Data Engineering in the Cloud

Hybrid Transactional Analytical Processing

Xuemao Zhang
East Stroudsburg University

January 18, 2025

Outline

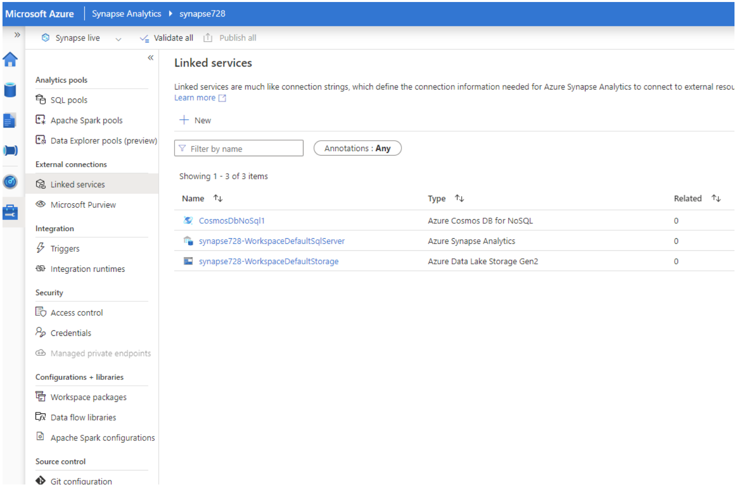
- In the labs, we Query Azure Cosmos DB with Apache Spark Pool and Serverless SQL Pools
- Lab 1: Querying Azure Cosmos DB with Apache Spark Pool
- Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

Prererquisites

- Prerequisites (10-15 minutes):
  - Create a datalake (storage account with hierarchical namespace enabled)
    - To lower the cost, you may choose `Redundancy` as `LRS`
    - Create a container in the storage account
  - Create an Azure Synapse workspace and
    - create a Apache spark pool in the `Manage` tab
  - Create an Azure Cosmos DB database with a containter
    - with `Analytical store` enabled and then create two key-value items
  - In the Synapse Studio, go to the `Manage` tab to add a new linked service for `Azure Cosmos DB for NoSQL`
  - Go to the `Data` tab of the Synapse Studio, click on + and choose "Integration dataset" to create a dataset.
    - We do not need the Copy pipeline

Lab 1: Querying Azure Cosmos DB with Apache Spark Pool

- ❶ Open "synapse studio", go to "Manage" tab, and add a new Cosmos DB `linked service`.



Linked services

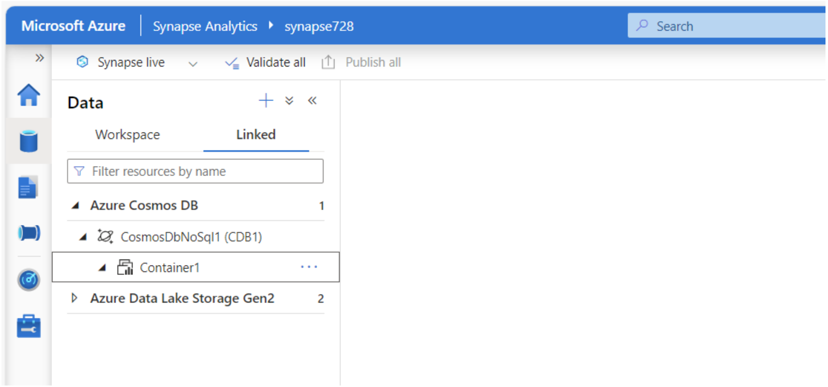
Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external resou
Learn more

Showing 1 - 3 of 3 items

Name	Type	Related
CosmosDbNoSql1	Azure Cosmos DB for NoSQL	0
synapse728-WorkspaceDefaultSqlServer	Azure Synapse Analytics	0
synapse728-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2	0

