Validate connectivity from Spark

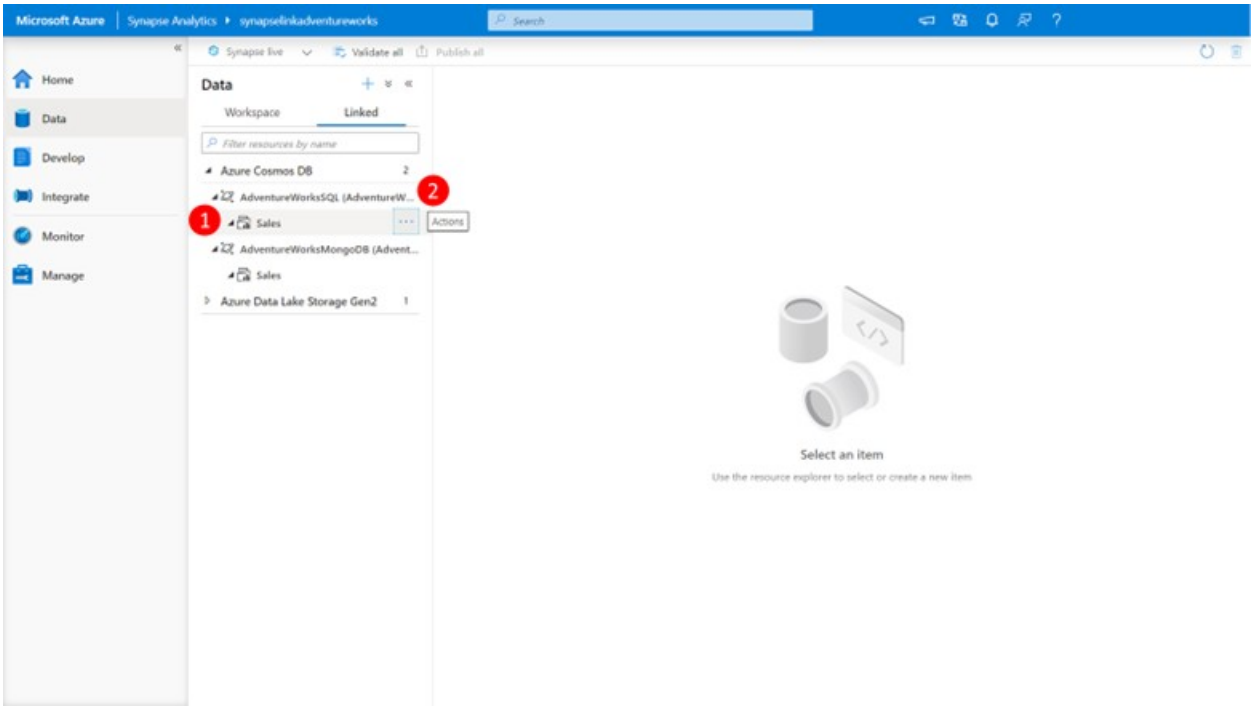
In this reading you can see the steps involved in the process of validating connectivity from Spark.

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.

