**Case Study**

**DBMS**

**By**

Name : Sidharth Bansal

Employee Id : 655314

**Codd’s Rule :**

1. The Foundation Rule
2. The Information Rule
3. The Guaranteed Access Rule
4. Systematic Treatment Of NULL Values
5. Active/Dynamic Online Catalog Based On The Relational Model
6. The Comprehensive Data Sublanguage Rule
7. The View Updating Rule
8. High-Level Insert, Update & Delete Rule
9. Physical Data Independence
10. Logical Data Independence
11. Integrity Independence
12. Distribution Independence
13. The Non-subversion Rule

**Attributes and Types:**

**Simple attribute :**

An attribute that cannot be further subdivided into components is a simple attribute.

**Composite attribute :**

An attribute that can be split into components is a composite attribute.

**Single-valued attribute :**

The attribute which takes up only a single value for each entity instance is a single-valued attribute.

**Multi-valued attribute :**

The attribute which takes up more than a single value for each entity instance is a multi-valued attribute.

**Derived attribute :**

An attribute that can be derived from other attributes is derived attributes.

**Entities and Types:**

One to One

When a single element of an entity is associated with single element of other entity

One to Many

When a single element of an entity is associated with multiple elements of other entity

Many to One

When multiple elements of an entity of and entity is associated with multiple elements of other entity

Many to Many

Multiple elements of an entity linked with multiple entities of other entity

**DDL (Data Definition Language):**

DDL is a subset of SQL used to define, modify, and manage the structure of a database. It includes commands like CREATE (to create tables, views, indexes), ALTER (to modify existing objects), DROP (to delete objects), and TRUNCATE (to remove all data from a table).

**DML (Data Manipulation Language):**

DML is used to manipulate data stored in the database. It includes commands like SELECT (to retrieve data), INSERT (to add new records), UPDATE (to modify existing records), and DELETE (to remove records).

**DQL (Data Query Language):**

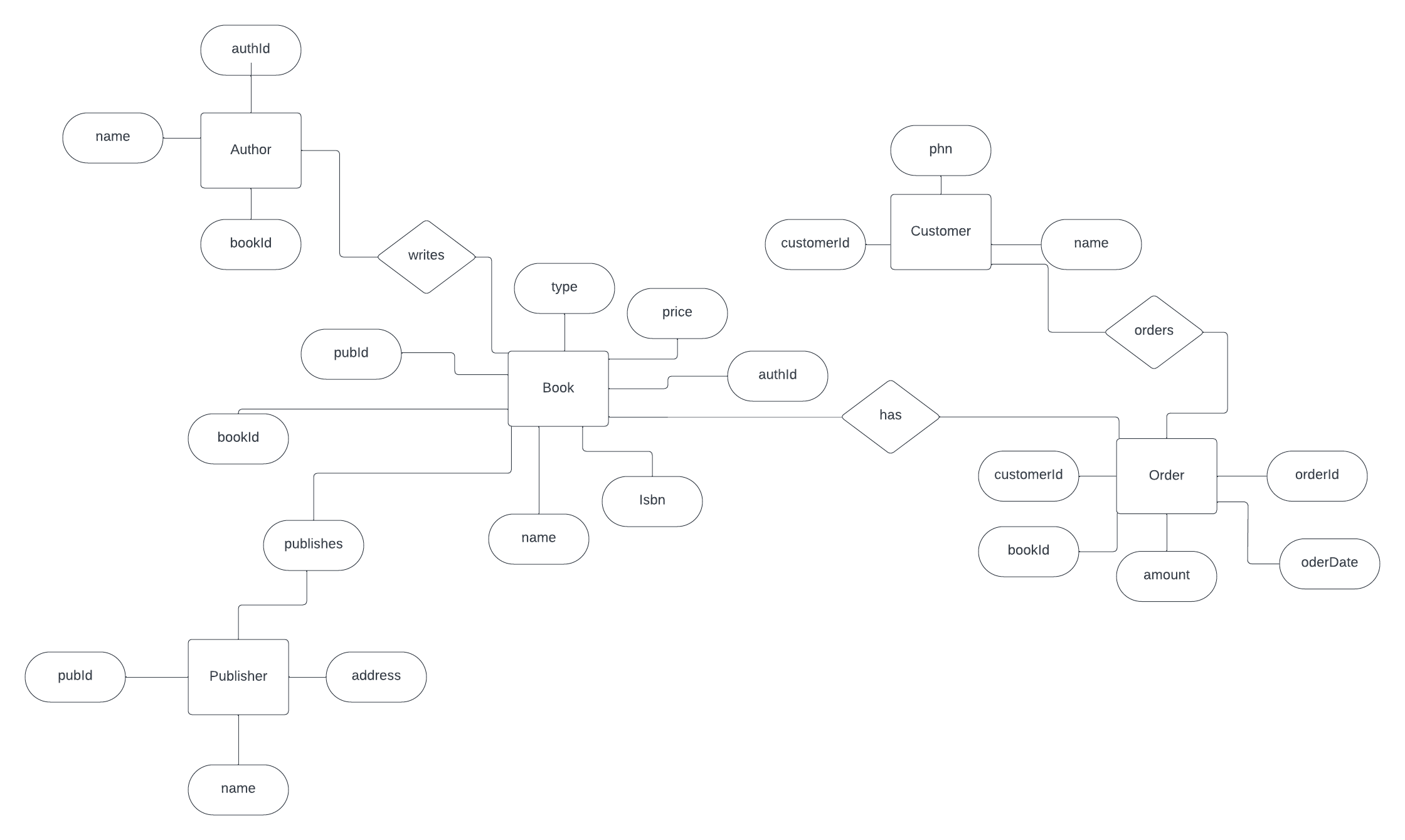
DQL is a subset of SQL focused on querying data from the database. The primary command is SELECT, which retrieves specific data based on conditions you specify in the query.

**DCL (Data Control Language):**

DCL is used to manage permissions and access control within the database. It includes commands like GRANT (to grant specific privileges to users or roles) and REVOKE (to remove privileges).

**TCL (Transaction Control Language):**

TCL is used to manage transactions within the database. It includes commands like COMMIT (to save changes made during the current transaction), ROLLBACK (to undo changes made during the current transaction), and SAVEPOINT (to set a point within a transaction to which you can later roll back).



Initial Tables:

Customer (customerId, name,phn)

Order (orderId, amount, customerId, bookId)

Author (authId, name, bookId)

Publisher (pubId, name, address)

Book(bookId,isbn,type,price,authId,pubId)

Relationships between tables

Customer and Order : One to Many

Order and Book : Many to Many

Author and Book : One to Many

Publisher and Book : One to Many

**1NF:** Given tables are already in 1 NF form.

**2NF:** In 2NF, we remove partial dependencies that are present in each of the tables

**Customer (customerId, name, phn)**

* + customerId (Primary Key)

1. **Order (orderId, amount, customerId, bookId)**
   * orderId (Primary Key)
   * customerId (Foreign Key referencing Customer)
   * bookId (Foreign Key referencing Book)

We need to split Order into two tables because amount and customerId are dependent on orderId

* + **OrderHeader (orderId, customerId)**
    - orderId (Primary Key)
    - customerId (Foreign Key referencing Customer)
  + **OrderDetail (orderId, amount, bookId)**
    - orderId and bookId together form the Primary Key of OrderDetail.

1. **Author (authId, name, bookId)**
   * authId (Primary Key)
   * bookId (Foreign Key referencing Book)
2. **Publisher (pubId, name, address)**
   * PublisherID (Primary Key)
3. **Book (bookId, isbn, name, price, authId, pubId)**
   * bookId (Primary Key)
   * authId (Foreign Key referencing Author)
   * pubId (Foreign Key referencing Publisher)

We need to split Book into two tables because : **authId** and **pubId** depend on **isbn**, but they should depend on the whole primary key, **bookId**.

* + **BookInfo (bookId, name, price, pubId)**
    - bookId (Primary Key)
    - pubId (Foreign Key referencing Publisher)
  + **BookAuthor (bookId, authId)**
    - bookId form the Primary Key of BookAuthor.

**3NF:** In 3NF, we remove transitive dependencies.

1. **Customer (customerId, name, phn)**
   * customerId (Primary Key)
2. **OrderHeader (orderId, customerId)**
   * orderId (Primary Key)
   * customerId (Foreign Key referencing Customer)
3. **OrderDetail (orderId, amount, bookId)**
   * orderId and bookId together form the Primary Key of OrderDetail.
4. **Author (authId, name)**
   * authId (Primary Key)
5. **Publisher (pubId, name, address)**
   * pubId (Primary Key)
6. **BookInfo (bookId, name, price, pubId)**
   * bookId (Primary Key)
   * pubId (Foreign Key referencing Publisher)
7. **BookAuthor (bookId, authId)**
   * bookId form the Primary Key of BookAuthor.

Datawarehouse is used for analytical purpose.

Database is used to transactions and day to day data.

RCD – Rapidly

It is a dimension which has attribute where values change often.

SCD – Slowly

It is a concept in data warehousing and database management that deals with managing changes to dimension data over time.

Type 1 : Only Latest state of data is maintained

UCD – Unchanging

Dimension Table

Nomenclature eg Table Employee so for dimension table name will be D\_Employee or Demployee or DimEmployeee or Dim\_Employee

**Big Data Terms**

Volume (large amount of data)

Value

Veracity (Refers to the quality and accuracy of the data)

Visualization

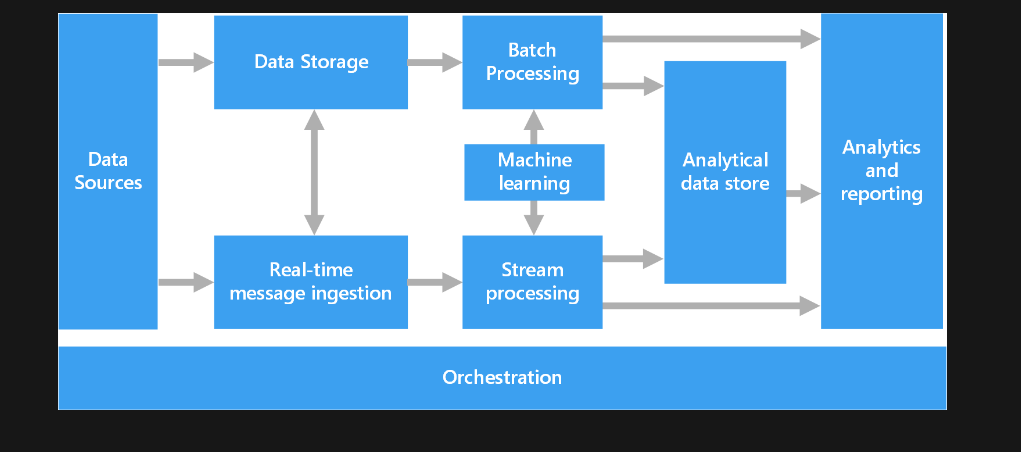
Variety (different type of data like audio,video and IOT devices)

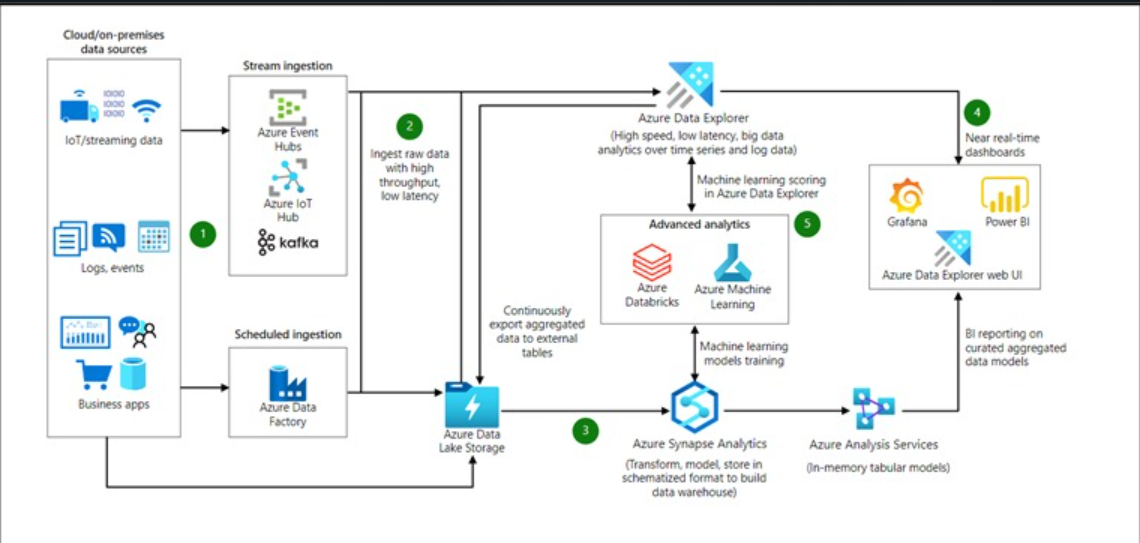
Velocity (Generating data very fast)

Virality

**Challenges of Big Data**

**Big Data Architecture**

****



HDFS

Master slave architecture

Single namenode,a master server that manages the file system namespace

File system namespace

A user or an application can create directories and can store files inside this directory

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon Instruction from the NameNode

Data Replication

HDFS is designed to reliable store very large files across machines in large cluster it store each file as sequence as blocks.The blocks of the file are replicated for fault tolerance

