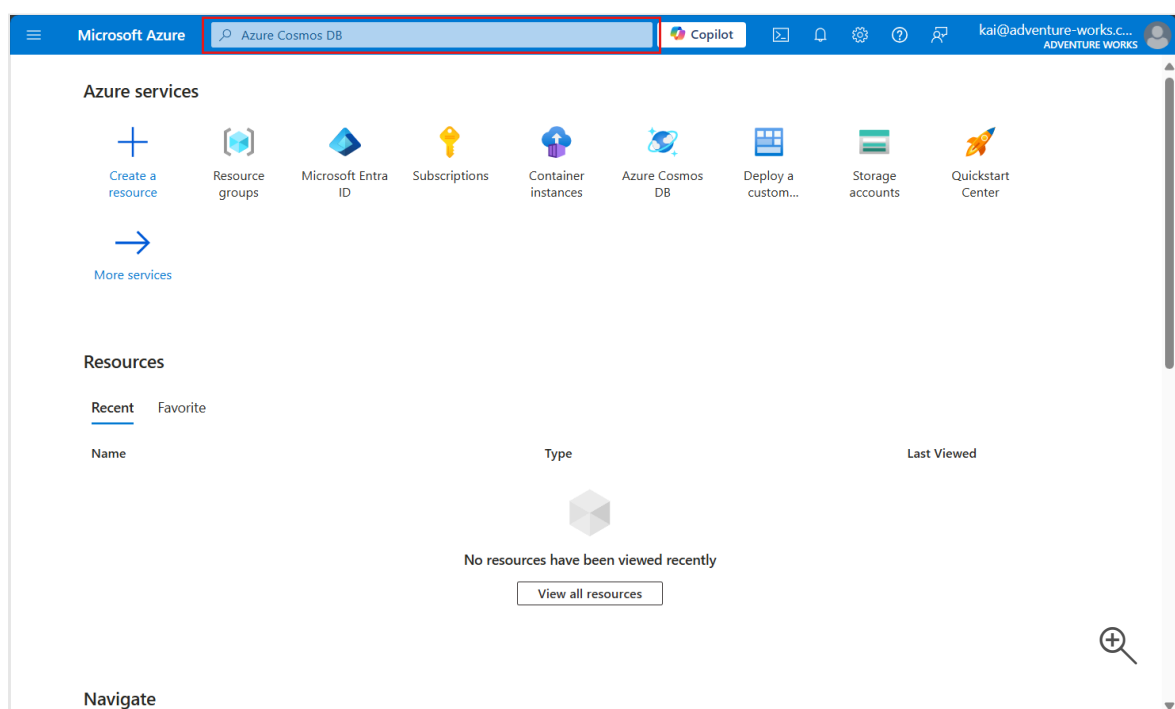
Prerequisites

- An Azure account with an active subscription. [Create an account for free](#) .

