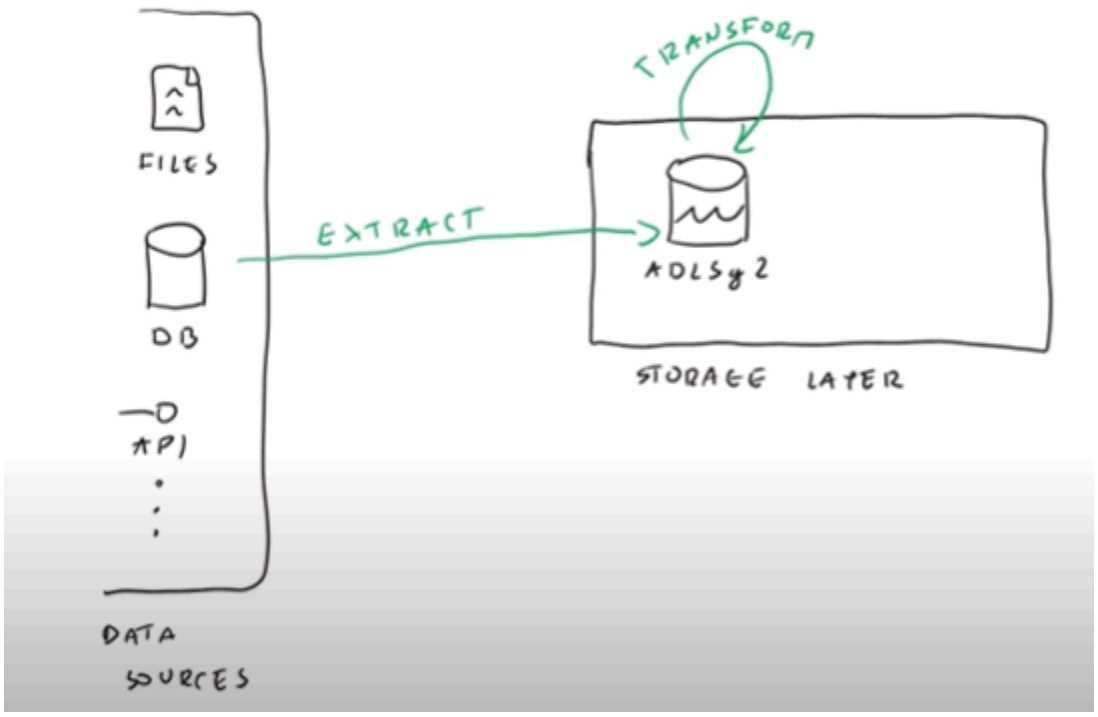
DP203: 09 - Data Lake Structure - Raw Layer

Basic Data flow(BI flow)

1. The basic data flow would be extracting data from multiple sources and storing it in our storage layer
2. inside Storage layer..we'll be having ADLSg2 service which is used to store data

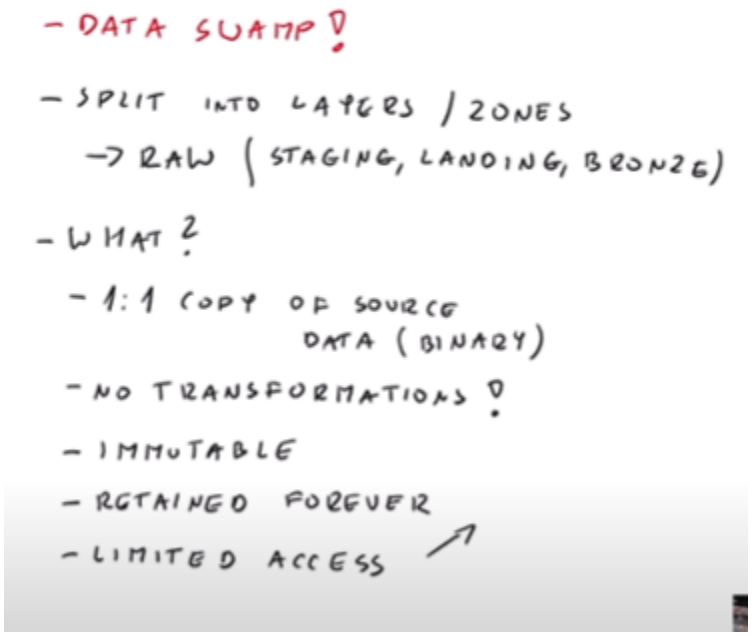


3. AFter getting the data…we can perform any transformation

Data Swamp

1. Data swamp - If we allow everyone to ingest data from different sources to our storage layer…then it would be a complete mess….like there may be duplicate data etc
2. So to avoid it we use data lake zones

Data Lake Zones

1. The solution to data swamp..is to split the layers
2. So Initial layer would be Raw/Staging/Landing/Bronze
3. Lets focus on raw layer today
4. 

5. So raw data will be 1:1 copy of source data…and here transformation will not be done
6. This raw layer data will acts as a source of truth and also this data is immutable
7. This raw layer data will be retained forever and it must have limited access …so any engineers will not make reports out of this data
8. Only data engineers/scientists will have access to this data
9. So here if the data is retained forever..then we would have to pay the storage costs as well

Why do we need raw layer?

- WHY ?
  - LIMIT IMPACT ON SOURCE SYSTEM
  - SIMPLIFY DEVELOPMENT
  - RERUN THE WHOLE PROCESS
  - BUGS

1. To limit the impact on the source system - While ingesting data we only approach sources systems once in a while and retrieve data and store it in raw layer…which consumes less compute resources
2. It simplifies the development
3. Rerun whole process - If we are storing all the source data in raw layer…then if anything bad happens in the transformations(due to bad logics)…then we can come back to raw layer and start the transformations again from scratch
4. We can also identify bugs

