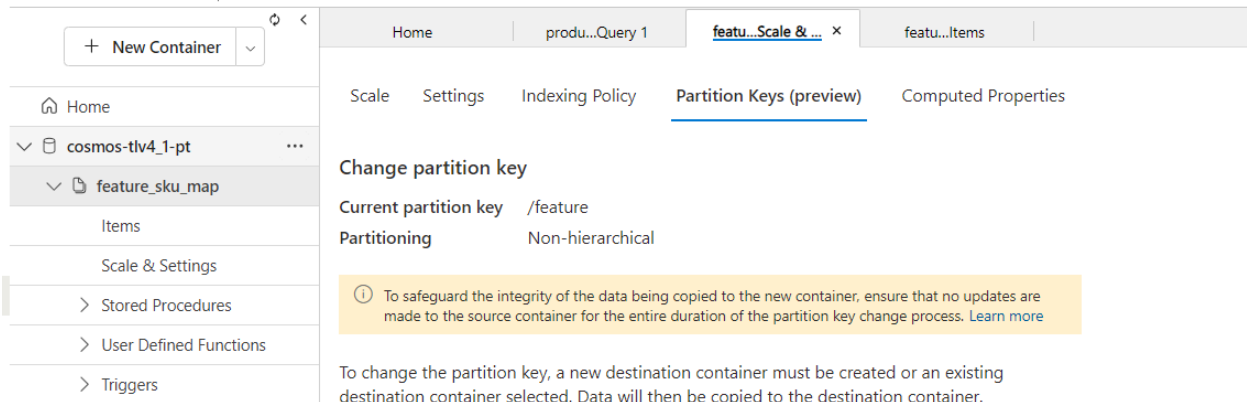
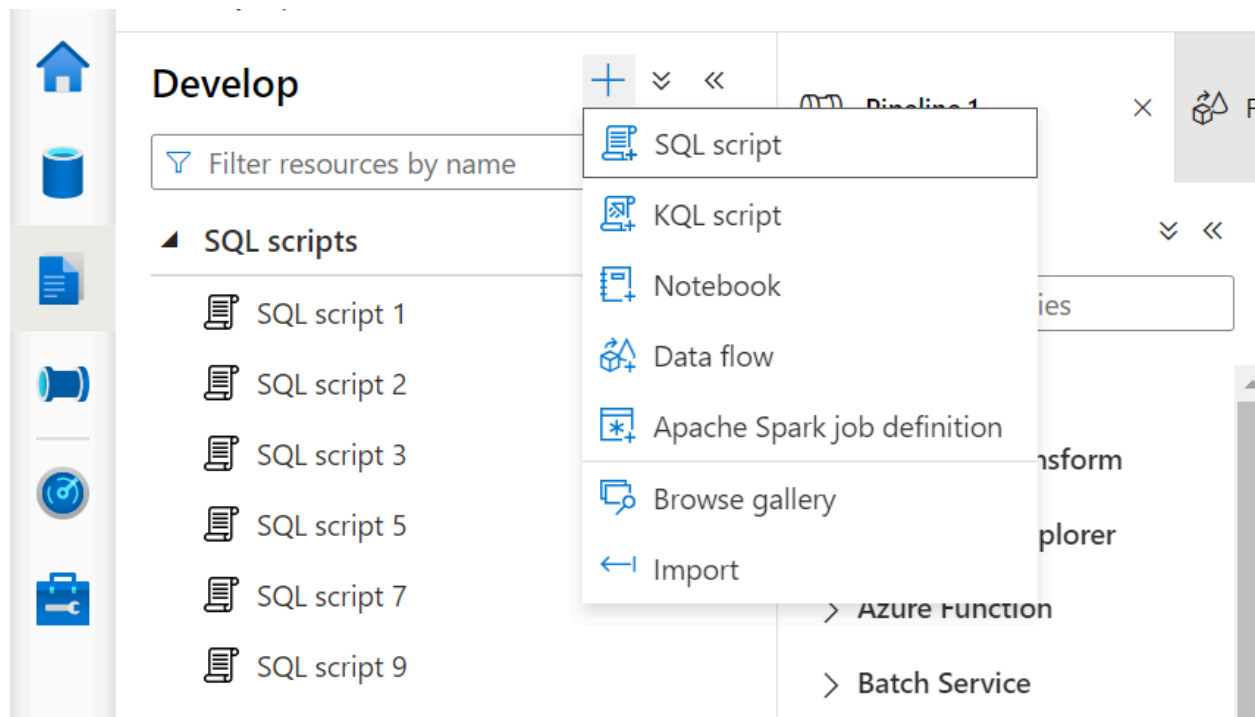


