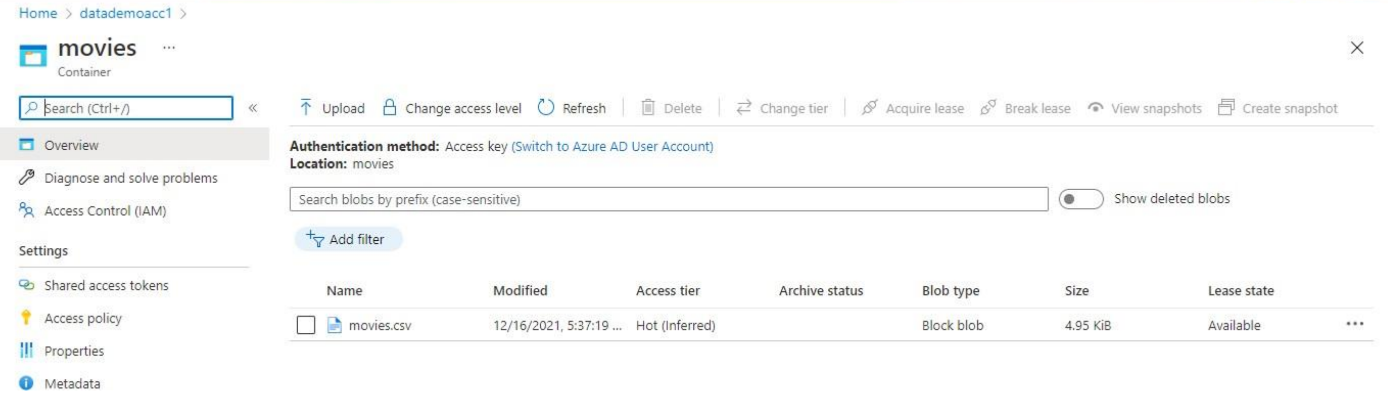
Building an ETL Pipeline using Azure Data Services

Prerequisites:

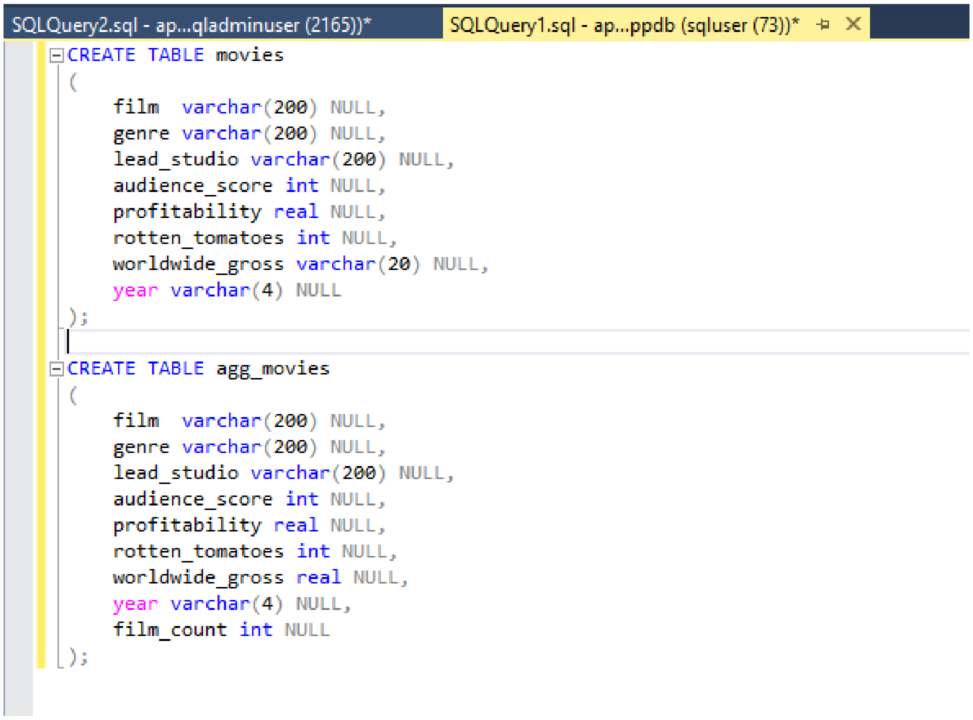
Step 1: Create a blob storage, then create a container named “movies” and put the dataset (movies.csv) into the container as below:



Authentication method: Access key (Switch to Azure AD User Account)
Location: movies

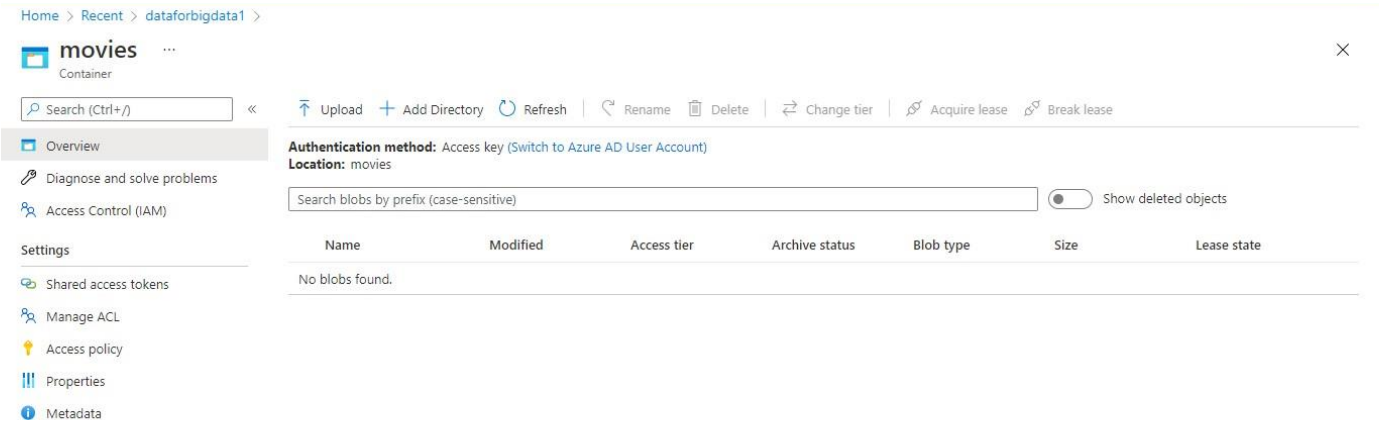
Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
movies.csv	12/16/2021, 5:37:19 ...	Hot (Inferred)		Block blob	4.95 KiB	Available

Step 2: Create the SQL database “appdb” and create a table named “movies” and “agg_movies” :


SQLQuery2.sql - ap...qladminuser (2165))*
SQLQuery1.sql - ap...ppdb (sqluser (73))*

```
CREATE TABLE movies
(
    film  varchar(200) NULL,
    genre varchar(200) NULL,
    lead_studio varchar(200) NULL,
    audience_score int NULL,
    profitability real NULL,
    rotten_tomatoes int NULL,
    worldwide_gross varchar(20) NULL,
    year varchar(4) NULL
);

CREATE TABLE agg_movies
(
    film  varchar(200) NULL,
    genre varchar(200) NULL,
    lead_studio varchar(200) NULL,
    audience_score int NULL,
    profitability real NULL,
    rotten_tomatoes int NULL,
    worldwide_gross real NULL,
    year varchar(4) NULL,
    film_count int NULL
);
```

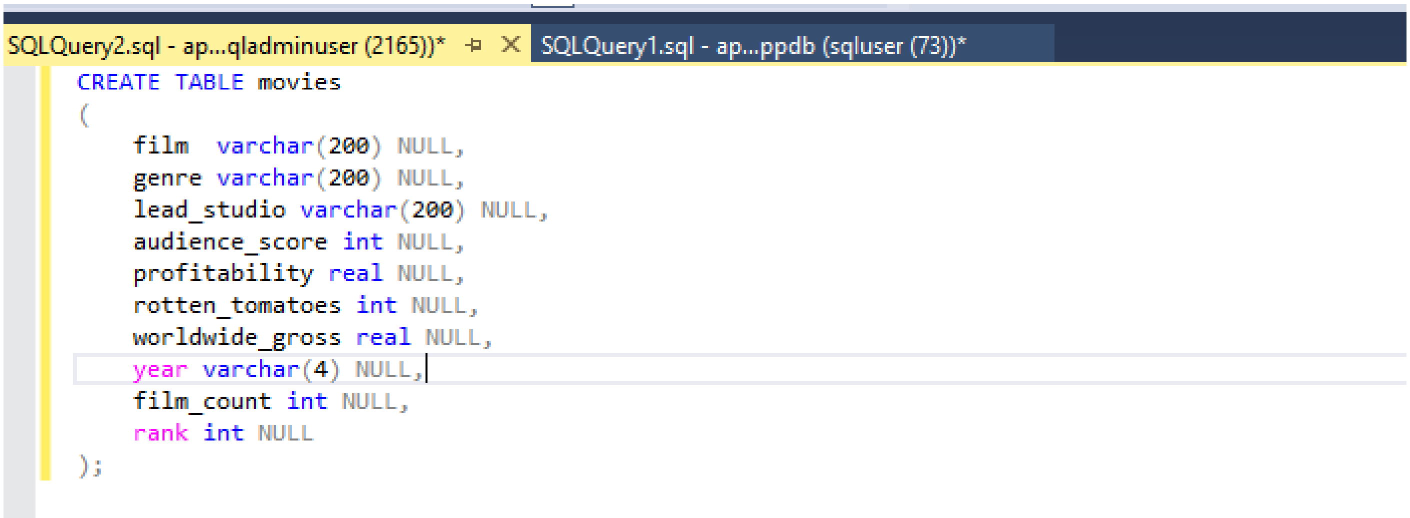
Step 3: Create a Data Lake Gen 2 Storage, then create a container named “movies” to store the processed data after cleaning and aggregating as shown below:



Authentication method: Access key (Switch to Azure AD User Account)
Location: movies

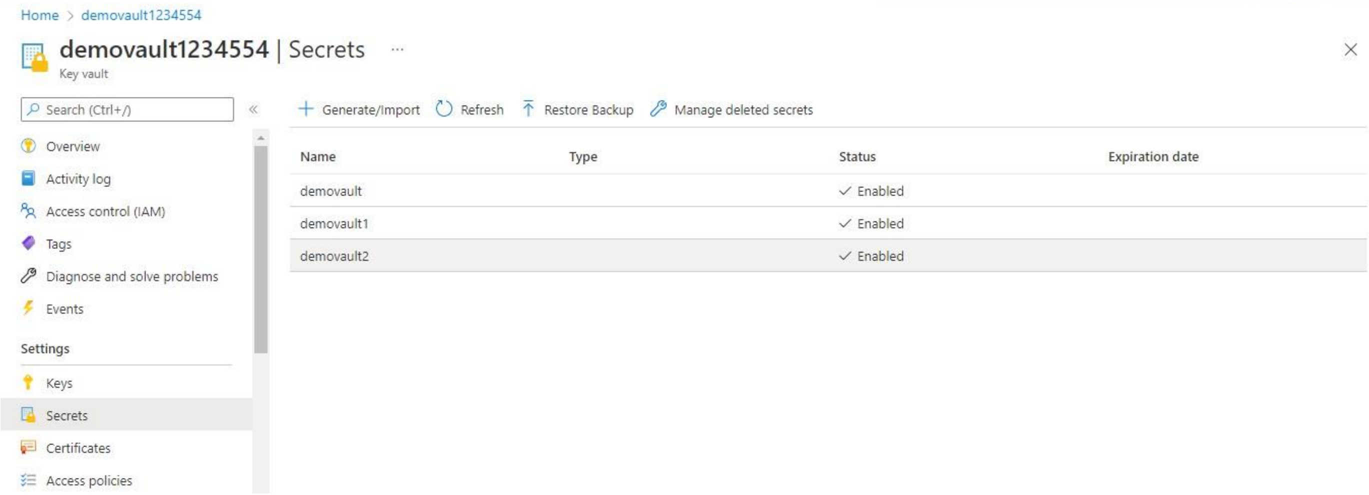
Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
No blobs found.						

