

Perform cross container queries

Note In this reading you can see the steps involved in the process of performing cross container queries.

Let's explore the second of our two containers; the SalesOrder container. This container hosts the AdventureWorks database within the Azure Cosmos DB API for MongoDB and contains sales order information.

For reference here is a sample of a sales order JSON document from the SalesOrder container:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
{
  "_id" : "F0FF0BA8-CEF4-454C-9173-FA7C546F7179",
  "customerId" : "A852CB99-DAA1-4348-A86F-622D4478A7D0",
  "orderDate" : "2013-07-11T00:00:00",
  "shipDate" : "2013-07-18T00:00:00",
  "details" : [
    {
      "sku" : "LJ-0192-M",
      "name" : "Long-Sleeve Logo Jersey, M",
      "price" : 49.99,
      "quantity" : 1
    },
    {
      "sku" : "SH-W890-M",
      "name" : "Women's Mountain Shorts, M",
```

```

        "price" : 69.99,
        "quantity" : 1
    }
}
]
}

```

Paste the following SQL into the query pane.

```

SELECT top 10 *
FROM OPENROWSET('CosmosDB',
                 'Account=adventureworks-
mongodb;Database=AdventureWorks;Key=v2mtZ85W0AMCv1ZrY7j...4g==',
                 SalesOrder) As SalesOrders

```

1
2
3
4

The screenshot shows the Microsoft Azure Synapse Analytics interface. The top bar indicates the environment is 'Synapse live' and the database is 'SynapseLinkAdventureWorks'. The query pane contains the following SQL code:

```

1 SELECT top 10 *
2 FROM OPENROWSET('CosmosDB',
3                  'Account=adventureworks-
4                  mongodb;Database=AdventureWorks;Key=v2mtZ85W0AMCv1ZrY7j...4g==',
5                  SalesOrder) As SalesOrders

```

The results pane shows a table with the following columns: ShipDate, _id, OrderDate, CustomerId, and Details. The first 10 rows of data are displayed. Red boxes labeled 'A' and 'B' highlight specific parts of the results.

ShipDate	_id	OrderDate	CustomerId	Details
5MEU...	["string":"2012-12-21T00:00:00"]	["string":"2012-12-14T00:00:00"]	["string":"38532...	["array":["object":{"sku":["string":"BK-M688-38"],"name":["string":"Mountain-200 Black 38"],"...
U2REML...	["string":"2012-06-06T00:00:00"]	["string":"2012-05-30T00:00:00"]	["string":"37597...	["array":["object":{"sku":["string":"FR-R388-52"],"name":["string":"LL Road Frame - Black, 52"],"...
gzNklt...	["string":"2013-10-30T00:00:00"]	["string":"2013-10-23T00:00:00"]	["string":"63171...	["array":["object":{"sku":["string":"CA-1098"],"name":["string":"AWC Logo Cap"],"price":["float...
ICQlgtQ...	["string":"2013-07-18T00:00:00"]	["string":"2013-07-11T00:00:00"]	["string":"A852...	["array":["object":{"sku":["string":"LI-0192-M"],"name":["string":"Long-Sleeve Logo Jersey, M"],"...
3SMERIN...	["string":"2011-06-13T00:00:00"]	["string":"2011-06-06T00:00:00"]	["string":"930F5...	["array":["object":{"sku":["string":"BK-R93R-44"],"name":["string":"Road-150 Red, 44"],"price":["...
4Mzgt...	["string":"2014-02-23T00:00:00"]	["string":"2014-02-16T00:00:00"]	["string":"DAB9...	["array":["object":{"sku":["string":"PK-7098"],"name":["string":"Patch Kit/8 Patches"],"price":["fl...
3yNOUT...	["string":"2013-11-25T00:00:00"]	["string":"2013-11-18T00:00:00"]	["string":"BEB68...	["array":["object":{"sku":["string":"FE-6654"],"name":["string":"Fender Set - Mountain"],"price"...
ISNzkt...	["string":"2013-10-18T00:00:00"]	["string":"2013-10-11T00:00:00"]	["string":"9DB7...	["array":["object":{"sku":["string":"SH-W890-L"],"name":["string":"Women's Mountain Shorts, ...
0MjEtQ...	["string":"2013-10-12T00:00:00"]	["string":"2013-10-05T00:00:00"]	["string":"306E6...	["array":["object":{"sku":["string":"TI-T723"],"name":["string":"Touring Tire"],"price":["float64"...
ZBNjYt...	["string":"2013-08-07T00:00:00"]	["string":"2013-07-31T00:00:00"]	["string":"BAA0...	["array":["object":{"sku":["string":"FR-M215-40"],"name":["string":"LL Mountain Frame - Silver...

At the bottom of the interface, a status message indicates: '00:00:03 Query executed successfully.'

Execute a cross-container query in Azure Synapse Studio
Click **run**.

You are presented with a result set of the first 10 rows. It is a row-based representation of the documents contained within the SalesOrder containers analytical store using Azure Cosmos Core API account for MongoDB. As a result, the data is represented using the full fidelity schema representation by default.

All top-level properties of the document are represented as columns with the associated property values. All properties are **represented as a structure of the type of values assigned to the properties; and the values themselves, (A) and (B)**. For complex types such

as objects and arrays, they will remain embedded within the structure, and expanded to include type encapsulation of each of their property values.

In this example, ignoring the system document properties for now, the `_id`, `customerId`, `orderDate`, and `shipDate` are all strings and have a **type encapsulation of string (A)**. The **details property is an embedded array (B)**, in turn with embedded properties `sku`, `name`, `price`, and `quantity` of each array element object.

Let's use what we observed here, and what we learned earlier about using `OPENROWSET` to combine this data with our `Customers` view that we created previously, to create a new `SalesOrders` view.

Create the following script:

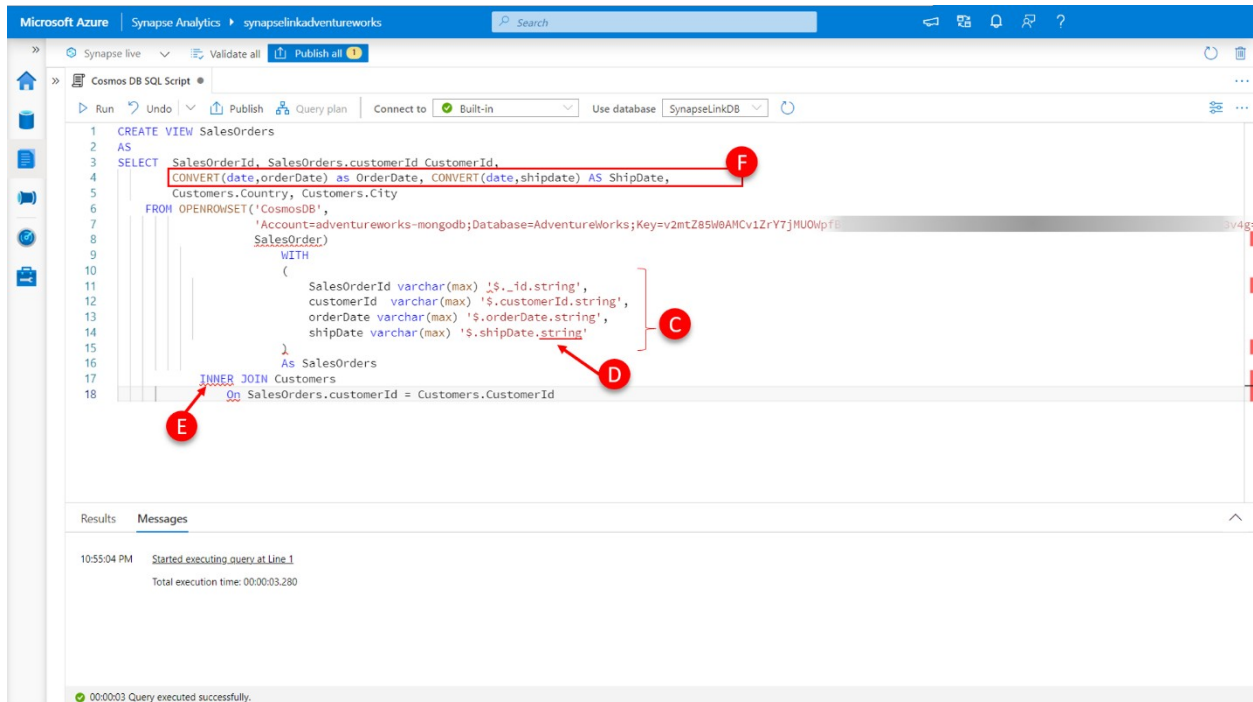
```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