Apache Kafka

Elastic Search

Kibana

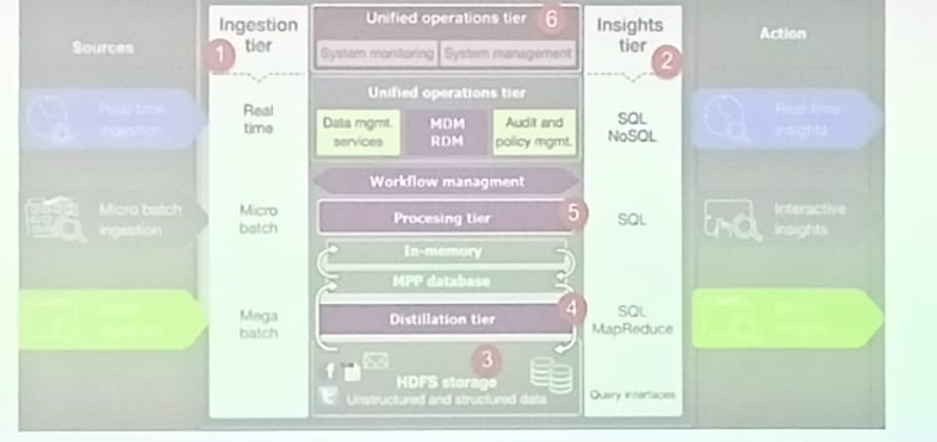
Data Lake

We can build data lakes solution powered by azure

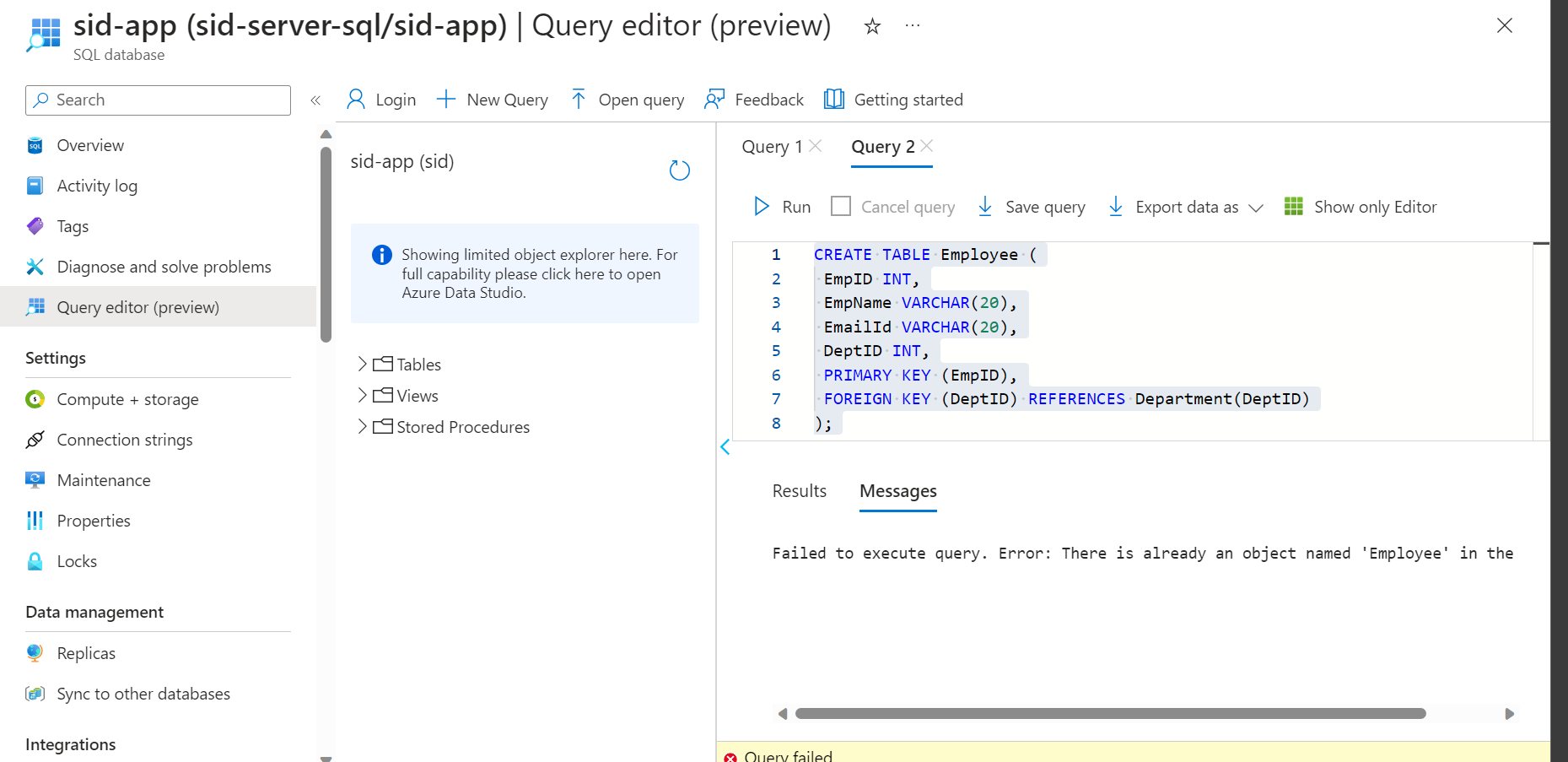
Azure HD insight

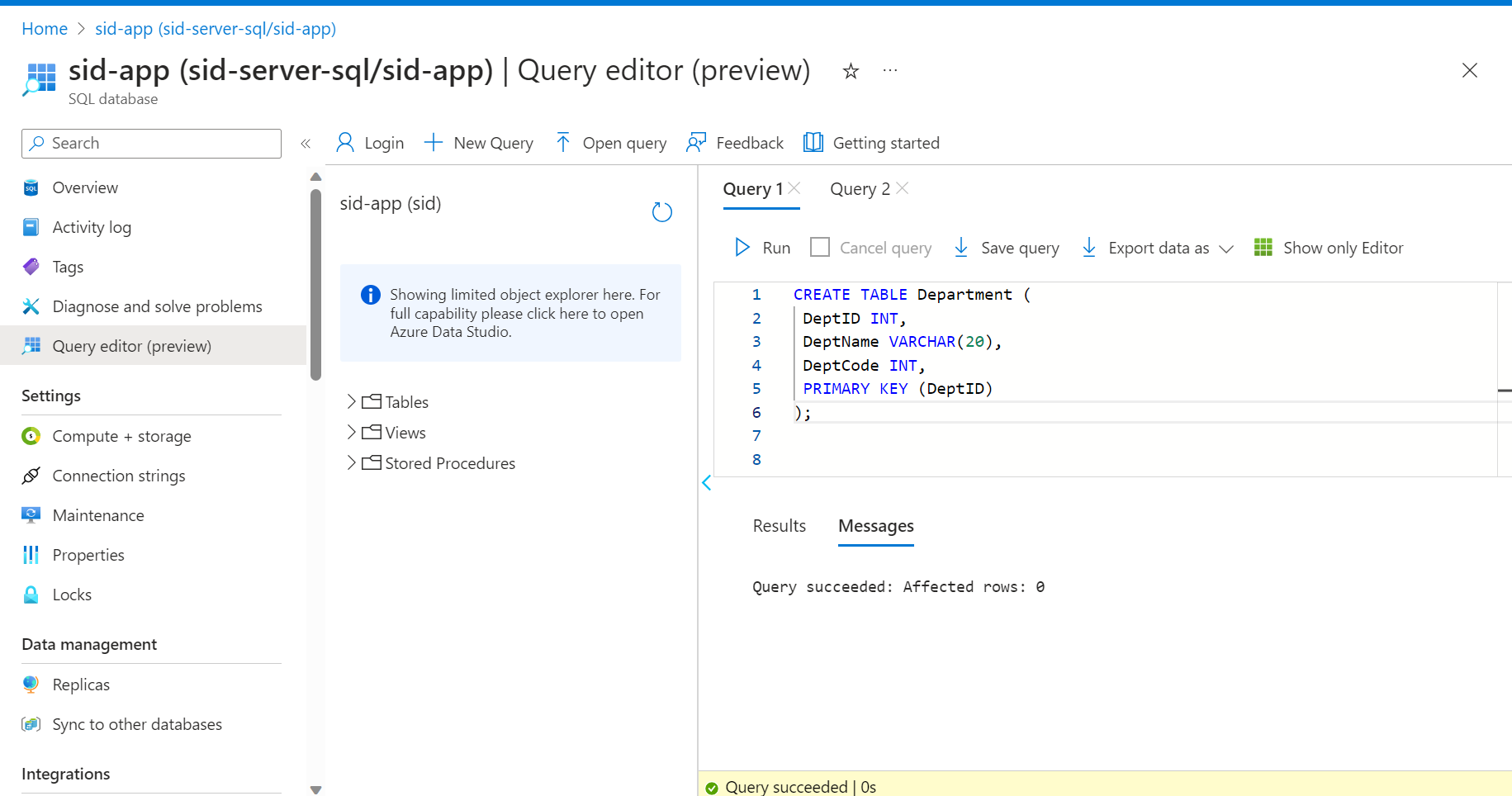
Azure data lake store

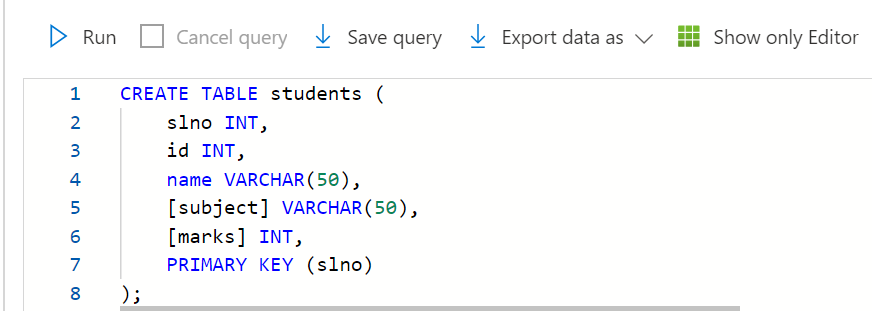
Azure datal lake analytics

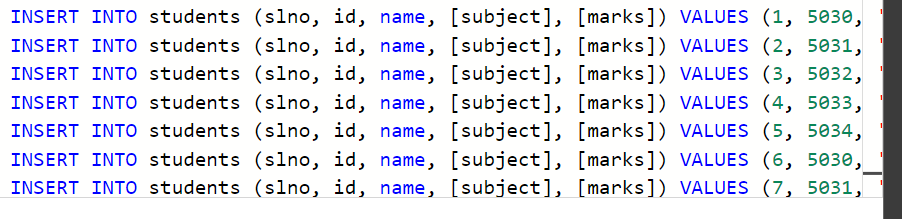


Azure SQL









With cte as(

SELECT DENSE\_RANK() OVER (PARTITION BY SUBJECT ORDER BY MARKS DESC) drn,

[subject],marks,[name] from students)

Select \* from cte where drn = 1

select [subject],[name],marks from students order by [subject],marks desc

select [name],sum(marks) as t from students group by [subject] order by t

correlated sub-query

Queries that return multiple columns as output depending on the information obtained from the parent query are called correlated sub-query

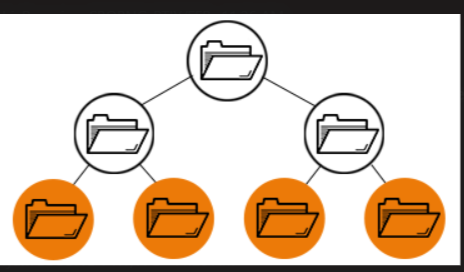
Create clustered index idx\_salesorderlinkekey on sales\_order\_data

9/4/202s3

**Types of storages in Azure**

**File storage**

It is a data stored as single piece of information inside a folder



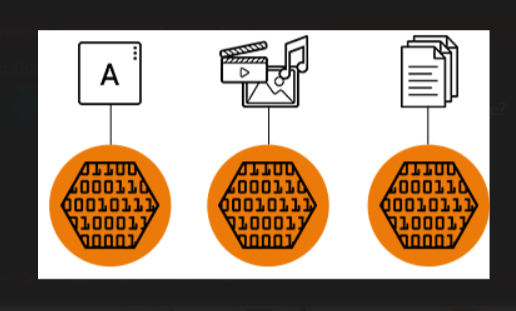
**Block Storage**

Chops data into blocks and stores data them as separate pieces. Each block is given an unique identifier,which follows the storage system to place smaller pieces of data wherever most convenient.It requires a simple http ai which is used by most clients in all languages.



**Object Storage**

Files are broken into pieces and spread across hardware. In this the data is broken into 3 units old object,discrete units and kept in a single repository instead of being kept as a file in folders or as blocks on servers.



Azure storage offers different access tier so that we can store blob data in most cost-effective manner.

Azure storage type includes

* **Hot tier** - An online tier optimized for storing data that is accessed or modified frequently. The hot tier has the highest storage costs, but the lowest access costs.
* **Cool tier** - An online tier optimized for storing data that is infrequently accessed or modified. Data in the cool tier should be stored for a minimum of **30** days. The cool tier has lower storage costs and higher access costs compared to the hot tier.
* **Cold tier** - An online tier optimized for storing data that is infrequently accessed or modified. Data in the cold tier should be stored for a minimum of **90** days. The cold tier has lower storage costs and higher access costs compared to the cool tier.
* **Archive tier** - An offline tier optimized for storing data that is rarely accessed, and that has flexible latency requirements, on the order of hours. Data in the archive tier should be stored for a minimum of 180 days.

**Azure Queue Storage** is a service for storing large numbers of messages. You access messages from anywhere in the world via authenticated calls using HTTP or HTTPS. A queue message can be up to 64 kb in size.A queue may contain millions of messages,up to the total capacity limit of a storage account.