Step 4: Create a Synapse Analytics, the create a SQL pool named “appexample1234” and create a table named “movies” using the below query:


SQLQuery2.sql - ap...qladminuser (2165))*
SQLQuery1.sql - ap...ppdb (sqluser (73))*

```
CREATE TABLE movies
(
    film  varchar(200) NULL,
    genre varchar(200) NULL,
    lead_studio varchar(200) NULL,
    audience_score int NULL,
    profitability real NULL,
    rotten_tomatoes int NULL,
    worldwide_gross real NULL,
    year varchar(4) NULL,
    film_count int NULL,
    rank int NULL
);
```

Step 5: Create a key vault, then navigate to secrets and create one secret to use in Azure Databricks



Home > demovault1234554

demovault1234554 | Secrets ...

Key vault

Search (Ctrl+/)

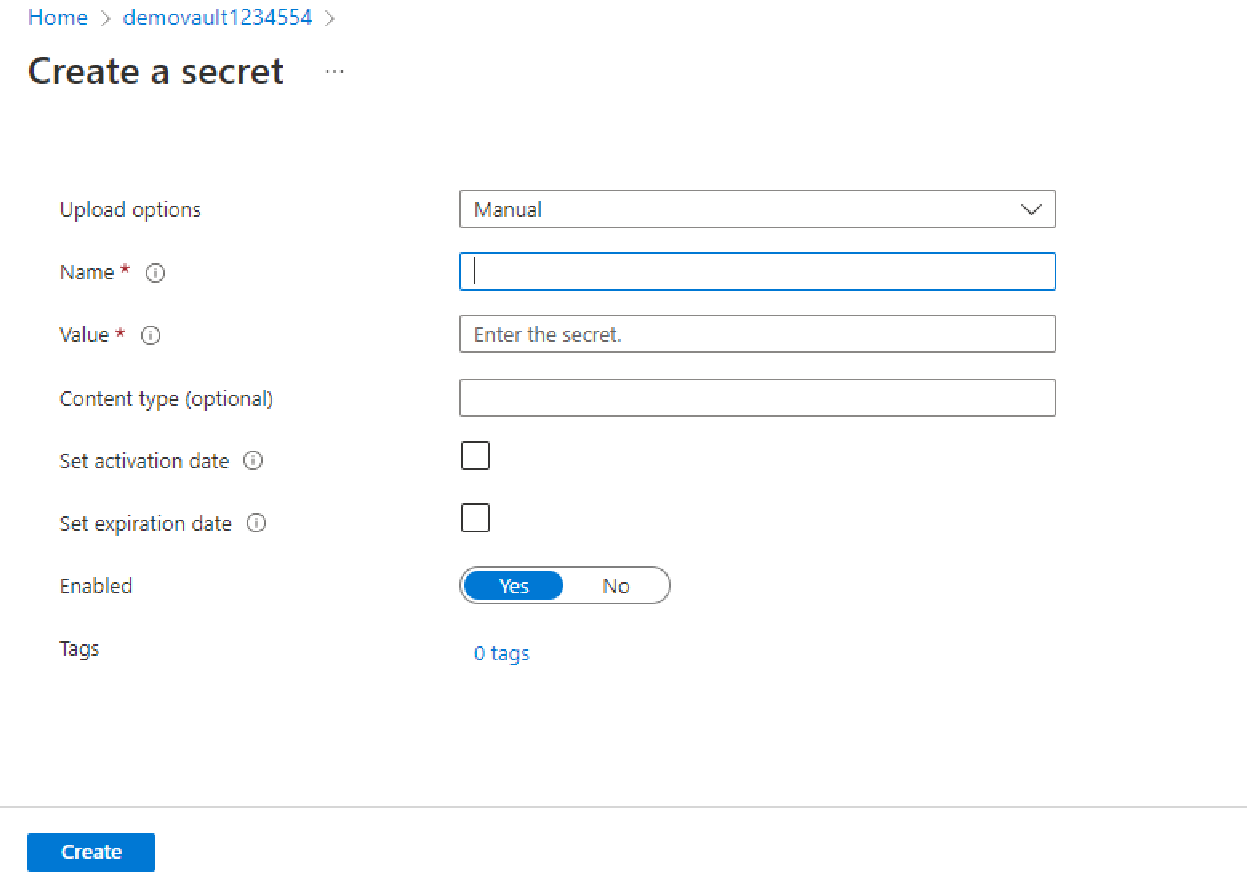
- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events

Settings

- Keys
- Secrets
- Certificates
- Access policies

+ Generate/Import   Refresh   Restore Backup   Manage deleted secrets

Name	Type	Status	Expiration date
demovault		✓ Enabled	
demovault1		✓ Enabled	
demovault2		✓ Enabled	



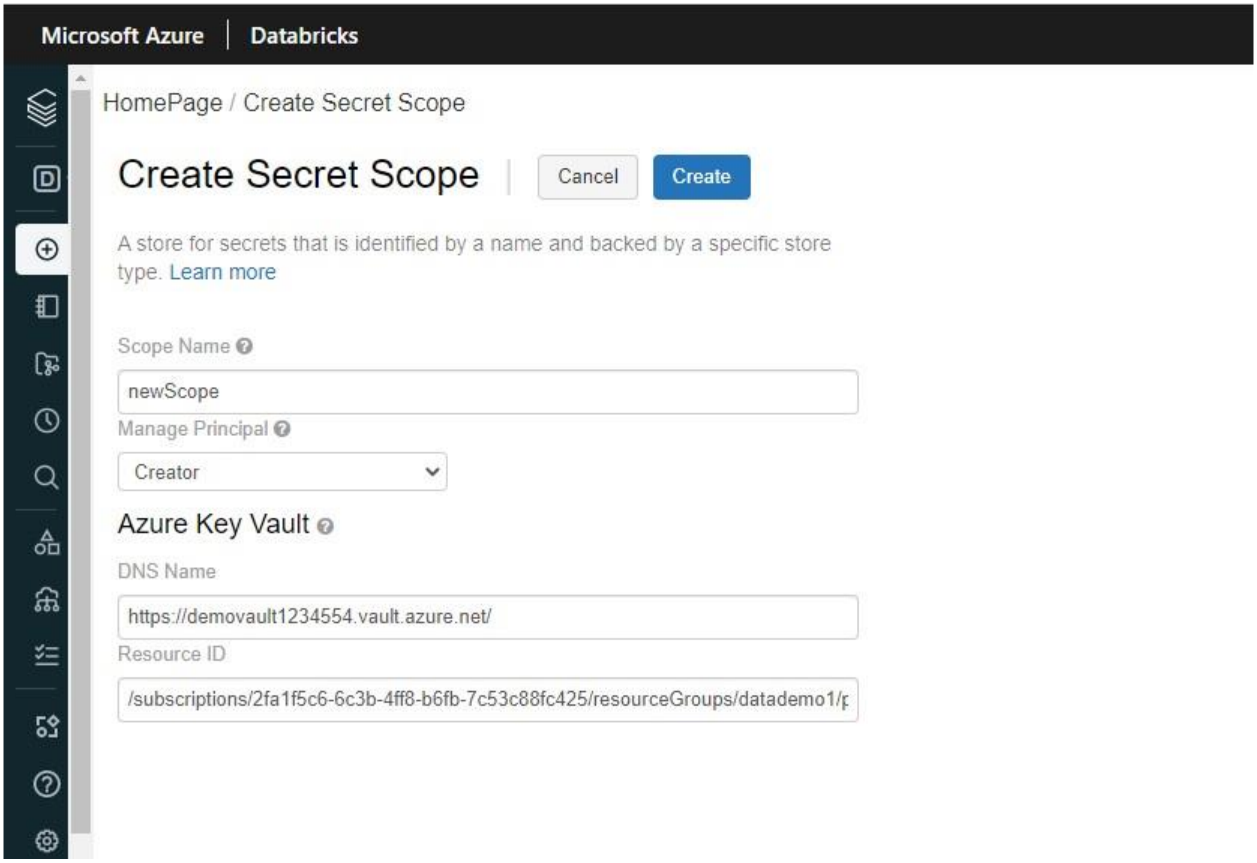
Home > demovault1234554 >

Create a secret ...

Upload options	Manual
Name * ⓘ	
Value * ⓘ	Enter the secret.
Content type (optional)	
Set activation date ⓘ	□
Set expiration date ⓘ	□
Enabled	[✓ Yes] No
Tags	0 tags

Create

- Once created, Now launch your Azure databricks and append this to the URI "/#secrets/createScope ", It will open up a window like this



Create Secret Scope

A store for secrets that is identified by a name and backed by a specific store type. Learn more

Scope Name: newScope

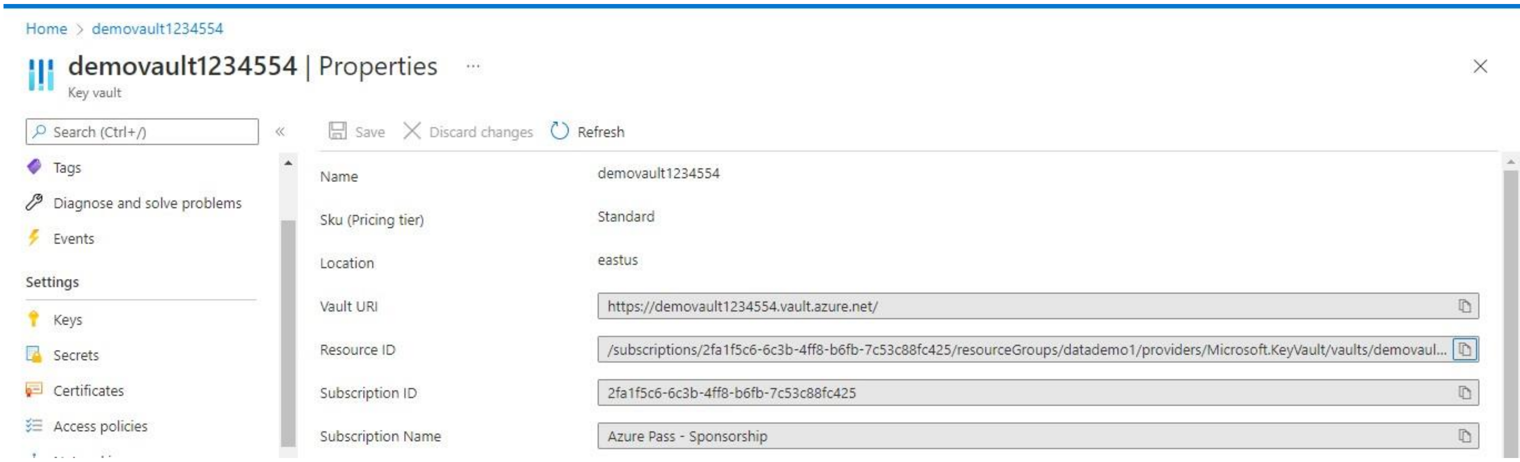
Manage Principal: Creator

Azure Key Vault

DNS Name: https://demovault1234554.vault.azure.net/

Resource ID: /subscriptions/2fa1f5c6-6c3b-4ff8-b6fb-7c53c88fc425/resourceGroups/datademo1/p

