

# Perform complex queries with JSON data

**Note** In this reading you can see the steps involved in performing complex queries with JSON data.

Let's focus on extracting the data from sales order details for now. To do that we are going to want to look at the SalesOrderId (contained within the `_id` property of the document) and the details property that contains the array of sales order details.

Paste the following SQL into the query pane.

```
1
2
3
4
5
6
7
8
SELECT TOP(10) SalesOrderId, details
FROM OPENROWSET('CosmosDB',
                 'Account=adventureworks-
mongodb;Database=AdventureWorks;Key=v2mtZ85W0AMCv1ZrY7jMU0WpfBTi1BrUz0Y3Rwmvj9SXS
SIKDU7EQVu5kdEMcwAQfvJBnmHSMxy50c3gD3v4g==',
                 SalesOrder)
WITH
(   SalesOrderId varchar(max) '$._id',
    details varchar(max) '$.details'
) As SalesOrders
```

Microsoft Azure | Synapse Analytics | synapselinkadventureworks

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

```

1 SELECT TOP(10) SalesOrderId, details
2 FROM OPENROWSET('CosmosDB',
3 'Account=adventureworks-mongodb;Database=AdventureWorks;Key=v2mtZ85W0AMCv1ZrY7jMUOWpfBTi1BrUz0Y3Rwmvj9SXS
4 SalesOrder)
5 WITH
6 ( SalesOrderId varchar(max) '$._id',
7 details varchar(max) '$.details'
8 ) As SalesOrders
9

```

Results | Messages

View | Table | Chart | Export results

SalesOrderId	Details
["string":"F8BC4659-6AD8-41FE-B...	["array":["object":{"sku":{"string":"RA-H123"},"name":{"string":"Hitch Rack - 4-Bike"},"price":{"float64":37.152},"quantity":{"int32":1}}]]
["string":"E32722EC-38EF-4378-0...	["array":["object":{"sku":{"string":"HB-M763"},"name":{"string":"ML Mountain Handlebars"},"price":{"float64":469.794},"quantity":{"int32":1}}]]
["string":"E5EF46DA-955A-4115-A...	["array":["object":{"sku":{"string":"BK-R508-58"},"name":{"string":"Road-650 Black, 58"},"price":{"float64":469.794},"quantity":{"int32":3}}]]
["string":"D87ACEB9-C743-4F1B-...	["array":["object":{"sku":{"string":"PK-7098"},"name":{"string":"Patch Kit/8 Patches"},"price":{"float64":2.29},"quantity":{"int32":1}}]]
["string":"F6A83AA2-9908-4A47-...	["array":["object":{"sku":{"string":"TT-T092"},"name":{"string":"Touring Tire Tube"},"price":{"float64":4.99},"quantity":{"int32":1}}]]
["string":"F703774A-9CFC-49CD-...	["array":["object":{"sku":{"string":"BK-R93R-52"},"name":{"string":"Road-150 Red, 52"},"price":{"float64":359.27},"quantity":{"int32":1}}]]
["string":"DEC1D1F1-801D-448B-...	["array":["object":{"sku":{"string":"BK-R898-48"},"name":{"string":"Road-250 Black, 48"},"price":{"float64":2181.5625},"quantity":{"int32":1}}]]
["string":"D7010F17-C886-4773-8...	["array":["object":{"sku":{"string":"TI-T723"},"name":{"string":"Touring Tire"},"price":{"float64":28.99},"quantity":{"int32":1}}]]
["string":"EC3A5882-828E-4823-9...	["array":["object":{"sku":{"string":"SJ-0194-L"},"name":{"string":"Short-Sleeve Classic Jersey, L"},"price":{"float64":53.99},"quantity":{"int32":1}}]]
["string":"EC2D853-998E-4597-9...	["array":["object":{"sku":{"string":"HL-U509-R"},"name":{"string":"Sport-100 Helmet, Red"},"price":{"float64":34.99},"quantity":{"int32":1}}]]

00:00:07 Query executed successfully.

Execute a JSON query in Azure Synapse Studio  
Click **run**.

Here we can see the **SalesOrderId column (A)** is returning a JSON fragment including the data type of the document `_id` property, in this case the string along with the property values. The **details column (B)** is also returning a JSON fragment in this case indicating that the data type is an **array (C)** and that it contains **multiple array elements (D)**.

Let's now access these properties by correctly specifying the path, including the type suffix for each property (in this case "string" for `_id` and "array" for the details attributes).

Paste the following SQL into the query pane.

```

SELECT TOP(10) SalesOrderId, details
FROM OPENROWSET('CosmosDB',
'Account=adventureworks-
mongodb;Database=AdventureWorks;Key=v2mtZ85W0AMCv1ZrY7jMUOWpfBTi1BrUz0Y3Rwmvj9SXS
SIKDU7EQVu5kdEMcwAQfvJBnmHSMxy50c3gd3v4g==',
SalesOrder)
WITH
( SalesOrderId varchar(max) '$._id.string',
details varchar(max) '$.details.array'

```

1  
2  
3  
4  
5  
6  
7  
8

## ) As SalesOrderDetails

The screenshot shows the Microsoft Azure Synapse Studio interface. The top bar indicates the environment is 'Synapse Analytics' and the database is 'synapselinkadventureworks'. The query editor shows a SQL query that uses `OPENROWSET` to connect to a Cosmos DB instance. The query selects the top 10 results from a subquery that joins `SalesOrder` and `SalesOrderDetails`. The results table is displayed below the query, showing two columns: `SalesOrderId` and `Details`. The `SalesOrderId` column contains unique identifiers, and the `Details` column contains JSON arrays of objects. Red callouts are used to highlight specific elements: 'E' points to the `As SalesOrderDetails` alias in the query; 'F' points to the `SalesOrderId` column header; 'G' points to the `Details` column header; 'H' points to the opening and closing brackets of a JSON array; and 'I' points to an object within the JSON array.