**Azure Storage vs Cosmos DB**

Data model

Scalability

Querying Capability

Storage options

Consistency

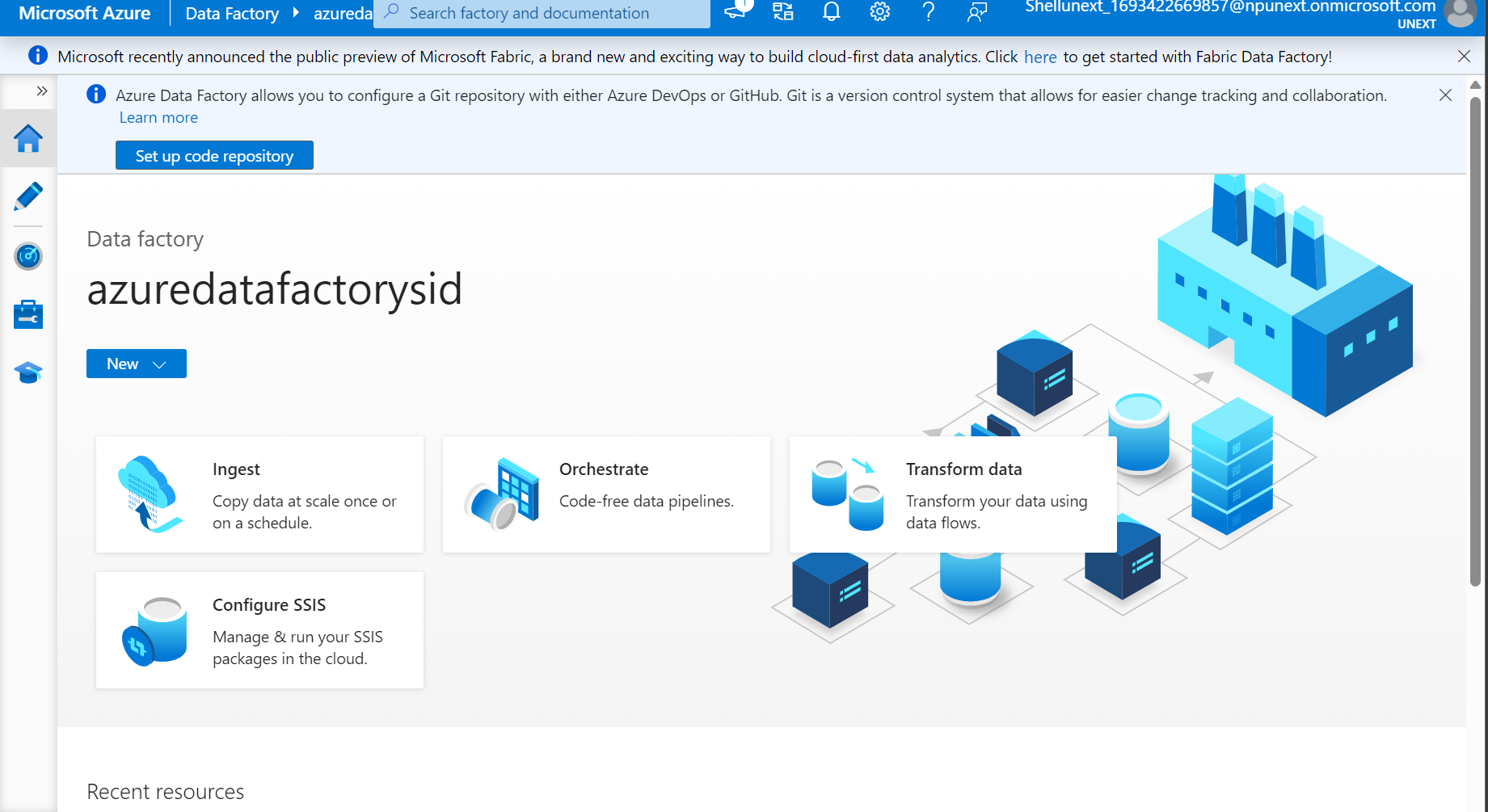
9/5/2023

**Azure Active Directory Integration**

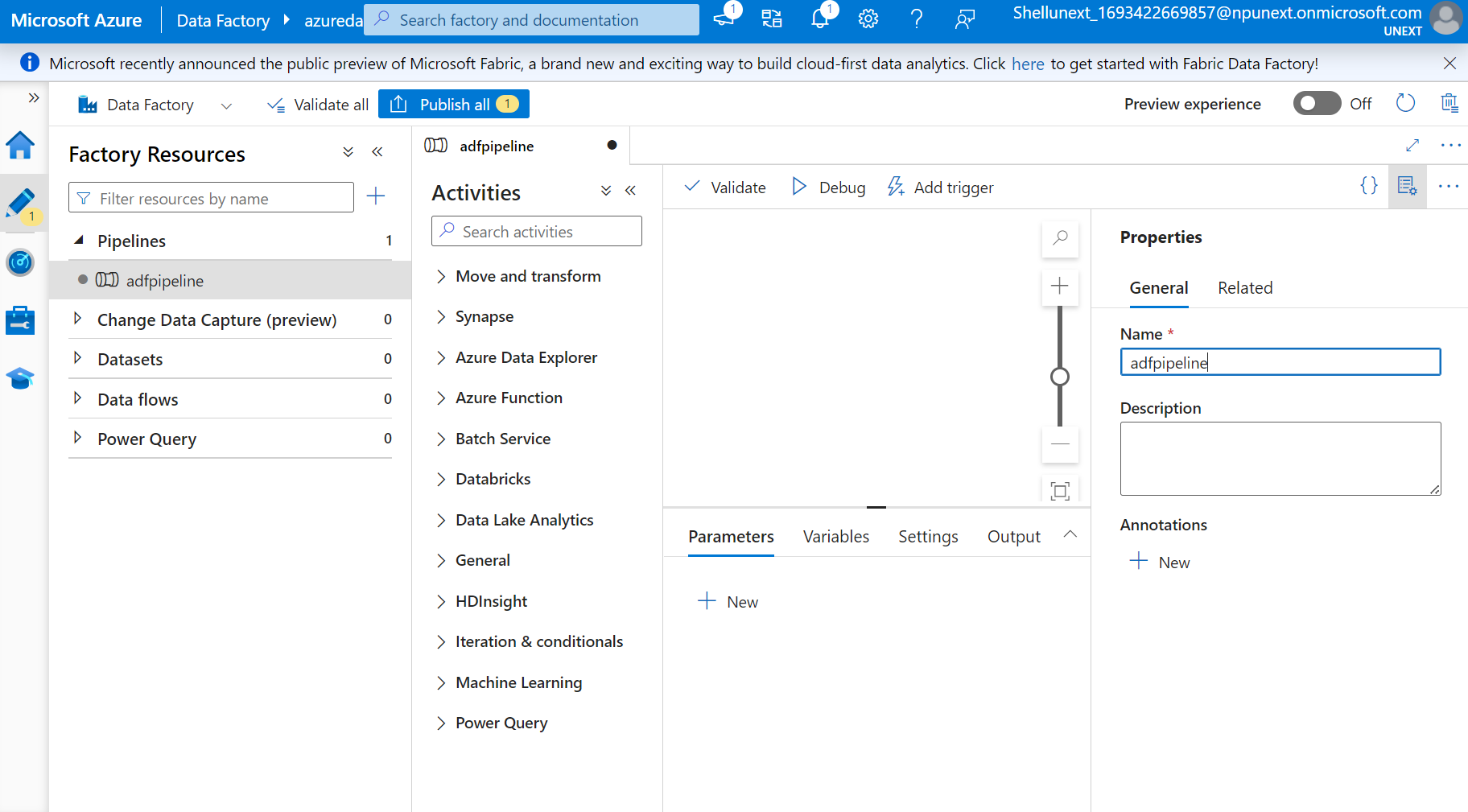
Allows you to use Azure AD identifies for authentication and authorization.