Spark Queries for Cosmos DB Core (SQL) API

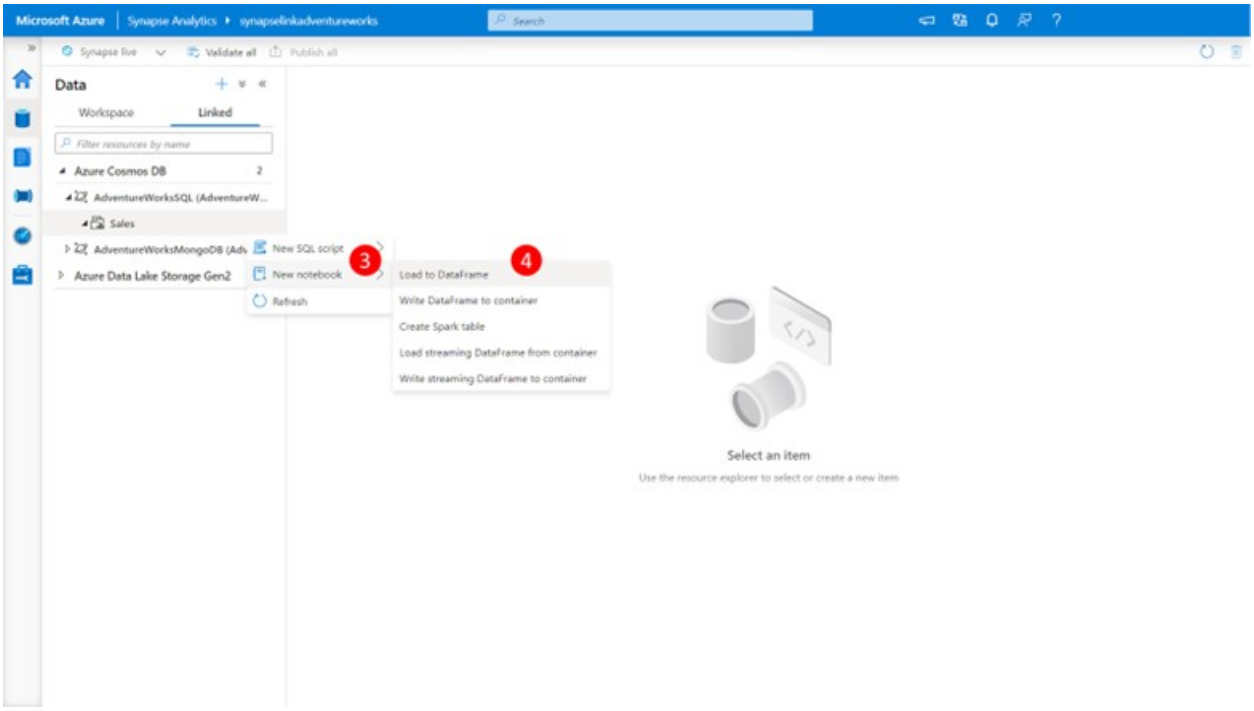
Let's now connect to our Cosmos DB Cosmos DB Core (SQL) API analytical store using Spark and retrieve some data by performing the following steps:



Retrieving data from a Cosmos DB analytical store.

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

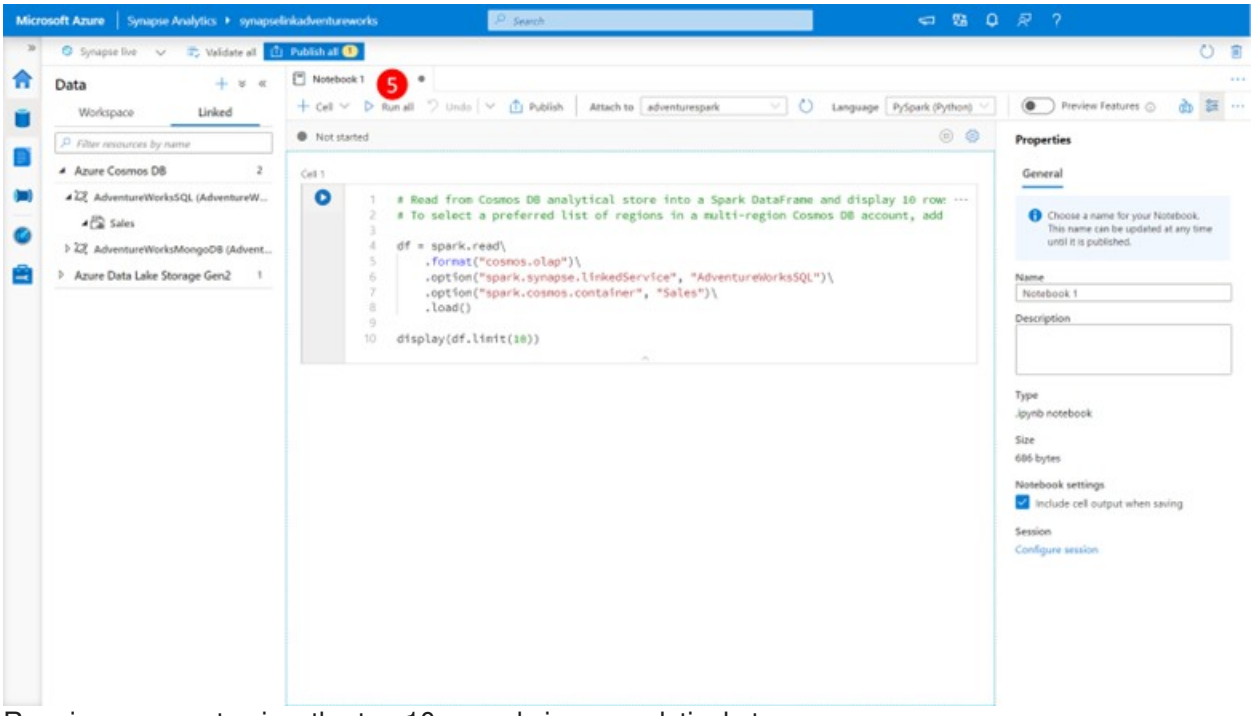
2. Click on the **Actions ellipsis "..."**