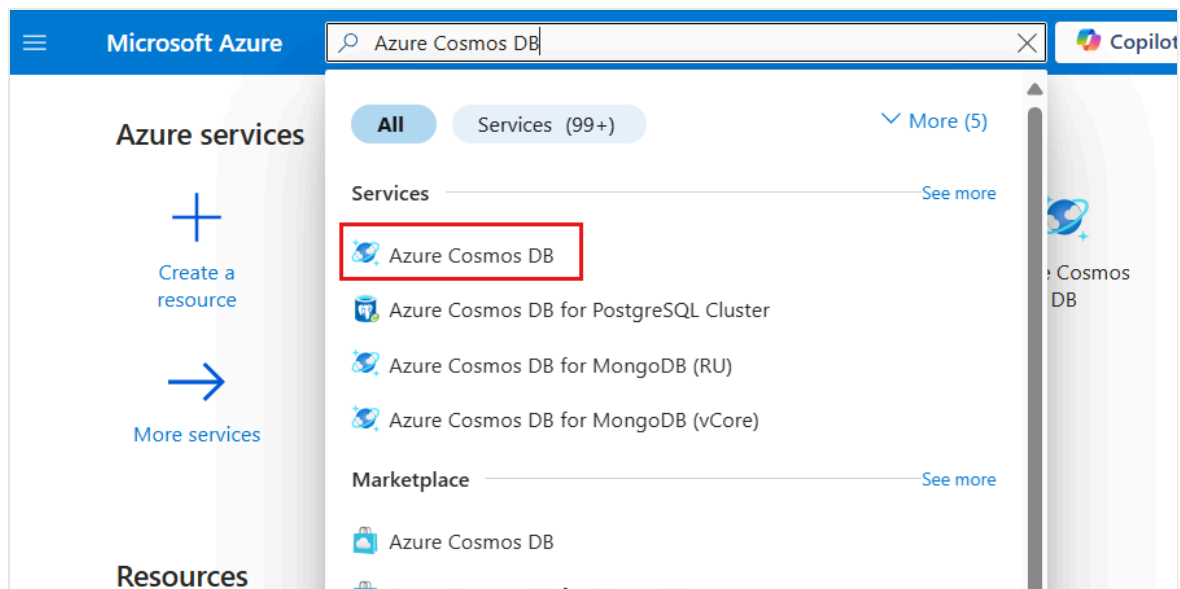
Create an account

Start by creating a new Azure Cosmos DB for NoSQL account

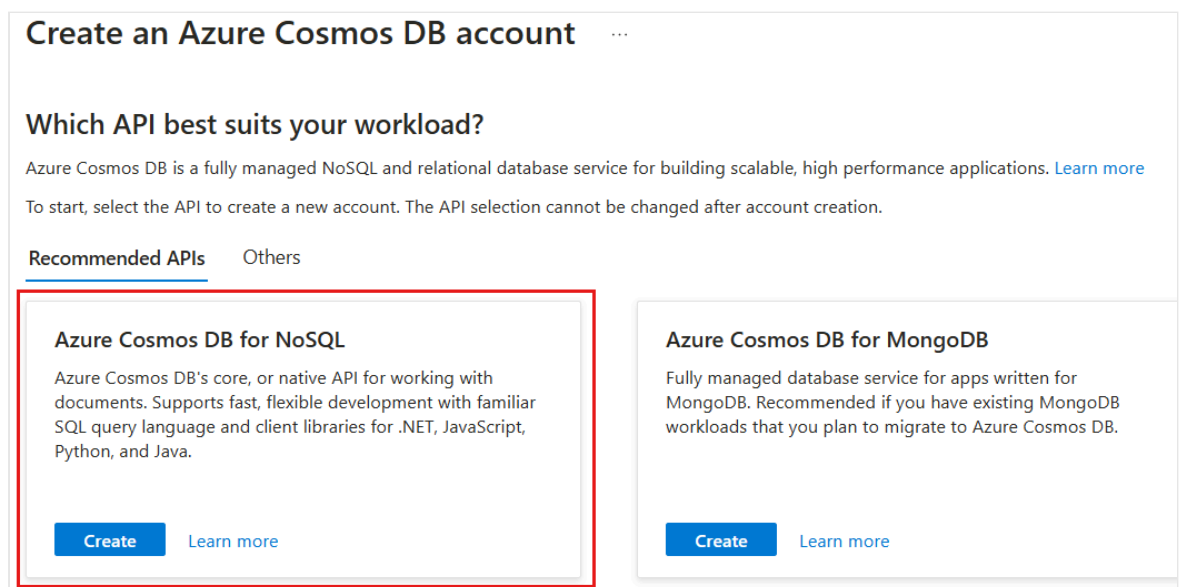
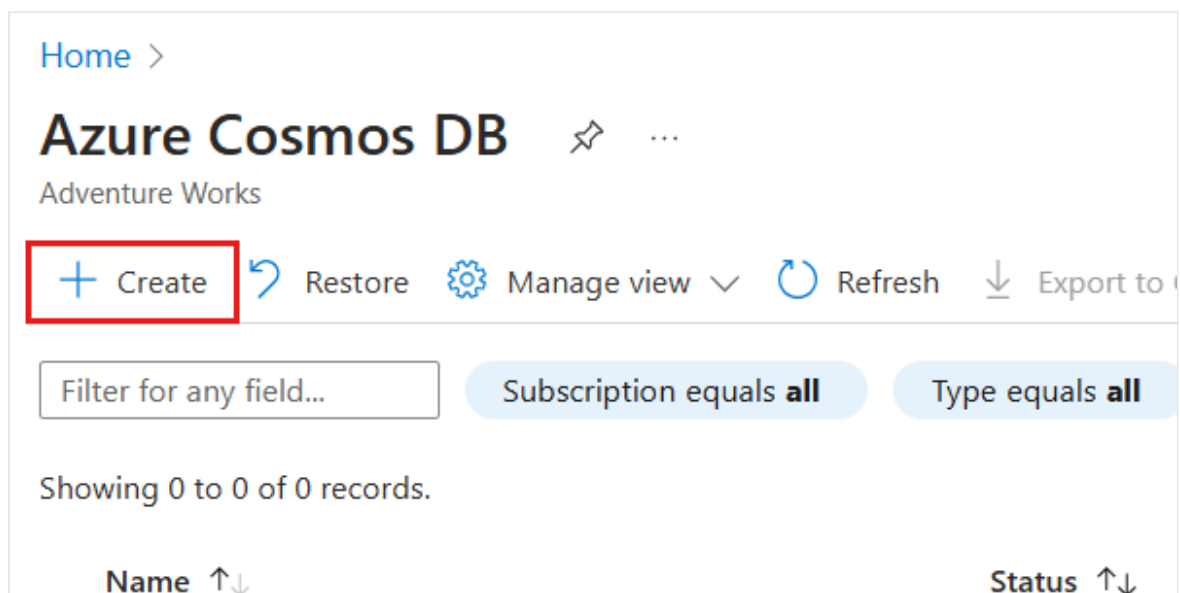
1. Sign in to the Azure portal (<https://portal.azure.com>).
2. Enter *Azure Cosmos DB* in the global search bar.



