

```
-- DAMG 7275 Lab 3

-- Question 1 (2 points)

/*
    Write a SQL query using "FOR JSON PATH" and AdventureWorks2017
    to get the total sales a salesperson has made. Use TotalDue in
    SalesOrderHeader to calculate the total sales. Return the data
    in the format described below.

    Use the provided format only for formatting purposes. Don't use
    the data contained in the format for validation purposes.

    Submit the code.
*/
SELECT
    SalesPersonID,
    CAST(SUM(TotalDue) AS decimal(10, 2)) AS TotalSales
FROM
    Sales.SalesOrderHeader
WHERE
    SalesPersonID IS NOT NULL
GROUP BY
    SalesPersonID
ORDER BY
    SalesPersonID
FOR JSON PATH

/*
[{"SalesPersonID":280,"TotalSales":3748246},
 {"SalesPersonID":281,"TotalSales":7259568},
 {"SalesPersonID":282,"TotalSales":6683537},
 {"SalesPersonID":283,"TotalSales":4207895},
 {"SalesPersonID":284,"TotalSales":2608116}
*/
```

```
/*
 Import the generated data into the Cosmos DB SQL API database.
 Submit a screenshot of importing results.
*/
```

The screenshot shows the Microsoft Azure portal with the URL [portal.azure.com/#@northeastern.onmicrosoft.com/resource/subscriptions/fd9b8c70-6014-461d-9938-fab03df3242a/resourceGroups/rc/providers/Microsoft...](https://portal.azure.com/#@northeastern.onmicrosoft.com/resource/subscriptions/fd9b8c70-6014-461d-9938-fab03df3242a/resourceGroups/rc/providers/Microsoft...). The user is signed in as [chotalia.r@northeastern.edu](#) from NORNEASTERN UNIVERSITY (N...). The page displays the Azure Cosmos DB Data Explorer for the 'ruchac' database account. The 'container - It...' tab is selected. The table 'ruchac' is shown with columns 'id' and '/id'. A specific row is selected, displaying its JSON representation:

```
1 {  
2   "SalesPersonID": 280,  
3   "TotalSales": 3748246.12,  
4   "id": "1d20b6f2-b698-486f-87f9-65a41cdad711",  
5   "_rid": "L0opAKlxJJQHAAAAAAA==",  
6   "_self": "dbs/L0opAKlxJJQ=/colls/L0opAKlxJJQ/_docs/L0opAKlxJJQHAAAAAAA==/",  
7   "_etag": "\"6a01b2d7-0000-0700-0000-63f2952b0000\"",  
8   "_attachments": "attachments/",  
9   "_ts": 1676842283  
10 }
```

```
/*
 Write a SQL query for the Cosmos DB SQL API to get
 the sum, average, maximum and minimum of all of the
 total sales made by all salespersons.
```

Submit the code and a screenshot of the executing results. \*/

```
SELECT  
SUM(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesSum',  
AVG(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesAvg',  
MAX(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesMax',  
MIN(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesMin'  
FROM  
r
```

The screenshot shows the Azure Cosmos DB Data Explorer interface. The left sidebar lists databases, containers, and items. The main area displays a query editor with the following SQL code:

```
1 SELECT
2 SUM(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesSum',
3 AVG(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesAvg',
4 MAX(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesMax',
5 MIN(IIF(IS_DEFINED(r.TotalSales), r.TotalSales, 0)) AS 'SalesMin'
6 FROM
7 r
```

The results pane shows the output of the query:

	1 - 1
	{"SalesSum": 90775447.02, "SalesAvg": 5339732.177647059, "SalesMax": 11695019.06, "SalesMin": 195528.78}

-- Question 2 (3 points)

```
/*
Write a SQL query using "FOR JSON PATH" and AdventureWorks2017
to get the products an order contains. Return the data in the
format described below. Return data only for orders in the
sales order id range between 43660 and 43680.
```

Use the provided format only for formatting purposes. Don't use
the data contained in the format for validation purposes.