**Shared Access Signature (SAS)**

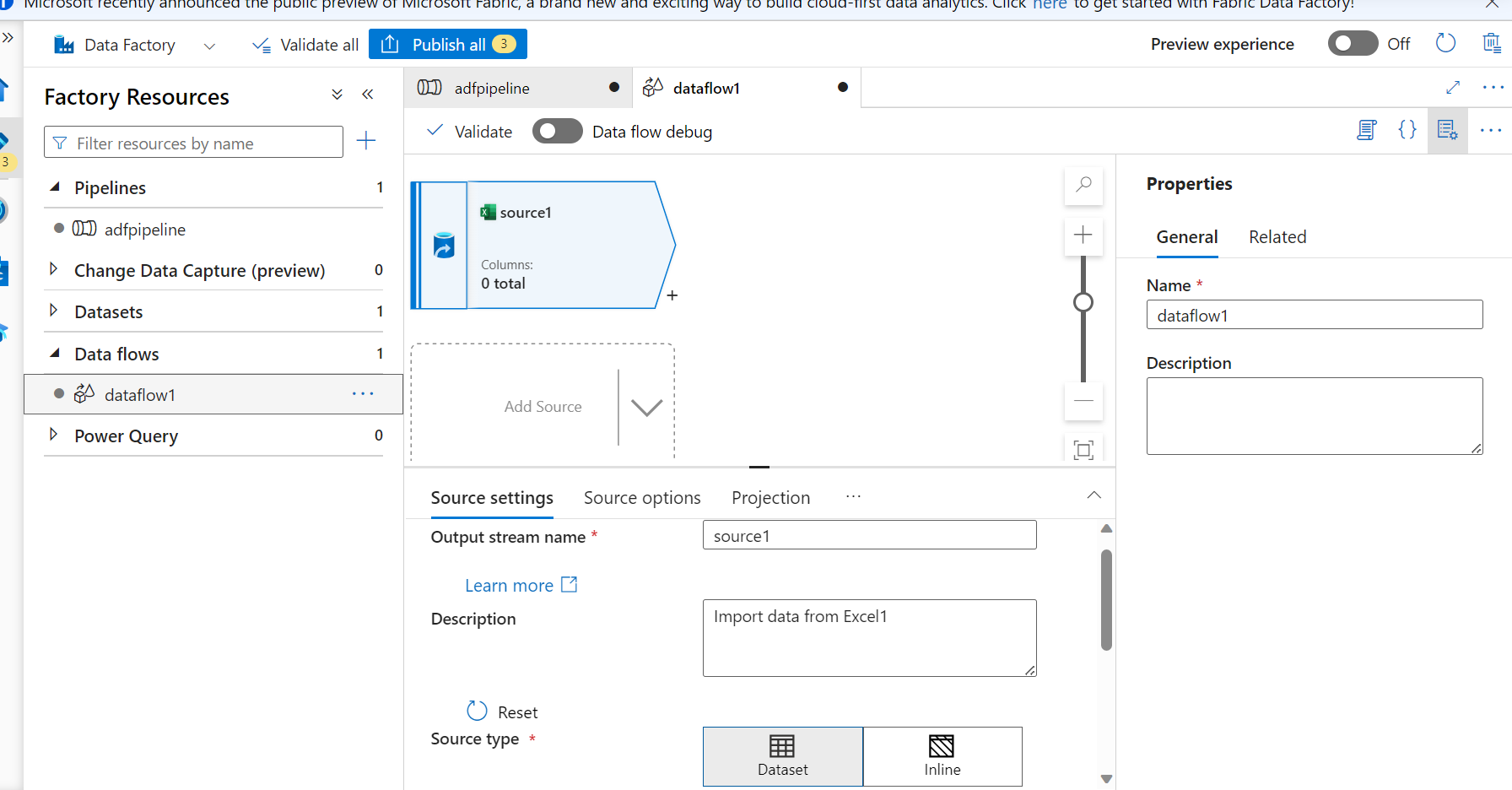
**Azure Data factory**

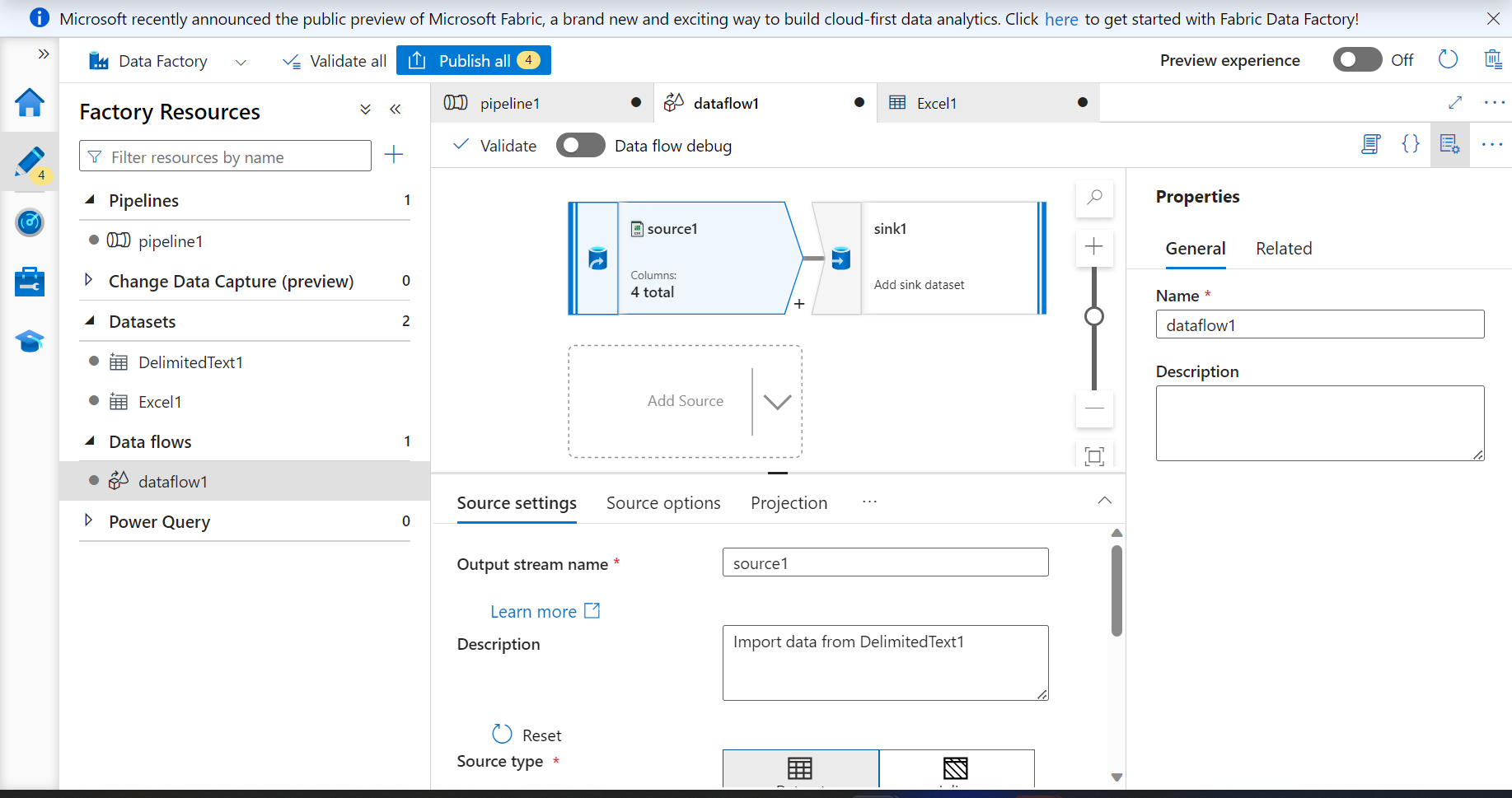
****

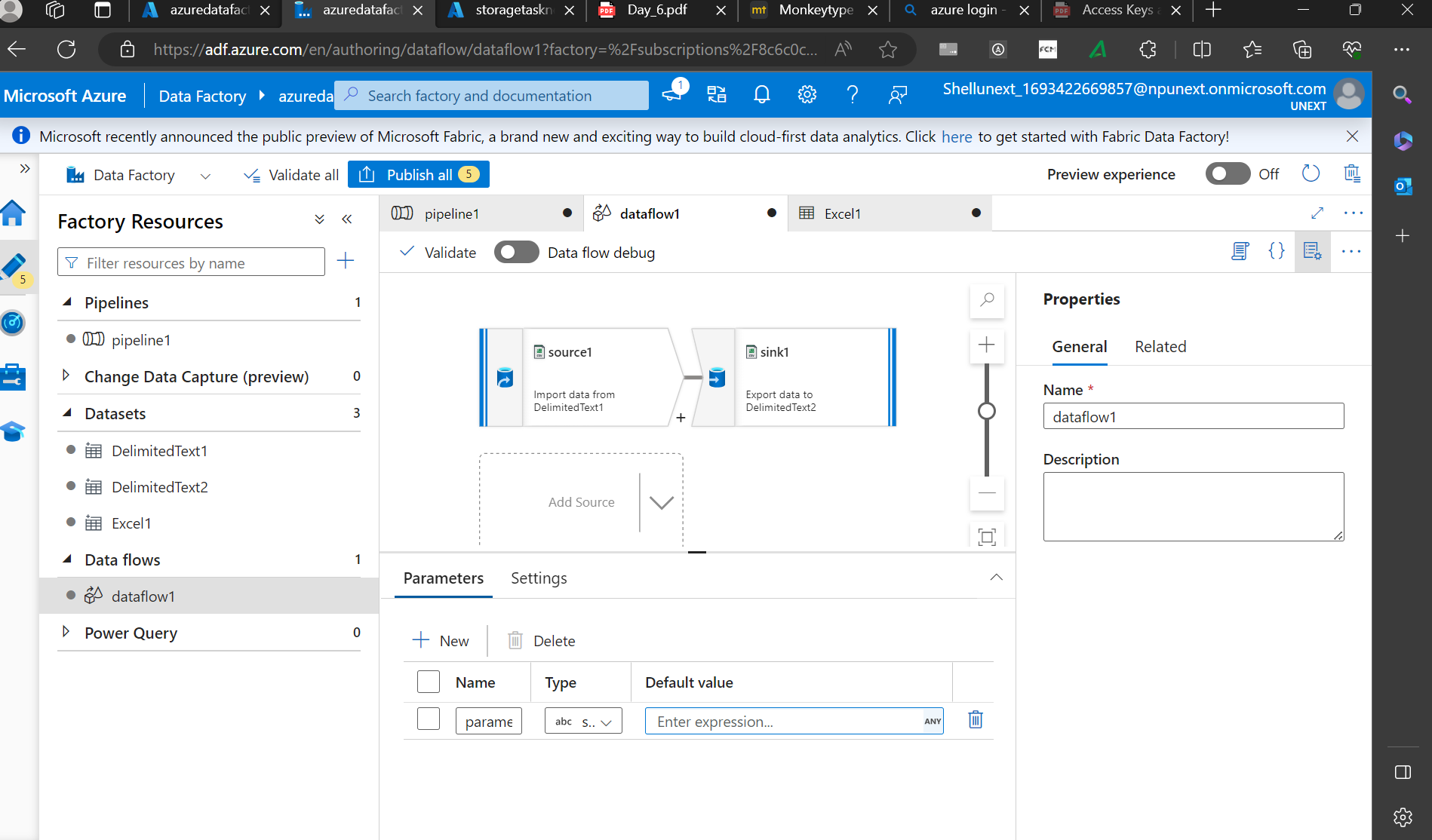
**Pipeline**

****

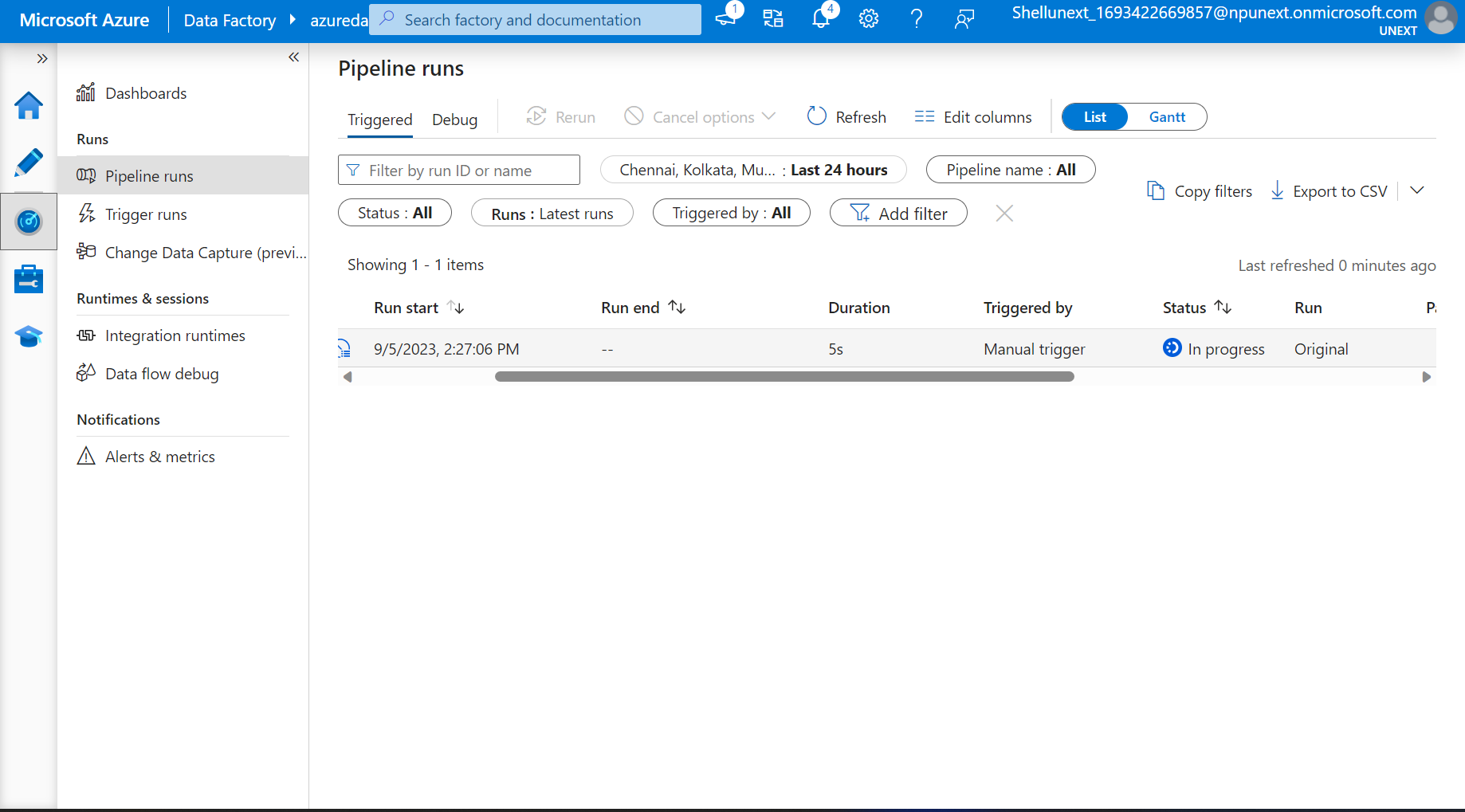
**Data Flow**

****

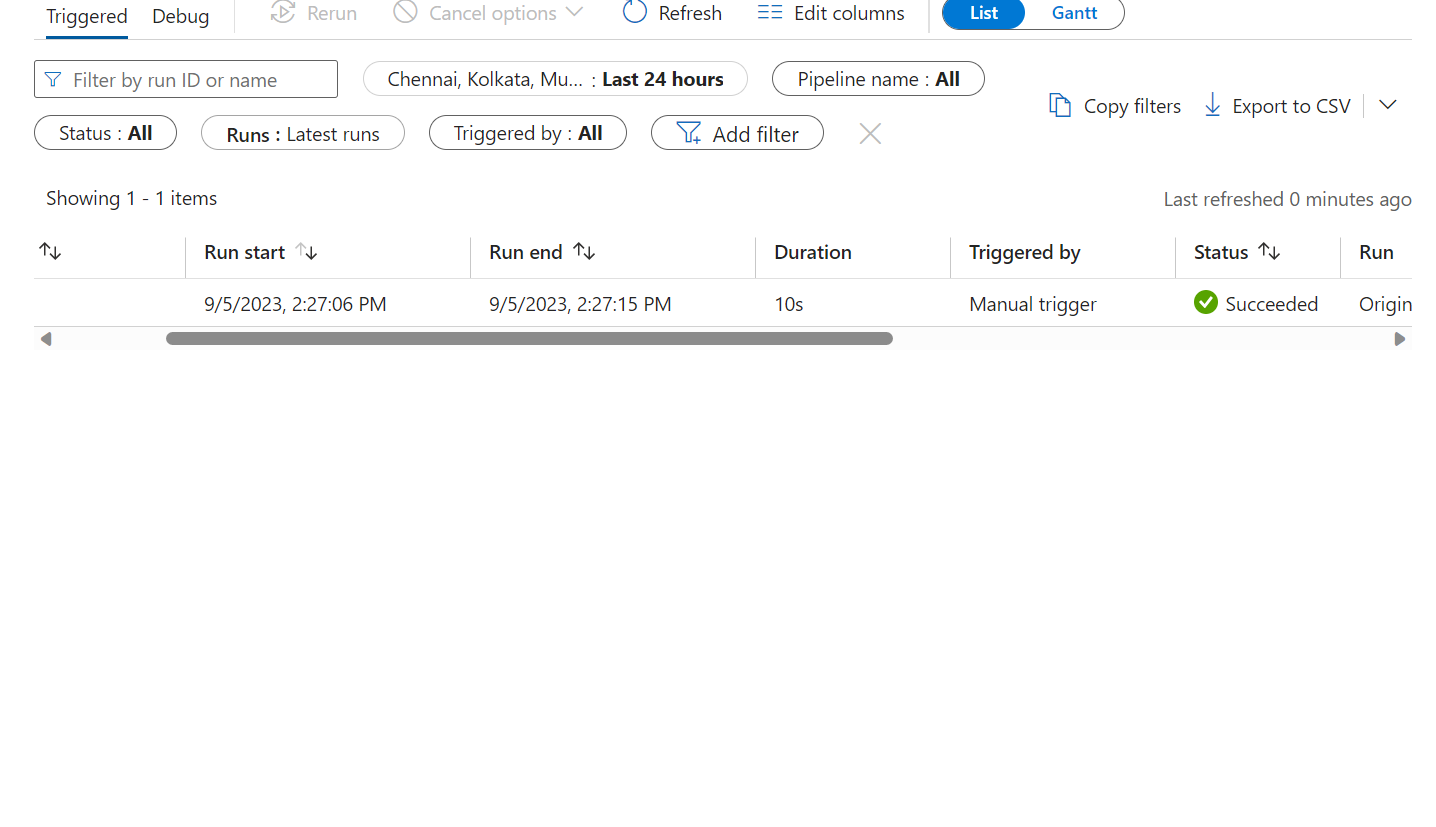