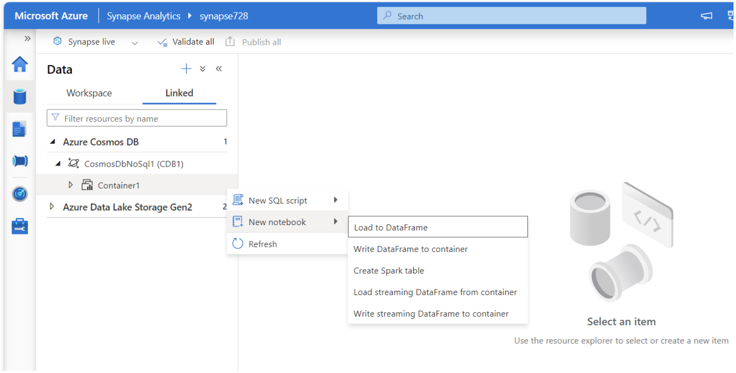
Lab 1: Querying Azure Cosmos DB with Apache Spark Pool

- 2 Go to “Data” tab, and choose “Linked”.



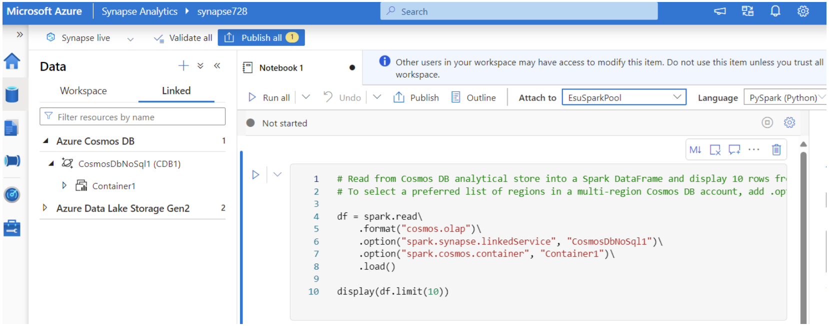
Lab 1: Querying Azure Cosmos DB with Apache Spark Pool

- 3. Expand the Azure Cosmos DB, choose the container “Container1” created in the Cosmos DB, click on three dots, choose “New notebook”, and choose “Load to Dataframe”.



Lab 1: Querying Azure Cosmos DB with Apache Spark Pool

- 4 Attach the spark pool to the notebook.



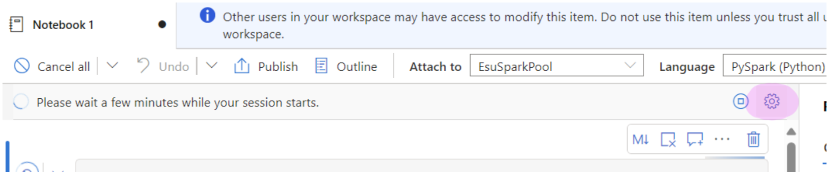
Other users in your workspace may have access to modify this item. Do not use this item unless you trust all workspace.

Attach to: EsuSparkPool
Language: PySpark (Python)

```
# Read from Cosmos DB analytical store into a Spark DataFrame and display 10 rows fr
# To select a preferred list of regions in a multi-region Cosmos DB account, add .op

df = spark.read\
    .format("cosmos.olap")\
    .option("spark.synapse.linkedService", "CosmosDbNoSql1")\
    .option("spark.cosmos.container", "Container1")\
    .load()

display(df.limit(10))
```

Lab 1: Querying Azure Cosmos DB with Apache Spark Pool

- 5 Click on "Run All". It takes some time to run.
  - If you see similar message, then update the configuration of the spark session