3. Within **Services**, select **Azure Cosmos DB**.



4. In the **Azure Cosmos DB** pane, select **Create**, and then **Azure Cosmos DB for NoSQL**.



5. Within the **Basics** pane, configure the following options, and then select **Review + create**:

 Expand table

	Value
Subscription	Select your Azure subscription
Resource Group	Create a new resource group or select an existing resource group
Account Name	Provide a globally unique name
Availability Zones	<i>Disable</i>
Location	Select a supported Azure region for your subscription

Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

Basics Global distribution Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Try it for free](#), for 30 days with unlimited renewals. Go to production starting at \$24/month per database, multiple containers included. [Learn more](#)

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource Group *
[Create new](#)

Instance Details

Account Name *

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones ☐ Enable ☒ Disable

Location *

Available locations are determined by your subscription's access and availability zone support (if that is enabled). If you don't see or cannot select your desired location, please open a support request for region access.

[Review + create](#) [Previous](#) [Next: Global distribution](#) [Feedback](#)

Tip

You can leave any unspecified options to their default values. You can also configure the account to limit total account throughput to 1,000 request units per second (RU/s) and enable free tier to minimize your costs.

6. On the **Review + create** pane, wait for validation of your account to finish successfully, and then select **Create**.

Validation Success

BasicsGlobal distributionNetworkingBackup PolicyEncryptionTagsReview + create

Creation Time

Estimated Account Creation Time (in minutes)2

The estimated creation time is calculated based on the location you have selected

Basics

Subscription

Adventure Works Subscription

Resource Group

(new) msdocs-cosmos-demo

Location

West US 2

Account Name

(new) msdocs-cosmos-demo-nosql

API

Azure Cosmos DB for NoSQL

Capacity mode

Provisioned throughput

Geo-Redundancy

Disable

Multi-region Writes

Disable

Availability Zones

Disable

CreatePreviousNextDownload a template for automationFeedback

7. The portal automatically navigates to the **Deployment** pane. Wait for the deployment to complete.

Deployment is in progress

Deployment name : Microsoft.Azure.Cosm...