- DNS Name and Resource ID can be copied from key vault properties "Vault URI" and "Resource ID"



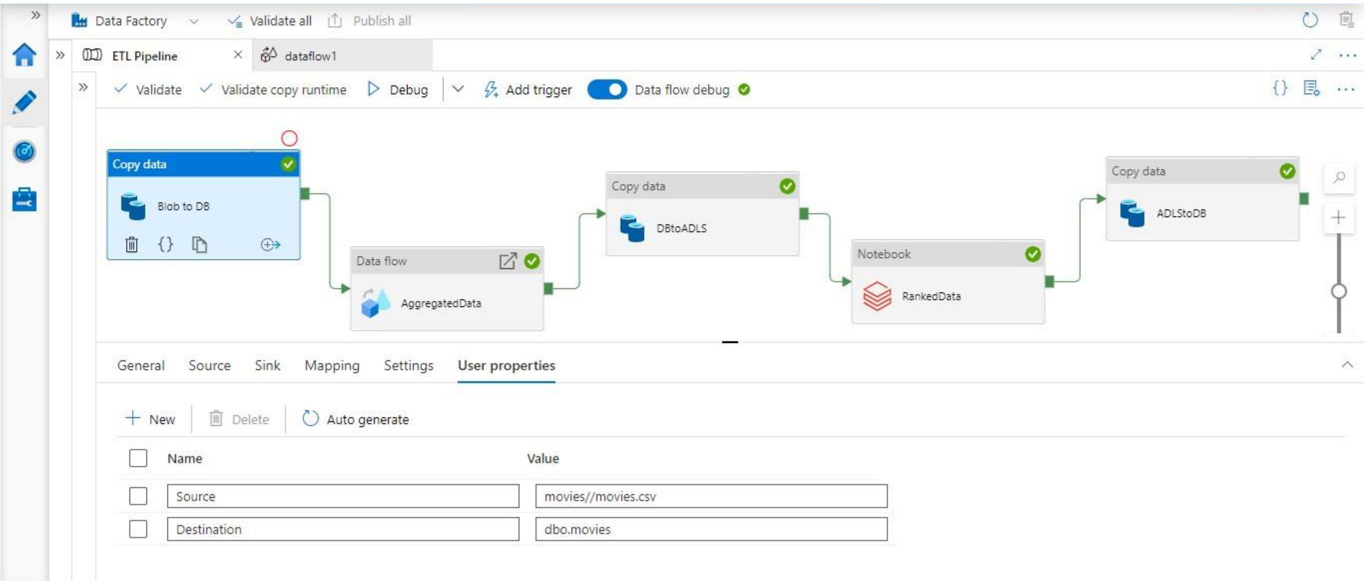
demovault1234554 | Properties

Name	demovault1234554
Sku (Pricing tier)	Standard
Location	eastus
Vault URI	https://demovault1234554.vault.azure.net/
Resource ID	/subscriptions/2fa1f5c6-6c3b-4ff8-b6fb-7c53c88fc425/resourceGroups/datademo1/providers/Microsoft.KeyVault/vaults/demovaul...
Subscription ID	2fa1f5c6-6c3b-4ff8-b6fb-7c53c88fc425
Subscription Name	Azure Pass - Sponsorship

- Make sure you remember the scope name created in databricks since it will be used to connect with ADLS in the later part.

Proceedings:

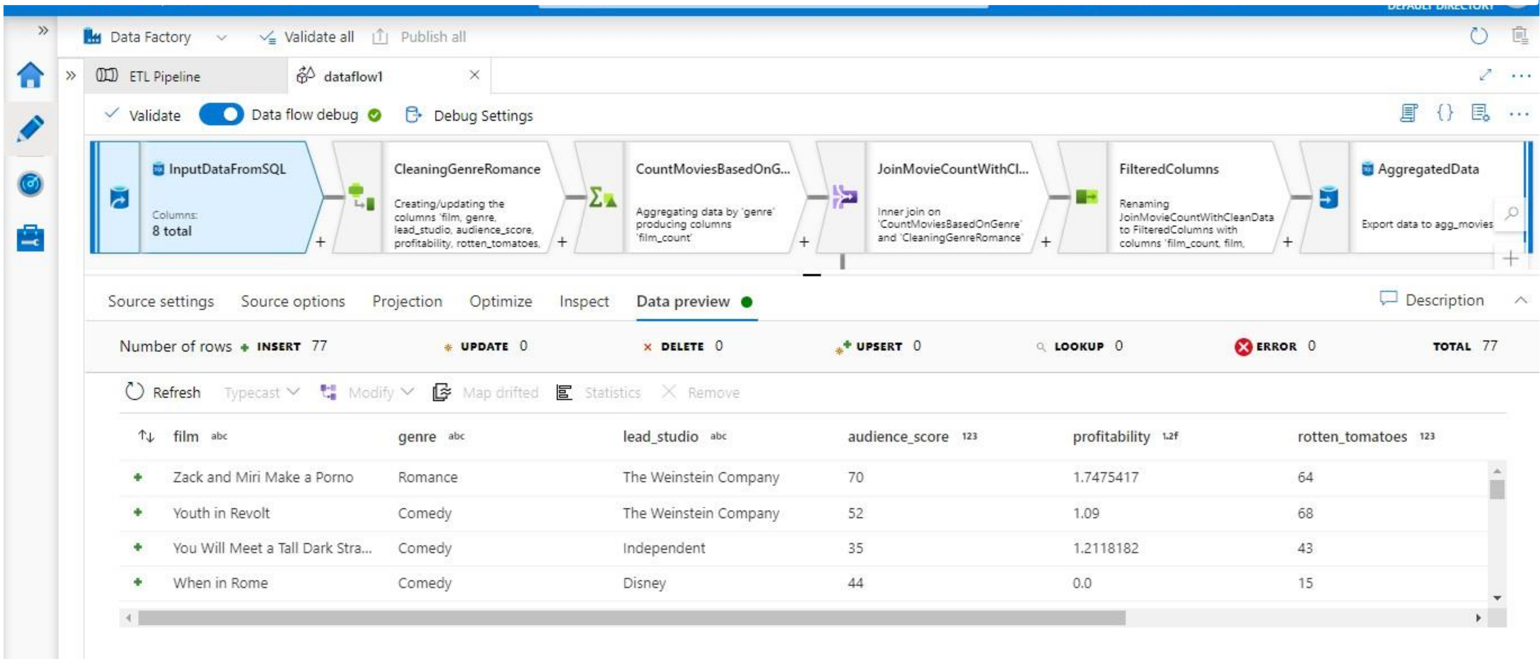
Step 1: Importing dataset from blob to DB



Name	Value
Source	movies//movies.csv
Destination	dbo.movies

Step 2: Create a dataflow to clean and join the table as required

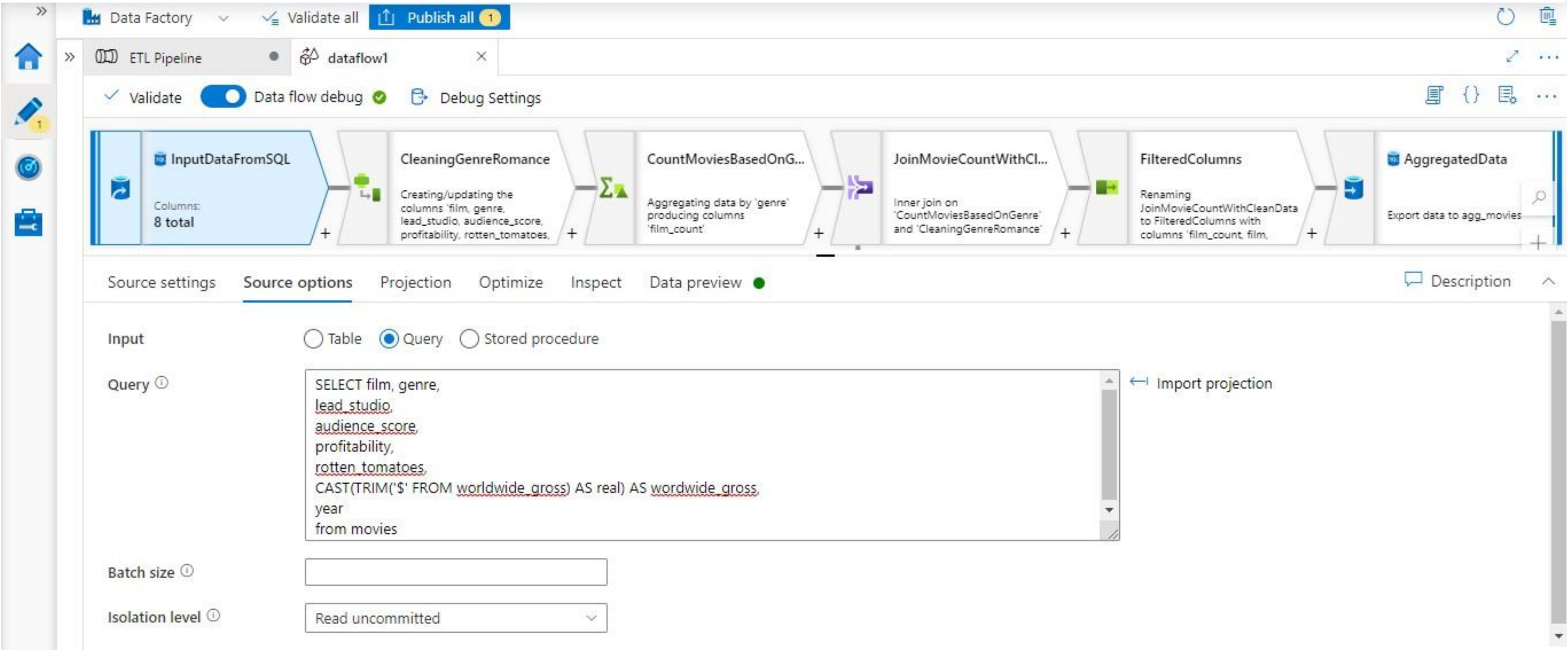
- Get the Input Data from SQL



Number of rows INSERT 77 UPDATE 0 DELETE 0 UPSERT 0 LOOKUP 0 ERROR 0 TOTAL 77

	film abc	genre abc	lead_studio abc	audience_score 123	profitability 1.2f	rotten_tomatoes 123
+	Zack and Miri Make a Porno	Romance	The Weinstein Company	70	1.7475417	64
+	Youth in Revolt	Comedy	The Weinstein Company	52	1.09	68
+	You Will Meet a Tall Dark Stra...	Comedy	Independent	35	1.2118182	43
+	When in Rome	Comedy	Disney	44	0.0	15

- Remove ‘$’ from worldwide_gross and make it float using query in source options


Data Factory
Validate all
Publish all 1
ETL Pipeline
dataflow1
Validate
Data flow debug
Debug Settings
InputDataFromSQL
Columns:
8 total
CleaningGenreRomance
Creating/updating the columns 'film, genre, lead_studio, audience_score, profitability, rotten_tomatoes,
CountMoviesBasedOnG...
Aggregating data by 'genre' producing columns 'film_count'
JoinMovieCountWithCl...
Inner join on 'CountMoviesBasedOnGenre' and 'CleaningGenreRomance'
FilteredColumns
Renaming JoinMovieCountWithCleanData to FilteredColumns with columns 'film_count, film,
AggregatedData
Export data to agg_movies
Source settings
Source options
Projection
Optimize
Inspect
Data preview
Description
Import projection