Data Flow pipeline for creating feature sku map container:

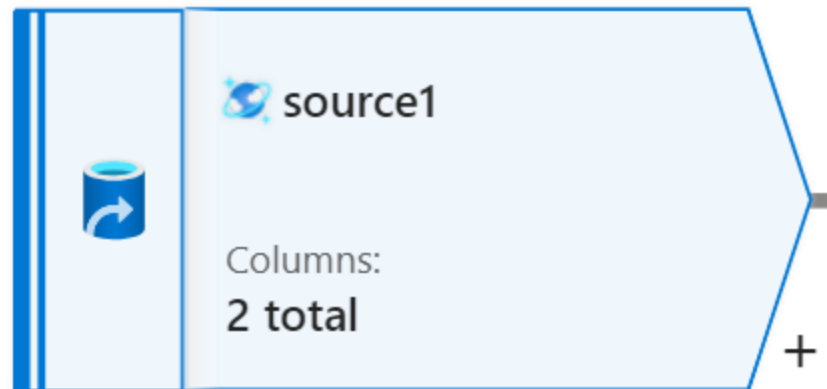
1. Create new container for feature_sku_map with feature as partition key.



2. Open synapse studio and go to integrates tab.
3. Create new data flow.



4. Add one source to read details from product_reader container.



5. Use query option to took only feature and sku fields.

Source settings **Source options** Projection Optimize Inspect Data preview ●

Input ☐ Container ☒ Query

Query * ⓘ



```
SELECT
  f AS feature,
  c.sku
FROM
  c
JOIN
  f IN c.features
```

Change feed ⓘ ☐

Include system columns ☐

Page size ⓘ

Throughput ⓘ

  Import p

Source settings	Source options	Projection	Optimize	Inspect	Data preview ●
-----------------	----------------	------------	----------	---------	----------------

Number of rows	+ INSERT 100	* UPDATE 0	× DELETE 0	+ UPSERT 0	LOOK
----------------	--------------	------------	------------	------------	------

Refresh	Typecast	Modify	Map drifted	Statistics	Remove	Export to CSV
---------	----------	--------	-------------	------------	--------	---------------

↑↓	feature	abc	↑↓	sku	abc	↑↓
+	FEA-SKP-SEF-GOLD			BS-SB-SKP-SEFAIRA		
+	FEA-CONN-MAX-C...			OR-LT		
+	FEA-CONN-MAX-FO...			OR-LT		
+	FEA-CONN-MAX-IN...			OR-LT		
+	FEA-CONN-MAX-PRJ			OR-LT		
+	FEA-CONN-MAX-ST...			OR-LT		
+	FEA-OR-LT			OR-LT		

6. It is important to import projection to get columns to pass to next flow.

Source settings	Source options	Projection	Optimize	Inspect	Data preview ●
-----------------	----------------	------------	----------	---------	----------------

Import projection	Reset schema	Overwrite schema
-------------------	--------------	------------------

