Validate connectivity from SQL Serverless

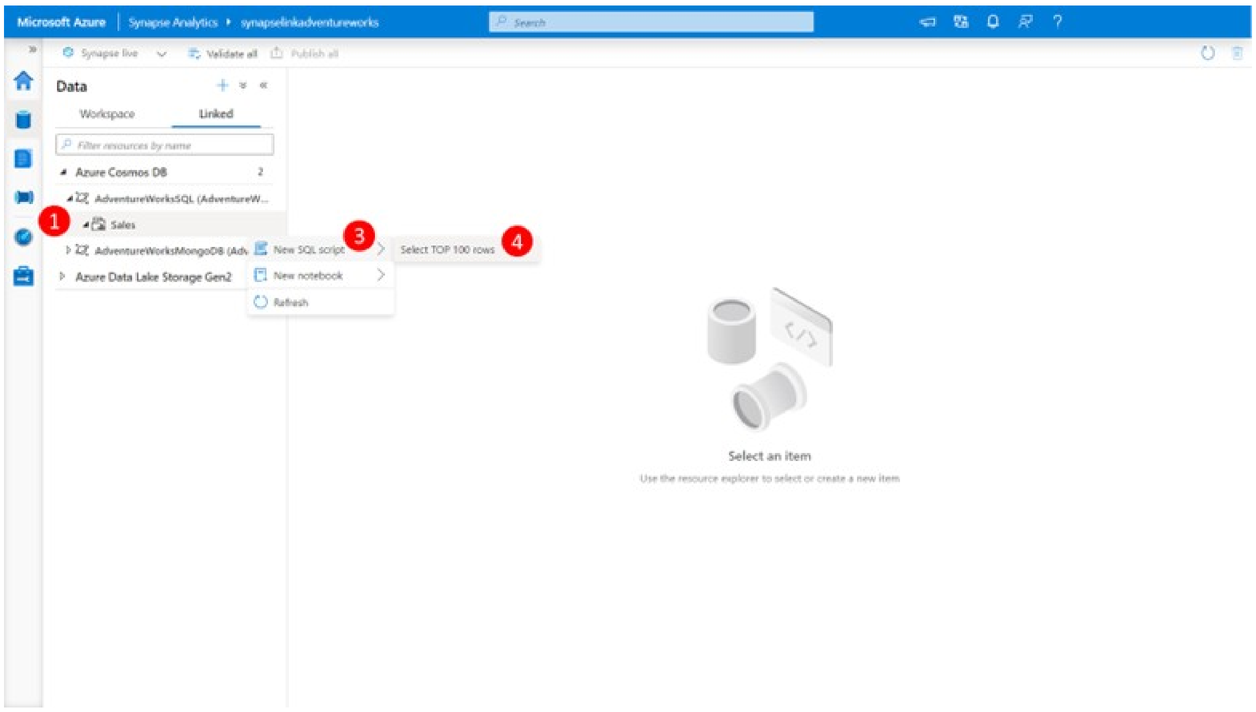
In this reading you can see the steps involved in the process of validating connectivity from SQL Serverless.

Note

You are not required to complete the processes, tasks, activities, or steps presented in this example. Your system set-up may differ from the system set-up in the demonstration in this reading. The various samples provided are for illustrative purposes only and it's likely that if you try this out you will encounter issues in your system.

SQL Queries for Cosmos DB Core (SQL) API

Let's now connect to our Cosmos DB Cosmos DB Core (SQL) API analytical store using the Synapse Analytics SQL Serverless capability and retrieve some data by performing the following steps:



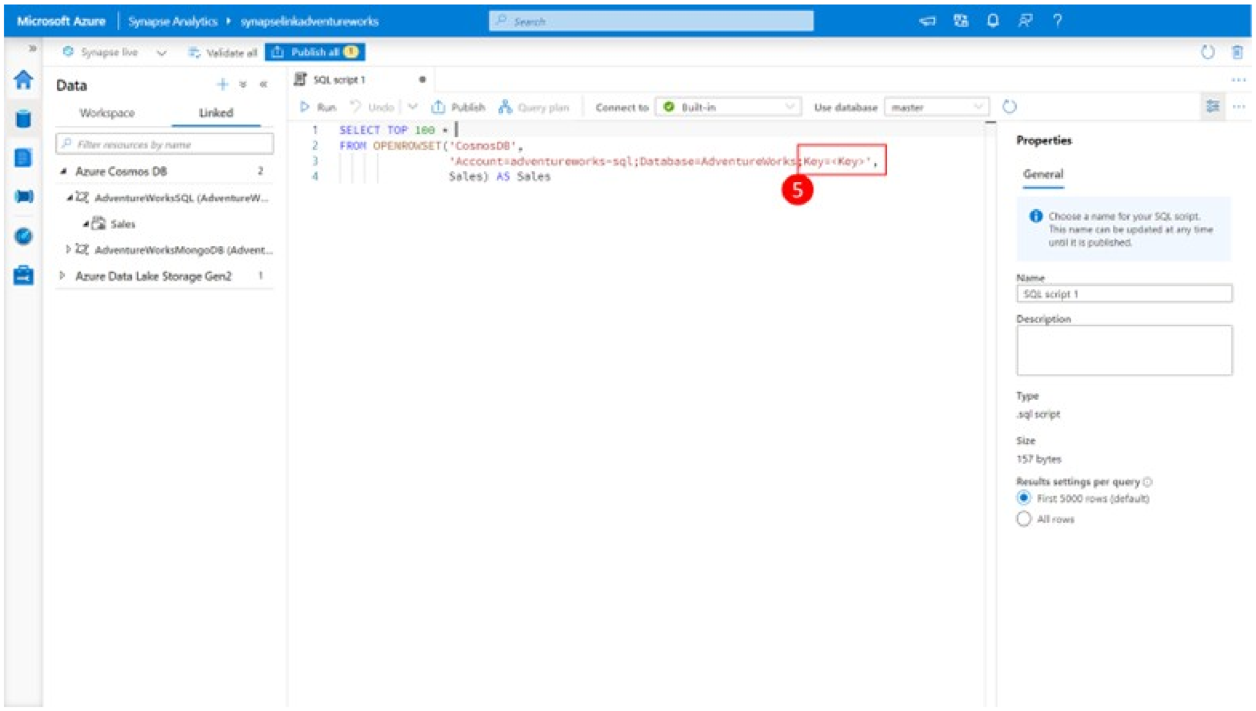
Selecting top 10 records with a sql serverless query.