Input ○ Table ● Query ○ Stored procedure

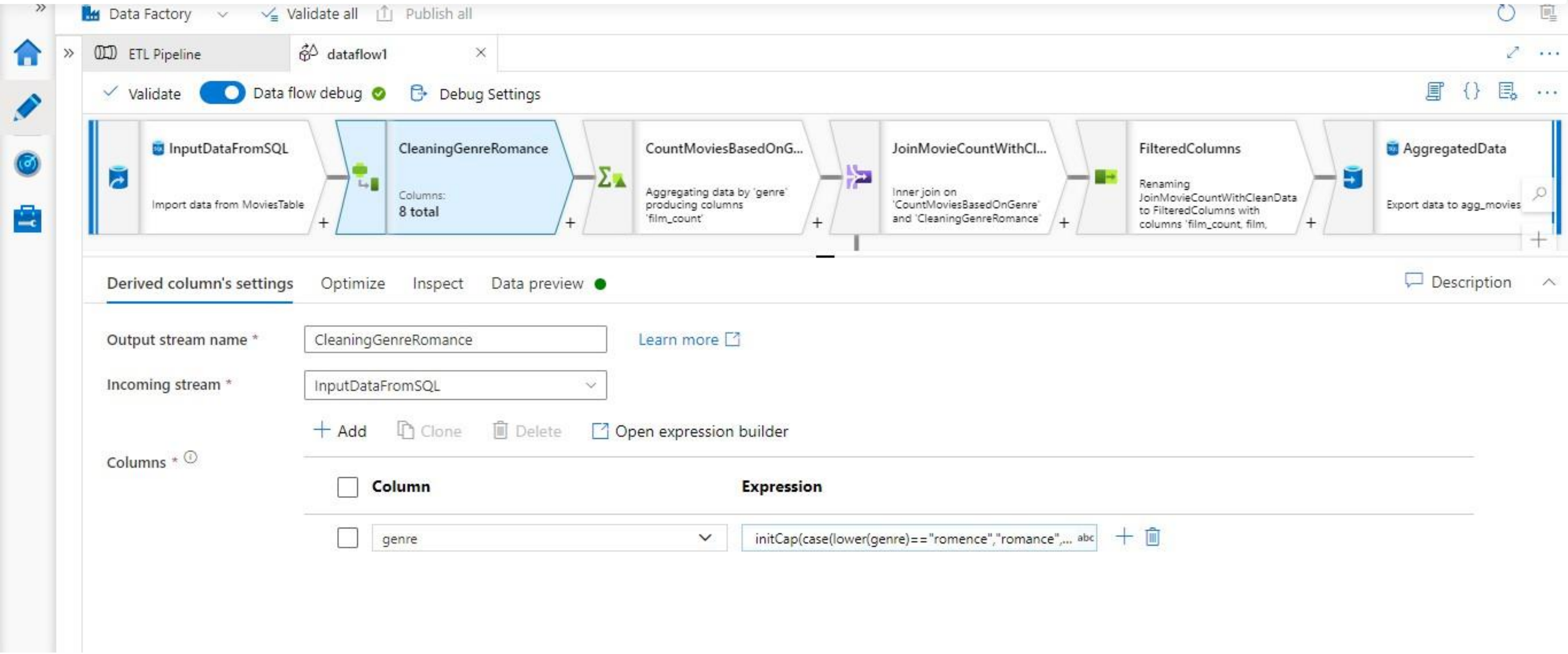
Query

```
SELECT film, genre,
lead_studio,
audience_score,
profitability,
rotten_tomatoes,
CAST(TRIM('$' FROM worldwide_gross) AS real) AS wordwide_gross,
year
from movies
```

Batch size

Isolation level: Read uncommitted

Step 3: Clean the Genre column which has typo mistakes as required



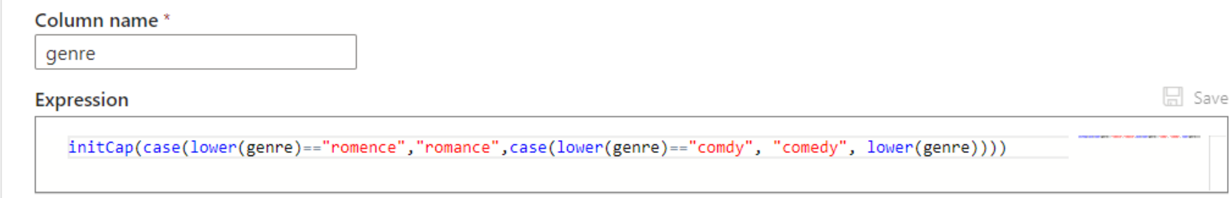
Output stream name *: CleaningGenreRomance

Incoming stream *: InputDataFromSQL

Columns *

Column	Expression
genre	initCap(case(lower(genre)=="romence","romance",...

- Cleaning using expression builder



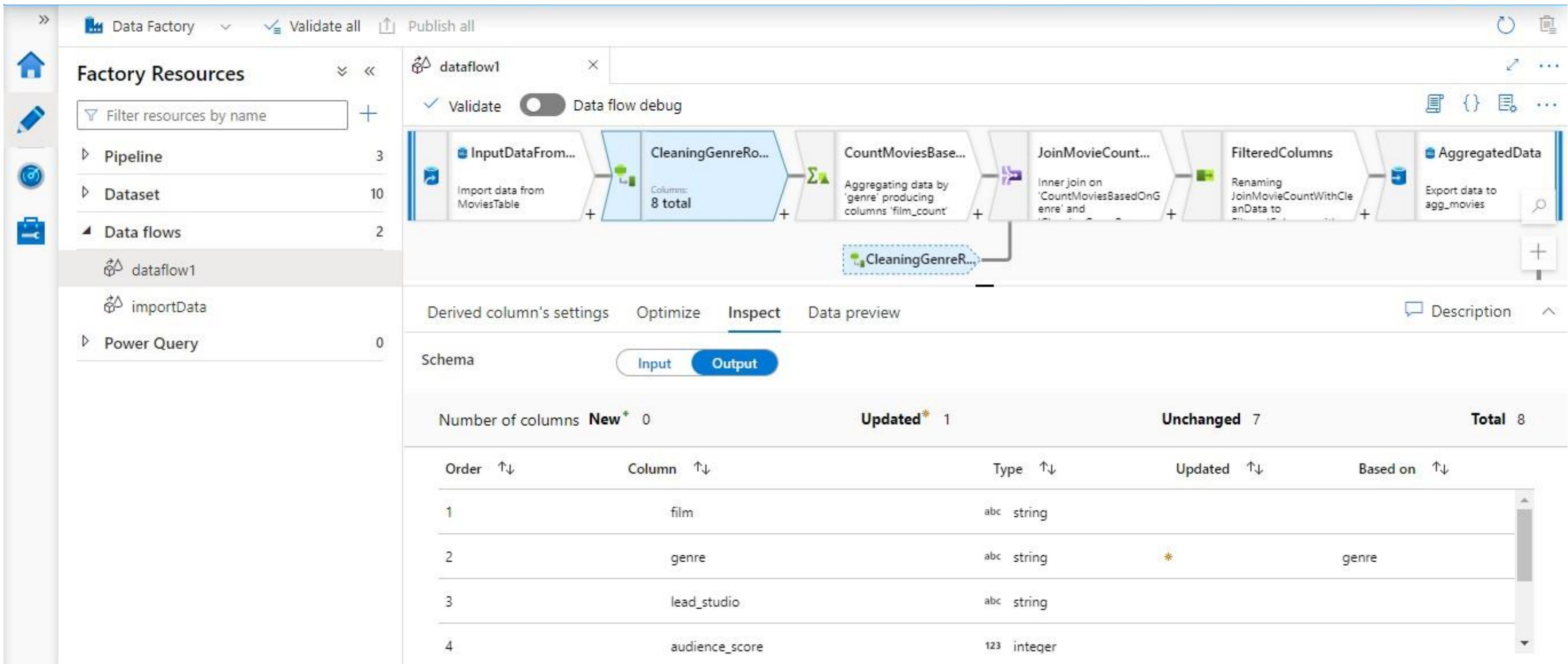
Column name *

genre

Expression

```
initCap(case(lower(genre)=="romence","romance",case(lower(genre)=="comdy", "comedy", lower(genre))))
```

- You can see that genre column is updated

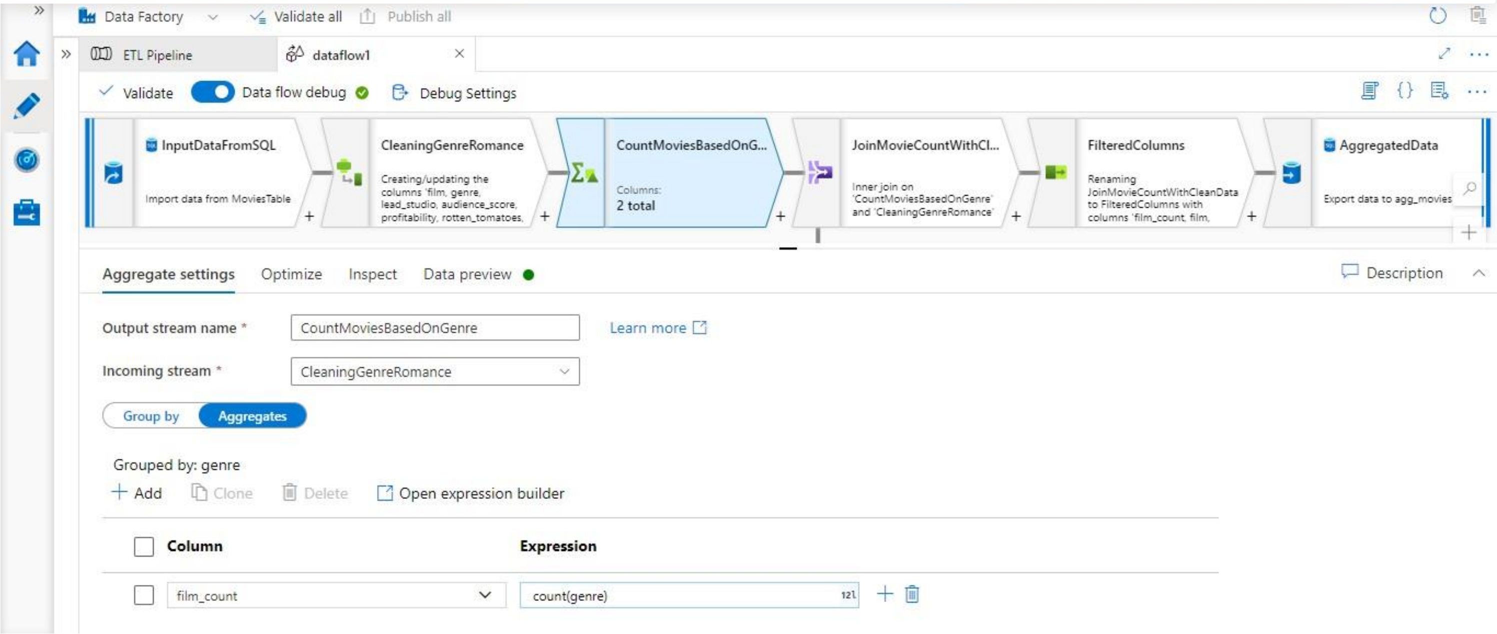


Pipeline	3
Dataset	10
Data flows	2
dataflow1	
importData	
Power Query	0

Number of columns New * 0 Updated * 1 Unchanged 7 Total 8

Order	Column	Type	Updated	Based on
1	film	abc string		
2	genre	abc string	*	genre
3	lead_studio	abc string		
4	audience_score	123 integer		