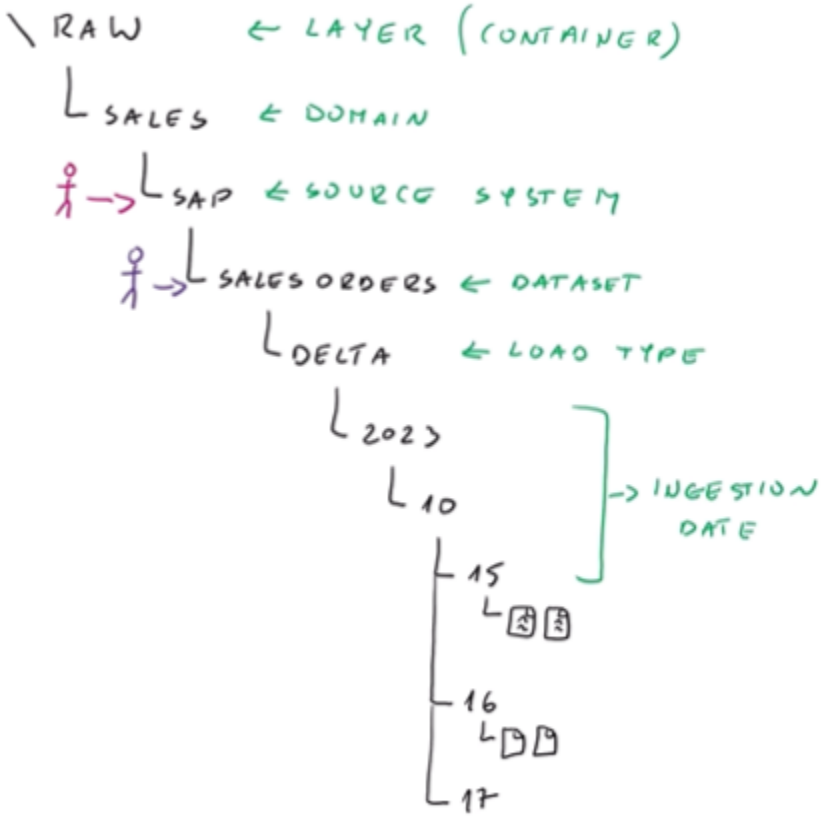
How to implement this?

1. We can create a raw container inside ADLSg2…or we can also use dedicated container to store raw data
2. Using dedicated container for different layers doesn't effect the pricing …as at the EOD we are paying for what we storing

   - HOW ?
     - RAW CONTAINER IN ADLSg2
     - DEDICATED OR SHARED ADLSg2
3. And inside the raw layer..we have to define hierarchical model of arranging source data

- DEFINE HIERARCHY
  - BUSINESS DOMAIN
  - SOURCE SYSTEM
  - DATASET
  - LOAD TYPE
    - FULL
    - DELTA
  - INGESTION DATA

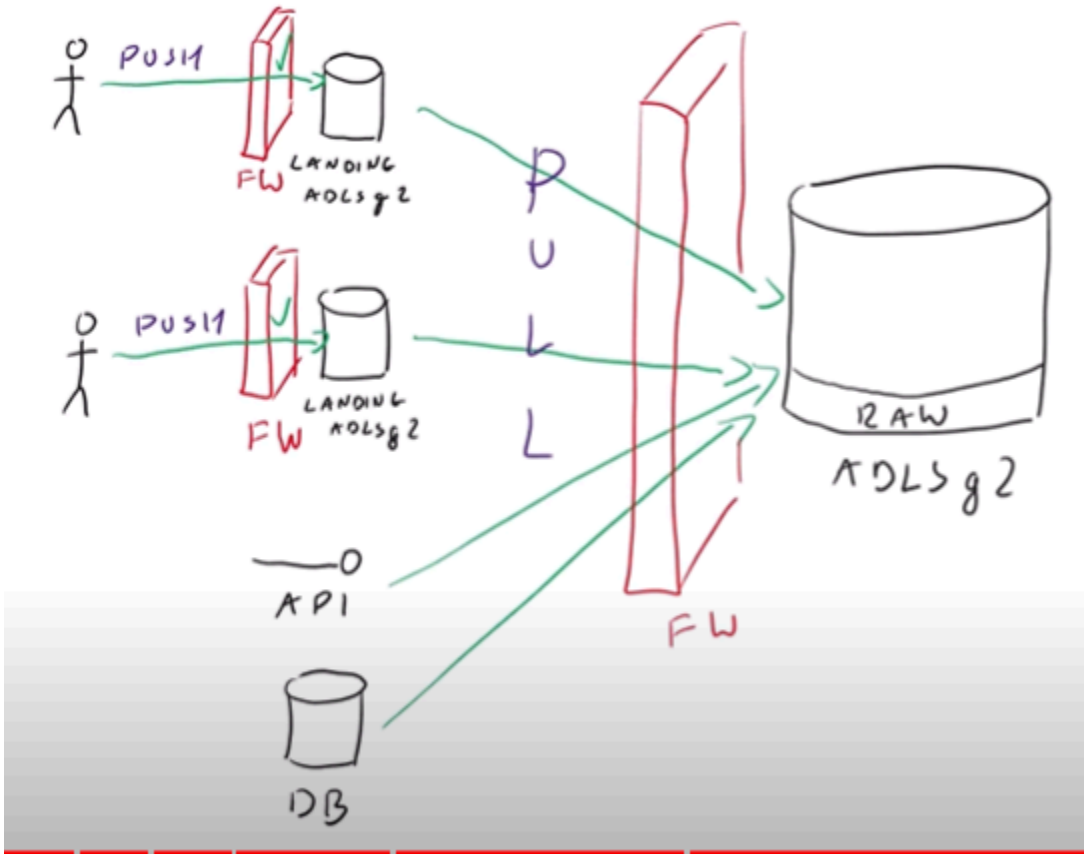
4.



5. This provides us security …like here we can just give certain people to have access to SAP folder
6. And inside the raw container..we store the file in there native format…like if we ingested CSV file…then we store them as CSV file

7. If we retrieving the entire DB ..then we will store it as parquet format.

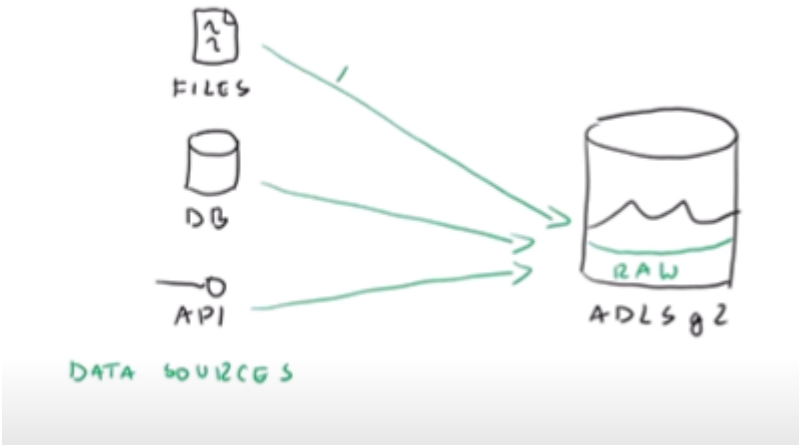
   - NATIVE FORMAT
     - CSV -> CSV
     - XML -> XML
     - DB -> PARQUET / DELTA
8. And also we have to define life cycle management policies..using access tiers to save costs
9. To retrieve PII(personal data) we need to have a consent team and perform ingestion/ or we can anonymize the data using hash values to hide PII data and handle them
10. Security/networking
11. We have two things push and pull