CREATE VIEW SalesOrders
AS
SELECT SalesOrderId, SalesOrders.customerId CustomerId,
       CONVERT(date,orderDate) as OrderDate, CONVERT(date,shipdate) AS ShipDate
,
       Customers.Country, Customers.City
FROM OPENROWSET('CosmosDB',
                'Account=adventureworks-
mongodb;Database=AdventureWorks;Key=v2mtZ85W0AMCv1Zr...D3v4g==',
                SalesOrder)
WITH
(
    SalesOrderId varchar(max) '$._id.string',
    customerId   varchar(max) '$.customerId.string',
```

```

        orderDate varchar(max) '$.orderDate.string',
        shipDate varchar(max) '$.shipDate.string'
    )
    As SalesOrders
INNER JOIN Customers
    On SalesOrders.customerId = Customers.CustomerId

```



Create a cross-container view in Azure Synapse Studio

Here you can see we are using the **WITH clause (C)** again to specify the path to our property values, notably **including the data type suffix (D)** of the property, as part of the path to the property, in order to access the values when using the full fidelity schema. In this example, we need to specify “.shipDate.string” rather than just “.shipDate”.

We are joining it back to our Customers view using the **CustomerId key present in both (E)**, and doing some explicit type conversion for our date values using the CONVERT function to project them as SQL date data types.

Let's see what this view returns by:

```
SELECT top 10 * FROM SalesOrders
```

Microsoft Azure | Synapse Analytics | synapselinkadventureworks

Synapse live | Validate all | Publish all

Cosmos DB SQL Script

Run | Undo | Publish | Query plan | Connect to Built-in | Use database SynapseLinkDB

```
1
2 SELECT top 10 * FROM SalesOrders
```

Results | Messages

View | Table | Chart | Export results

SalesOrderId	CustomerId	OrderDate	ShipDate	Country	City
967E2756-5A9C-4F7C-A204-22C...	00015962-05D8-46E5-A2D9-AE6...	2013-08-22T00:00:00.0000000	2013-08-29T00:00:00.0000000	AU	East Brisbane
A1C87FDD-1823-463A-9787-E46...	000C5E58-E5B6-4958-83C0-B82...	2011-10-13T00:00:00.0000000	2011-10-20T00:00:00.0000000	US	Burbank
33D323A1-1523-449D-86D9-855...	00182FAD-5C64-4A77-81C8-DEF...	2014-06-30T00:00:00.0000000	2014-07-07T00:00:00.0000000	US	Redwood City
7FB380E2-A2E5-46D0-8EF3-C6EF...	001C8C08-9891-47A5-A198-877...	2014-03-17T00:00:00.0000000	2014-03-24T00:00:00.0000000	DE	Hamburg
E76DA6A1-AF1B-4B71-87B6-816...	00307EA7-625F-4C5D-A99F-32B...	2013-11-19T00:00:00.0000000	2013-11-26T00:00:00.0000000	AU	Melbourne
6A7C863B-3F98-426D-9FC3-719...	0031725D-6D6C-40C8-943F-4C2...	2013-11-13T00:00:00.0000000	2013-11-20T00:00:00.0000000	DE	Paderborn
F01C558D-29F3-4D56-8AA7-AB...	00528816-95B9-45DE-964A-C89...	2014-05-20T00:00:00.0000000	2014-05-27T00:00:00.0000000	US	Seattle
630BCA3F-FFA0-4E58-8B78-426...	0092801F-9D23-4818-A8AD-3F6...	2014-06-19T00:00:00.0000000	2014-06-26T00:00:00.0000000	US	Woodland Hills
9E46328D-83C9-4428-8561-FF71...	00A61365-0110-4469-81C9-3F0E...	2013-07-08T00:00:00.0000000	2013-07-15T00:00:00.0000000	DE	Berlin
5528F49A-D031-4C64-8FF1-E04...	00D81835-F5F7-4970-B650-03C...	2013-11-21T00:00:00.0000000	2013-11-28T00:00:00.0000000	GB	Billerica

00:00:18 Query executed successfully.

Query a view in Azure Synapse Studio

Here you can see the results from our SalesOrders view, returning a flattened columnar view of the JSON documents that reside in the SalesOrder container within the Azure Cosmos DB API for MongoDB account.