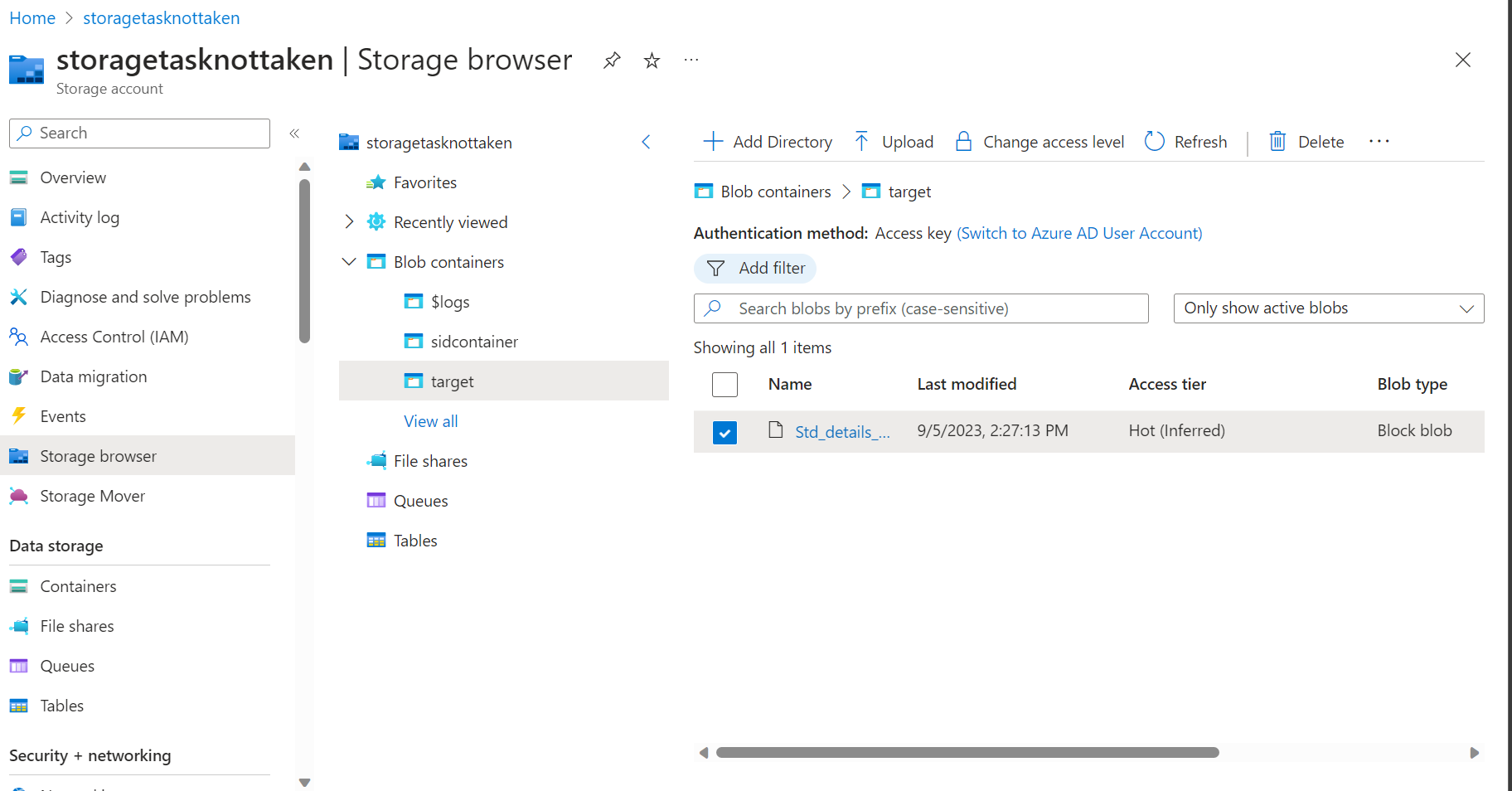
**Running Pipeline**



Pipeline Trigger Successful



Data Copied to Target

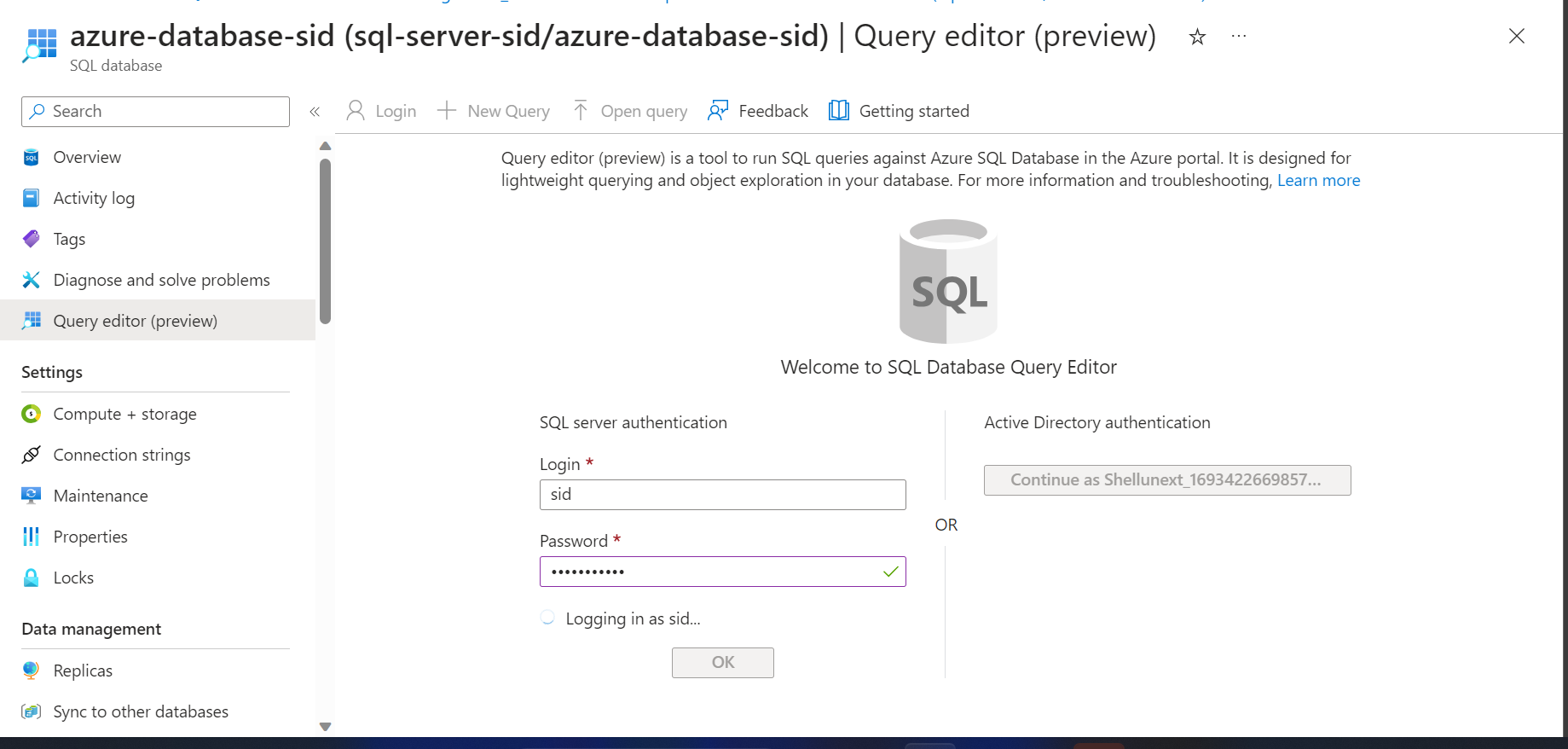


9/6/2023

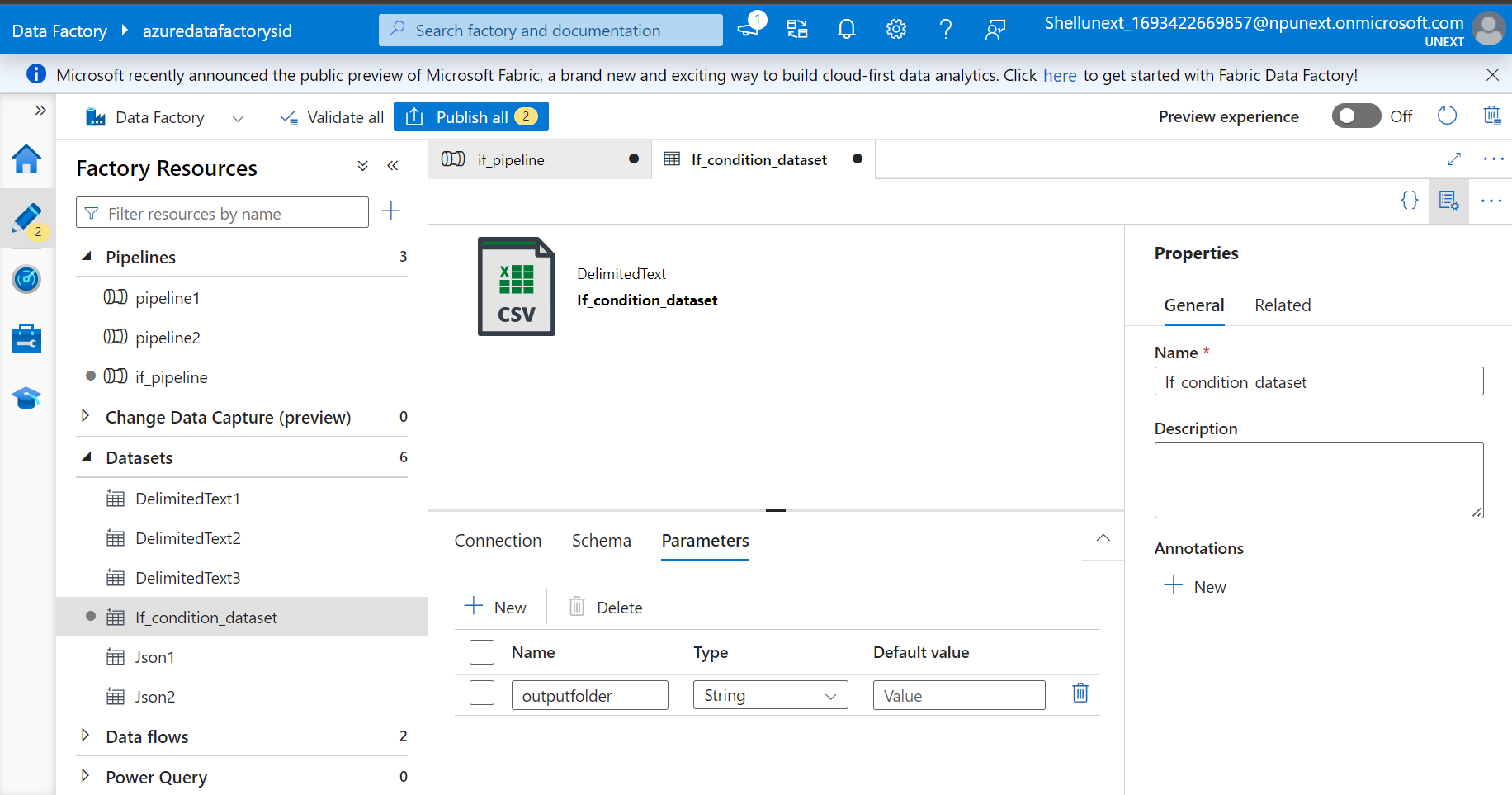
**Create Sql Server**



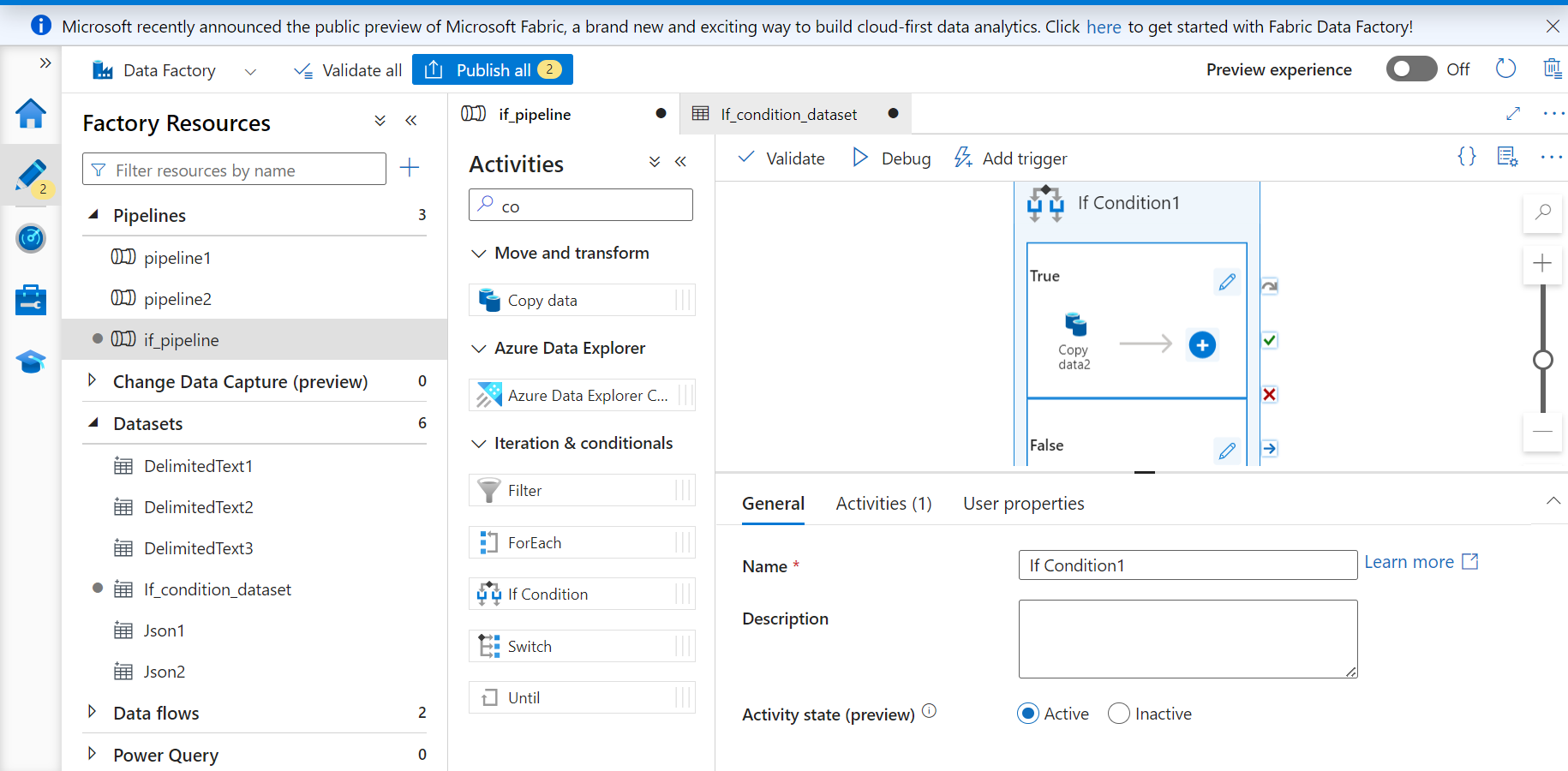