Loading data into a DataFrame.

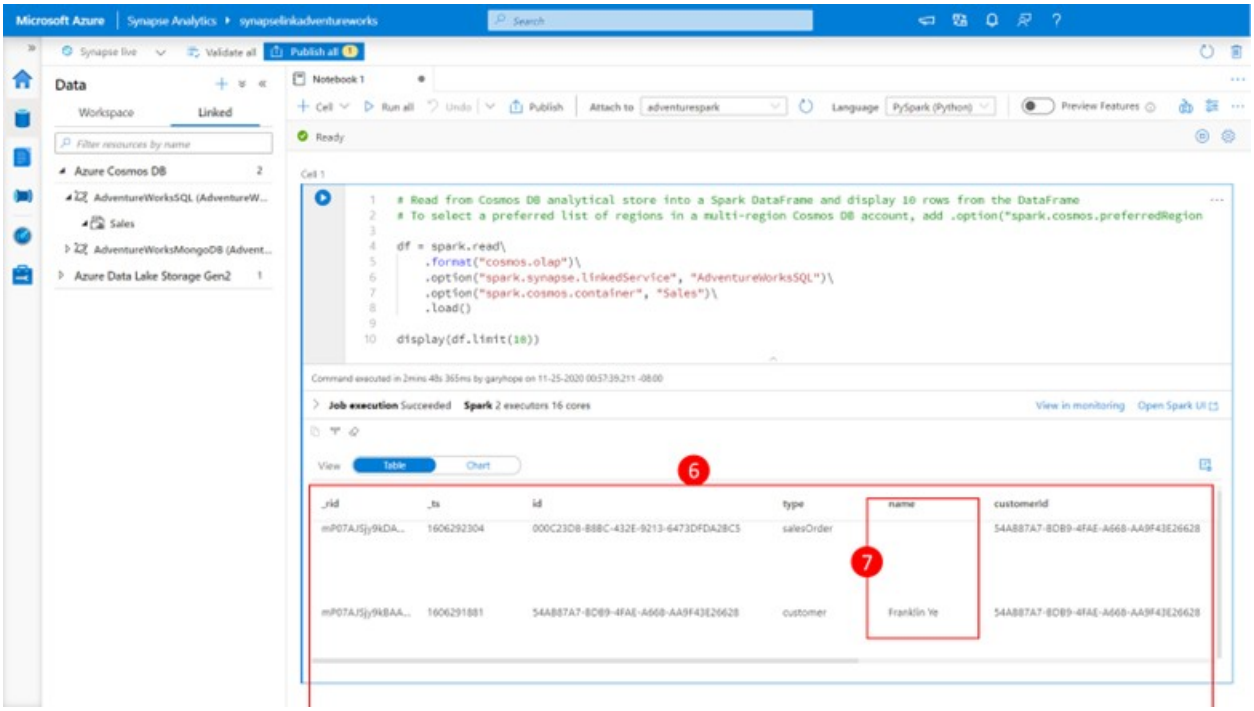
3. Click on **new notebook** to expose the list of notebooks actions.

4. Click on **Load to DataFrame**, to load a prepopulated notebook with a spark query to retrieve the top 10 records from the Linked Server and its associated analytical store.