1. Expand the **AdventureWorksSQL linked service** in the explorer view and click on the **Sale container (1)**

2. Click on the **Actions ellipsis "…"**

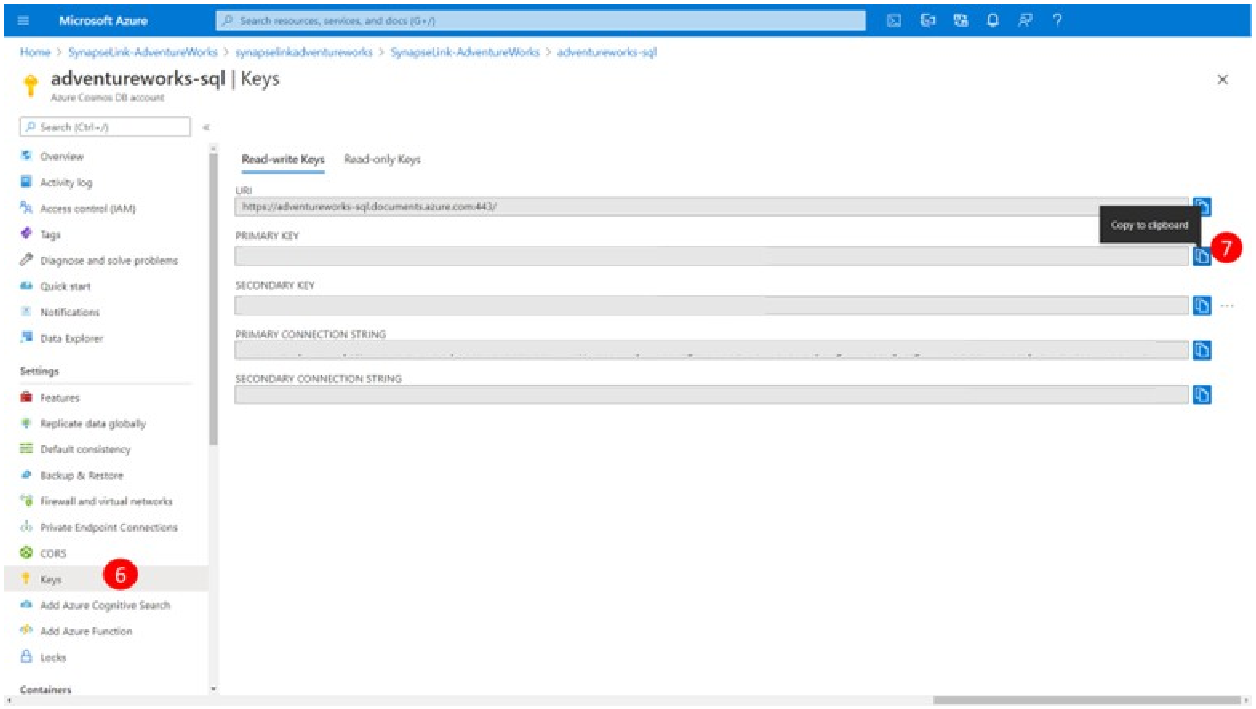
3. Click on **new notebook** to expose the list of New SQL Script actions **(3)**

4. Click on Select TOP 100 rows **(4)**, to load a SQL script window to retrieve the top 100 records from the Linked Server and its associated analytical store.



```
SELECT TOP 100 *
FROM OPENROWSET('CosmosDB',
                'Account=adventureworks-sql;Database=AdventureWorks;Key=<Key>',
                Sales) AS Sales
```

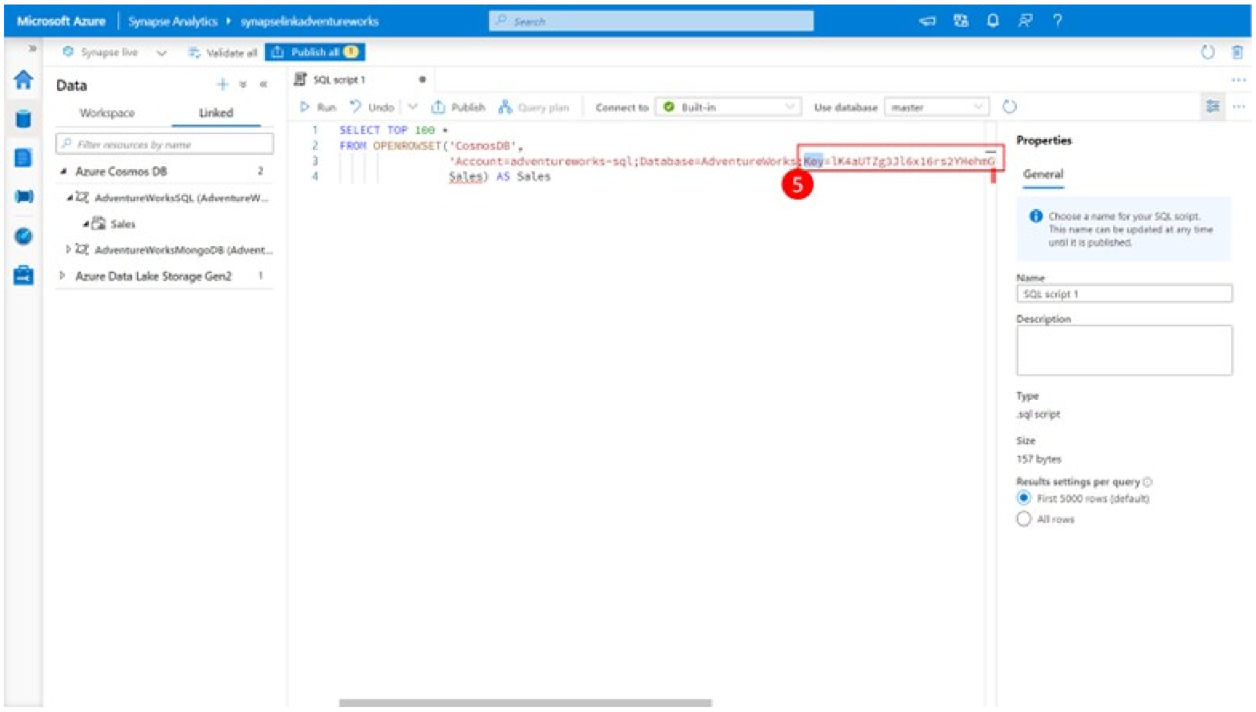
Embedding access keys in a sql serverless query.
You will note that the SQL script template requires the Azure Cosmos DB account key **(5)** for the account we are trying to connect to.