**Create Azure Database**



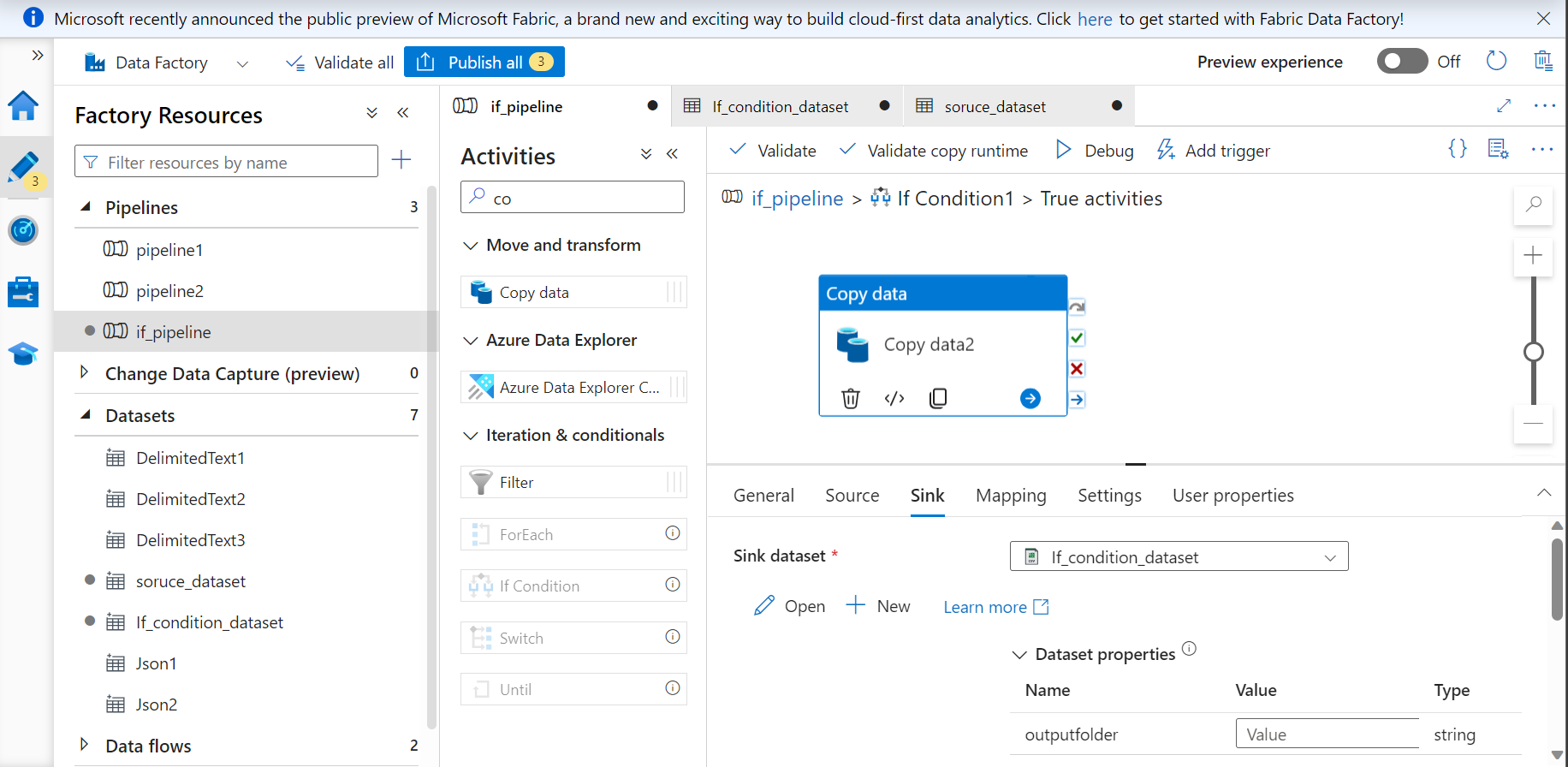
If\_condition\_dataset creation



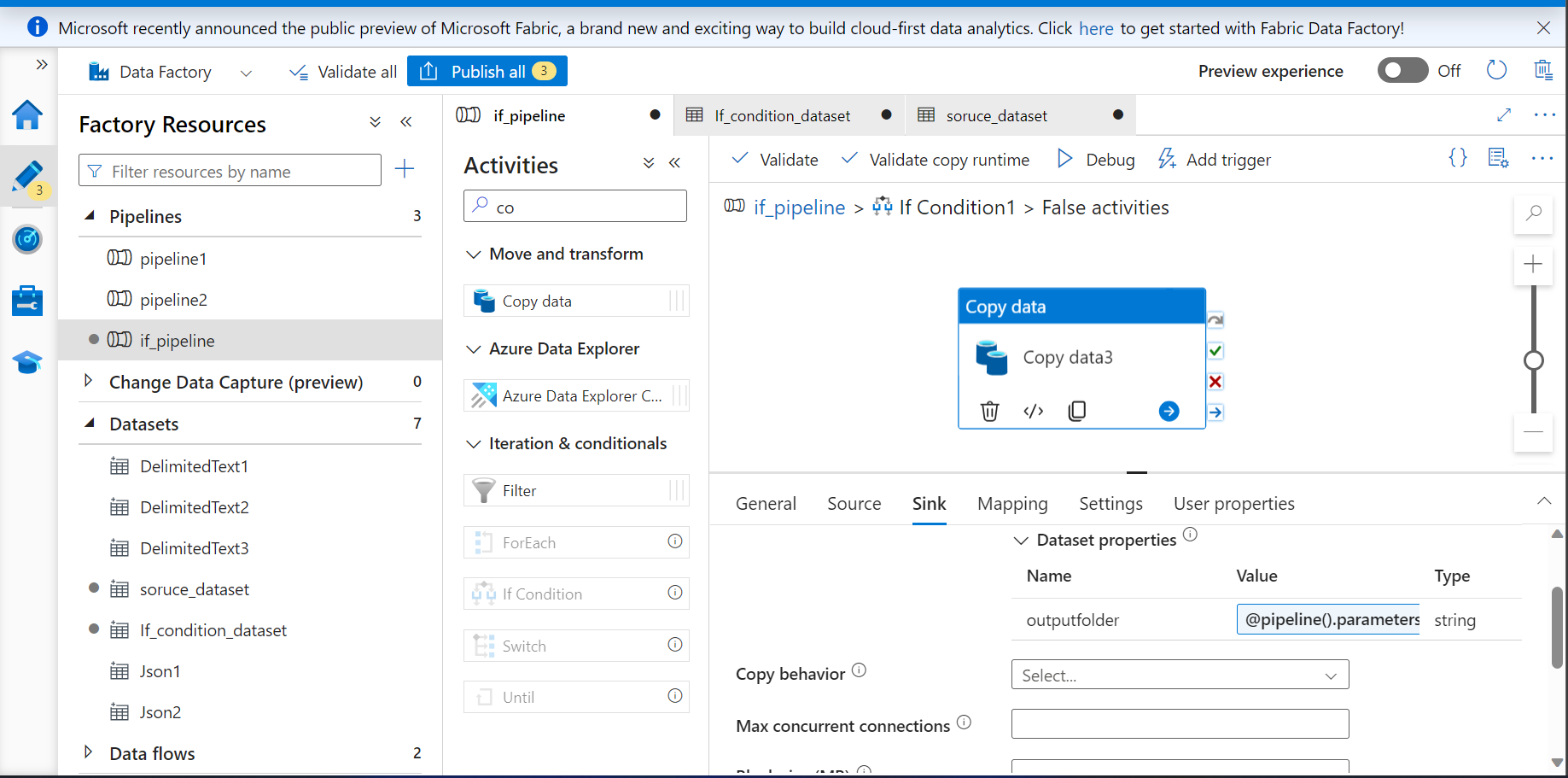
If\_pipeline creation

****

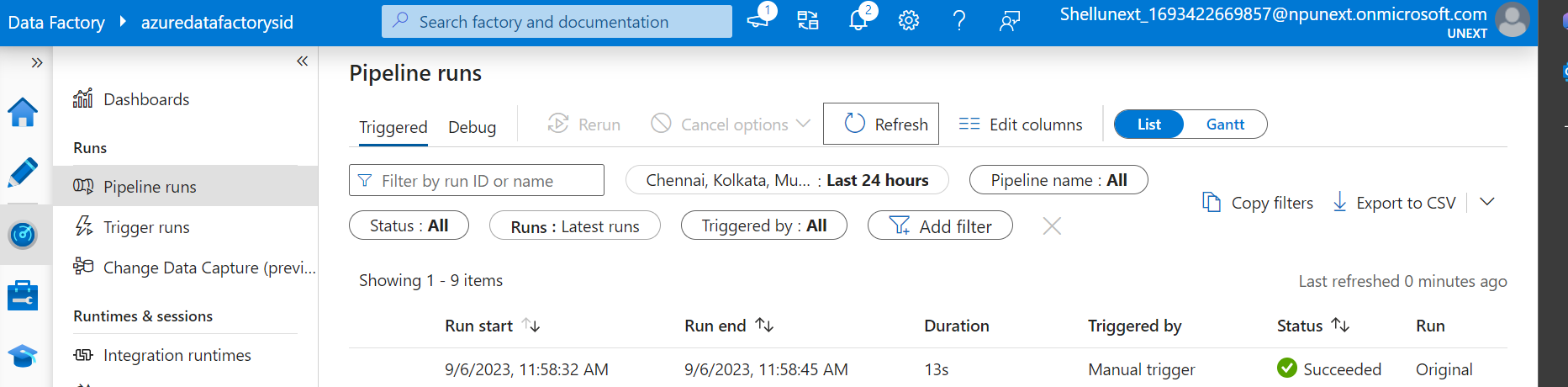
**Copy data Adding Sink and Source For True Condition**