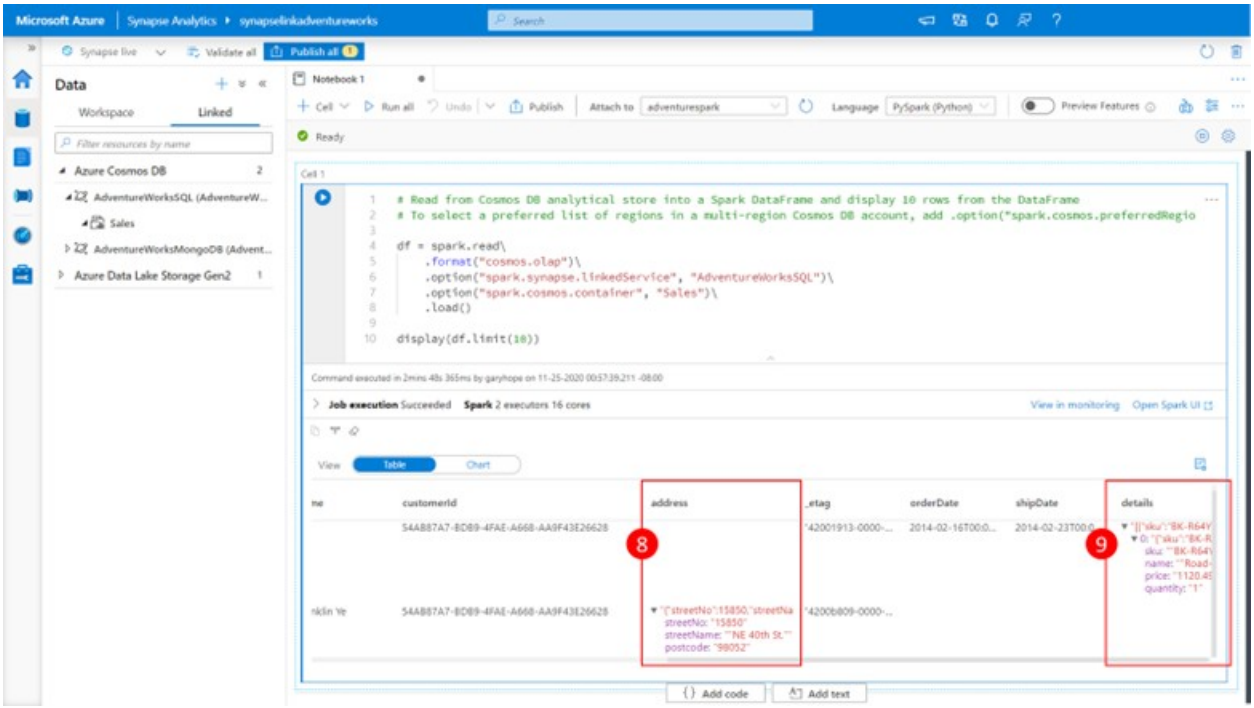
Running a query to view the top 10 records in an analytical store.

5. Click the **Run All** button on the ribbon to execute the notebook.



Viewing the top 10 records in an analytical store.
You should almost immediately see the query begin to execute and then shortly thereafter receive back a result set **(6)**

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



Viewing JSON results in an analytical store.
6. Scroll right though the result set

You will see that columns that contain JSON objects, such as address **(8)** and JSON arrays such as detail **(9)** has the JSON as their column content.