Microsoft Azure
Home > SynapseLink-AdventureWorks > synapselinkadventureworks > SynapseLink-AdventureWorks > adventureworks-sql
adventureworks-sql | Keys
Azure Cosmos DB account
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Quick start
Notifications
Data Explorer
Settings
Features
Replicate data globally
Default consistency
Backup & Restore
Firewall and virtual networks
Private Endpoint Connections
CORS
Keys
6
Add Azure Cognitive Search
Add Azure Function
Locks
Containers
Read-write Keys
Read-only Keys
URI
https://adventureworks-sql.documents.azure.com:443/
PRIMARY KEY
Copy to clipboard
7
SECONDARY KEY
PRIMARY CONNECTION STRING
SECONDARY CONNECTION STRING

Getting access keys for a storage account.

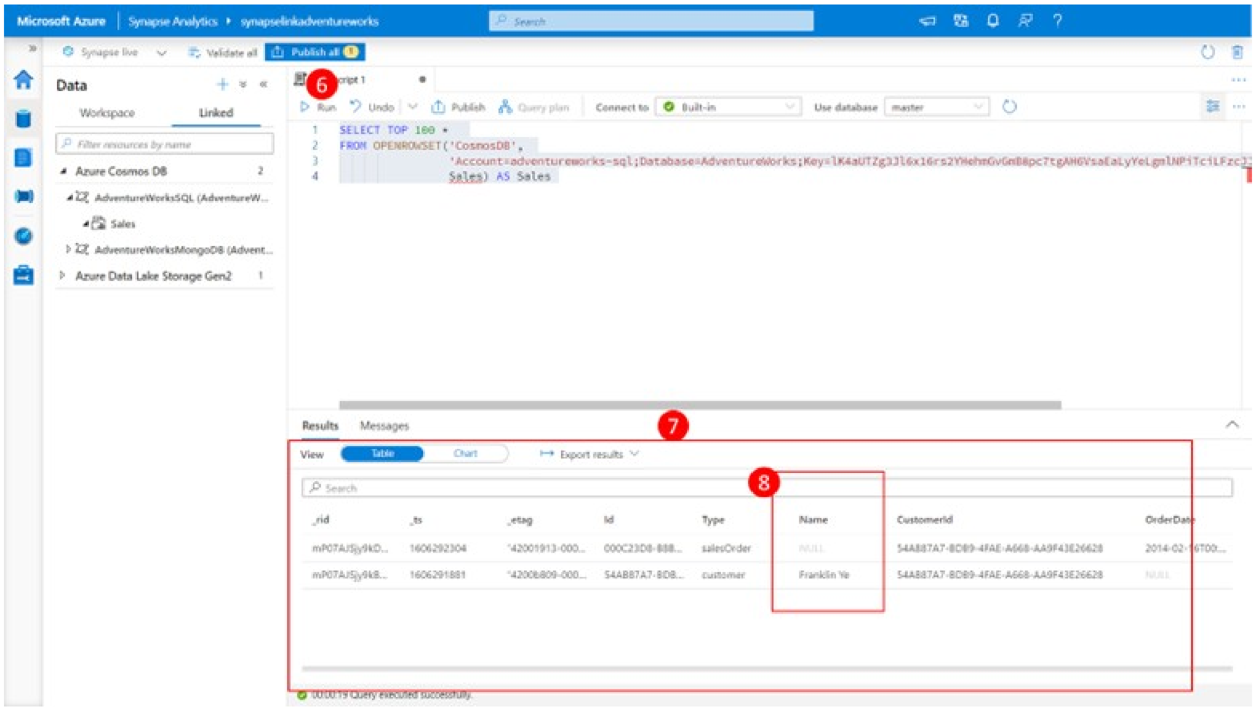
5. You can retrieve this from the Azure Cosmos DB account by a. Clicking on the **Keys** in the left-hand menu **(6)** b. Clicking the **copy icon** next to the PRIMARY KEY value **(7)**



```
SELECT TOP 100 *
FROM OPENROWSET('CosmosDB',
                'Account=adventureworks-sql;Database=AdventureWorks;Key=lK4aUTZg3Jl6x16rs2YHehmG
                Sales) AS Sales
```