Submit the code. \*/

```
SELECT
    rr.SalesOrderID,
    JSON_QUERY(
        CONCAT('[',
            STRING_AGG(
                JSON_QUERY(CONCAT('{',
                    '"ProductID":', CAST(d.ProductID AS VARCHAR), ',',
                    '"OrderQty":', CAST(d.OrderQty AS VARCHAR),
                    '}')), ,
                ',')),
            ']')
    ) AS Products
FROM
    Sales.SalesOrderHeader AS rr
    JOIN Sales.SalesOrderDetail AS d ON rr.SalesOrderID = d.SalesOrderID
WHERE
    rr.SalesOrderID BETWEEN 43660 AND 43680
GROUP BY
    rr.SalesOrderID
FOR JSON PATH

/*
[{"SalesOrderID":43660,
    "Products":[{"ProductID":762,"OrderQty":1},
                 {"ProductID":758,"OrderQty":1}],
    {"SalesOrderID":43661,
        "Products":[{"ProductID":745,"OrderQty":1},
                    {"ProductID":743,"OrderQty":1},
                    {"ProductID":747,"OrderQty":2},
                    {"ProductID":712,"OrderQty":4},
                    {"ProductID":715,"OrderQty":4},
                    {"ProductID":742,"OrderQty":2},
                    {"ProductID":775,"OrderQty":3},
                    {"ProductID":778,"OrderQty":2},
                    {"ProductID":711,"OrderQty":2},
                    {"ProductID":741,"OrderQty":2},
```

```
/*  
 {"ProductID":776,"OrderQty":4},  
 {"ProductID":773,"OrderQty":2},  
 {"ProductID":716,"OrderQty":2},  
 {"ProductID":777,"OrderQty":2},  
 {"ProductID":708,"OrderQty":5}}]  
*/
```

```
/*  
 Import the generated data into the Cosmos DB SQL API database.  
 */
```

The screenshot shows the Microsoft Azure Data Explorer interface for an Azure Cosmos DB account named 'ruchac'. The interface includes a top navigation bar with various links like Gmail, Outlook Mail, Student Hub, Internship Summer, ChatGPT, YouTube, Tutoring GSS, BPL - LIBBY, Tome, NUSource Commu..., CampusLabs-NEU, and a search bar. Below the navigation bar is a toolbar with icons for Create, Restore, and other database management functions. The main area is titled 'ruchac | Data Explorer' and shows a table with data from a container named 'Items'. The table has columns for 'Id' and '/Id'. A specific row is selected, displaying its detailed JSON structure. The JSON object contains fields like 'SalesOrderID', 'Products', and 'OrderQty'. The bottom of the interface shows a pagination control ('Page 1 of 1') and a toolbar with various application icons.

Id	/Id
0b883c...	0b883c...
93ee29...	93ee29...
7b7d85f...	7b7d85f...
1f046c7...	1f046c7...
3bd130...	3bd130...
0316f70...	0316f70...
9edefa7...	9edefa7...
939036...	939036...
ab2ade...	ab2ade...
c05185...	c05185...
7bcb3d...	7bcb3d...
3692ba...	3692ba...

```
1 {  
 2   "SalesOrderID": 43660,  
 3   "Products": [  
 4     {  
 5       "ProductID": 762,  
 6       "OrderQty": 1  
 7     },  
 8     {  
 9       "ProductID": 758,  
10       "OrderQty": 1  
11     }  
12   ],  
13   "_id": "0b883cbd-557e-430e-a73a-8d6c58682d8b",  
14   "_rid": "L0opANKr9YIBAAAAAAA==",  
15   "_self": " dbs/L0opAA==/colls/L0opANKr9YI=/docs/L0opANKr9YIBAAAAAAA==/",  
16   "_etag": "\\"6b015747-0000-0700-0000-63f29df90000\\\"",  
17   "_attachments": "attachments/",  
18   "_ts": 1676844537  
19 }
```

```
/*
 Write a SQL query for the Cosmos DB SQL API to get
 the number of unique products contained in each order.
 */
```

```
SELECT
    r.SalesOrderID,
    COUNT( pp.ProductID ) AS DistinctProducts
FROM r
    JOIN pp IN r.Products
GROUP BY
    r.SalesOrderID
```

The screenshot shows the Azure Cosmos DB Data Explorer interface. The left sidebar lists databases (ruchac), containers (lab3), and items. The right pane shows the NOSQL API interface with a query editor and results viewer.