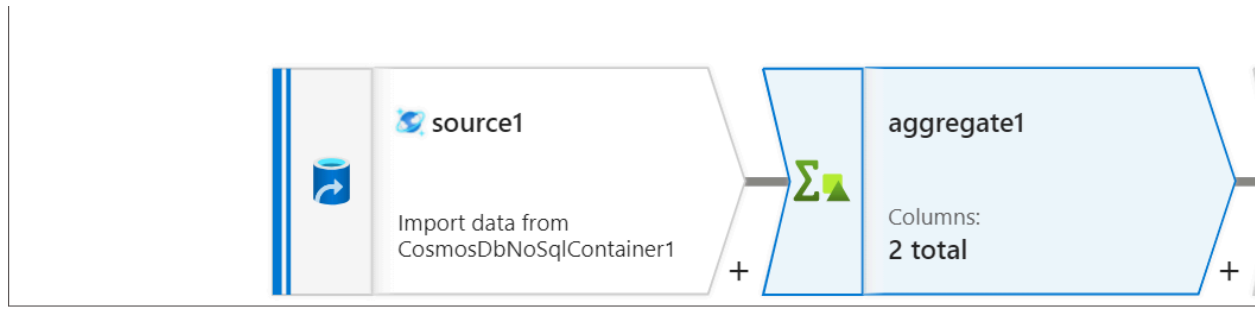
Column name	↑↓	Type
feature	abc	string
sku	abc	string

Source settings	Source options	Projection	Optimize	Inspect	Data preview ●
-----------------	----------------	------------	----------	---------	----------------

Number of columns	Total 2
-------------------	---------

Order	↑↓	Column	↑↓	Type	↑↓
1		feature		abc	string
2		sku		abc	string

7. Once the data has been ready, next do the transformation. Here, we have to do aggregation.



Aggregate settings Optimize Inspect Data preview ●

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

8. Now there is two fields to provide input: Group by and Arregates. For group by, provide “feature”.

Group by Aggregates

Columns	Name as
<input type="text" value="abc feature"/>	<input type="text" value="feature"/>

9. For aggregates, give sku column and expression as collectUnique.

Aggregate settings Optimize Inspect Data preview ●

Group by **Aggregates**

Grouped by: feature

Add Clone Delete Open expression builder

<input type="checkbox"/> Column	Expression
<input type="checkbox"/> <input type="text" value="sku"/>	<input type="text" value="collectUnique(sku)"/>

Aggregate settings

Optimize

Inspect

Data preview

Number of rows

+

INSERT

100

*

UPDATE

0

✕

DELETE

0

+

UPSERT

↺

Refresh

|

▼

Typecast

▼

🔗

Modify

▼

📄

Map drifted

📄

Statistics

✕

Remove

↓

Export

↕

feature

abc

↕

sku

[]

↕

+

FEA-3DW-BPM-GOLD

[...]

+

FEA-CANVAS-BASE

[...]

+

FEA-CANVAS-BASE-P

[...]

+

FEA-CANVAS-PRO

[...]

+

FEA-CANVAS-PRO-P

[...]

+

FEA-CEC-SITEWORKS

[...]

+

FEA-CEC-WORKSOS...

[...]

+

FEA-CEC-WORKSOS...

[...]

10. Now, the final step is to sink the data into result container.

source1

Import data from CosmosDbNoSqlContainer1

+

aggregate1

Aggregating data by 'feature' producing columns 'sku'

+

sink1

Columns: 3 total

Sink

Settings

Errors

Mapping

Optimize

Inspect

Data preview ●

Output stream name *

sink1

Learn more

Description

Export data to CosmosDbNoSqlContainer2


Reset

Incoming stream *

aggregate1

11. For dataset, click new and select Azure Cosmos DB for NoSQL.

New integration dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. [Learn more](#) 

Select a data store

< All **Azure** Database File Generic protocol >



Azure Blob Storage



Azure Cosmos DB for
NoSQL



Azure Data Explorer
(Kusto)