Subscription : Adventure Works Sub...

Resource group : msdocs-cosmos-demo


Deployment details

Resource	Type	Status
msdocs-cosmo...	Microsoft.DocumentDb/databas	OK

8. Once the deployment is complete, select **Go to resource** to navigate to the new Azure Cosmos DB for NoSQL account.

✓

Your deployment is complete



Deployment name : Microsoft.Azure.Cosm...

Subscription : [Adventure Works Sub...](#)

Resource group : [msdocs-cosmos-demo](#)

>

Deployment details

✓




Next steps

Go to resource

Create a database and container

Next, use the Data Explorer to create a database and container in-portal.

1. In the account resource pane, select **Data Explorer** in the service menu.

 **msdocs-cosmos-demo-nosql**   ...

Azure Cosmos DB account

Search


×


«

+ Add Container

↻ Refresh

→ Move ▾

 Data Explorer

 Enable

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Cost Management

Quick start

Data Explorer

Settings

Essentials

Status
Online

Resource group ([move](#))
[msdocs-cosmos-demo](#)

Subscription ([move](#))
[Adventure Works Subscription](#)

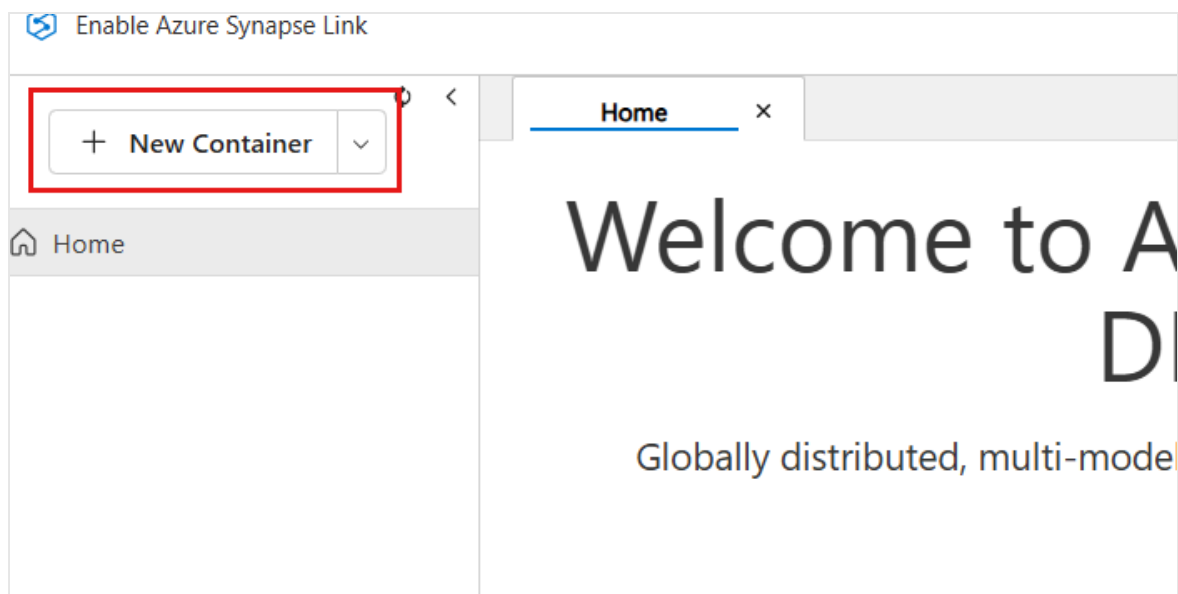
Subscription ID
aaaa0a0a-bb1b-cc2c-dd3d-eeeeee4e4e4e

Total throughput limit
[1000 RU/s](#)

[See more](#)