**Query Editor:**

```
1 SELECT
2     r.SalesOrderID,
3     COUNT( pp.ProductID ) AS DistinctProducts
4 FROM
5     r
6     JOIN pp IN r.Products
7 GROUP BY
8     r.SalesOrderID
```

**Results View:**

SalesOrderID	DistinctProducts
43680	15
43679	15
43678	15
43677	15
43676	15
43675	15
43674	15
43673	15
43672	15
43671	15
43670	15
43669	15
43668	15
43667	15
43666	15
43665	15
43664	15
43663	15
43662	15
43661	15
43660	15
43659	15
43658	15
43657	15
43656	15
43655	15
43654	15
43653	15
43652	15
43651	15
43650	15
43649	15
43648	15
43647	15
43646	15
43645	15
43644	15
43643	15
43642	15
43641	15
43640	15
43639	15
43638	15
43637	15
43636	15
43635	15
43634	15
43633	15
43632	15
43631	15
43630	15
43629	15
43628	15
43627	15
43626	15
43625	15
43624	15
43623	15
43622	15
43621	15
43620	15
43619	15
43618	15
43617	15
43616	15
43615	15
43614	15
43613	15
43612	15
43611	15
43610	15
43609	15
43608	15
43607	15
43606	15
43605	15
43604	15
43603	15
43602	15
43601	15
43600	15
43599	15
43598	15
43597	15
43596	15
43595	15
43594	15
43593	15
43592	15
43591	15
43590	15
43589	15
43588	15
43587	15
43586	15
43585	15
43584	15
43583	15
43582	15
43581	15
43580	15
43579	15
43578	15
43577	15
43576	15
43575	15
43574	15
43573	15
43572	15
43571	15
43570	15
43569	15
43568	15
43567	15
43566	15
43565	15
43564	15
43563	15
43562	15
43561	15
43560	15
43559	15
43558	15
43557	15
43556	15
43555	15
43554	15
43553	15
43552	15
43551	15
43550	15
43549	15
43548	15
43547	15
43546	15
43545	15
43544	15
43543	15
43542	15
43541	15
43540	15
43539	15
43538	15
43537	15
43536	15
43535	15
43534	15
43533	15
43532	15
43531	15
43530	15
43529	15
43528	15
43527	15
43526	15
43525	15
43524	15
43523	15
43522	15
43521	15
43520	15
43519	15
43518	15
43517	15
43516	15
43515	15
43514	15
43513	15
43512	15
43511	15
43510	15
43509	15
43508	15
43507	15
43506	15
43505	15
43504	15
43503	15
43502	15
43501	15
43500	15
43499	15
43498	15
43497	15
43496	15
43495	15
43494	15
43493	15
43492	15
43491	15
43490	15
43489	15
43488	15
43487	15
43486	15
43485	15
43484	15
43483	15
43482	15
43481	15
43480	15
43479	15
43478	15
43477	15
43476	15
43475	15
43474	15
43473	15
43472	15
43471	15
43470	15
43469	15
43468	15
43467	15
43466	15
43465	15
43464	15
43463	15
43462	15
43461	15
43460	15
43459	15
43458	15
43457	15
43456	15
43455	15
43454	15
43453	15
43452	15
43451	15
43450	15
43449	15
43448	15
43447	15
43446	15
43445	15
43444	15
43443	15
43442	15
43441	15
43440	15
43439	15
43438	15
43437	15
43436	15
43435	15
43434	15
43433	15
43432	15
43431	15
43430	15
43429	15
43428	15
43427	15
43426	15
43425	15
43424	15
43423	15
43422	15
43421	15
43420	15
43419	15
43418	15
43417	15
43416	15
43415	15
43414	15
43413	15
43412	15
43411	15
43410	15
43409	15
43408	15
43407	15
43406	15
43405	15