DP203: 10 - Azure Data Factory

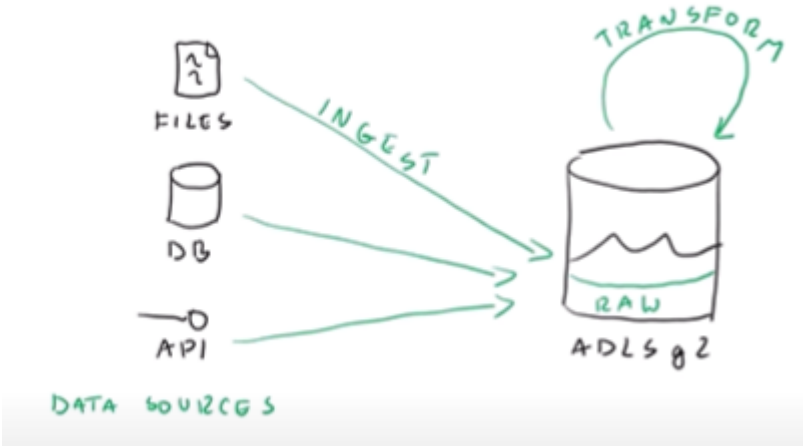
ADF place in BI flow

1. Here our first step would be to ingest all the data from sources and store it in RAW Layer



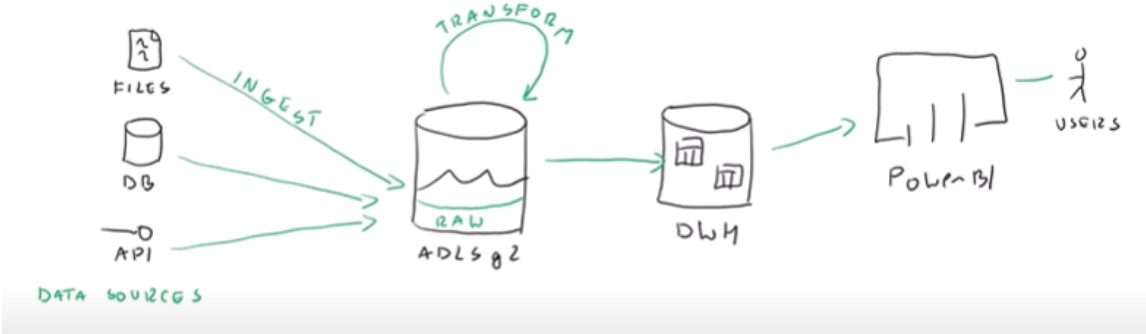
inside ADLSg2

2. Now we'll transform this data present in RAW layer..and store it in another layer inside



ADLSg2

3. After that we'll load this data into datawarehouses…and from there we can generate the powerBI reports(PowerBi gets data from datawarehouse)....and the end user can see reports



4. ADF can connect to many data sources using ADF connectors to get the data
5. It is also used for orchestration(a data flow) …so ADF copies data and orchestrate it

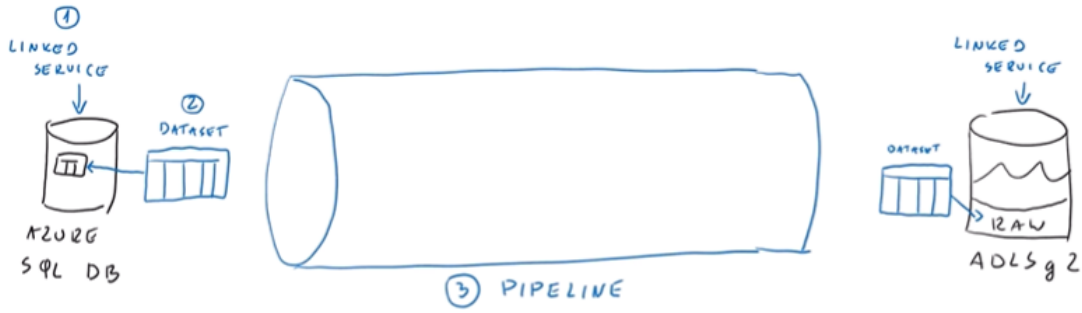
ADF components

1. We have our data(CSV) in azureSQL DB ..this is our data source
2. And our destination is raw layer in ADLSg2

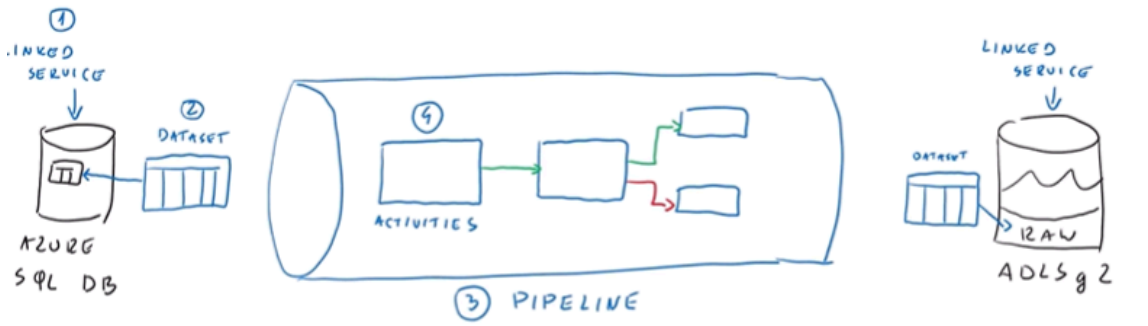
3. So here first thing we need is LinkedService which is used to connect bw source - ADF and ADF - destination

4. Next we will create a dataset for source and destination

5. Now we need a pipeline to connect these two things
6. Inside pipeline we can create activities that we need to transfer migrate the data