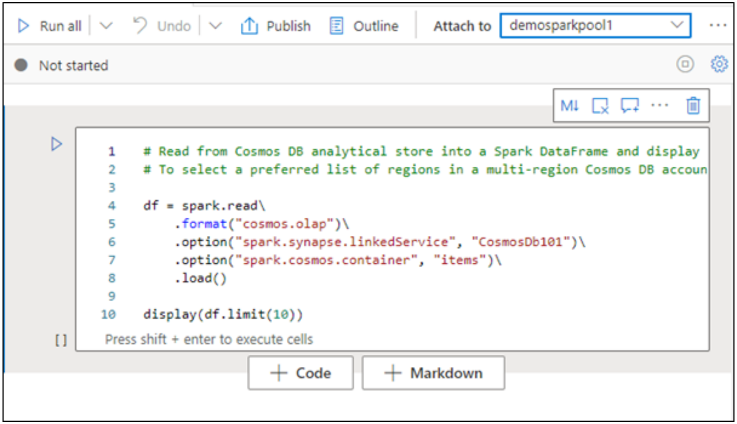
Notebook 1

Other users in your workspace may have access to modify this item. Do not use this item unless you trust all u workspace.

Please wait a few minutes while your session starts.

```
InvalidHttpRequestToLivy: Your Spark job requested 48 vcores.
However, the workspace has a 12 core limit. Try reducing the
numbers of vcores requested or increasing your vcore quota.
Quota can be increased using Azure Support request https://learn.microsoft.com/en-us
```

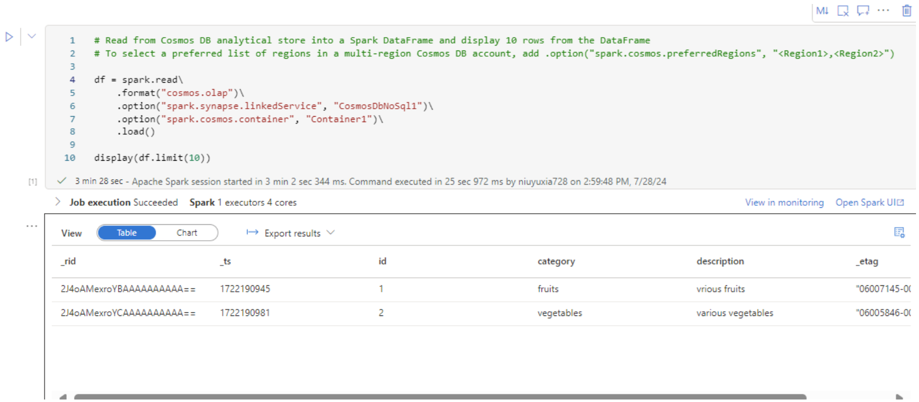
Lab 1: Querying Azure Cosmos DB with Apache Spark Pool



```
# Read from Cosmos DB analytical store into a Spark DataFrame and display
# To select a preferred list of regions in a multi-region Cosmos DB accoun

df = spark.read\
    .format("cosmos.olap")\
    .option("spark.synapse.linkedService", "CosmosDb101")\
    .option("spark.cosmos.container", "items")\
    .load()

display(df.limit(10))
```

Lab 1: Querying Azure Cosmos DB with Apache Spark Pool

- 6 You must be able to see the output as given below:



```
# Read from Cosmos DB analytical store into a Spark DataFrame and display 10 rows from the DataFrame
# To select a preferred list of regions in a multi-region Cosmos DB account, add .option("spark.cosmos.preferredRegions", "<Region1>,<Region2>")

df = spark.read\
    .format("cosmos.olap")\
    .option("spark.synapse.linkedService", "CosmosDbNoSql1")\
    .option("spark.cosmos.container", "Container1")\
    .load()

display(df.limit(10))
```

[1] ✓ 3 min 28 sec - Apache Spark session started in 3 min 2 sec 344 ms. Command executed in 25 sec 972 ms by niuyuxia728 on 2:59:48 PM, 7/28/24

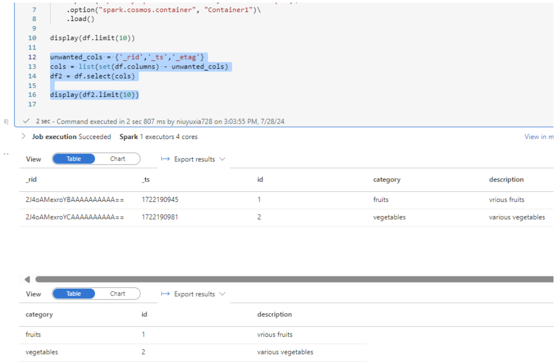
Job execution Succeeded **Spark** 1 executors 4 cores