****

**False Condition**

****

**Pipeline Triggered**

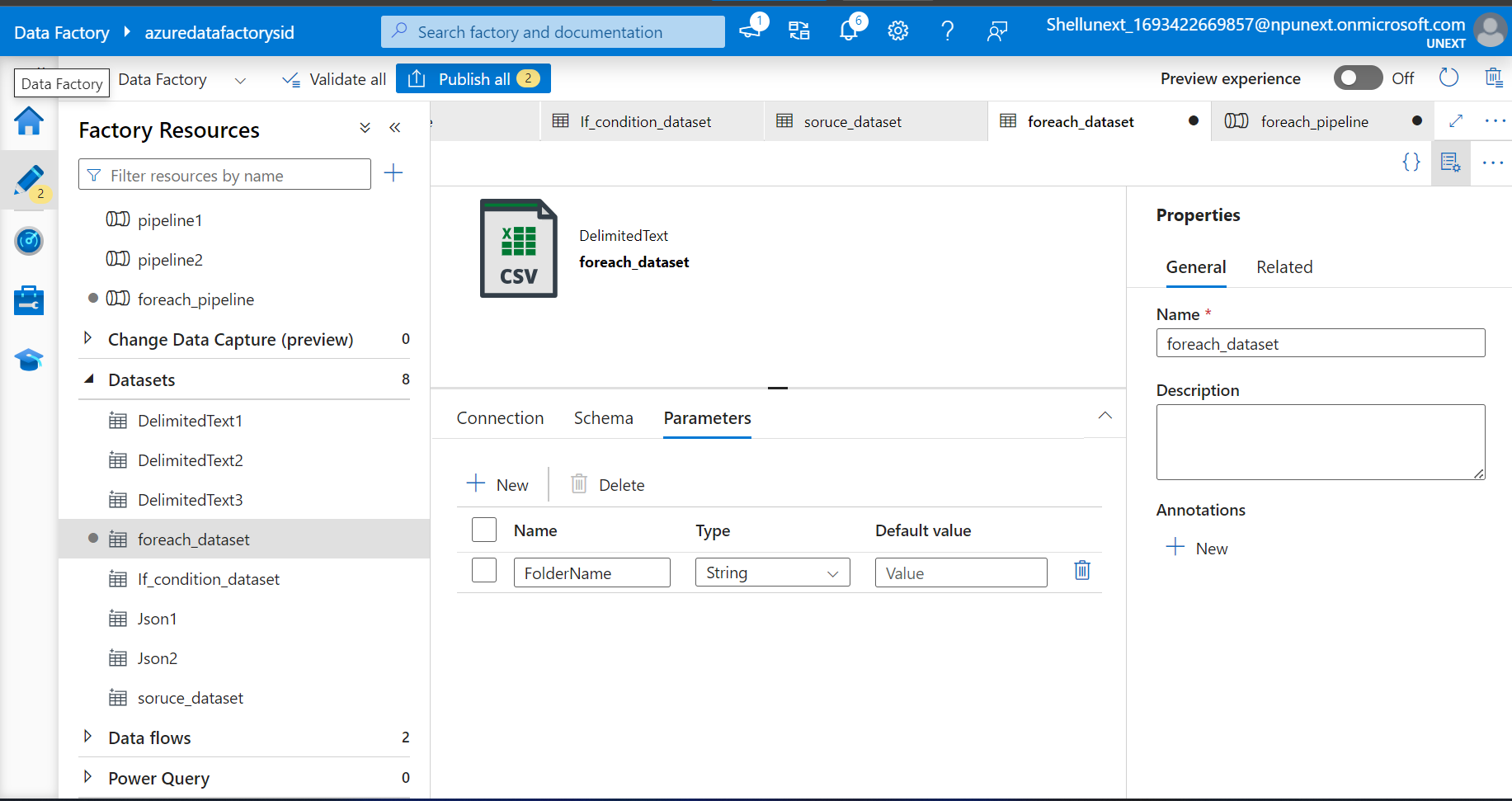
****

**Output 1 folder created because condition was true**

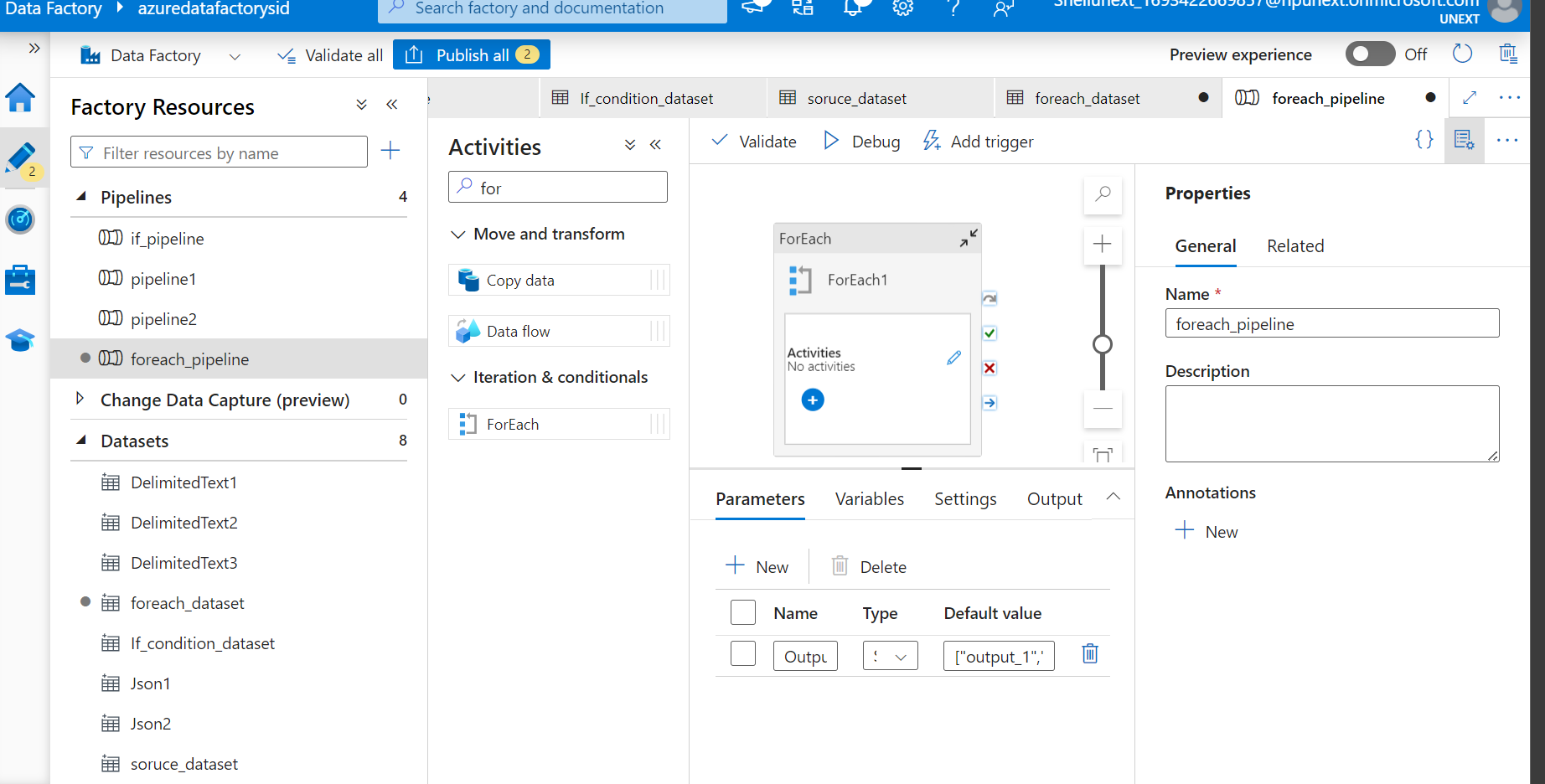
****

**Foreach**

**Dataset Creation**

****

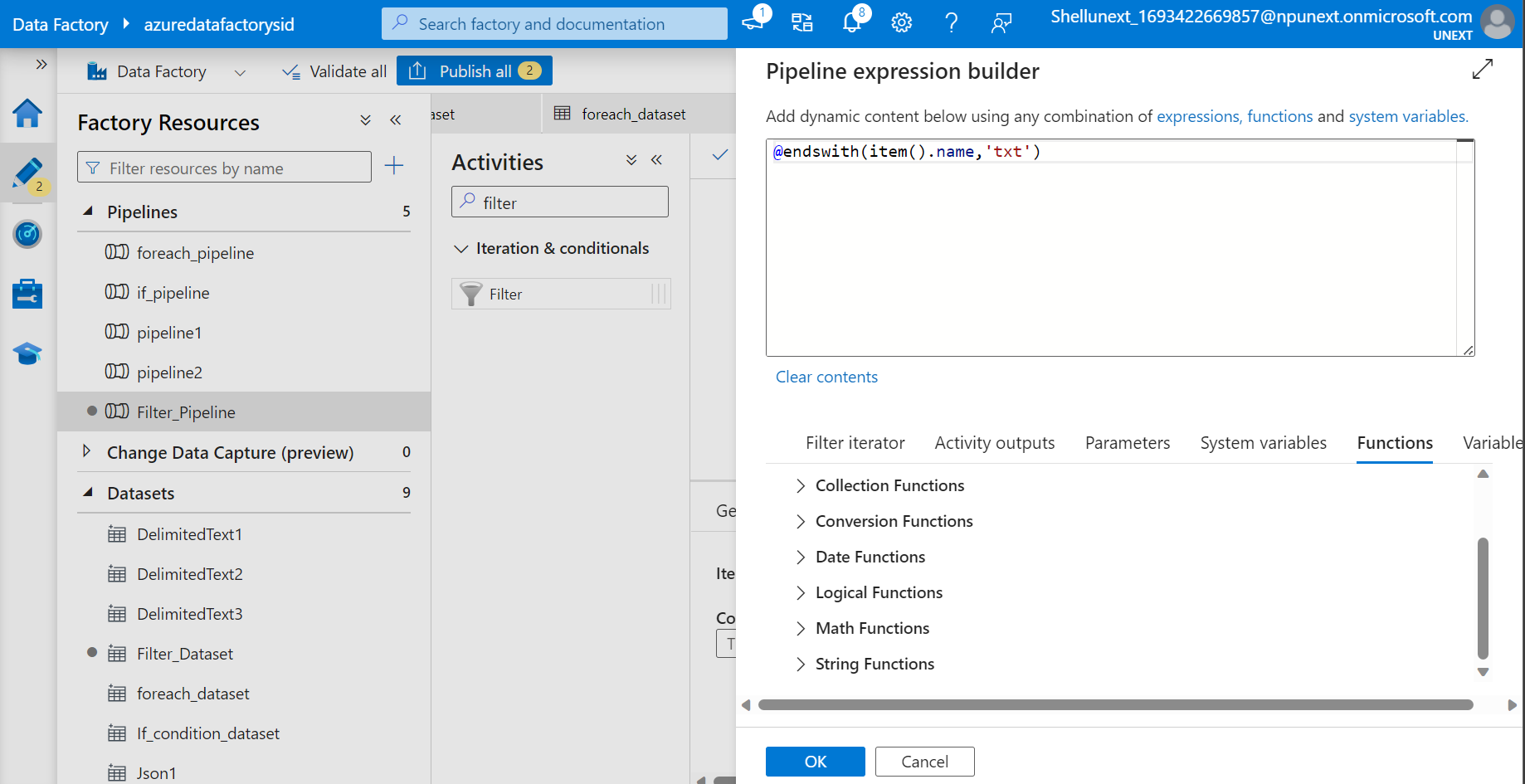
**Pipeline Creation**

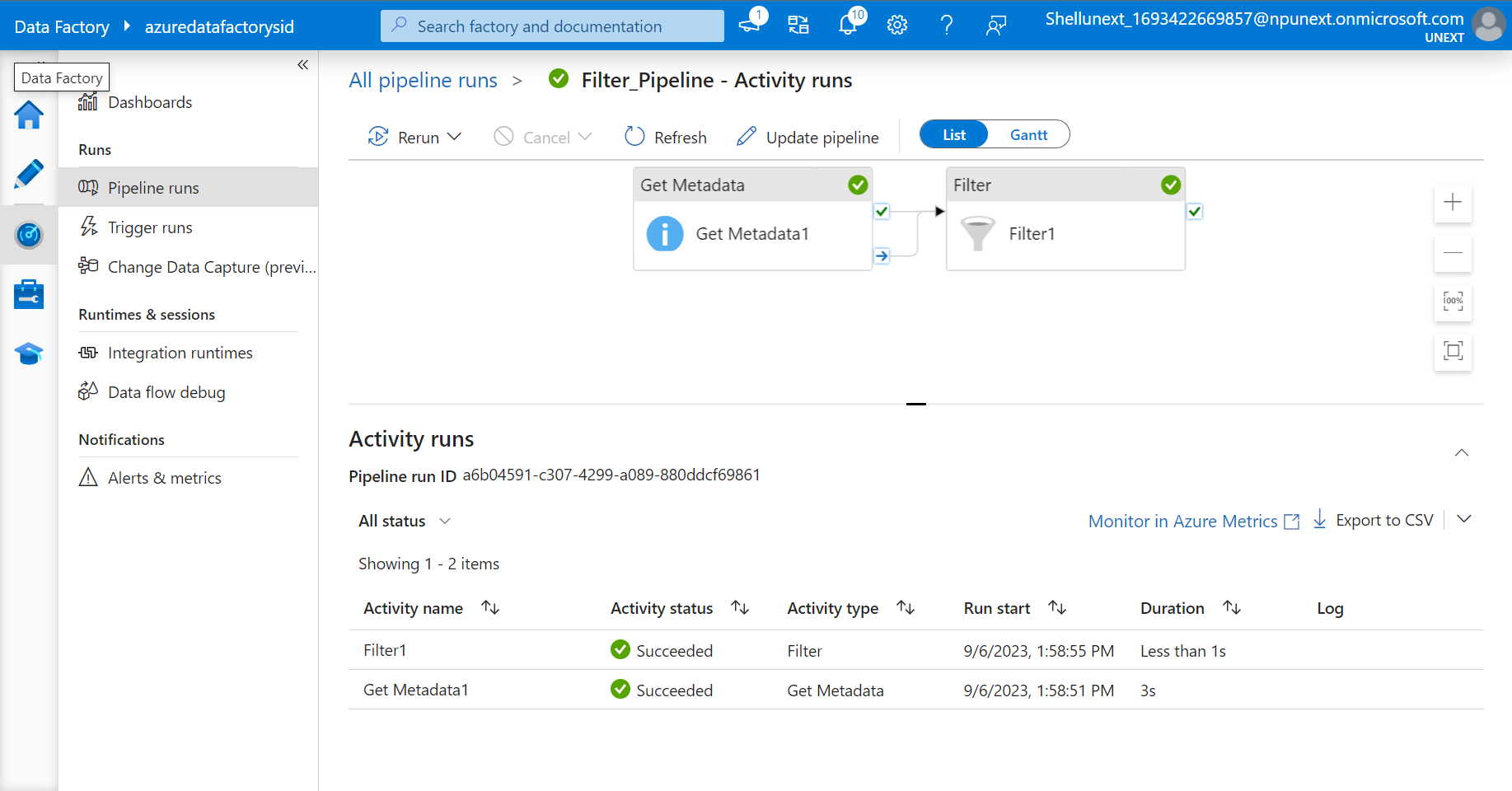
****

**Data Flow Creation**

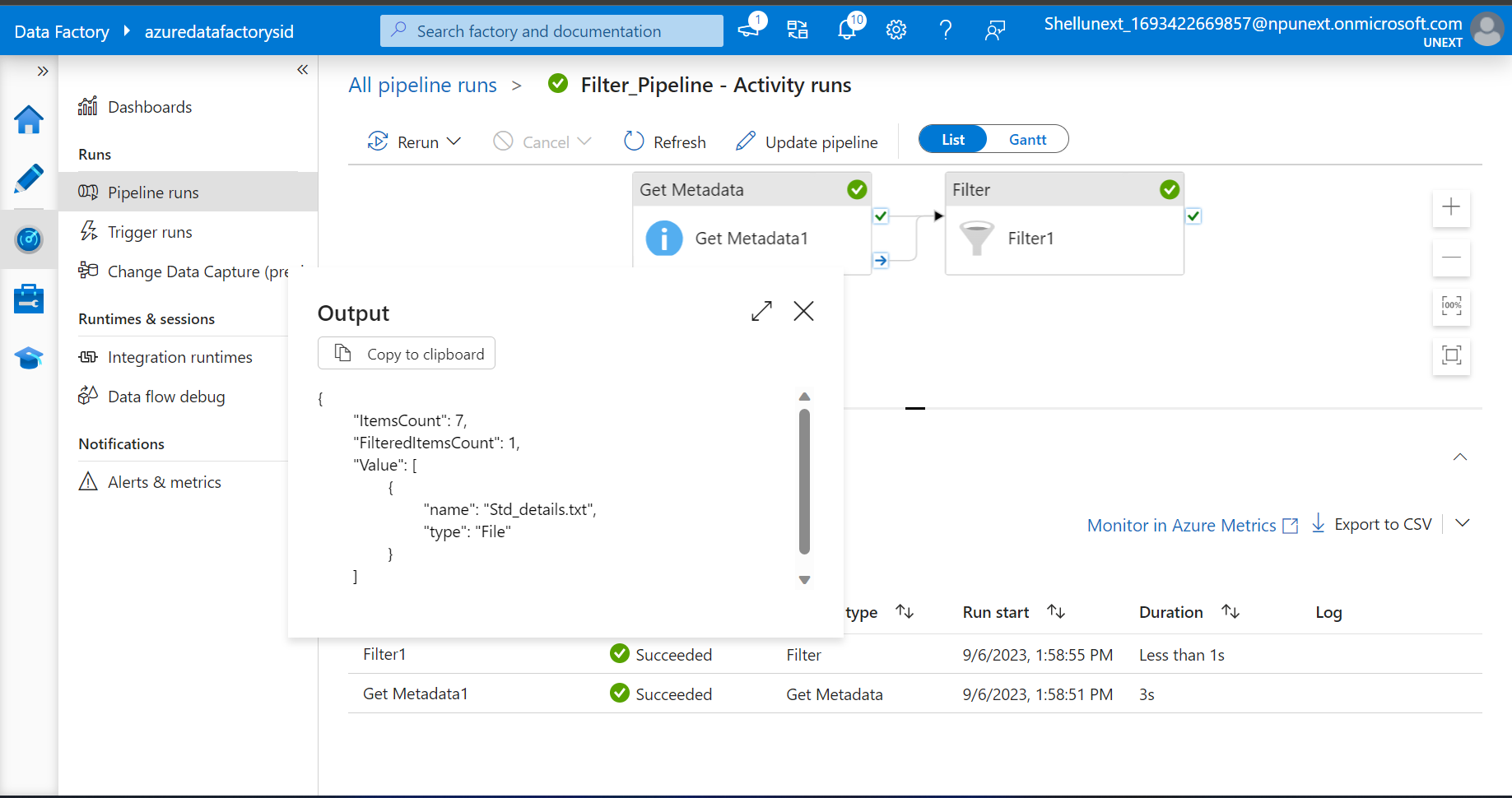
****

**Filter Condition**

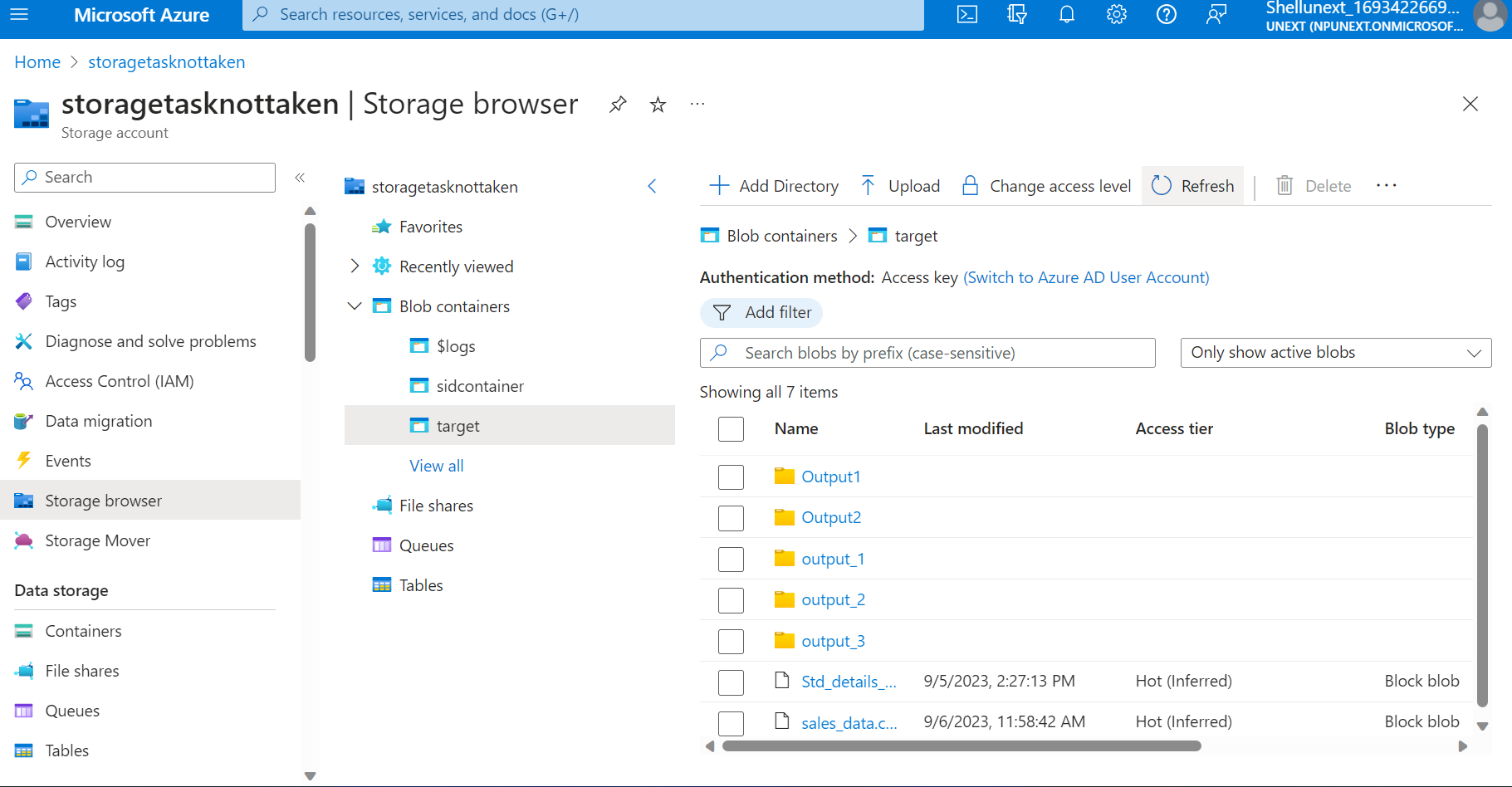