7. Inside the pipeline we have created 4 activites

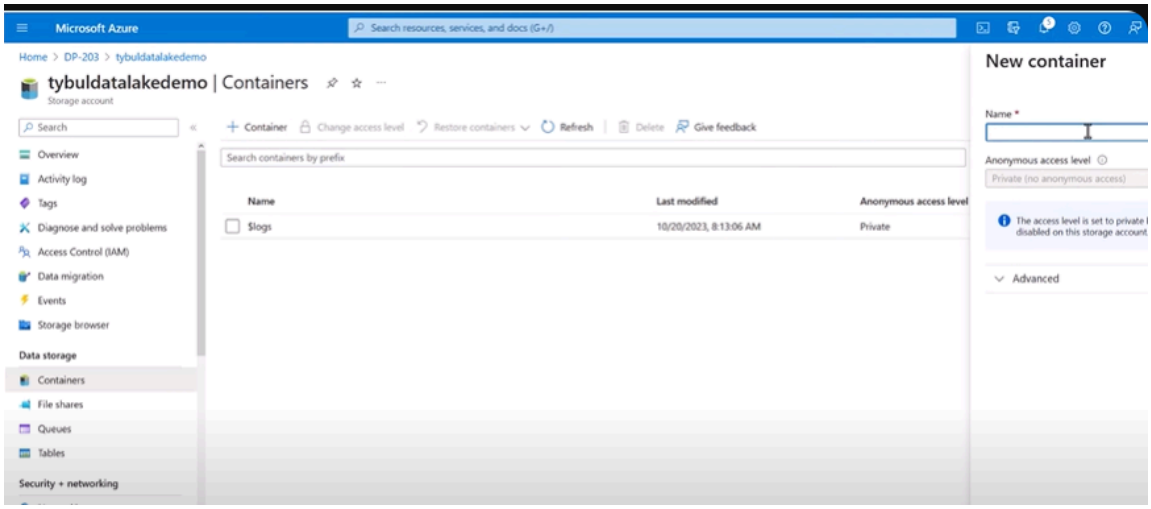


If activity one is passed then its result will be passed to 2nd activity….similarly if activity 2 fails…then it sends output to red activity

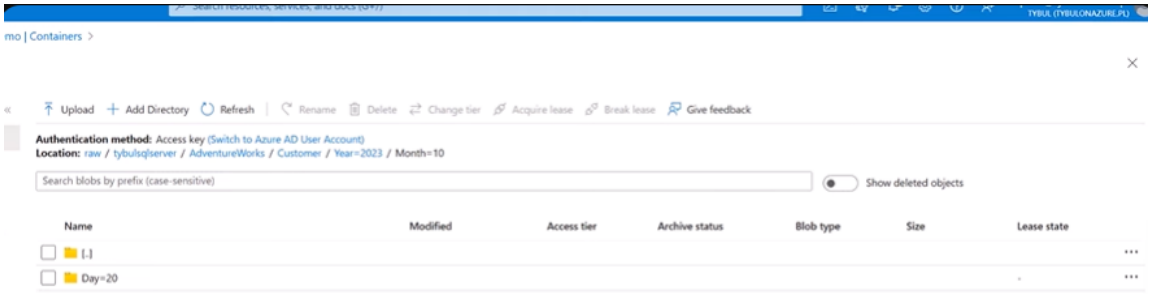
8. We can also perform transform activities in the pipeline
9. Here in our example we just need copy activity which copies the data from source and sinks in destination

Practical

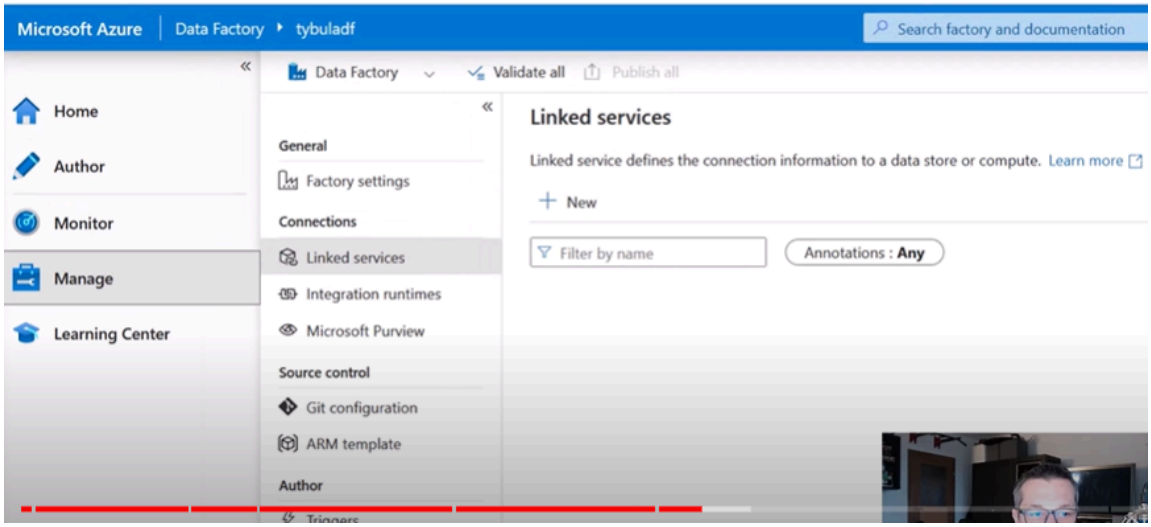
1. Here we create AzureSQL server DB with sample data and azure storage account for ADLSg2
2. Inside ADLSg2 we will create a row layer using “create container” inside row layer