Spark queries for Azure Cosmos DB API for MongoDB

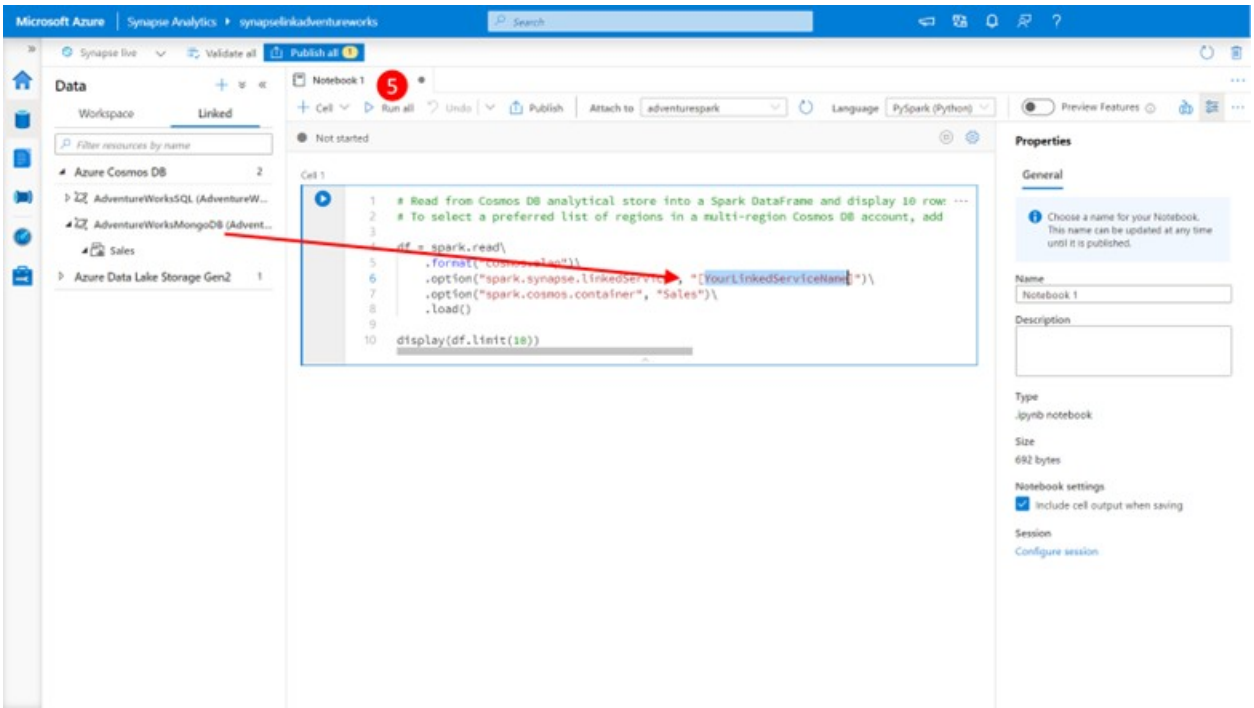
Let's now connect to our Azure Cosmos DB API for MongoDB analytical store using Spark and retrieve some data by performing the following steps:

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

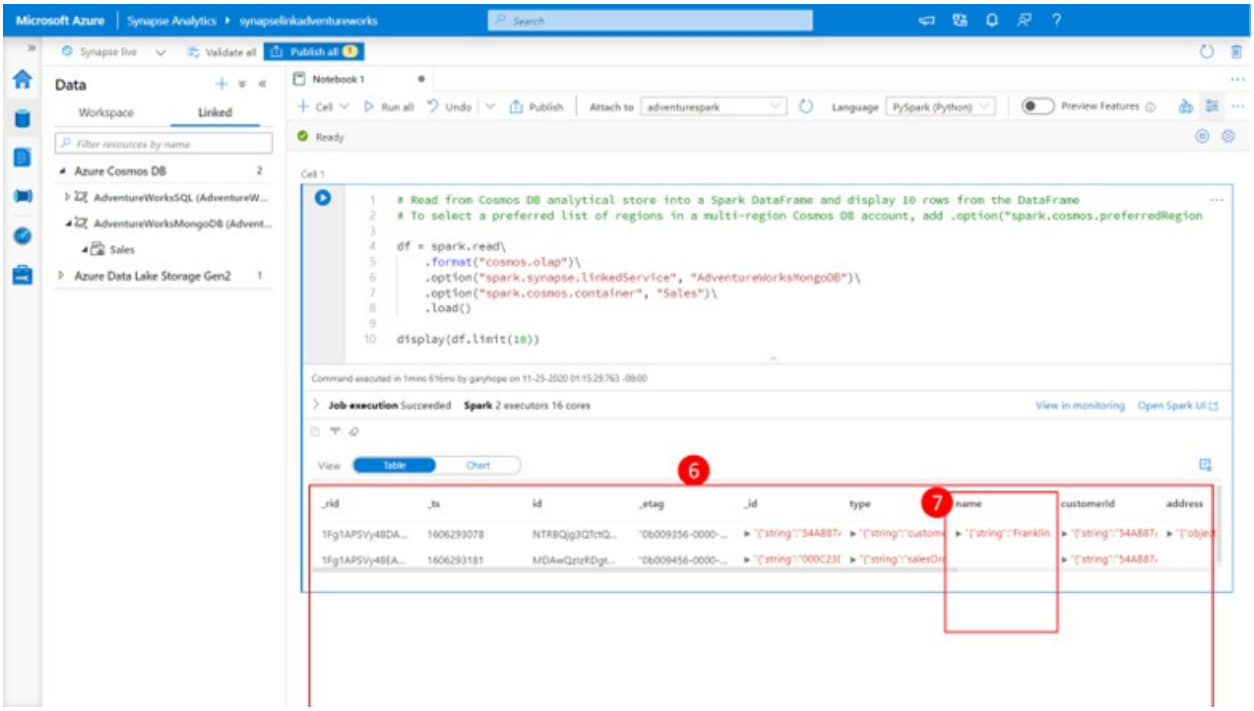
2. Click on the **Actions ellipsis "…"**