adding access keys into a query.

6. And paste the key value from the clipboard back into the query **(5)**



```
SELECT TOP 100 *
FROM OPENROWSET('CosmosDB',
                'Account=adventureworks-sql;Database=AdventureWorks;Key=lK4aUTZg3Jl6x16rs2YHehmGvGmB8pc7tgAH6VsaEaLyYeLgmlNPiTciLFzcJ
                Sales) AS Sales
```

_rid	_ts	_etag	id	Type	Name	CustomerId	OrderDate
mP07AJSjy9kD...	1606292304	"42001913-000...	000C23D8-888...	salesOrder	NULL	54A887A7-8D89-4FAE-A668-AA9F43E26628	2014-02-16T00...
mP07AJSjy9k8...	1606291881	"42008809-000...	54A887A7-8D8...	customer	Franklin Ye	54A887A7-8D89-4FAE-A668-AA9F43E26628	NULL

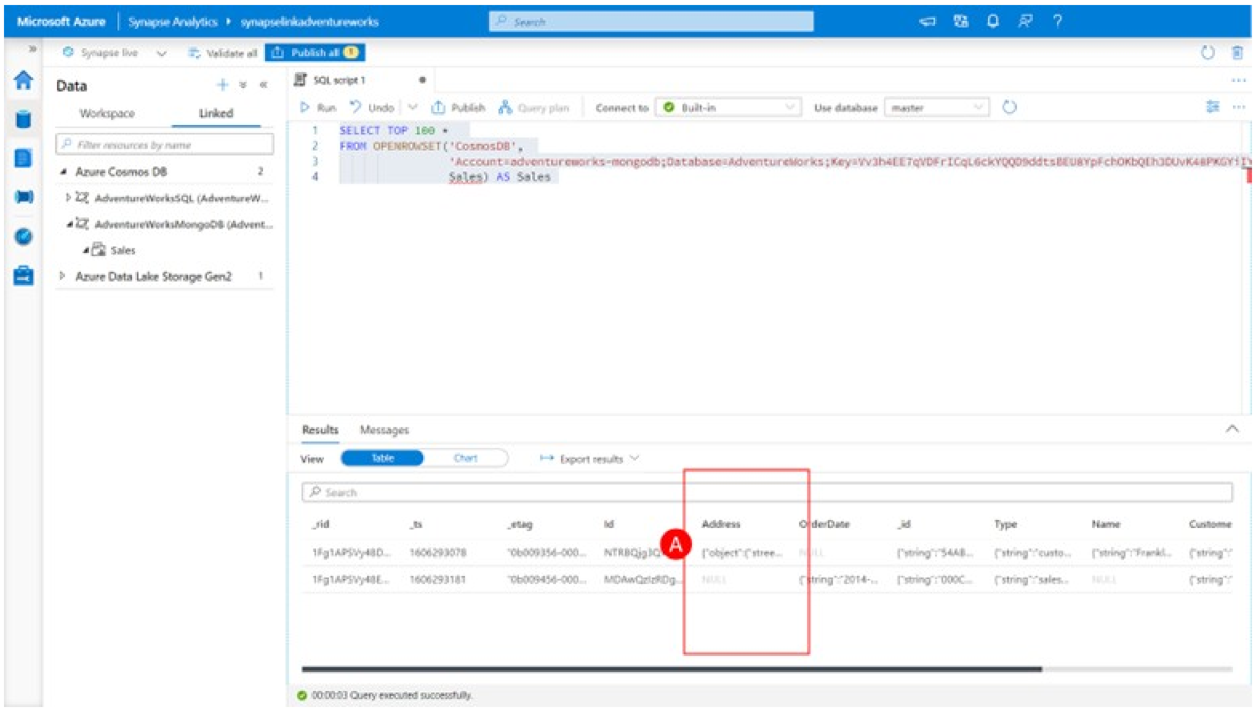
00:00:19 Query executed successfully.