3. Here we have created hierarchy inside raw container

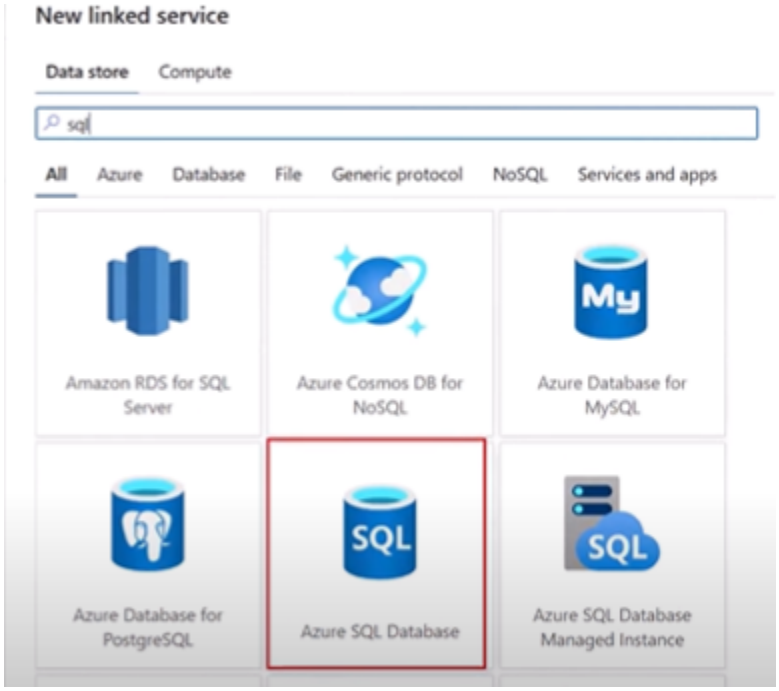


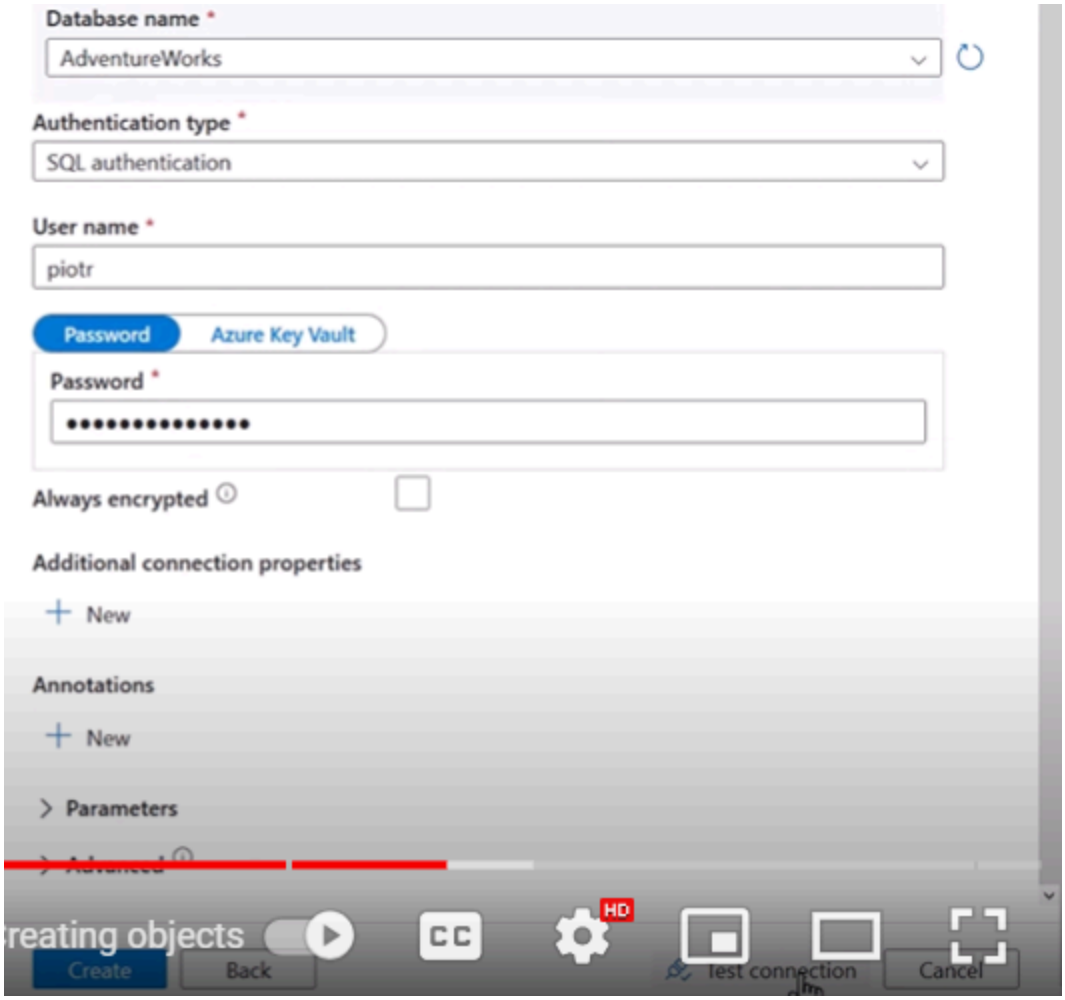
4. Next we create a new ADF…and launch the adf
5. First we have to create Linked Services



to create that we go to manage tab

6. And we create AzureSQL linked service

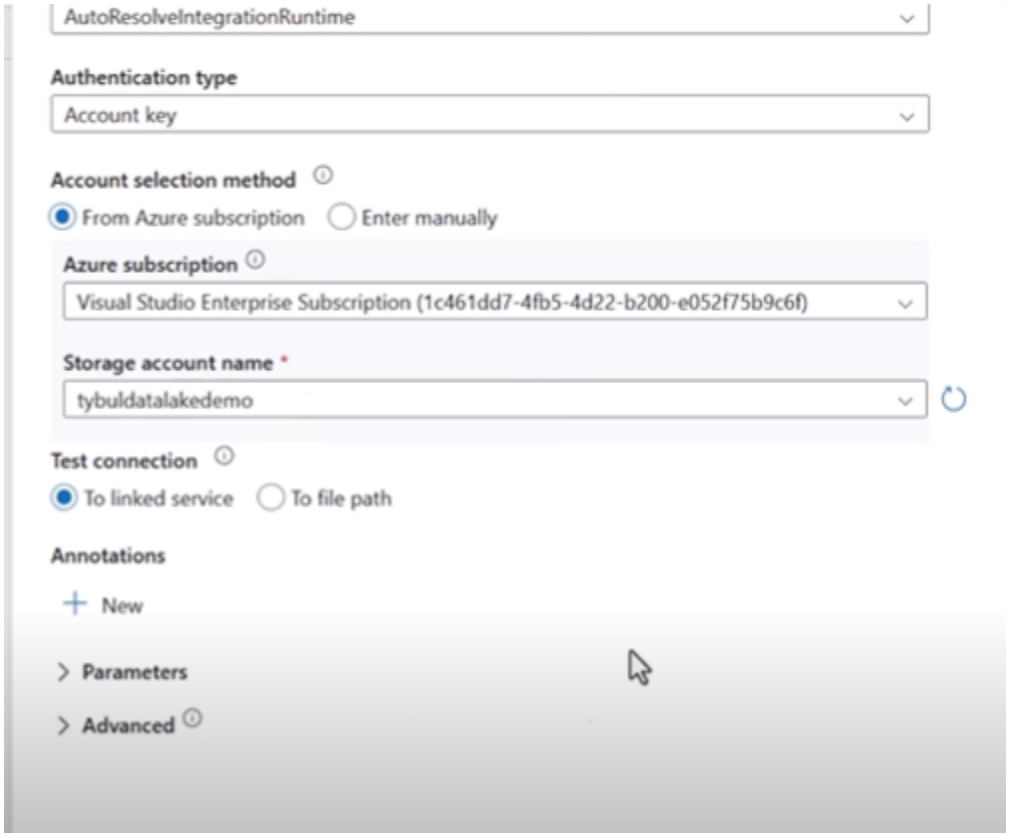




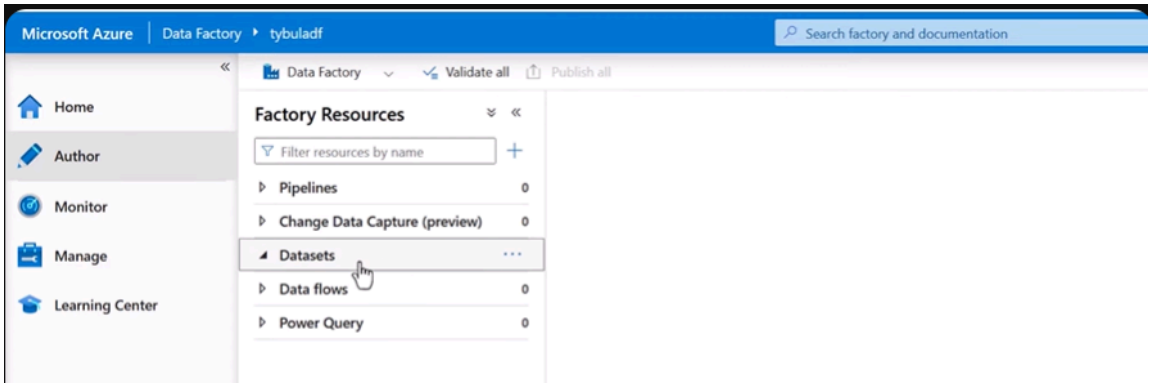
we provide all the required details and then test connection