****

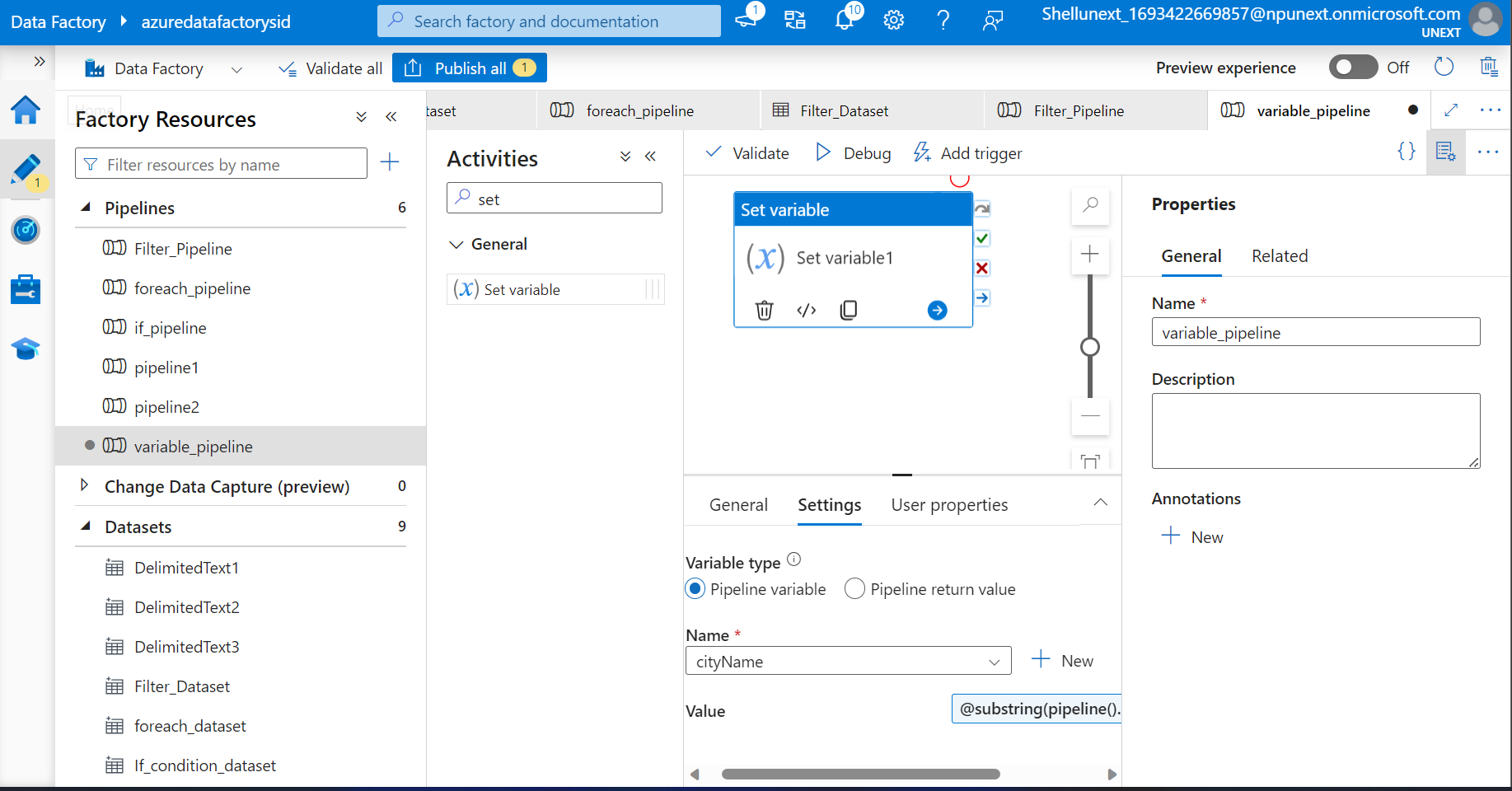
****

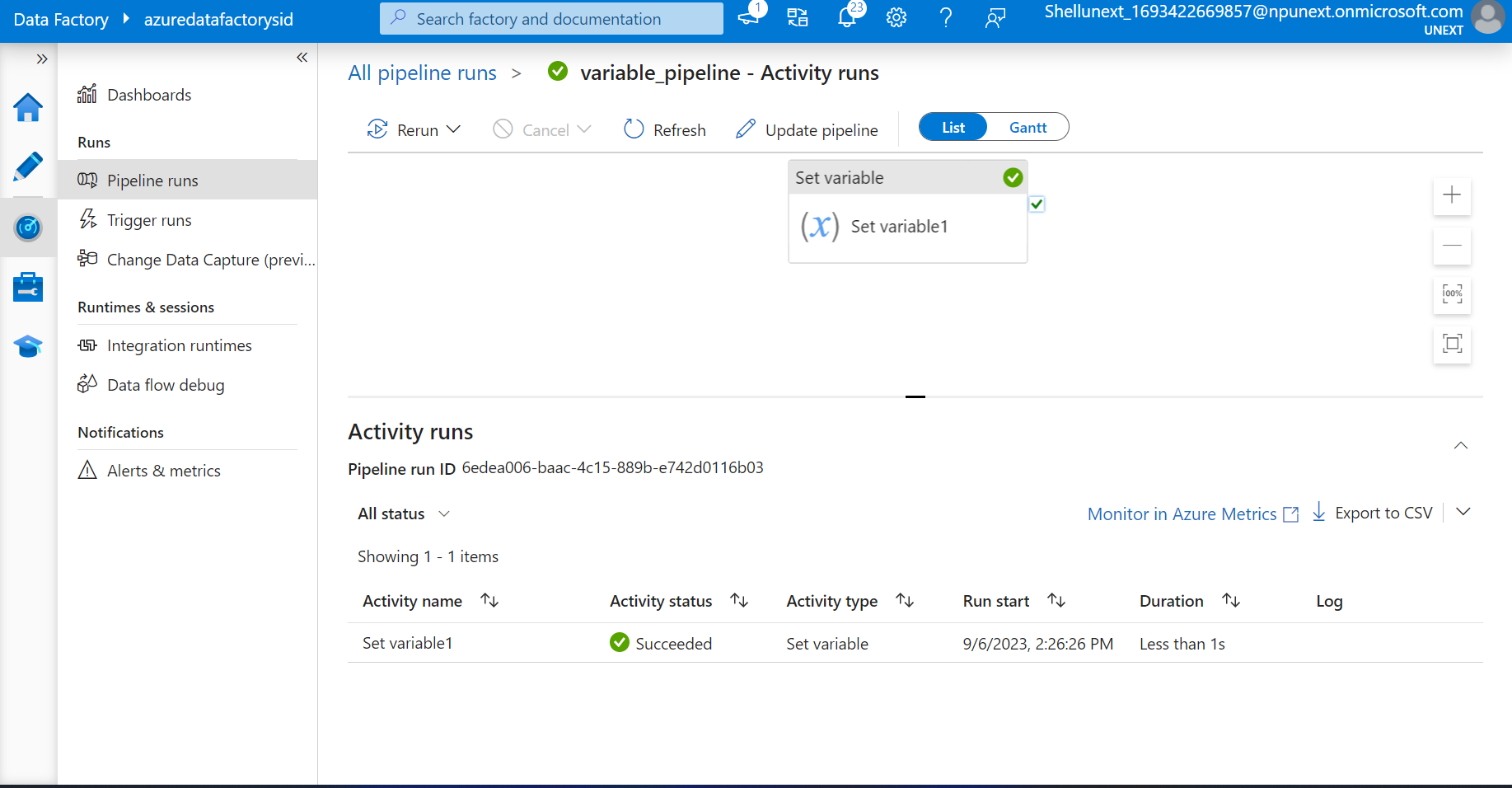
**Only Text Files given as output**

****

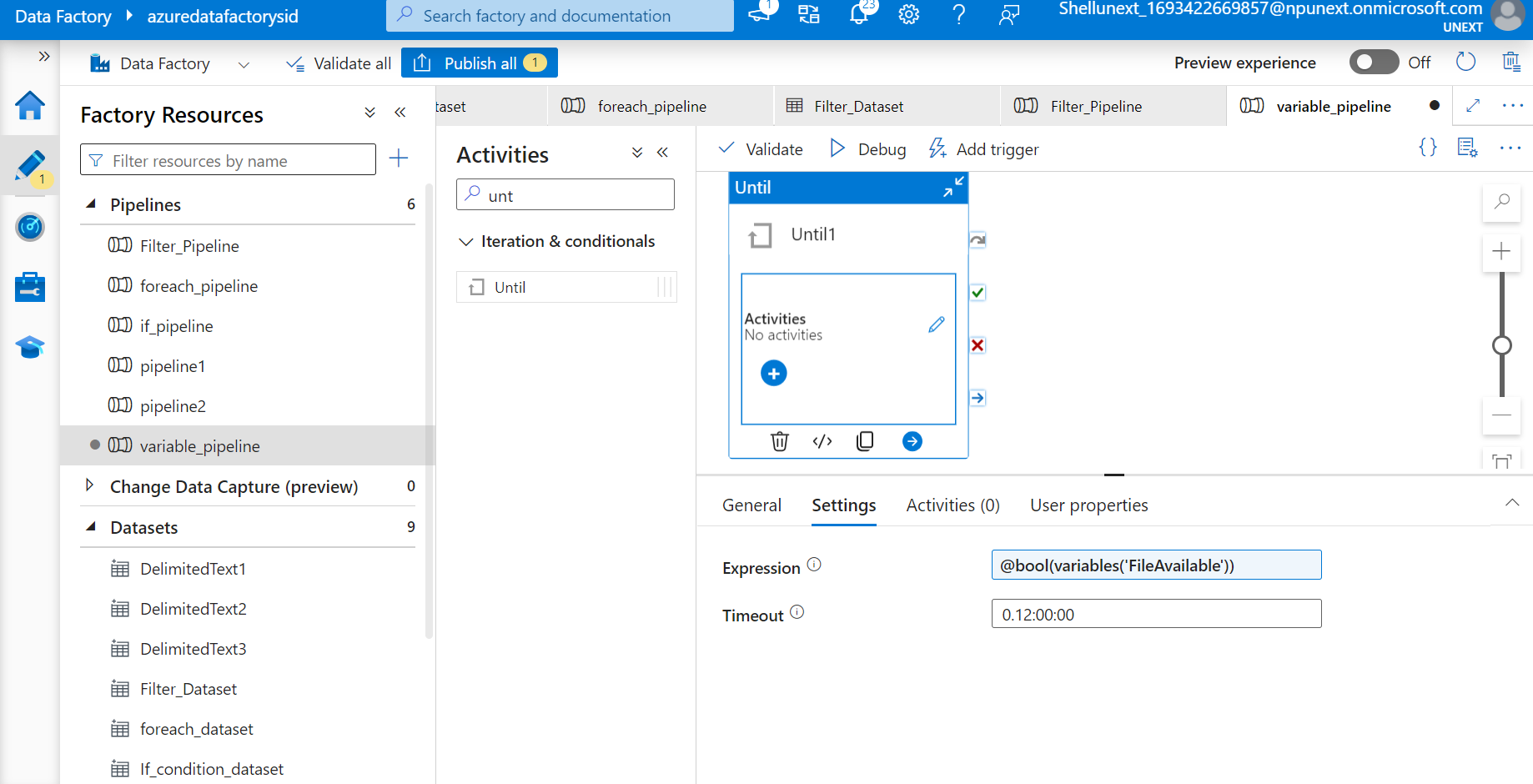
**3 folders created after triggering in target container**

****

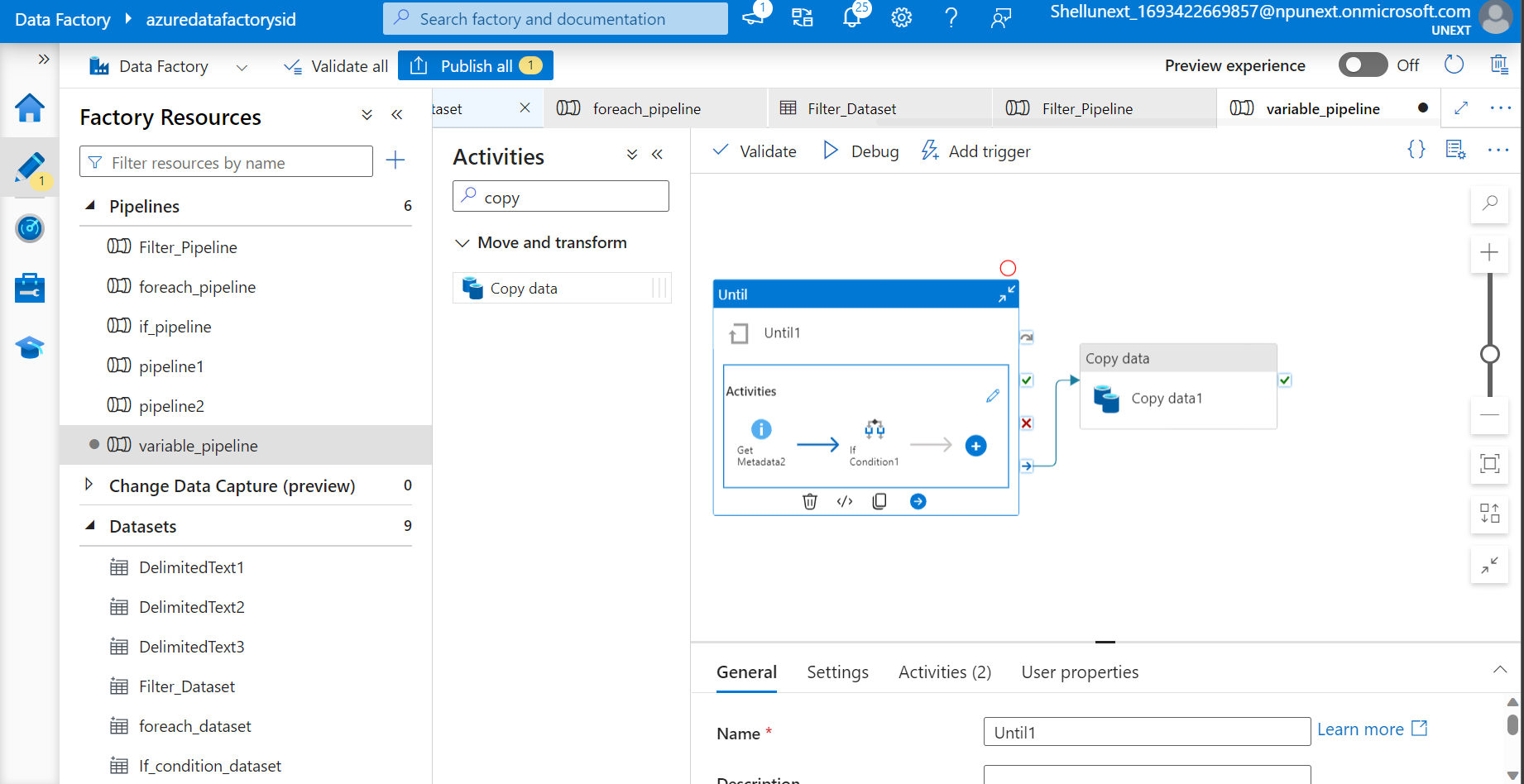
****

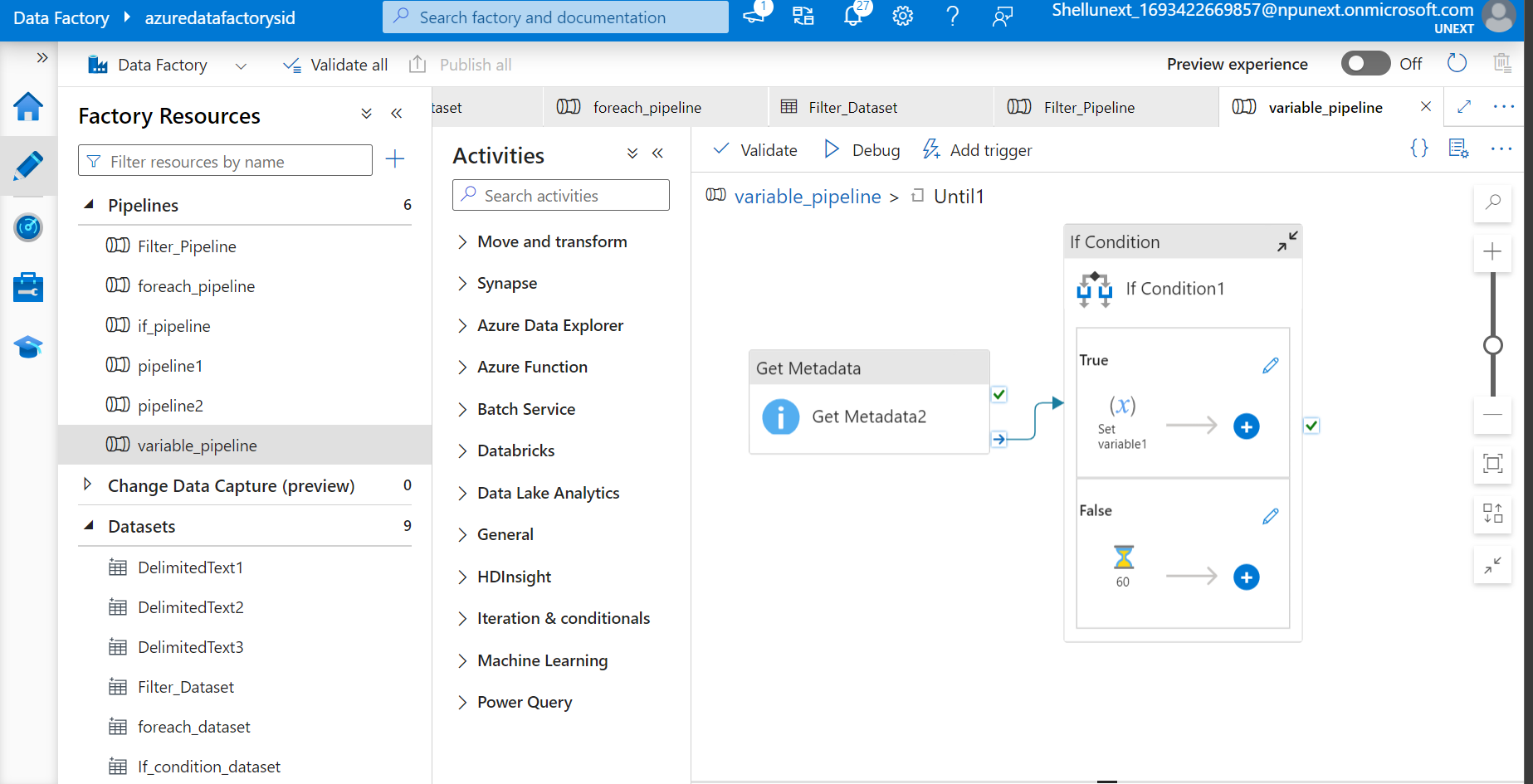
****

****

****

**Created pipeline to check if file is available then data is transferred from source to sink**

****

****

****