Viewing SQL serverless query results.

7. Now we can click the **Run** button.

You should almost immediately see the query begin to execute and then shortly thereafter receive back a result set **(7)**

Note That for records where data was not defined, such as the name column for the salesOrder record we get back a null value **(8)**



```
SELECT TOP 100 *
FROM OPENROWSET('CosmosDB',
                'Account=adventureworks-mongodb;Database=AdventureWorks;Key=Vv3h4EE7qVDFrICqL6ckYQQD9ddtsBEU8YpFchOKbQEh3DUvK48PKGY!I
                Sales) AS Sales
```

_rid	_ts	_etag	Id	Address	OrderDate	_id	Type	Name	Custome
1Fg1APSVy48D...	1606293078	'0b009356-000...	NTRBQjg3Q...	{"object":{"stree...	NULL	{"string":"54AB...	{"string":"custo...	{"string":"Frankl...	{"string":"
1Fg1APSVy48E...	1606293181	'0b009456-000...	MDAwQzlzRDg...	NULL	{"string":"2014-...	{"string":"000C...	{"string":"sales...	NULL	{"string":"

00:00:03 Query executed successfully.

Viewing JSON objects in the query results.
You will see that columns that contain JSON objects, such as address **(A)** and JSON arrays such a detail has the JSON as their column value content.

8. You can now close the SQL script and discard the changes by clicking the Close and discard changes on the dialog that pops up.

SQL Queries for Cosmos DB API for MongoDB

Let’s now connect to our Cosmos DB Cosmos DB API for MongoDB analytical store using the Synapse Analytics SQL Serverless capability and retrieve some data by performing the following steps:

1. Expand the **AdventureWorksMongoDB linked service** in the explorer view and click on the **Sale container (1)**

2. Click on the **Actions ellipsis “…”**

3. Click on **new notebook** to expose the list of New SQL Script actions **(3)**

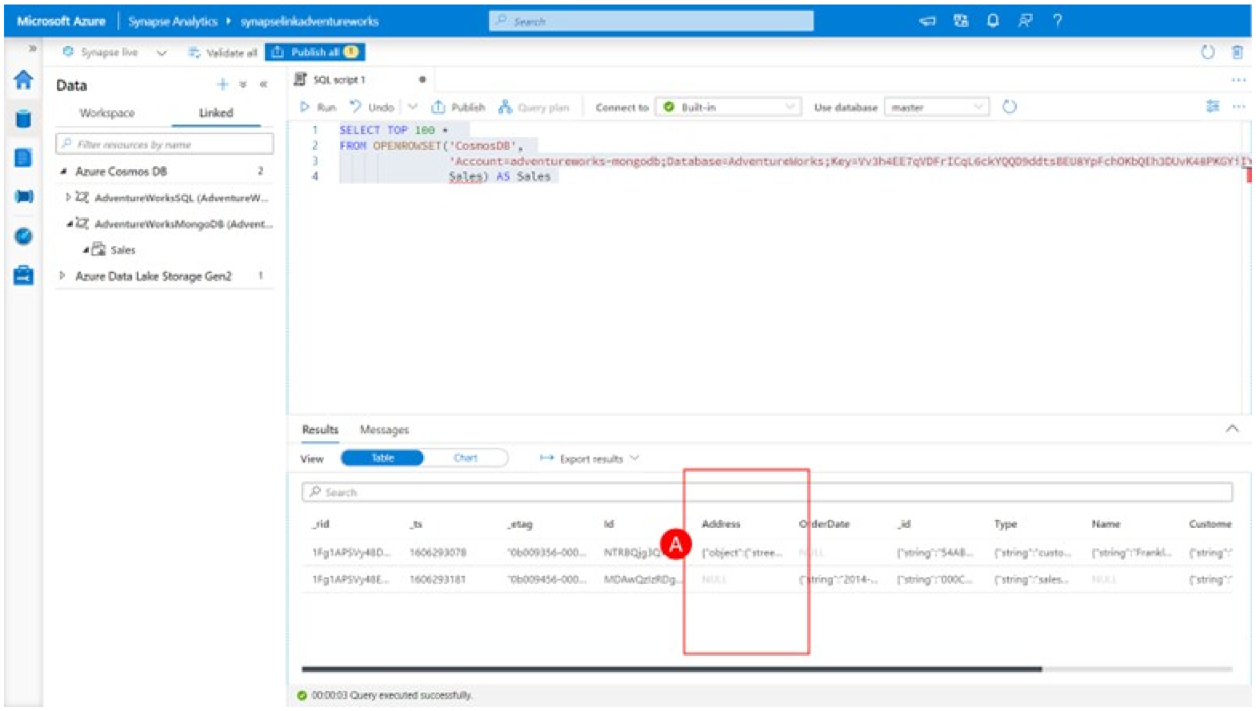
4. Click on **Select TOP 100 rows (4)**, to load a SQL script window to retrieve the top 100 records from the Linked Server and its associated analytical store.

You will see that the SQL script template requires the Azure Cosmos DB account key **(5)** for the account we are trying to connect to.

5. You can retrieve this from the Azure Cosmos DB account by a. Click on the **Keys** in the left-hand menu b. Click the **copy icon** next to the PRIMARY KEY value

6. And paste the key value from the clipboard back into the query

7. Now we can click the **Run** button.



```
SELECT TOP 100 *
FROM OPENROWSET('CosmosDB',
                'Account=adventureworks-mongodb;Database=AdventureWorks;Key=Vv3h4EE7qVDFrICqL6ckYQQD9ddtsBEU8YpFchOKbQEh3DUvK48PKGY!I
                Sales) AS Sales
```

_rid	_ts	_etag	id	Address	OrderDate	_id	Type	Name	Custome
1Fg1APSVy48D...	1606293078	"0b009356-000...	NTRBQjg3Q...	{"object":{"stree...	NULL	{"string":"54AB...	{"string":"custo...	{"string":"Frankl...	{"string":"
1Fg1APSVy48E...	1606293181	"0b009456-000...	MDAwQzIzRDg...	NULL	{"string":"2014-...	{"string":"000C...	{"string":"sales...	NULL	{"string":"

00:00:03 Query executed successfully.

Viewing JSON objects in the MondoDB query results.
You should almost immediately see the query begin to execute and then shortly thereafter receive back a result set.

Note That for records where data was not defined, such as the name column for the salesOrder record we get back a null value and that the other column values are returned as JSON and expanded to include the properties data type.