7. Similarly we create one for destination

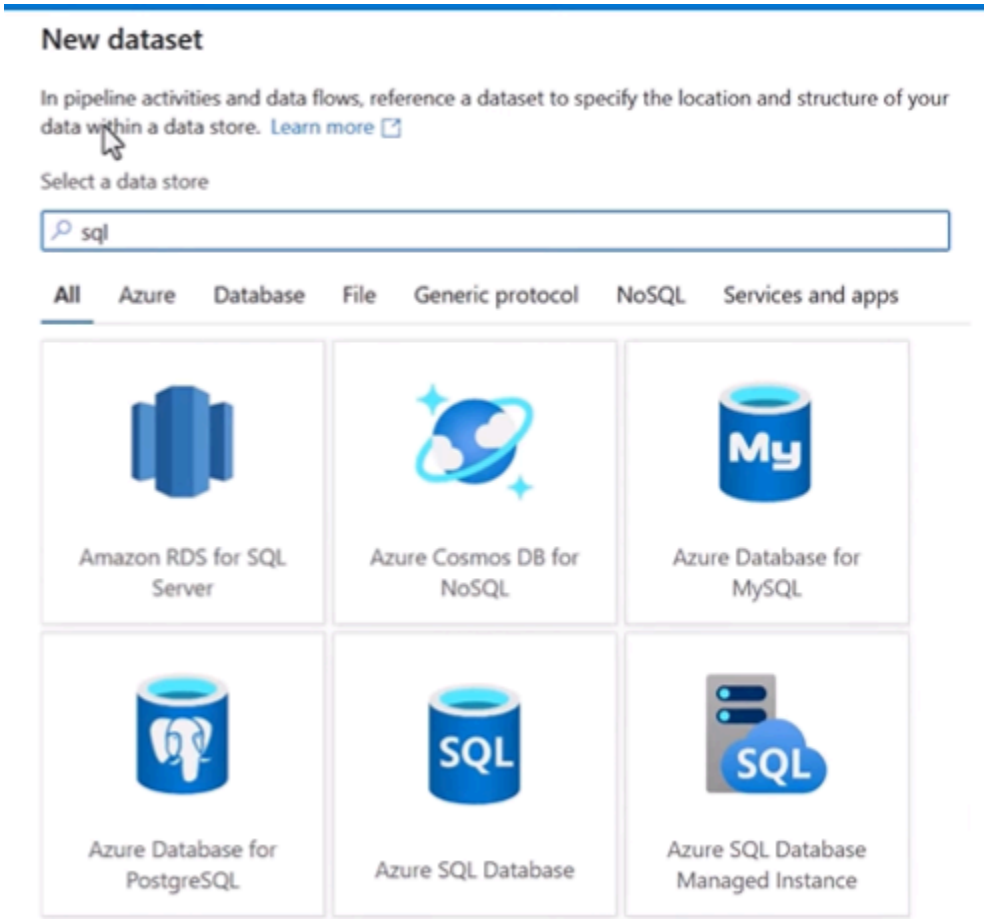
8. We select ADLSg2 linked service and create it by giving required details



9. Next we create data sets
10. To create them we go to author tab and select datasets



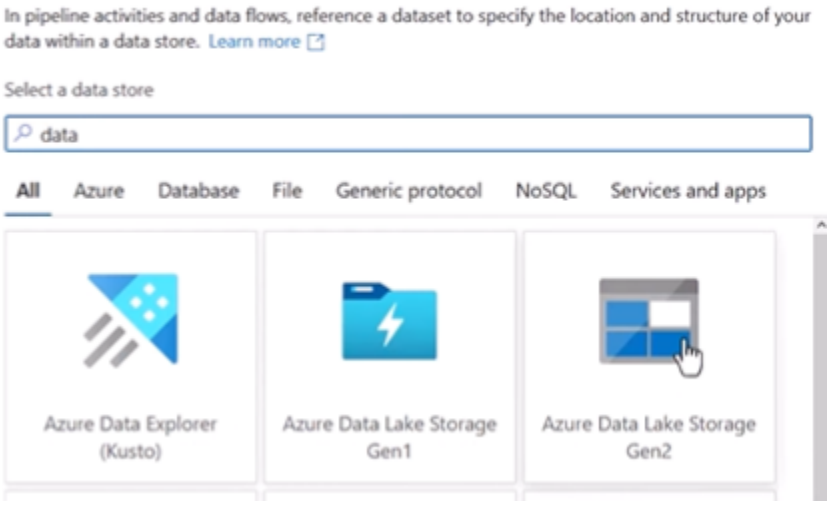
11. We choose dataset for azureSQL DB



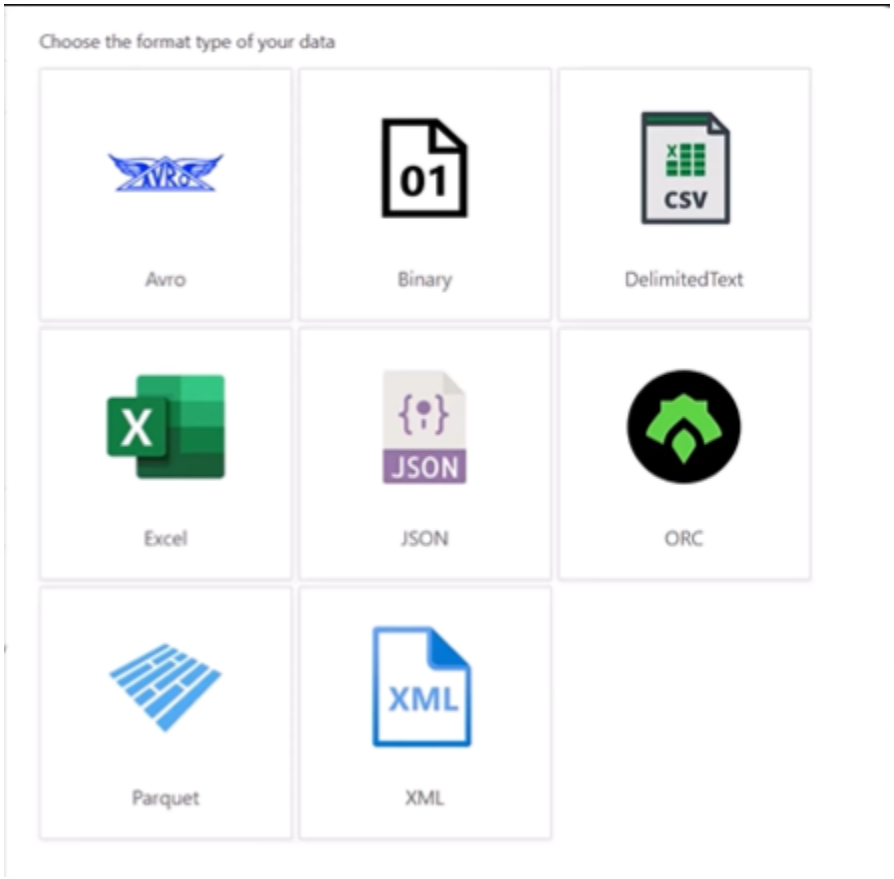
12. Next we give name of dataset and linkedservice