Accessing JSON Properties in a query in Azure Synapse Studio  
Click **run**.

Now, we can see that the `SalesOrderId` column contains just the appropriate value, and the details column just contains the JSON array, starting with the array designator “[” and ending with with “]” and can contain **multiple array elements (H)** of object type, as indicated in the JSON with **object type suffix (I)**.

Let's now create a `SalesOrderDetails` view.

Paste the following SQL into the query pane.

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

```

CREATE VIEW SalesOrderDetails
AS
SELECT SalesOrderId, SalesOrderArray.[key]+1 as SalesOrderLine, SKUCode, SKUName
,Price, Quantity
    FROM OPENROWSET('CosmosDB',
                    'Account=adventureworks-
mongodb;Database=AdventureWorks;Key=v2mtZ85W0AMCv1ZrY7jMUOWpfBTi1BrUz0Y3Rwmvj9SXS
SIKDU7EQVu5kdEMcwAQfvJBnmHSMxy50c3gD3v4g==',
                    SalesOrder)
    WITH
        ( SalesOrderId varchar(max) '$._id.string',
          details varchar(max) '$.details.array'
        ) As SalesOrders
CROSS APPLY OPENJSON(SalesOrders.details) AS SalesOrderArray
CROSS APPLY OPENJSON(SalesOrderArray.[value])
WITH
    (SKUCode varchar(max) '$.object.sku.string',
     SKUName varchar(max) '$.object.name.string',
     Price decimal(10,4) '$.object.price.float64' ,
     Quantity int '$.object.quantity.int32'
    ) As SalesOrderDetails

```

The first OPENJSON clause provides the SalesOrders.details value we extracted using the WITH clause of the OPENROWSET. When you call OPENJSON without a WITH clause and provide it with a JSON fragment that represents an array (as is the case in our example) the function returns a table with the following columns:

- Key- A value that contains the zero-based index of the element in the specified array.
- Value - A varchar(max) value that contains the value of the property itself. This will be the object value for each of the array elements in our example.
- Type - An int value that contains the type of the value, which we don't use in our example.

The second OPENJSON clause is provided with an input value of the value returned by the first OPENJSON clause, in our example the JSON fragment that represents the object of each element within the array. In this case, we use the WITH clause to further specify the column alias, and the column data type, and the element path we want to access (remembering to include the type suffix).

Lastly, we will project all of the needed columns including the key value returned by the first OPENJSON function (L), which will provide us with the SalesOrderLine, which by convention starts at 1 for each order at AdventureWorks, so needs to be adjusted from its zero-based value.

Let's see the result of all this transformation work that the view now contains.

Paste the following SQL into the query pane.

```
SELECT TOP(10) * FROM SalesOrderDetails
```

The screenshot shows the Azure Synapse Studio interface. At the top, the header bar displays 'Microsoft Azure', 'Synapse Analytics', and the workspace name 'synapselinkadventureworks'. Below the header, the 'Cosmos DB SQL Script' editor is active, showing the query: `1 SELECT TOP(10) * FROM SalesOrderDetails`. The 'Run' button is highlighted. Below the editor, the 'Results' tab is selected, showing a table view of the query results. The table has six columns: SalesOrderId, SalesOrderLine, SKUCode, SKUName, Price, and Quantity. The first 10 rows of data are displayed, including details for various mountain bike components and frames. A red box highlights the first 10 rows of the results table. The status bar at the bottom indicates '00:00:13 Query executed successfully.'

SalesOrderId	SalesOrderLine	SKUCode	SKUName	Price	Quantity
F88C4659-6AD8-41FE-8883-6B3...	1	RA-H123	Hitch Rack - 4-Bike	120.0000	1
E32722EC-38EF-4378-9291-ESAF...	1	HB-M763	ML Mountain Handlebars	37.1520	1
E32722EC-38EF-4378-9291-ESAF...	2	SE-M940	HL Mountain Seat/Saddle	31.5840	1
E32722EC-38EF-4378-9291-ESAF...	3	FR-M635-40	ML Mountain Frame-W - Silver, 40	218.4540	5
E32722EC-38EF-4378-9291-ESAF...	4	SH-W890-L	Women's Mountain Shorts, L	41.9940	8
E32722EC-38EF-4378-9291-ESAF...	5	BK-M685-38	Mountain-200 Silver, 38	1391.9940	1
E32722EC-38EF-4378-9291-ESAF...	6	FR-M215-52	LL Mountain Frame - Silver, 52	158.4300	1
E32722EC-38EF-4378-9291-ESAF...	7	CS-4759	LL Crankset	105.2940	4
E32722EC-38EF-4378-9291-ESAF...	8	CS-9183	HL Crankset	242.9940	1
E32722EC-38EF-4378-9291-ESAF...	9	FR-M218-42	LL Mountain Frame - Black, 42	149.8740	2

View results from a query in Azure Synapse Studio  
Click **run**.

As you can see, we have now extracted the order details information from within the sales order details array including its crucial revenue and unit sales data.

We are now able to create the statistical information needed to answer the questions we set to resolve.