43404	15
43403	15
43402	15
43401	15
43400	15
43399	15
43398	15
43397	15
43396	15
43395	15
43394	15
43393	15
43392	15
43391	15
43390	15
43389	15
43388	15
43387	15
43386	15
43385	15
43384	15
43383	15
43382	15
43381	15
43380	15
43379	15
43378	15
43377	15
43376	15
43375	15
43374	15
43373	15
43372	15
43371	15
43370	15
43369	15
43368	15
43367	15
43366	15
43365	15
43364	15
43363	15
43362	15
43361	15
43360	15
43359	15
43358	15
43357	15
43356	15
43355	15
43354	15
43353	15
43352	15
43351	15
43350	15
43349	15
43348	15
43347	15
43346	15
43345	15
43344	15
43343	15
43342	15
43341	15
43340	15
43339	15
43338	15
43337	15
43336	15
43335	15
43334	15
43333	15
43332	15
43331	15
43330	15
43329	15
43328	15
43327	15
43326	15
43325	15
43324	15
43323	15
43322	15
43321	15
43320	15
43319	15
43318	15
43317	15
43316	15
43315	15
43314	15
43313	15
43312	15
43311	15
43310	15
43309	15
43308	15
43307	15
43306	15
43305	15
43304	15
43303	15
43302	15
43301	15
43300	15
43299	15
43298	15
43297	15
43296	15
43295	15
43294	15
43293	15
43292	15
43291	15
43290	15
43289	15
43288	15
43287	15
43286	15
43285	15
43284	15
43283	15
43282	15
43281	15
43280	15
43279	15
43278	15
43277	15
43276	15
43275	15
43274	15
43273	15
43272	15
43271	15
43270	15
43269	15
43268	15
43267	15
43266	15
43265	15
43264	15
43263	15
43262	15
43261	15
43260	15
43259	15
43258	15
43257	15
43256	15
43255	15
43254	15
43253	15
43252	15
43251	15
43250	15
43249	15
43248	15
43247	15
43246	15
43245	15
43244	15
43243	15
43242	15
43241	15
43240	15
43239	15
43238	15
43237	15
43236	15
43235	15
43234	15
43233	15
43232	15
43231	15
43230	15
43229	15
43228	15
43227	15
43226	15
43225	15
43224	15
43223	15
43222	15
43221	15
43220	15
43219	15
43218	15
43217	15
43216	15
43215	15
43214	15
43213	15
43212	15
43211	15
43210	15
43209	15
43208	15
43207	15
43206	15
43205	15
43204	15
43203	15
43202	15
43201	15
43200	15
43199	15
43198	15
43197	15
43196	15
43195	15
43194	15
43193	15
43192	15
43191	15
43190	15
43189	15
43188	15
43187	15
43186	15
43185	15
43184	15
43183	15
43182	15
43181	15
43180	15
43179	15
43178	15
43177	15
43176	15
43175	15
43174	15
43173	15
43172	15
43171	15
43170	15
43169	15
43168	15
43167	15
43166	15
43165	15
43164	15
43163	15
43162	15
43161	15
43160	15
43159	15
43158	15
43157	15

-- Question 3 (3 points)

/\*

Write a SQL query using "FOR JSON PATH" and AdventureWorks2017 to get the top 3 products in a sales territory and a product's total sold quantity. Return the data in the format described below. TotalProductQuantity is the total sold quantity of a product.

The top 3 products have the 3 highest total sold quantities in a territory. If there is a tie, the tie needs to be retrieved.

The product quantity of an order is stored in SalesOrderDetail.