13. Next we give table name of Customer

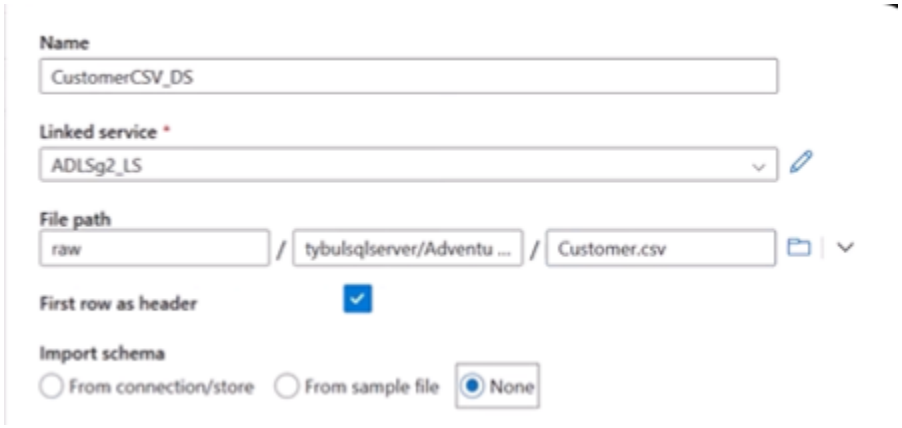
14. After that we create a dataset for destination(ADLSg2) as well….



Next we choose file format

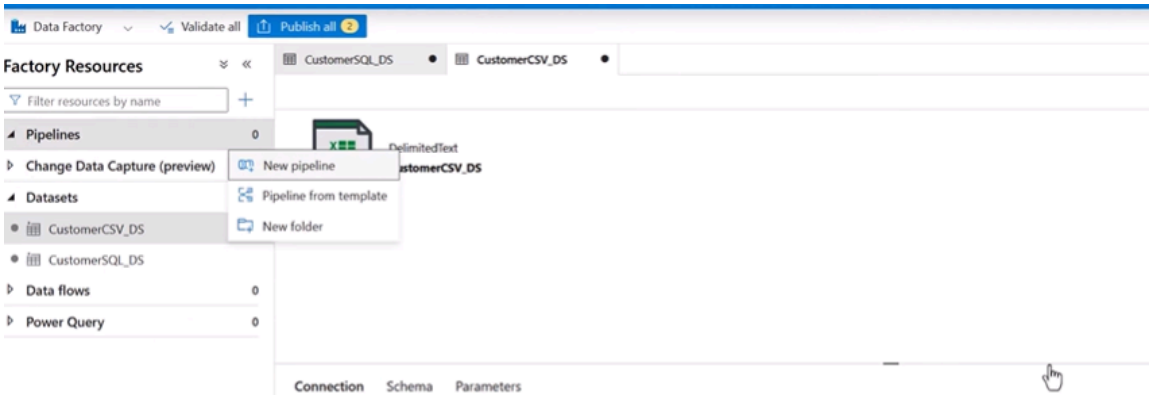


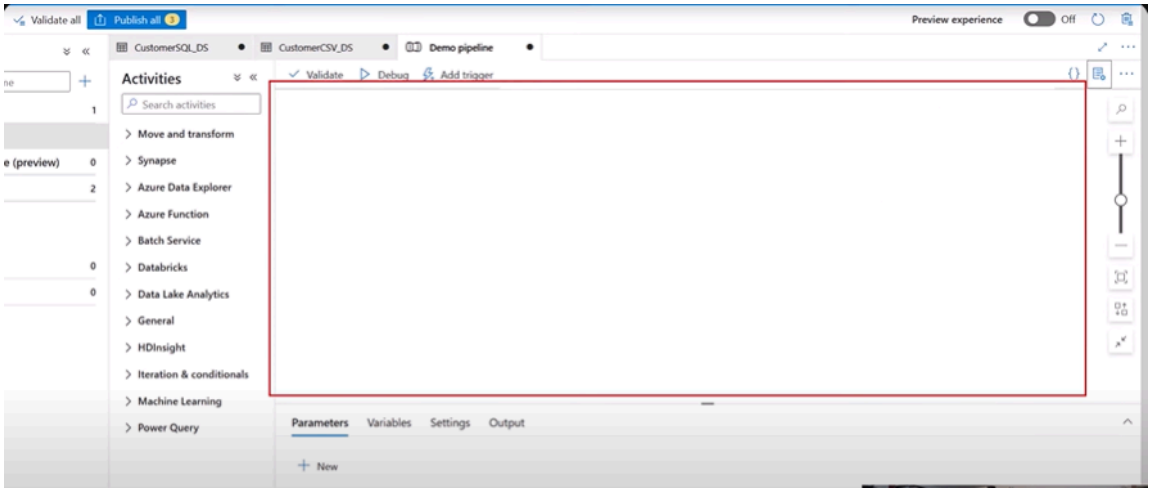
15. Next we create a dataset name,assign the linked service and file path