Step 4: Find the count of films based on genre of the film using aggregate functions



Output stream name * CountMoviesBasedOnGenre

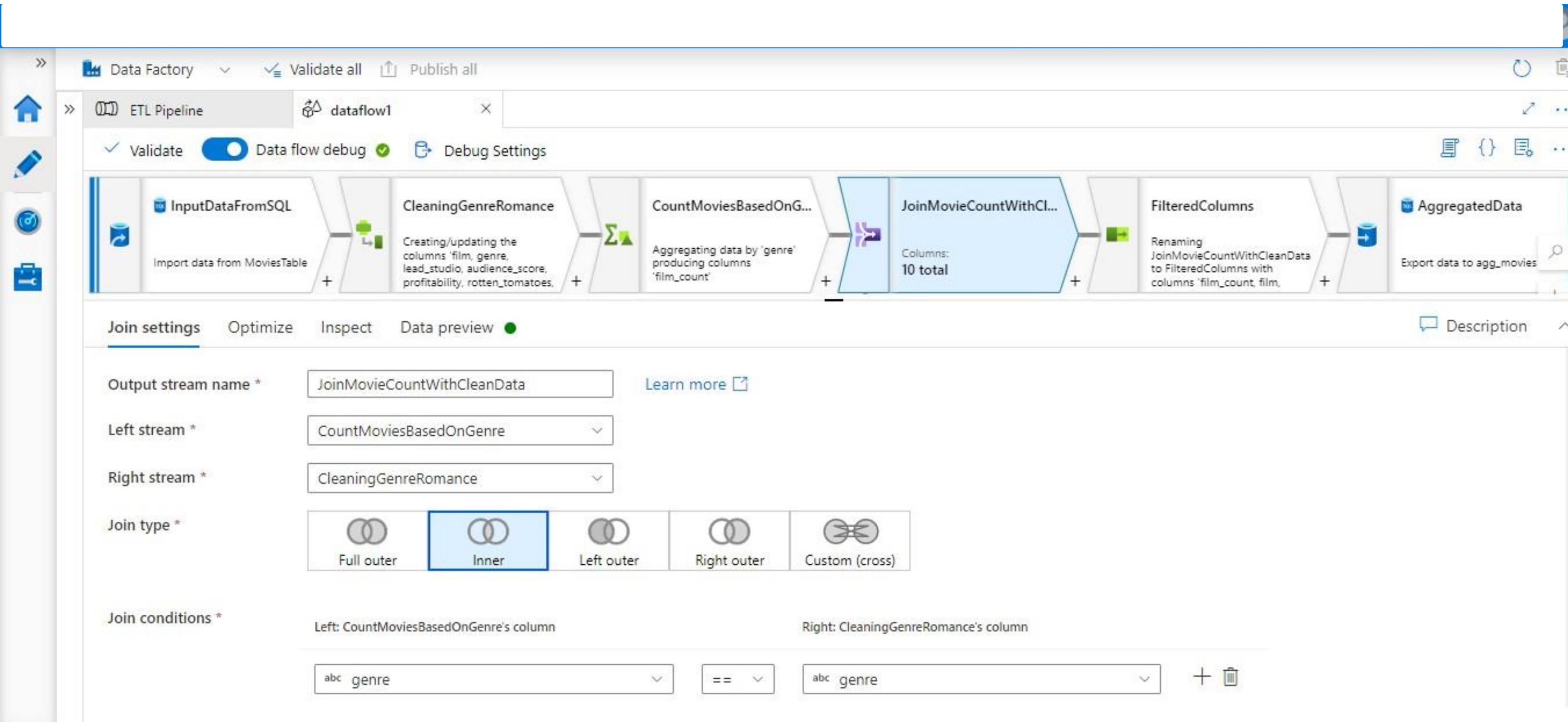
Incoming stream * CleaningGenreRomance

Group by | Aggregates

Grouped by: genre

Column	Expression
film_count	count(genre)

Step 5: Join the film count(step 4) with existing clean data(step 3) using genre column


Data Factory
Validate all
Publish all
ETL Pipeline
dataflow1
Validate
Data flow debug
Debug Settings
InputDataFromSQL
Import data from MoviesTable
CleaningGenreRomance
Creating/updating the columns 'film, genre, lead_studio, audience_score, profitability, rotten_tomatoes,
CountMoviesBasedOnG...
Aggregating data by 'genre' producing columns 'film_count'
JoinMovieCountWithCl...
Columns:
10 total
FilteredColumns
Renaming JoinMovieCountWithCleanData to FilteredColumns with columns 'film_count, film,
AggregatedData
Export data to agg_movies
Join settings
Optimize
Inspect
Data preview
Description
Learn more

Output stream name * JoinMovieCountWithCleanData

Left stream * CountMoviesBasedOnGenre

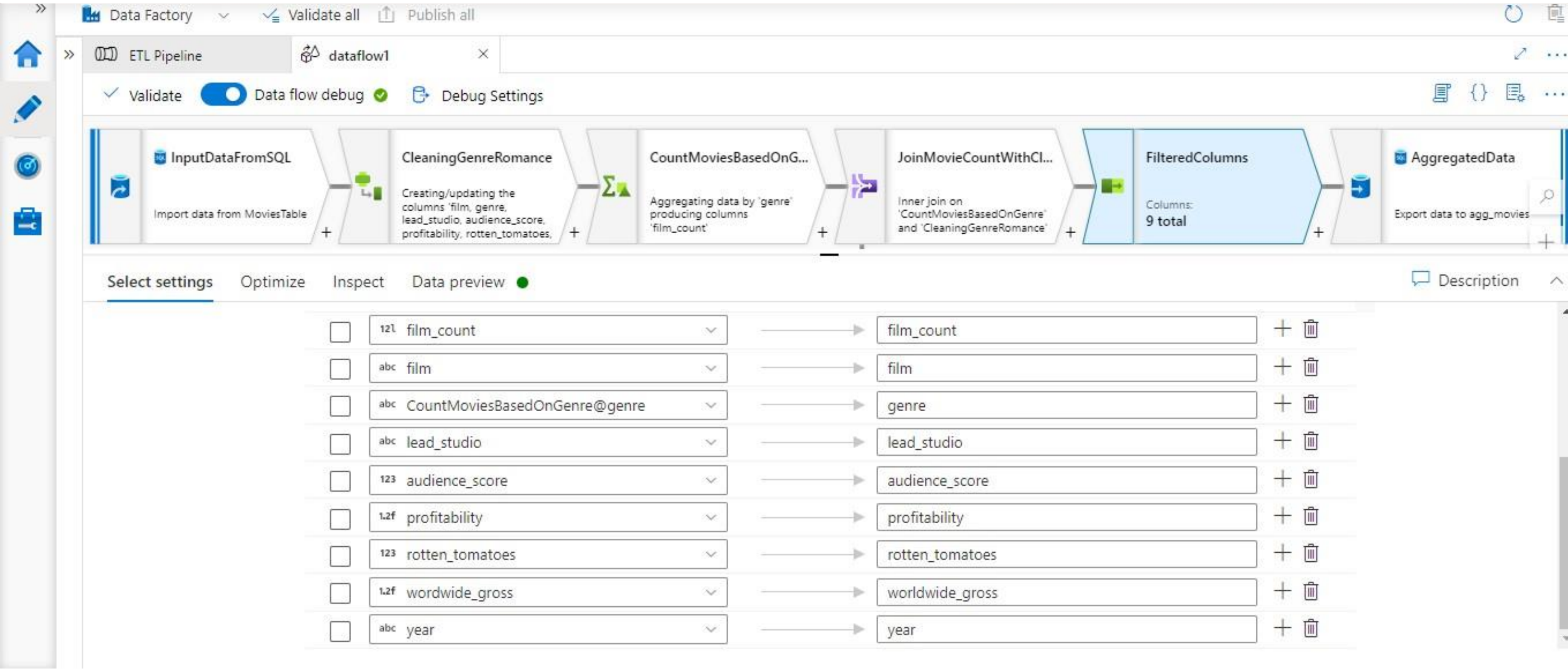
Right stream * CleaningGenreRomance

Join type * Full outer [✓ Inner] Left outer Right outer Custom (cross)

Join conditions *

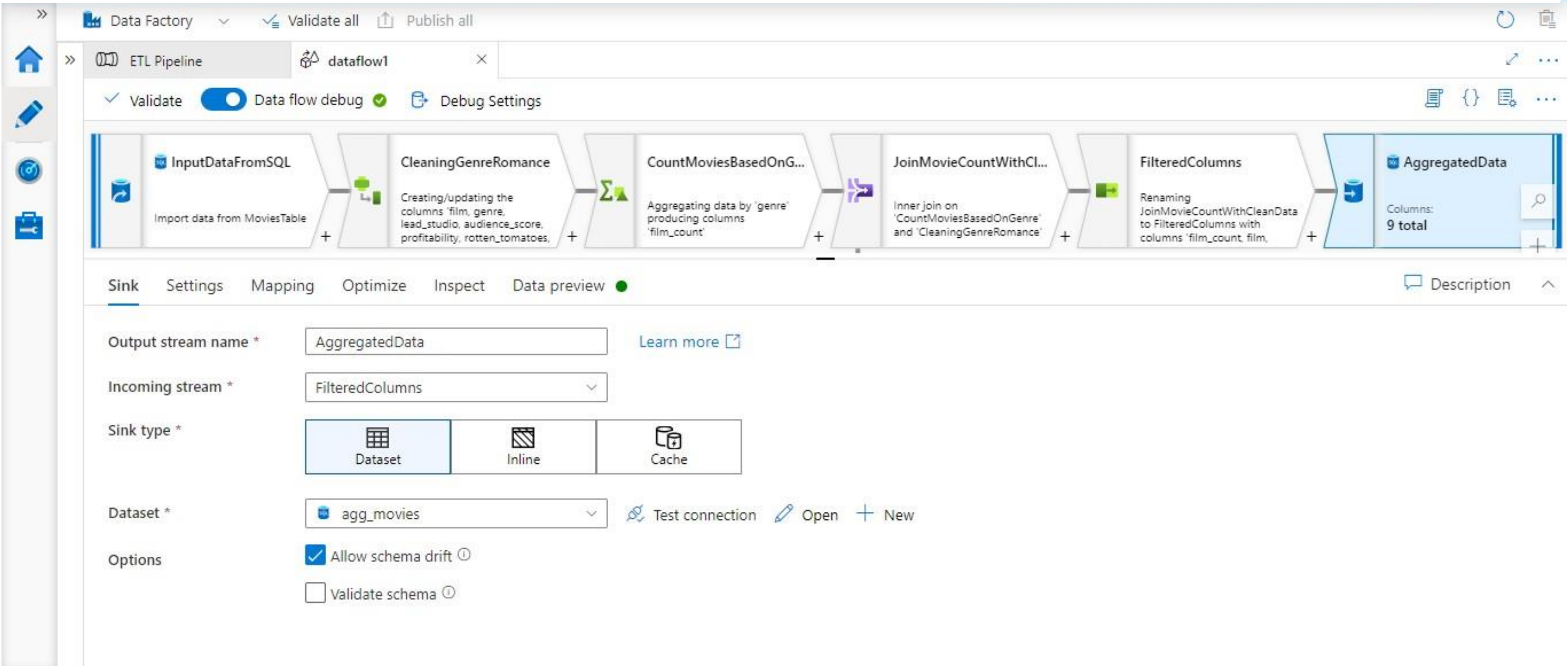
Left: CountMoviesBasedOnGenre's column		Right: CleaningGenreRomance's column
abc genre	==	abc genre