Continue

Cancel



Set properties

Name

CosmosDbNoSqlContainer3

Linked service *

CosmosDbNoSql2

Connect via integration runtime * ⓘ

✓ AutoResolveIntegrationRuntime

Container

feature_sku_map

☐

Enter manually

Import schema



From connection/store



None

> Advanced

OK

Back

Cancel



12. Fill the remaining settings and other tabs:

Update method ⓘ

- ☒ Allow insert
- ☐ Allow delete
- ☐ Allow upsert
- ☐ Allow update

Container action ⓘ

- ☒ None
- ☐ Recreate container ⓘ

Batch size ⓘ

Partition key ⓘ

Throughput ⓘ

Write throughput budget ⓘ

Options

- ☐ Skip duplicate input columns ⓘ
- ☐ Skip duplicate output columns ⓘ
- ☐ Auto mapping ⓘ
- Reset
- Add mapping
- Delete

<input type="checkbox"/>	Input columns		Output columns		
<input type="checkbox"/>	abc feature	→	feature	+	
<input type="checkbox"/>	sku	→	sku	+	
<input type="checkbox"/>	abc feature	→	id	+	

Schema

InputOutput

Number of columns		Updated* 1		Dropped 0		Unchanged 2		Total 3	
Order	↑↓	Column	↑↓	Type	↑↓	Updated	↑↓	Input column	↑↓
1		feature		abc	string			feature	
2		sku		[]	string			sku	
3		id		abc	string	*		feature	

Pipeline runs

Triggered Debug Rerun Cancel options Refresh Edit columns List Gantt

Filter by run ID or name Chennai, Kolkata, Mu... : Last 24 hours Pipeline name : Pipeline 1

Copy filters Export to CSV

Status : All Runs : Latest runs Triggered by : All Add filter

Showing 1 - 1 items Last refreshed 0 minutes ago

<input type="checkbox"/> Pipeline name ↑↓	Run start ↑↓	Run end ↑↓	Duration	Triggered by	Status ↑↓
<input type="checkbox"/> Pipeline 1	10/10/2024, 10:40:22 PM	10/10/2024, 10:44:28 PM	4m 6s	Manual trigger	✓ Succeed

New Container

Home

cosmos-trlv4.1-pt

feature_sku_map

Items

Scale & Settings

Stored Procedures

User Defined Functions

Triggers

license_reader_v2

product_reader

role_table

usage_activity_reader

SELECT * FROM c

id

FEA-SC-CH_CALCULATION

APPL_PP_Data_APX_Airborne...

APPL_PP_Data_APX_Airborne...

APPL_PP_Data_APX_Automotive

APPL_PP_Data_APX_HumanPo...

APPL_PP_Data_APX_Mapping...

APPL_PP_Data_APX_OffroadVe...

APPL_PP_Data_AV

APPL_PP_Data_Generic_GNSS...

APPL_PP_Data_LV

APPL_PP_Data_MV

APPL_PP_Data_RINEX

APPL_PP_Data_TG

APPL_PP_Max_Num_Projects

/feature

FEA-SC-CH_CALCUL...

APPL_PP_Data_APX...

APPL_PP_Data_APX...

APPL_PP_Data_APX...

APPL_PP_Data_APX...

APPL_PP_Data_APX...

APPL_PP_Data_APX...

APPL_PP_Data_AV

APPL_PP_Data_Gene...

APPL_PP_Data_LV

APPL_PP_Data_MV

APPL_PP_Data_RINEX

APPL_PP_Data_TG

APPL_PP_Max_Num ...

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

{

"feature": "FEA-SC-CH_CALCULATION",

"sku": [

"SUB-SCAD-MLTP-BNL",

"SUB-SCAD-MAP-BNL",

"SUB-SCAD-MFP-BNL",

"SUB-DEV-SCAD-TFR"

],

"id": "FEA-SC-CH_CALCULATION",

"_rid": "wsISAKFSERoBAAAAAAAAA==",

"_self": "dbs/wsISAA==/colls/wsISAKFSERo/docs/wsISAKFSERoBAAAAAAAAA==/",

"etag": "\000069e4-0000-0000-0000-67000b500000\"",

"attachments": "attachments/",

"ts": 1728580432

}