_rid	_ts	id	category	description	_etag
2J4oAMexroYBAAAAAAAAAA==	1722190945	1	fruits	vrious fruits	"06007145-0(
2J4oAMexroYCAAAAAAAAAA==	1722190981	2	vegetables	various vegetables	"06005846-0(

Lab 1: Querying Azure Cosmos DB with Apache Spark Pool

- 7. You can remove the unwanted column using the below command:

```
unwanted_cols = {'_rid','_ts','_etag'}
cols = list(set(df.columns) - unwanted_cols)
df2 = df.select(cols)

display(df2.limit(10))
```



```
    .option("spark.cosmos.container", "Container1")\
    .load()

display(df.limit(10))

unwanted_cols = {'_rid','_ts','_etag'}
cols = list(set(df.columns) - unwanted_cols)
df2 = df.select(cols)

display(df2.limit(10))
```

8] ✓ 2 sec - Command executed in 2 sec 807 ms by niuyuxia728 on 3:03:55 PM, 7/28/24

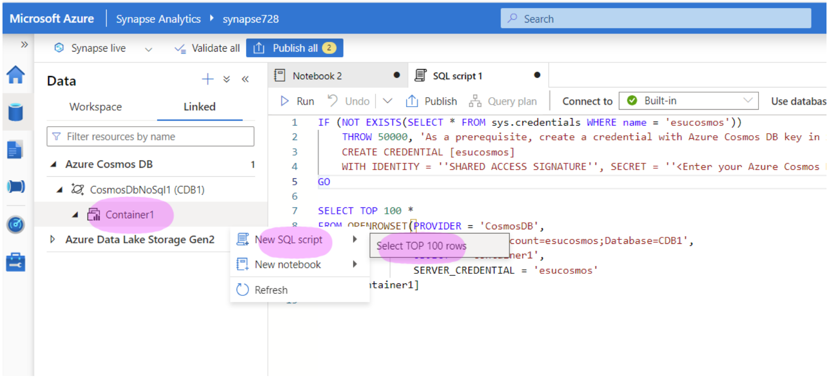
Job execution Succeeded Spark 1 executors 4 cores

_rid	_ts	id	category	description
2J4oAMexroYBAAAAAAAAAA==	1722190945	1	fruits	vrious fruits
2J4oAMexroYCAAAAAAAAAA==	1722190981	2	vegetables	various vegetables

category	id	description
fruits	1	vrious fruits
vegetables	2	various vegetables

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

- 1. Go to "Data" in the Synapse Studio and right click `Container1` to add an SQL script.



```
IF (NOT EXISTS(SELECT * FROM sys.credentials WHERE name = 'esucosmos'))
    THROW 50000, 'As a prerequisite, create a credential with Azure Cosmos DB key in
    CREATE CREDENTIAL [esucosmos]
    WITH IDENTITY = ''SHARED ACCESS SIGNATURE'', SECRET = ''<Enter your Azure Cosmos
GO

SELECT TOP 100 *
FROM OPENROWSET(PROVIDER = 'CosmosDB',
                        count=esucosmos;Database=CDB1',
                        ner1',
                SERVER_CREDENTIAL = 'esucosmos'
       ntainer1]
```

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

- 2 Run the first block query see if a credential is needed.

```
IF (NOT EXISTS(SELECT * FROM sys.credentials WHERE name = 'esucosmos'))
    THROW 50000, 'As a prerequisite, create a credential with Azure Cosmos DB key in SECRET option:
    CREATE CREDENTIAL [esucosmos]
    WITH IDENTITY = ''SHARED ACCESS SIGNATURE'', SECRET = ''a4WXnMntqa0yJfMR4D84rcfNgDCDR33dfo70PMeum8n0vNeshAEqkD0TdyycfyHUFyrDrQUAb
GO
```