Step 6: Now that we have joined two tables, we'll have two genre columns which is there in both table, we can remove this using select function


Data Factory
Validate all
Publish all
ETL Pipeline
dataflow1
Validate
Data flow debug
Debug Settings
InputDataFromSQL
Import data from MoviesTable
CleaningGenreRomance
Creating/updating the columns 'film, genre, lead_studio, audience_score, profitability, rotten_tomatoes,
CountMoviesBasedOnG...
Aggregating data by 'genre' producing columns 'film_count'
JoinMovieCountWithCl...
Inner join on 'CountMoviesBasedOnGenre' and 'CleaningGenreRomance'
FilteredColumns
Columns:
9 total
AggregatedData
Export data to agg_movies
Select settings
Optimize
Inspect
Data preview
Description

	Input column		Name as
□	121 film_count	→	film_count
□	abc film	→	film
□	abc CountMoviesBasedOnGenre@genre	→	genre
□	abc lead_studio	→	lead_studio
□	123 audience_score	→	audience_score
□	1.2f profitability	→	profitability
□	123 rotten_tomatoes	→	rotten_tomatoes
□	1.2f wordwide_gross	→	worldwide_gross
□	abc year	→	year

Step 7: Store the resultant data in "agg_movies" table which we already created


Data Factory
Validate all
Publish all
ETL Pipeline
dataflow1
Validate
Data flow debug
Debug Settings
InputDataFromSQL
Import data from MoviesTable
CleaningGenreRomance
Creating/updating the columns 'film, genre, lead_studio, audience_score, profitability, rotten_tomatoes,
CountMoviesBasedOnG...
Aggregating data by 'genre' producing columns 'film_count'
JoinMovieCountWithCl...
Inner join on 'CountMoviesBasedOnGenre' and 'CleaningGenreRomance'
FilteredColumns
Renaming JoinMovieCountWithCleanData to FilteredColumns with columns 'film_count, film,
AggregatedData
Columns:
9 total
Sink
Settings
Mapping
Optimize
Inspect
Data preview
Description
Learn more
Test connection
Open
New

Output stream name * AggregatedData

Incoming stream * FilteredColumns

Sink type * Dataset | Inline | Cache

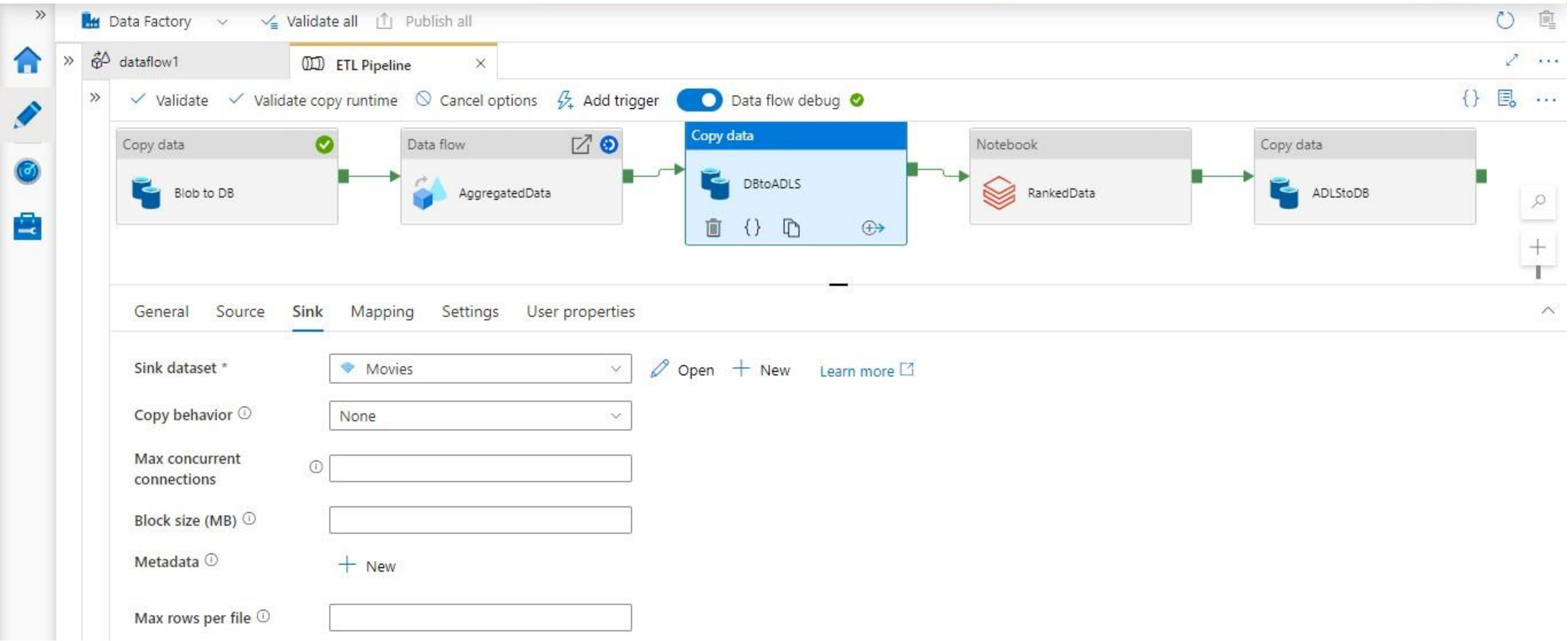
Dataset * agg_movies

Options

☑ Allow schema drift

☐ Validate schema

Step 8: Grab the dataflow to the existing pipeline, then add a copy data to transfer the table data to ADLS in PARQUET FORMAT



Sink dataset * Movies

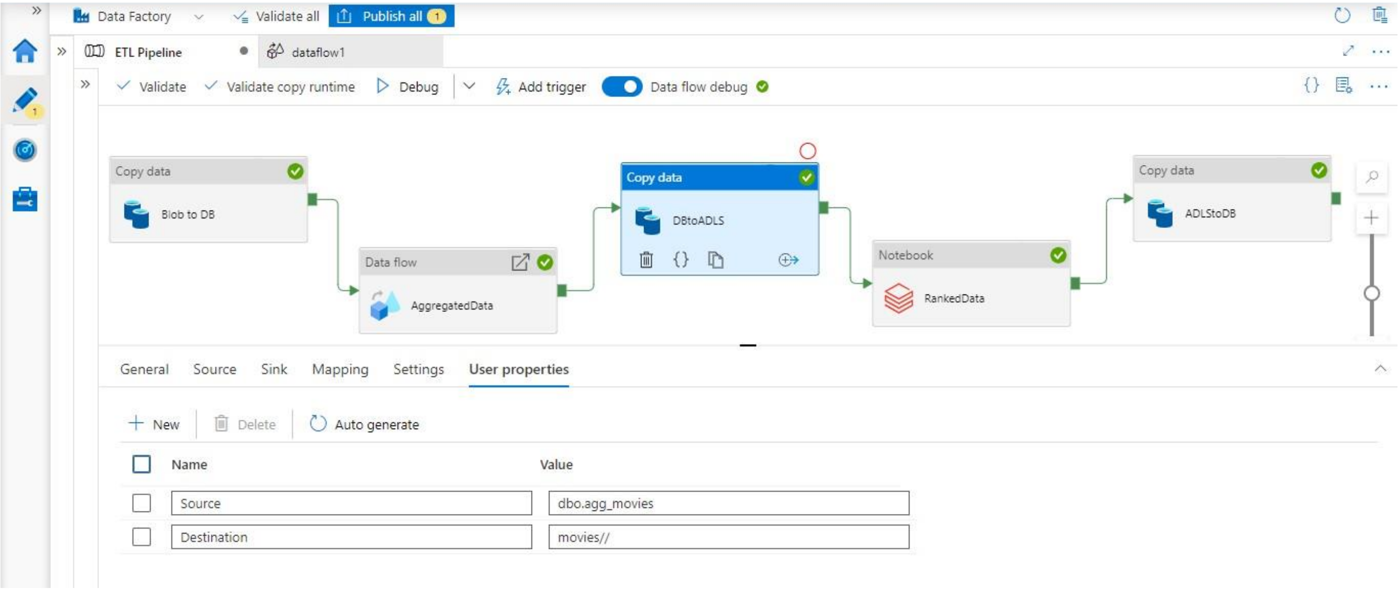
Copy behavior None

Max concurrent connections

Block size (MB)