Read
West
Write
West
URI
https:
Free
Opte
Capa
Provi

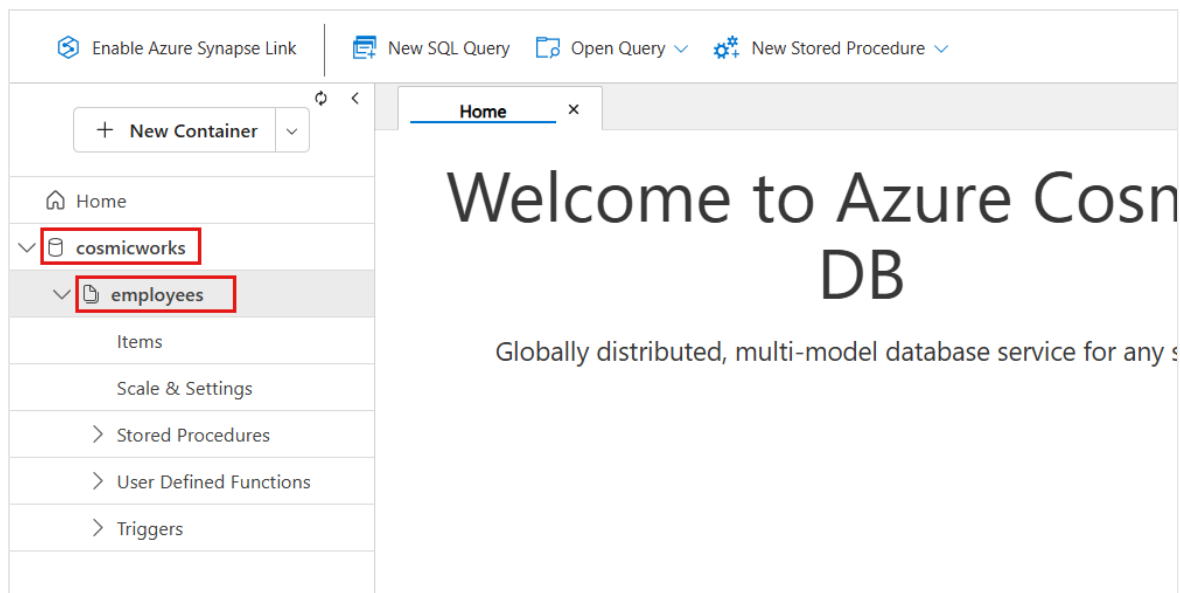
2. In the **Data Explorer** pane, select the **New Container** option.



3. In the **New Container** dialog, configure the following values and then select **OK**:

[Expand table](#)

	Value
Database	Create new
Database id	cosmicworks
Share throughput across containers	Don't select
Container id	employees
Partition key	department/name
Container throughput (autoscale)	Autoscale
Container Max RU/s	1000