Use the provided format only for formatting purposes. Don't use the data contained in the format for validation purposes.

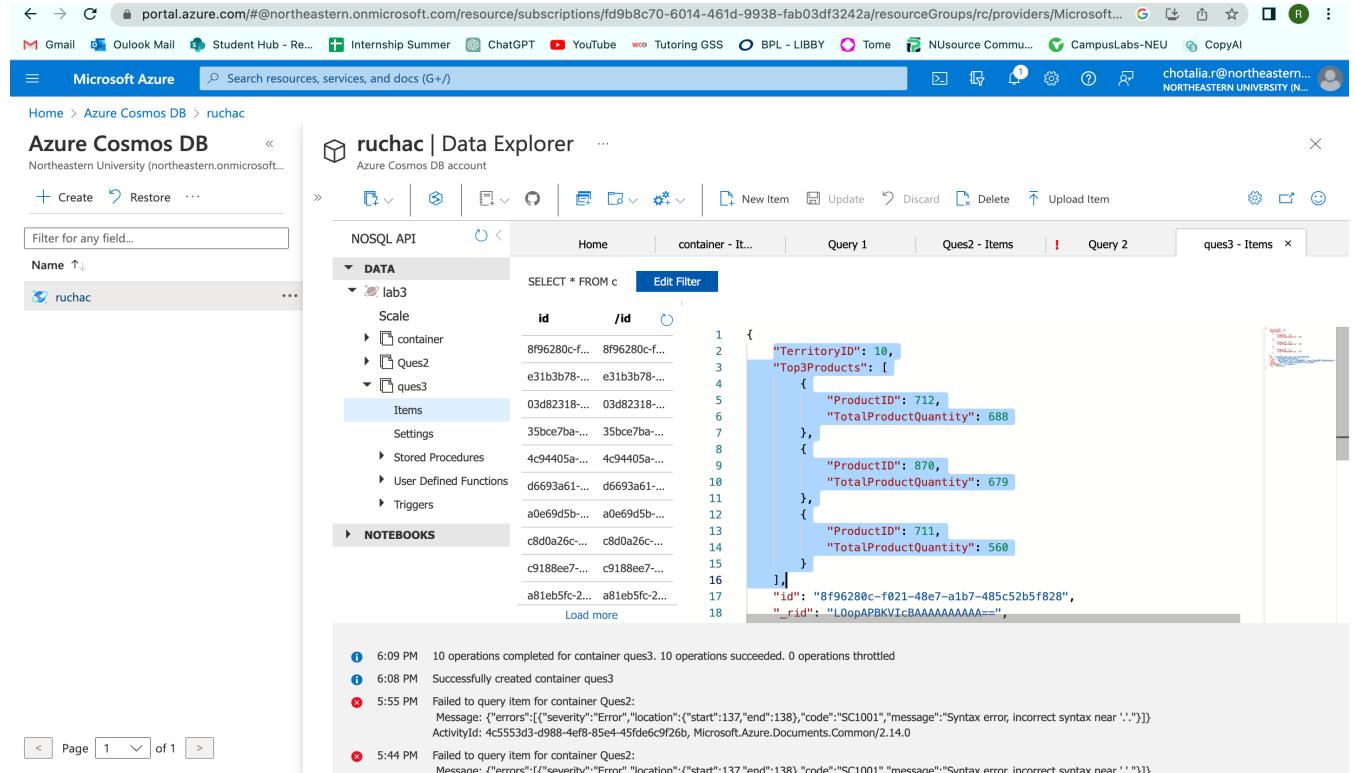
Submit the code.