Metadata + New

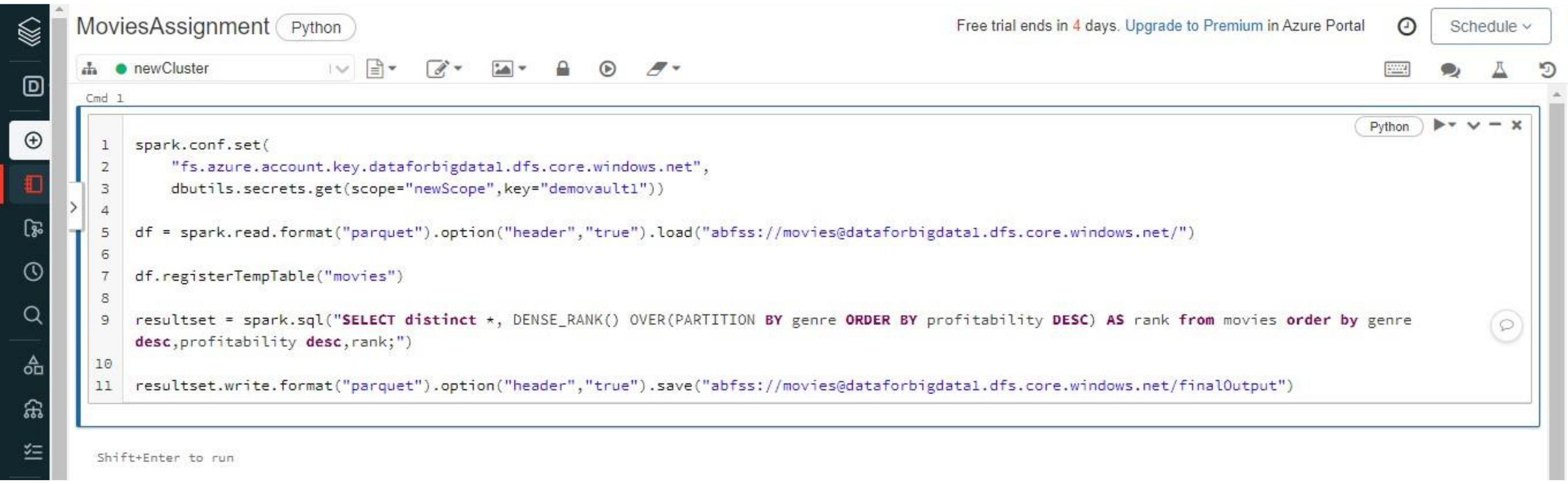
Max rows per file



Name	Value
Source	dbo.agg_movies
Destination	movies//

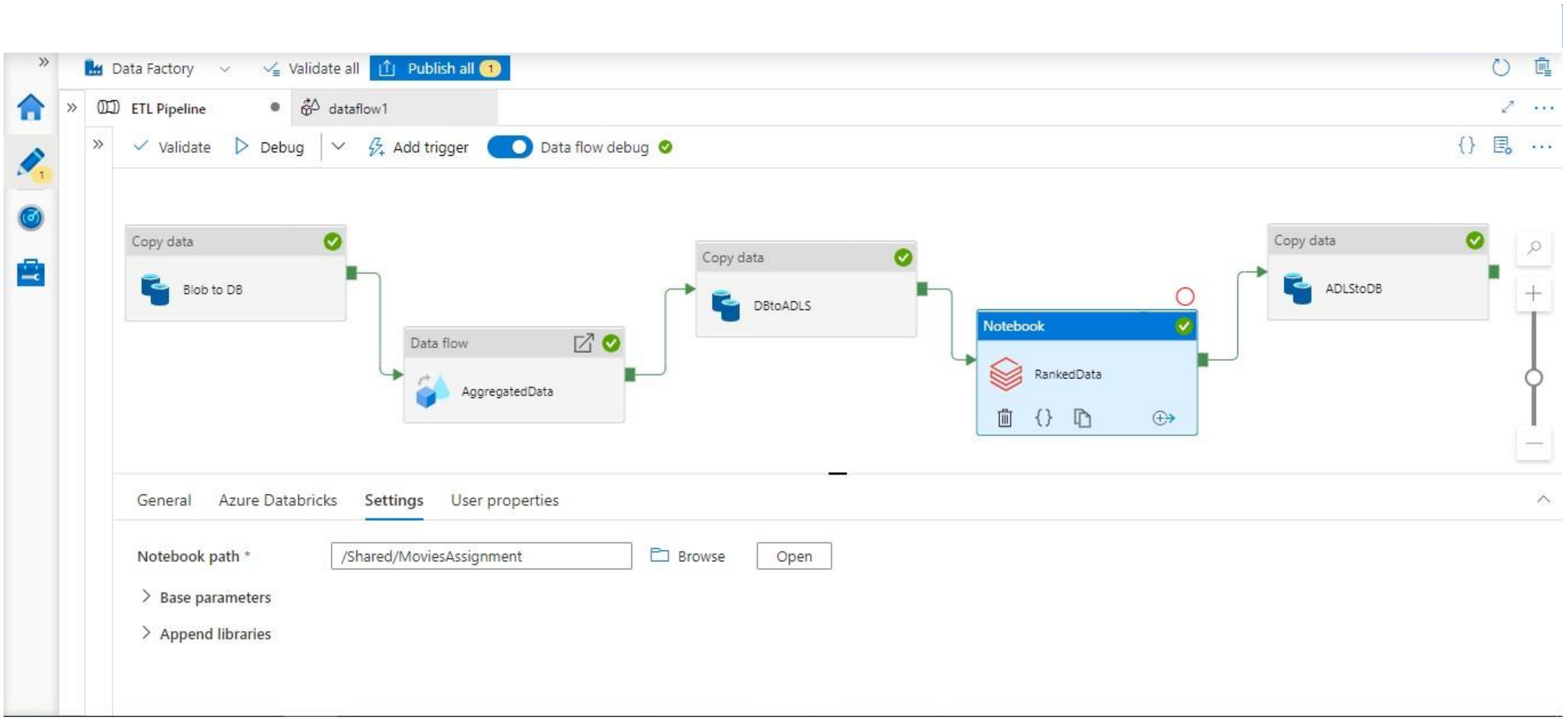
Step 9: Create a notebook to rank the data existing in ADLS and store it back in ADLS in the same parquet format

- To do this, you'll need to create a key vault and connect it to the notebook to make the ADLS accessible
- A new folder "finalOutput" will be created inside the movies container in ADLS.
- Scope and key mentioned here are created as part of prerequisites



```
spark.conf.set(
    "fs.azure.account.key.dataforbigdatal.dfs.core.windows.net",
    dbutils.secrets.get(scope="newScope",key="demovault1"))

df = spark.read.format("parquet").option("header","true").load("abfss://movies@dataforbigdatal.dfs.core.windows.net/")

df.registerTempTable("movies")

resultset = spark.sql("SELECT distinct *, DENSE_RANK() OVER(PARTITION BY genre ORDER BY profitability DESC) AS rank from movies order by genre desc,profitability desc,rank;")

resultset.write.format("parquet").option("header","true").save("abfss://movies@dataforbigdatal.dfs.core.windows.net/finalOutput")
```

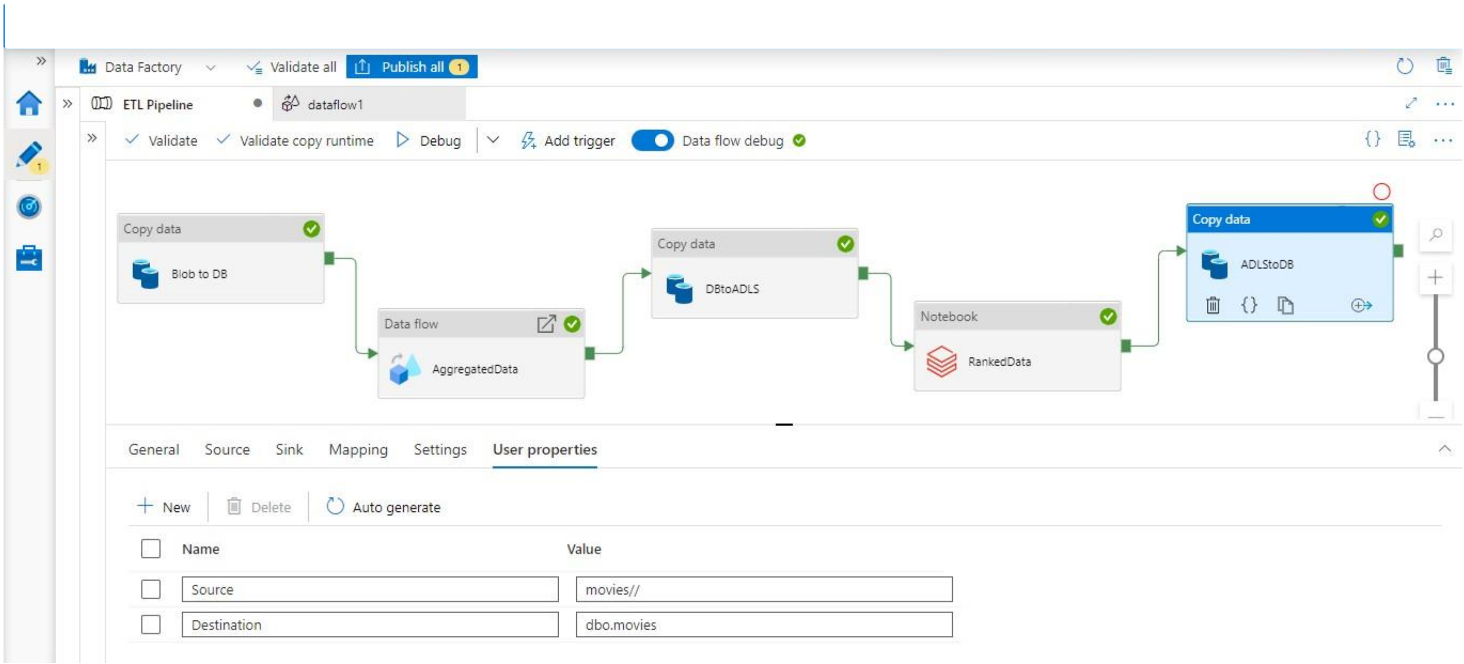
Step 10: Now grab the notebook created and connect it to the existing pipeline such that the notebook execution happens only in the sequential order of the pipeline (applies to all the steps)



Notebook path * /Shared/MoviesAssignment

Step 11: Grab a copy data which performs copying the dataset in ADLS to SQL POOL in Synapse Analytics

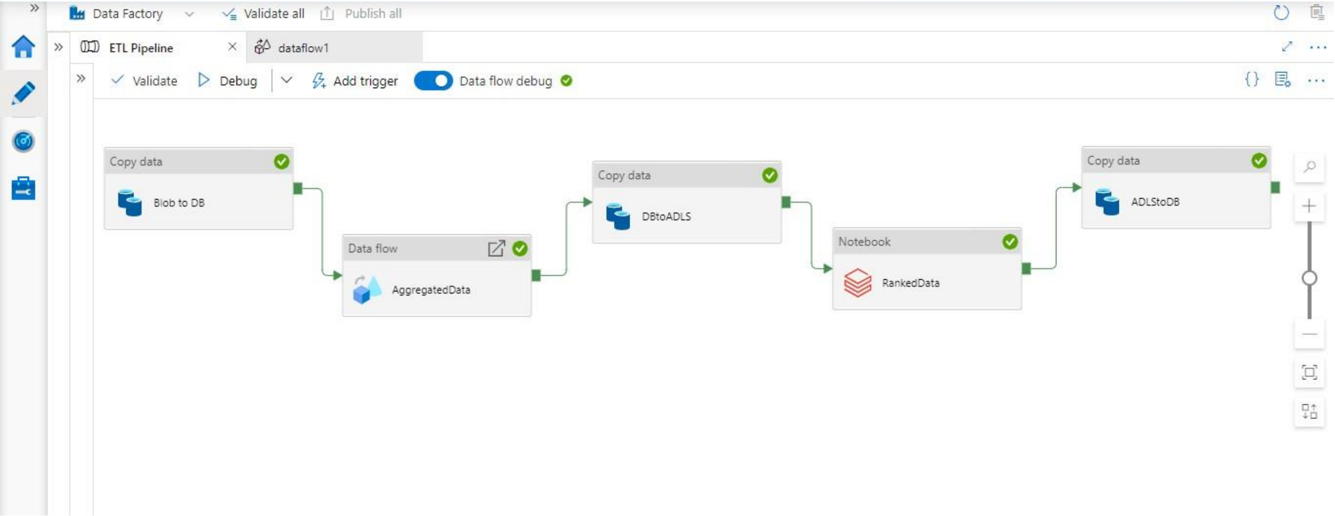
- Data will be inserted to the table “movies” that is already created in SQL POOL



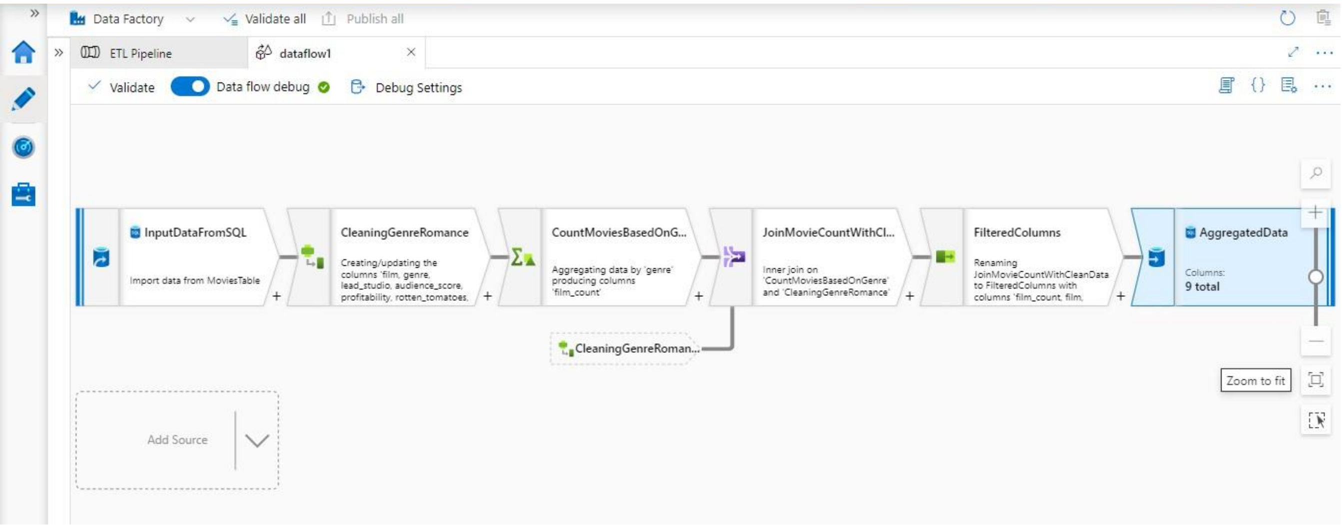
□	Name	Value
□	Source	movies//
□	Destination	dbo.movies