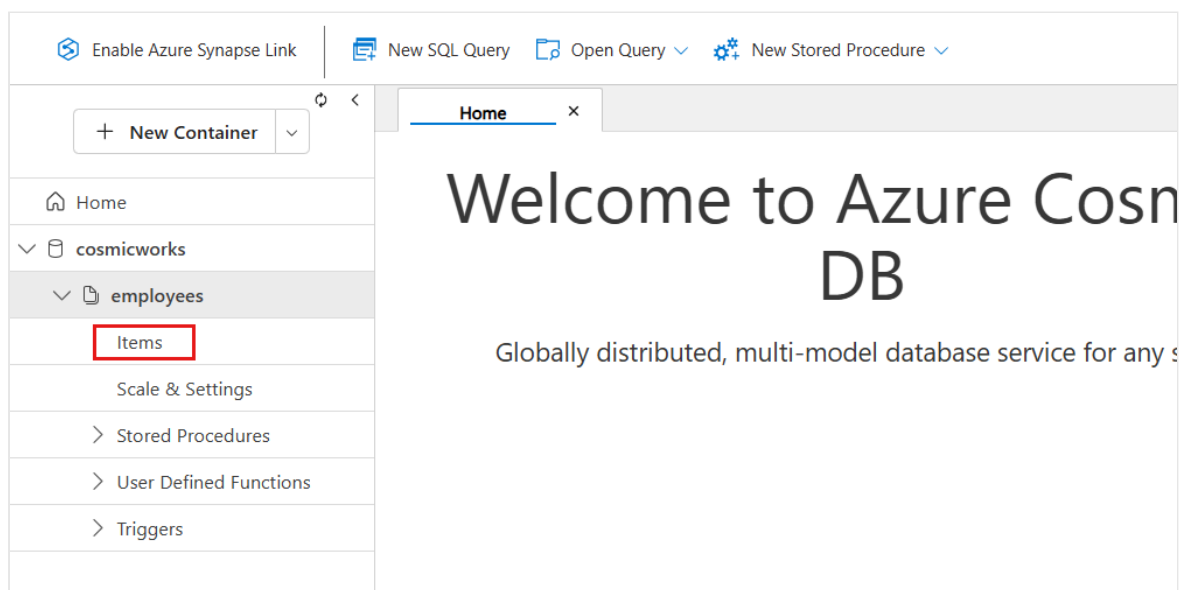
💡 Tip

Optionally, you can expand the container node to observe additional properties and configuration settings.

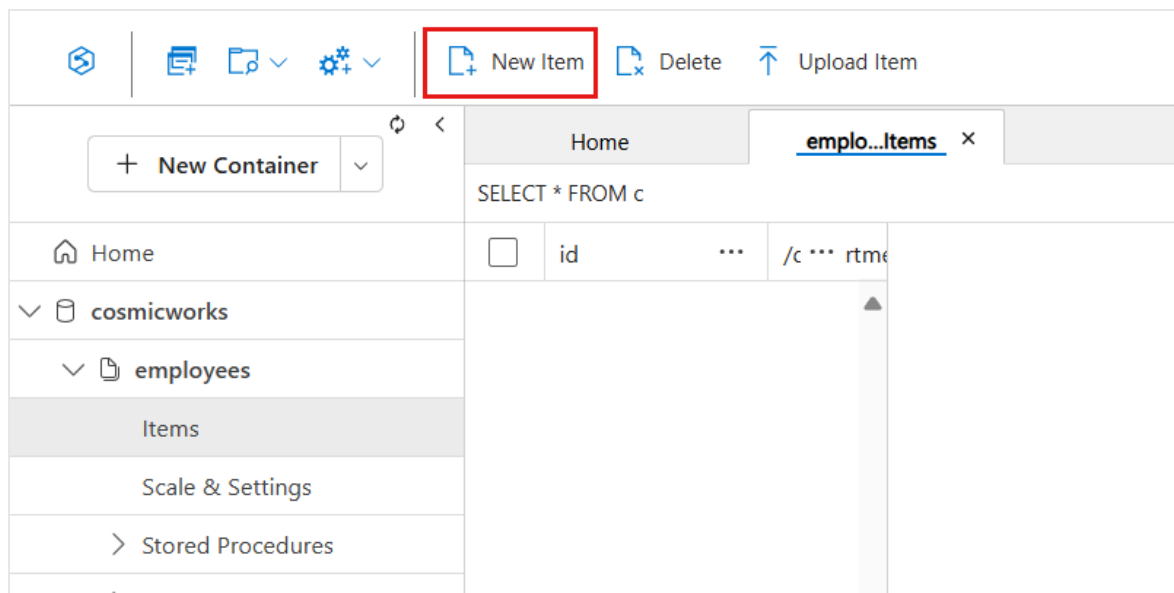
Add and query sample data

Finally, use the Data Explorer to create a sample item and then issue a basic query to the container.