3. Click on **new notebook** to expose the list of notebooks actions.

4. Click on **Load to DataFrame**, to load a prepopulated notebook with a spark query to retrieve the top 10 records from the Linked Server and its associated analytical store.

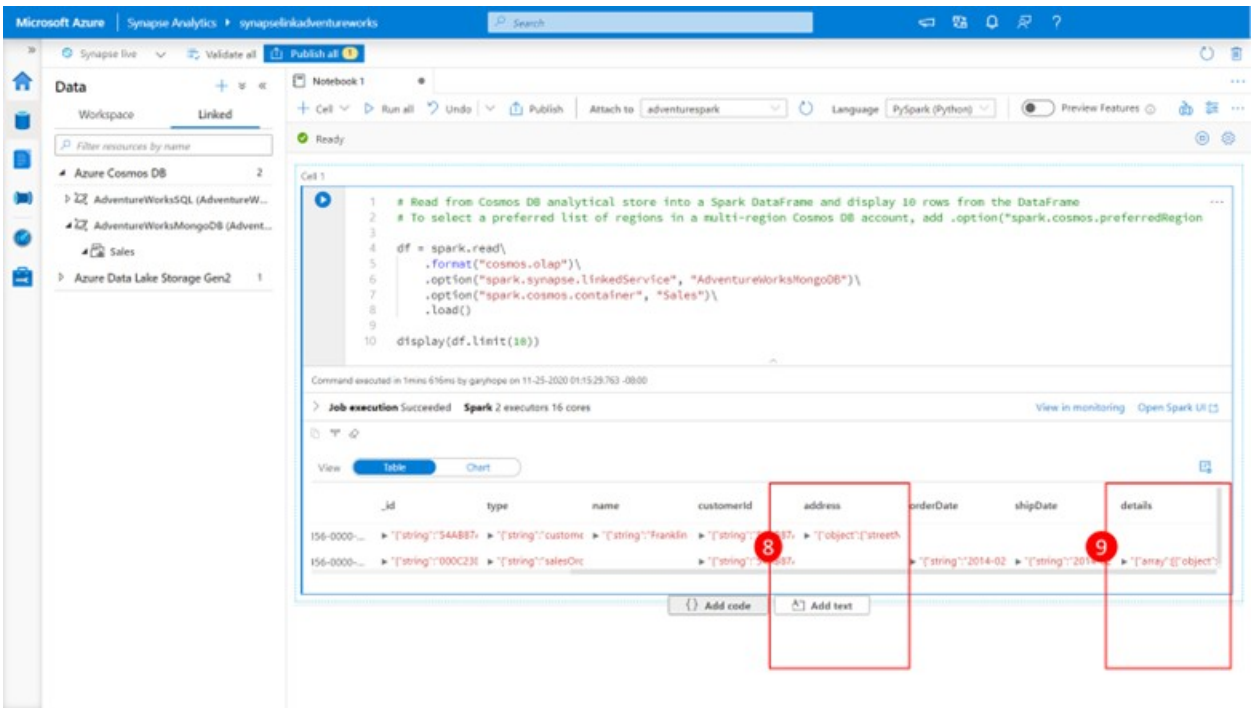


Defining a linked service in a Spark query.
5. Update the linked service name and click the **Run All** button on the ribbon to execute the notebook.



Executing a Spark query.
You should almost immediately see the query begin to execute and then shortly thereafter receive back a result set **(6)**

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 name column now contains a JSON fragment with both the data type and value since the MongoDB API uses full fidelity schema mode by default.



Viewing Spark query results.

6. Scroll right though the result set

You will see that columns that contain JSON objects, such as address **(8)** and JSON arrays such a detail **(9)** has the JSON as their column content as well, however this too is expanded to include the data type information.