Result Screenshots:

- **Pipeline**



- **Data Flow**



Import data from MoviesTable

Creating/updating the columns 'film, genre, lead_studio, audience_score, profitability, rotten_tomatoes,

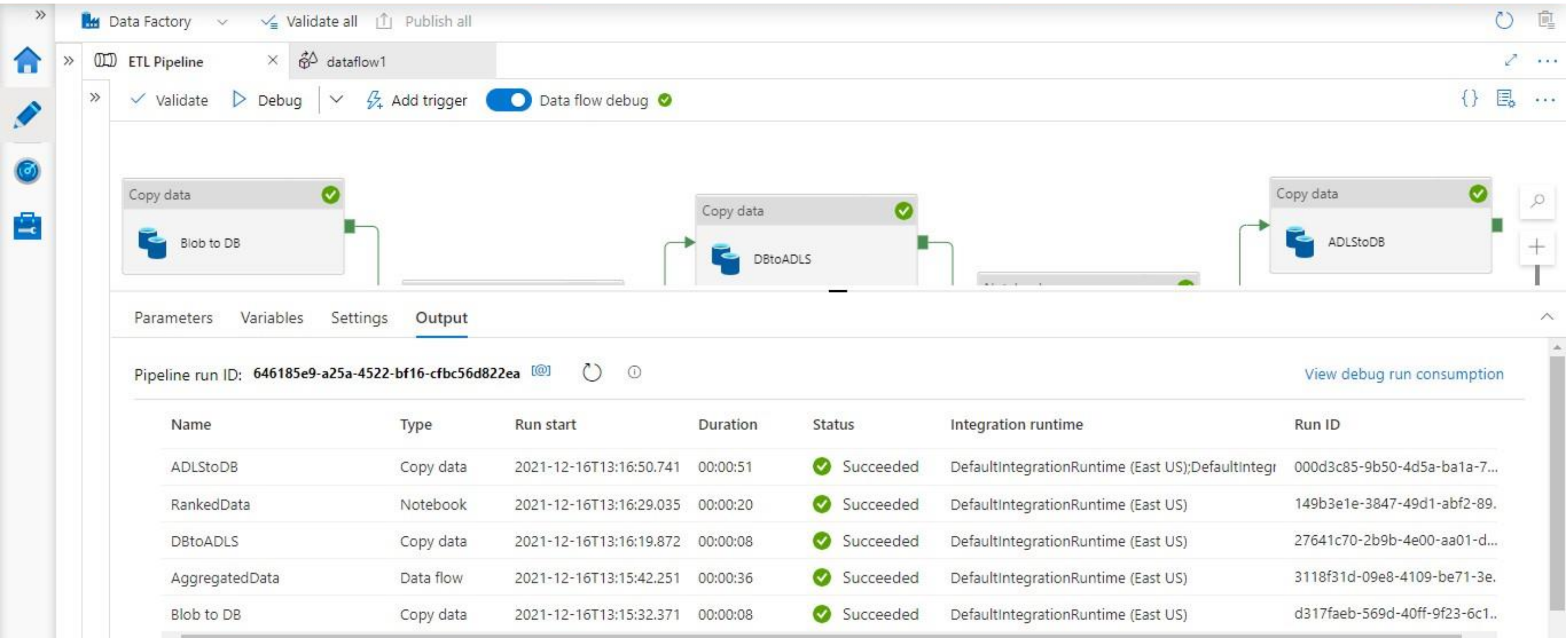
Aggregating data by 'genre' producing columns 'film_count'

Inner join on 'CountMoviesBasedOnGenre' and 'CleaningGenreRomance'

Renaming JoinMovieCountWithCleanData to FilteredColumns with columns 'film_count, film,

Columns:
9 total

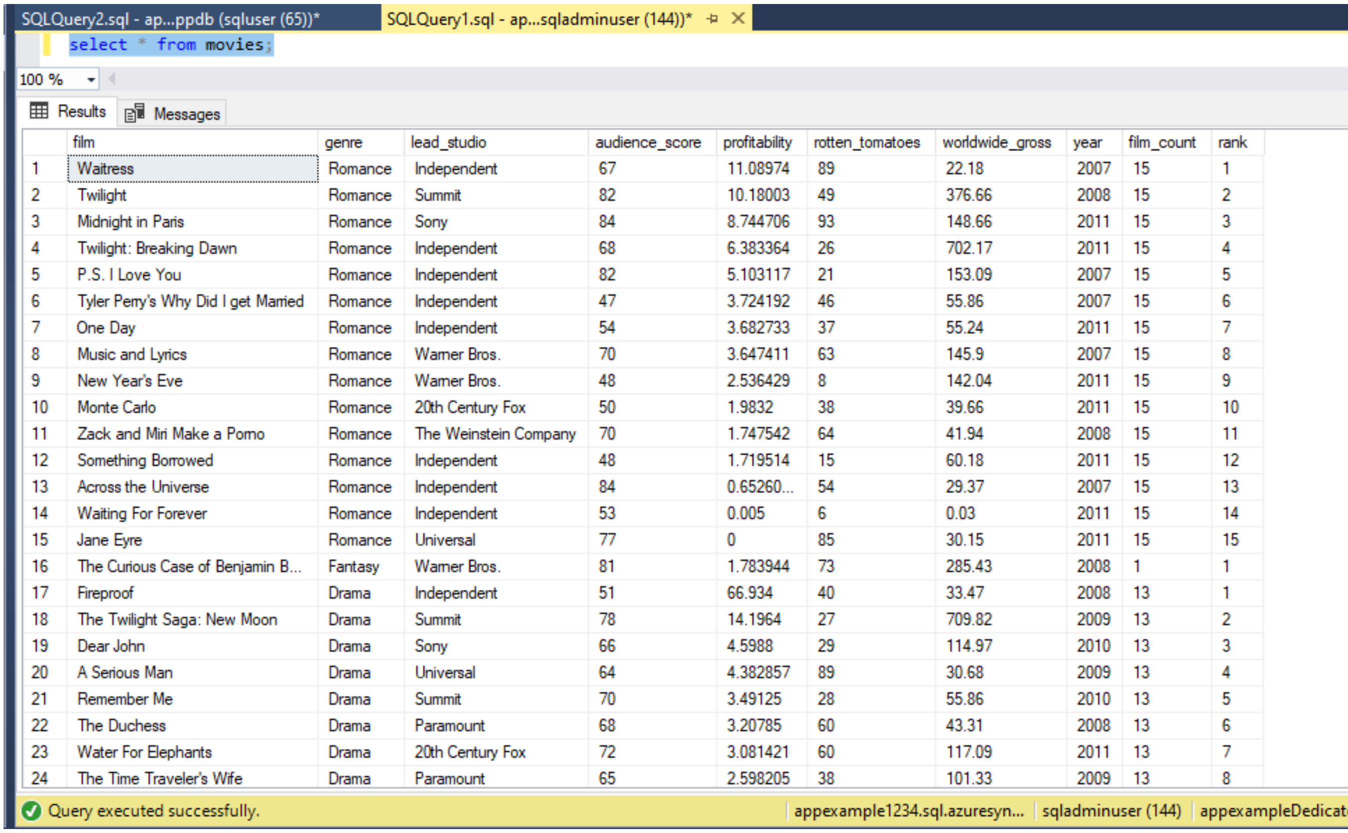
- **Pipeline Successful**



Pipeline run ID: 646185e9-a25a-4522-bf16-cfbc56d822ea

Name	Type	Run start	Duration	Status	Integration runtime	Run ID
ADLStoDB	Copy data	2021-12-16T13:16:50.741	00:00:51	✓ Succeeded	DefaultIntegrationRuntime (East US);DefaultIntegr	000d3c85-9b50-4d5a-ba1a-7...
RankedData	Notebook	2021-12-16T13:16:29.035	00:00:20	✓ Succeeded	DefaultIntegrationRuntime (East US)	149b3e1e-3847-49d1-abf2-89.
DBtoADLS	Copy data	2021-12-16T13:16:19.872	00:00:08	✓ Succeeded	DefaultIntegrationRuntime (East US)	27641c70-2b9b-4e00-aa01-d...
AggregatedData	Data flow	2021-12-16T13:15:42.251	00:00:36	✓ Succeeded	DefaultIntegrationRuntime (East US)	3118f31d-09e8-4109-be71-3e.
Blob to DB	Copy data	2021-12-16T13:15:32.371	00:00:08	✓ Succeeded	DefaultIntegrationRuntime (East US)	d317faeb-569d-40ff-9f23-6c1..

- **Result data**


SQLQuery2.sql - ap...ppdb (sqluser (65))*
SQLQuery1.sql - ap...sqladminuser (144))*
100 %
Results
Messages

```
select * from movies;
```

	film	genre	lead_studio	audience_score	profitability	rotten_tomatoes	worldwide_gross	year	film_count	rank
1	Waitress	Romance	Independent	67	11.08974	89	22.18	2007	15	1
2	Twilight	Romance	Summit	82	10.18003	49	376.66	2008	15	2
3	Midnight in Paris	Romance	Sony	84	8.744706	93	148.66	2011	15	3
4	Twilight: Breaking Dawn	Romance	Independent	68	6.383364	26	702.17	2011	15	4
5	P.S. I Love You	Romance	Independent	82	5.103117	21	153.09	2007	15	5
6	Tyler Perry's Why Did I get Married	Romance	Independent	47	3.724192	46	55.86	2007	15	6
7	One Day	Romance	Independent	54	3.682733	37	55.24	2011	15	7
8	Music and Lyrics	Romance	Warner Bros.	70	3.647411	63	145.9	2007	15	8
9	New Year's Eve	Romance	Warner Bros.	48	2.536429	8	142.04	2011	15	9
10	Monte Carlo	Romance	20th Century Fox	50	1.9832	38	39.66	2011	15	10
11	Zack and Miri Make a Porno	Romance	The Weinstein Company	70	1.747542	64	41.94	2008	15	11
12	Something Borrowed	Romance	Independent	48	1.719514	15	60.18	2011	15	12
13	Across the Universe	Romance	Independent	84	0.65260...	54	29.37	2007	15	13
14	Waiting For Forever	Romance	Independent	53	0.005	6	0.03	2011	15	14
15	Jane Eyre	Romance	Universal	77	0	85	30.15	2011	15	15
16	The Curious Case of Benjamin B...	Fantasy	Warner Bros.	81	1.783944	73	285.43	2008	1	1
17	Fireproof	Drama	Independent	51	66.934	40	33.47	2008	13	1
18	The Twilight Saga: New Moon	Drama	Summit	78	14.1964	27	709.82	2009	13	2
19	Dear John	Drama	Sony	66	4.5988	29	114.97	2010	13	3
20	A Serious Man	Drama	Universal	64	4.382857	89	30.68	2009	13	4
21	Remember Me	Drama	Summit	70	3.49125	28	55.86	2010	13	5
22	The Duchess	Drama	Paramount	68	3.20785	60	43.31	2008	13	6
23	Water For Elephants	Drama	20th Century Fox	72	3.081421	60	117.09	2011	13	7
24	The Time Traveler's Wife	Drama	Paramount	65	2.598205	38	101.33	2009	13	8

✓ Query executed successfully. | appexample1234.sql.azuresyn... | sqladminuser (144) | appexampleDedicat