- 3 Create a credential first

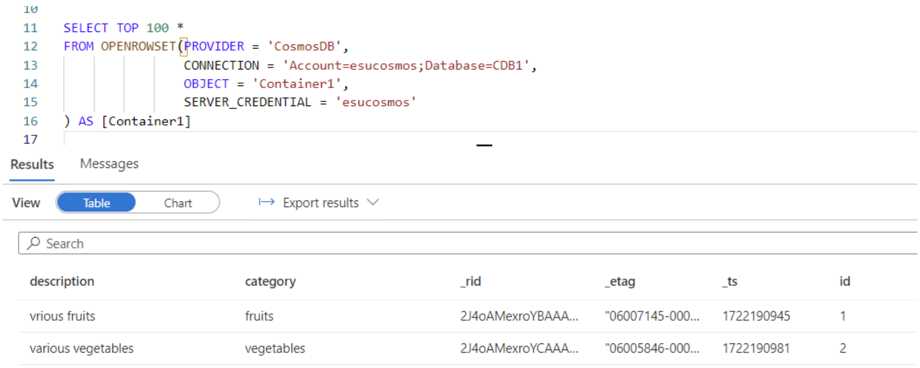
```
CREATE CREDENTIAL [esucosmos]
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
SECRET = 'a4WXnMntqa0yJfMR4D84rcfNgDCDR33dfo70
PMeum8n0vNeshAEqkD0TdyycfyHUFyrDrQUAbWUNACDb8oZYVg==';
```

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

- 4. Run the query:

```
SELECT TOP 100 *
FROM OPENROWSET( PROVIDER = 'CosmosDB',
                CONNECTION = 'Account=esucosmos;Database=CDB1',
                OBJECT = 'Container1',
                SERVER_CREDENTIAL = 'esucosmos'
) AS [Container1]
```

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools



```sql
SELECT TOP 100 *
FROM OPENROWSET(PROVIDER = 'CosmosDB',
                CONNECTION = 'Account=esucosmos;Database=CDB1',
                OBJECT = 'Container1',
                SERVER_CREDENTIAL = 'esucosmos'
) AS [Container1]
```

description	category	_rid	_etag	_ts	id
vrious fruits	fruits	2J4oAMexroYBAAA...	"06007145-000...	1722190945	1
various vegetables	vegetables	2J4oAMexroYCAAA...	"06005846-000...	1722190981	2

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

- 5. Now let's creat a database in Azure Synapse

```
-- Create the Profiles database if it does not exist
USE master;
GO
IF DB_ID(N'Profiles') IS NULL
BEGIN
    CREATE DATABASE Profiles;
END
GO

-- Switch to the Profiles database
USE Profiles;
GO

-- Drop the items view if it exists
DROP VIEW IF EXISTS items;
GO
```

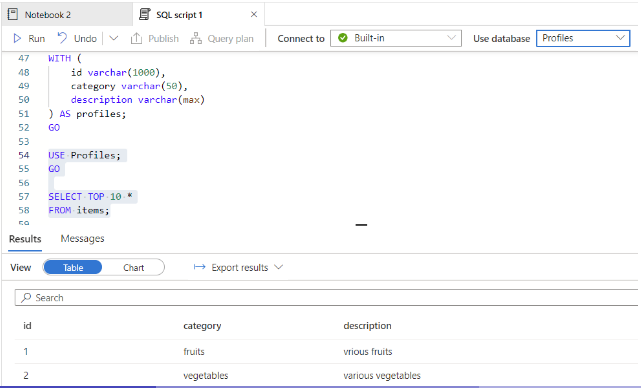
Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

```
-- Create the items view to access Cosmos DB
CREATE VIEW items
AS
SELECT
    *
FROM OPENROWSET(
PROVIDER = 'CosmosDB',
CONNECTION = 'Account=esucosmos;Database=CDB1',
OBJECT = 'Container1',
SERVER_CREDENTIAL = 'esucosmos'
)
WITH (
    id varchar(1000),
    category varchar(50),
    description varchar(max)
) AS profiles;
GO
```