1. Expand the node for the **employees** container in the tree of the Data Explorer. Then, select the **Items** option.



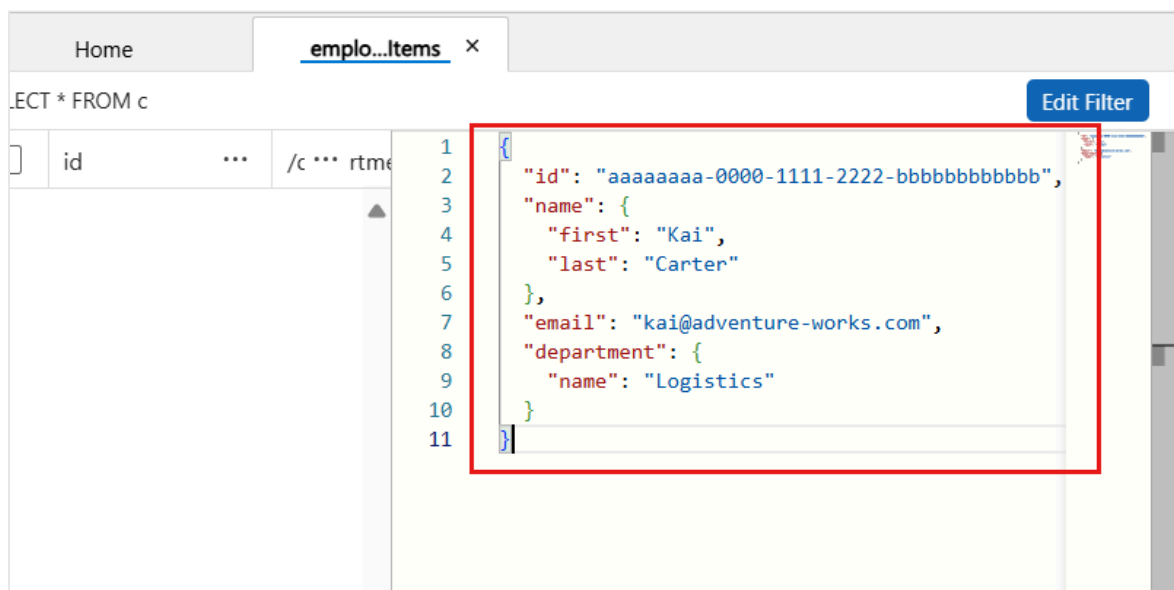
2. In the Data Explorer's menu, select **New Item**.



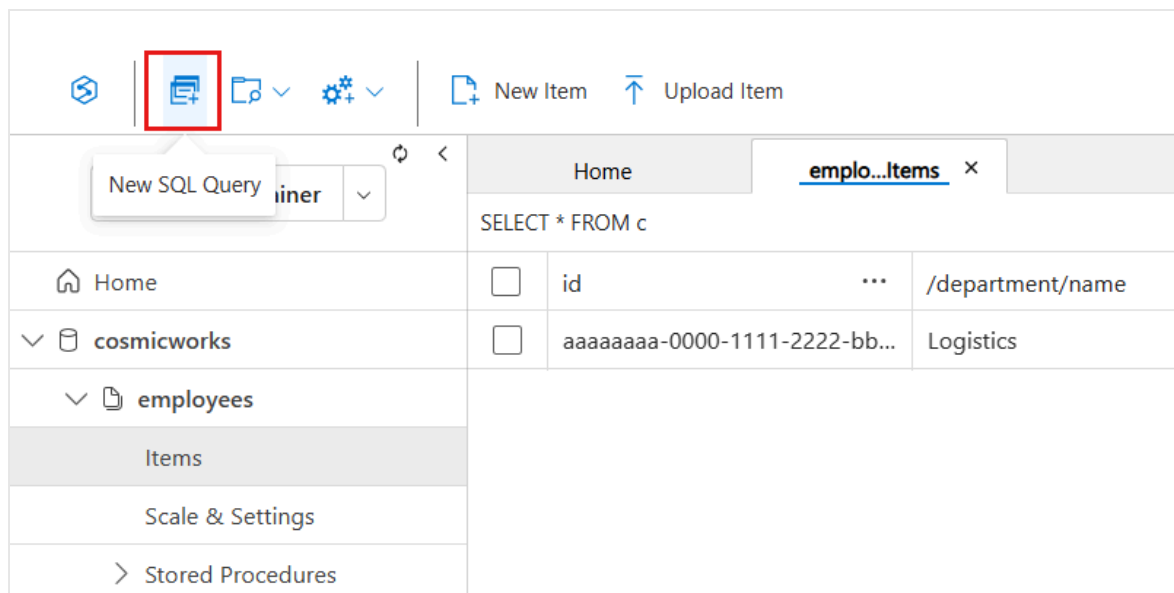
3. Now, insert the following JSON for a new item in the **employees** container and then select **Save**:

```
JSON

{
  "id": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb",
  "name": {
    "first": "Kai",
    "last": "Carter"
  },
  "email": "<kai@adventure-works.com>",
  "department": {
    "name": "Logistics"
  }
}
```