\*/

```
SELECT
    rr.TerritoryID,
    (
        SELECT TOP 3
            p.ProductID,
            SUM(salesdt.OrderQty) AS TotalProductQuantity
        FROM
            Sales.SalesOrderDetail AS salesdt
        JOIN Sales.SalesOrderHeader AS saleshead ON salesdt.SalesOrderID =
saleshead.SalesOrderID
        JOIN Production.Product AS p ON salesdt.ProductID = p.ProductID
        WHERE
            saleshead.TerritoryID = rr.TerritoryID
        GROUP BY
            p.ProductID
        ORDER BY
            TotalProductQuantity DESC,
            p.ProductID ASC
        FOR JSON PATH
    ) AS Top3Products
FROM
    Sales.SalesTerritory AS rr
FOR JSON PATH;

/*
[{"TerritoryID":1,
    "Top3Products": [{"ProductID":870,"TotalProductQuantity":947},
                    {"ProductID":712,"TotalProductQuantity":887},
                    {"ProductID":711,"TotalProductQuantity":794}]},
 {"TerritoryID":2,
    "Top3Products": [{"ProductID":715,"TotalProductQuantity":549},
                    {"ProductID":712,"TotalProductQuantity":514},
                    {"ProductID":711,"TotalProductQuantity":423}]}
*/
```

```
*/  
  
/*  
 Import the generated data into a Cosmos DB SQL API database.  
 Submit a screenshot of importing results.  
*/
```



The screenshot shows the Microsoft Azure Data Explorer interface for the 'ruchac' database. The left sidebar lists containers: 'lab3' (Scale, Items) and 'ques3' (Settings, Stored Procedures, User Defined Functions, Triggers). The main pane displays a query result for container 'ques3'. The query is:

```
SELECT * FROM c
```

The result shows a JSON array of documents. One document is highlighted, showing details for a product in territory 10:

```
1 {  
2   "TerritoryID": 10,  
3   "Top3Products": [  
4     {  
5       "ProductID": 712,  
6       "TotalProductQuantity": 688  
7     },  
8     {  
9       "ProductID": 870,  
10      "TotalProductQuantity": 679  
11     },  
12     {  
13       "ProductID": 711,  
14       "TotalProductQuantity": 560  
15     }  
16   ],  
17   "id": "8f96280c-f021-48e7-a1b7-485c52b5f828",  
18   "_rid": "L0opAPBKViCBAAAAAAA=="  
]
```

Below the result, a log shows activity history:

- 6:09 PM 10 operations completed for container ques3. 10 operations succeeded. 0 operations throttled
- 6:08 PM Successfully created container ques3
- 5:55 PM Failed to query item for container Ques2:  
Message: {"errors": [{"severity": "Error", "location": {"start": 137, "end": 138}, "code": "SC1001", "message": "Syntax error, incorrect syntax near '.'"}]}  
ActivityId: 4c5553d3-d988-4ef8-85e4-45fde6c9f26b, Microsoft.Azure.Documents.Common/2.14.0
- 5:44 PM Failed to query item for container Ques2:  
Message: {"errors": [{"severity": "Error", "location": {"start": 137, "end": 138}, "code": "SC1001", "message": "Syntax error, incorrect syntax near '.'"}]}

```
/*  
 Write a SQL query for the Cosmos DB SQL API to get  
 the totals of TotalProductQuantity regardless of the  
 sales territory for each product.  
 Submit the code and a screenshot of the executing results.  
*/
```

```
SELECT  
    pp.ProductID,  
    SUM(pp.TotalProductQuantity) AS TotalProdQty  
FROM rr  
    JOIN pp IN rr.Top3Products  
GROUP BY  
    pp.ProductID
```

The screenshot shows the Azure Cosmos DB Data Explorer interface. The left sidebar lists databases: 'ruchac' (selected), 'lab3', 'Ques2', and 'ques3'. The 'ques3' database is currently active. The main area displays a query editor with the following T-SQL code:

```
1 SELECT
2     pp.ProductID,
3     SUM(pp.TotalProductQuantity) AS TotalProdQty
4 FROM rr
5 JOIN pp IN rr.Top3Products
6 GROUP BY
7     pp.ProductID
```

The results pane shows two rows of data:

ProductID	TotalProdQty
873	674
708	1700