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

- 6 Check the new database

```
USE Profiles;
GO

SELECT TOP 10 *
FROM items;
```

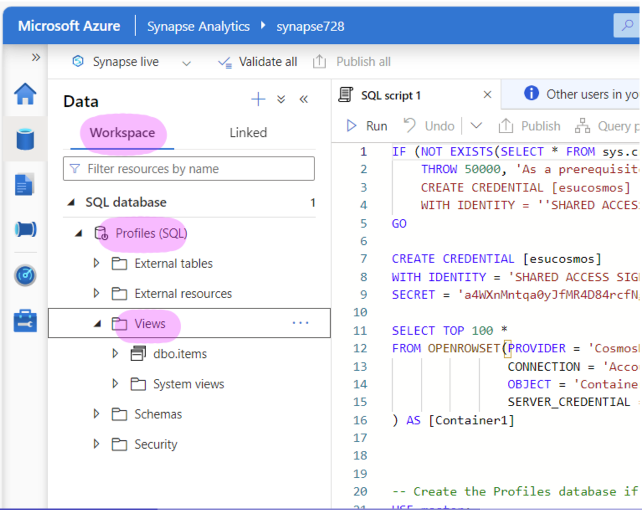


```
WITH (
    id varchar(1000),
    category varchar(50),
    description varchar(max)
) AS profiles;
GO

USE Profiles;
GO

SELECT TOP 10 *
FROM items;
```

id	category	description
1	fruits	vrious fruits
2	vegetables	various vegetables

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

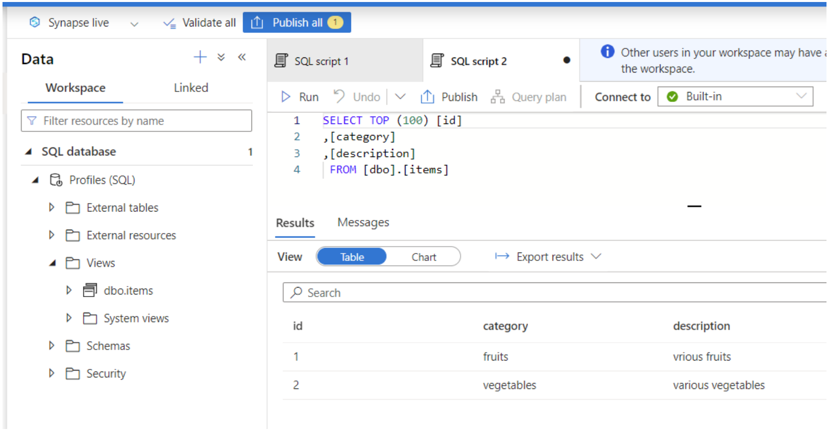
- ❼ Check the new database: Refresh the Synapse Studio, then go to “Data” tab, expand the “Profiles (SQL)” and under “Views” there must be a view created.



```
IF (NOT EXISTS(SELECT * FROM sys.c
    THROW 50000, 'As a prerequisit
    CREATE CREDENTIAL [esucosmos]
    WITH IDENTITY = ''SHARED ACCES
GO

CREATE CREDENTIAL [esucosmos]
WITH IDENTITY = 'SHARED ACCESS SIG
SECRET = 'a4WXnMntqa0yJfMR4D84rcfN

SELECT TOP 100 *
FROM OPENROWSET(PROVIDER = 'Cosmos
                CONNECTION = 'Acco
                OBJECT = 'Containe
                SERVER_CREDENTIAL
) AS [Container1]



-- Create the Profiles database if
```

Lab 2: Querying Azure Cosmos DB with Serverless SQL Pools

- 8 Click on "dbo.items", choose "New SQL script", click "Select Top 100 rows" and you will be able to see the auto generated query. Run the query.



Other users in your workspace may have a
the workspace.

```sql
SELECT TOP (100) [id]
,[category]
,[description]
 FROM [dbo].[items]
```

id	category	description
1	fruits	vrious fruits
2	vegetables	various vegetables

License