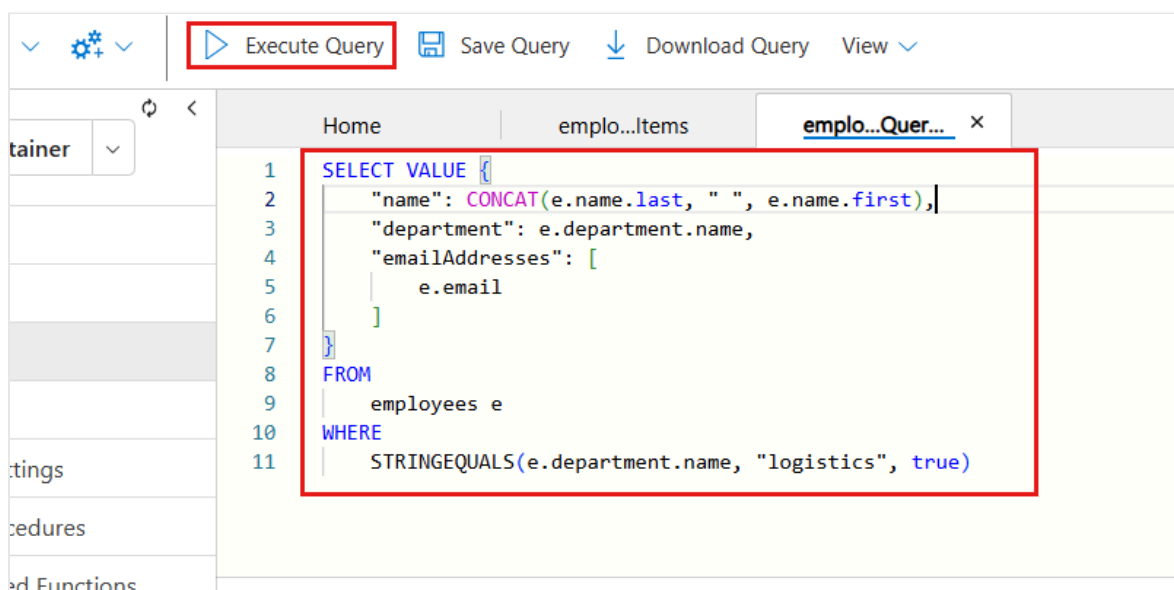
4. In the Data Explorer's menu, select **New SQL Query**.



- Now, insert the following NoSQL query to get all items for the `logistics` department using a case-insensitive search. The query then formats the output as a structured JSON object. Run the query by selecting **Execute Query**:

```
NoSQL

SELECT VALUE {
  "name": CONCAT(e.name.last, " ", e.name.first),
  "department": e.department.name,
  "emailAddresses": [
    e.email
  ]
}
FROM
  employees e
WHERE
  STRINGEQUALS(e.department.name, "logistics", true)
```



- Observe the JSON array output from the query.

JSON

```
[
  {
    "name": "Carter Kai",
    "department": "Logistics",
    "emailAddresses": [
      "kai@adventure-works.com"
    ]
  }
]
```

Results Query Status

1 - 1

```
[
  {
    "name": "Carter Kai",
    "department": "Logistics",
    "emailAddresses": [
      "kai@adventure-works.com"
    ]
  }
]
```