Pipeline

1. After creating Linkedservices,datasets..we create a pipeline



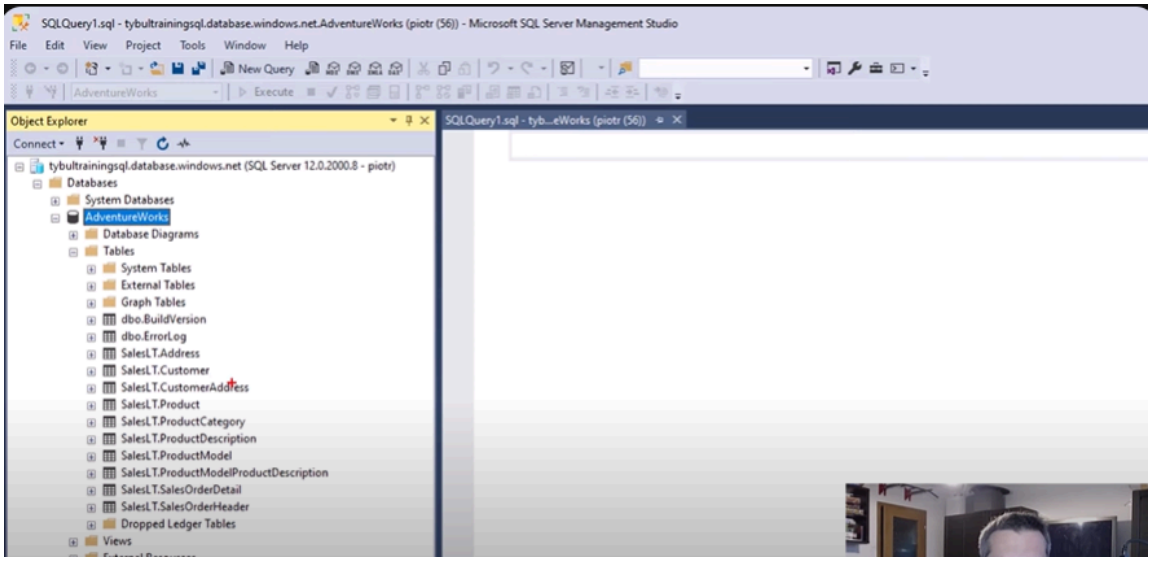
2. 

   in the white space …we drag the activities
3. Now we'll be using copy data activity..refer Mr . K video

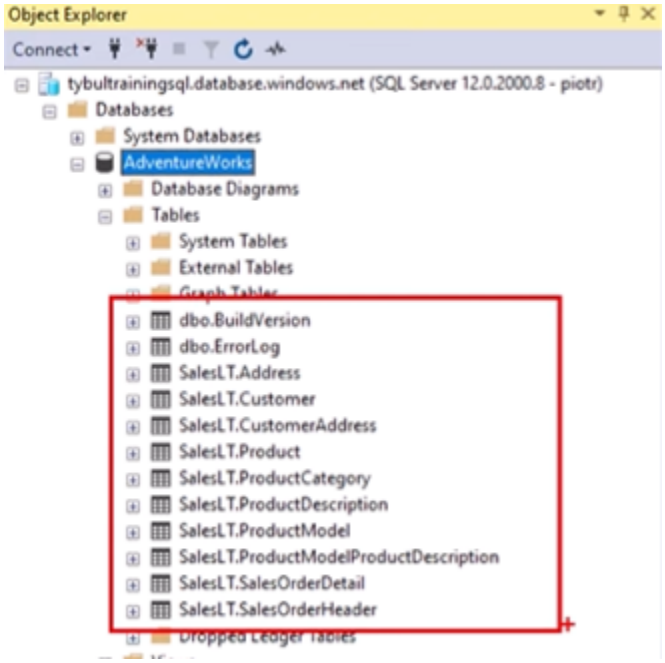
DP203: 11 - Dynamic Azure Data Factory

Business Scenario

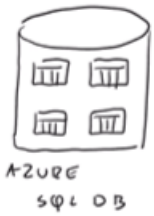
1. Here inside our AzureSQL DB we have multiple tables



2. Now we'll ingest all this tables dynamically



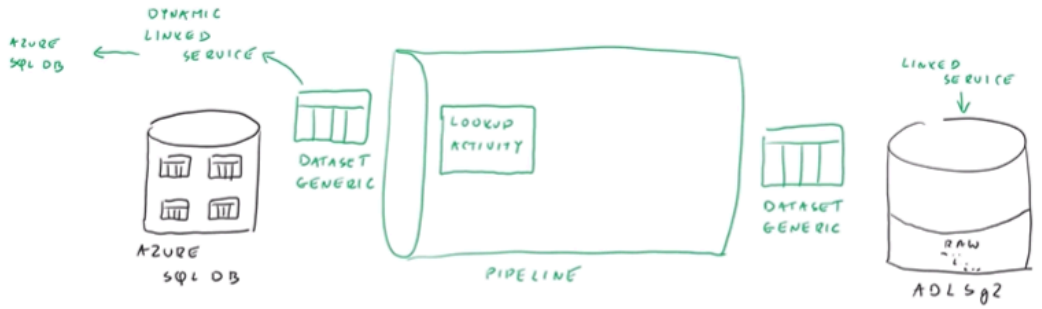
3. Hre we'll migrate all the table present in Azure SQLDB to rwa container inside ADLSg2



raw container has hierarchical folders inside it

4. We'll use linked services to connect to AzureSQL DB and we'll do this dynamically…and we need another linked service for dest
5. After that we create datasets…dataset are nothing but representation of our data….here we want to have generic dataset

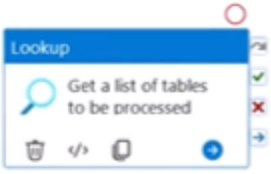
6. And to connect all these services…we need pipelines (it contains activity to move and transform data)
7. We'll use lookup activity that connects to the source DB

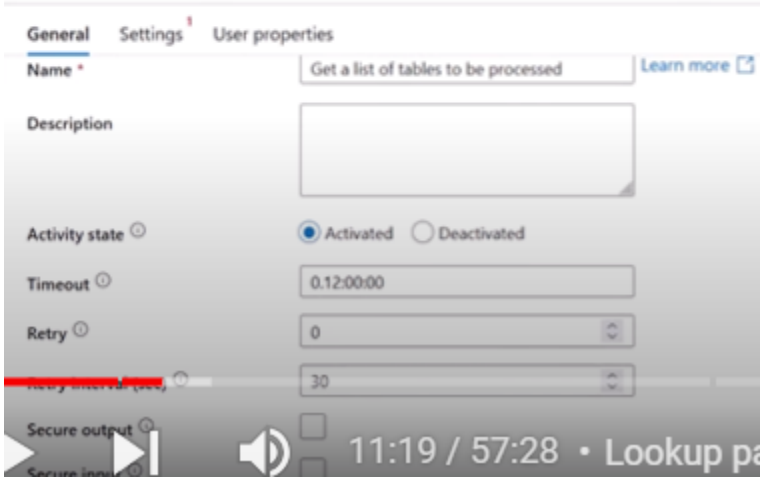


Practical - LookUp

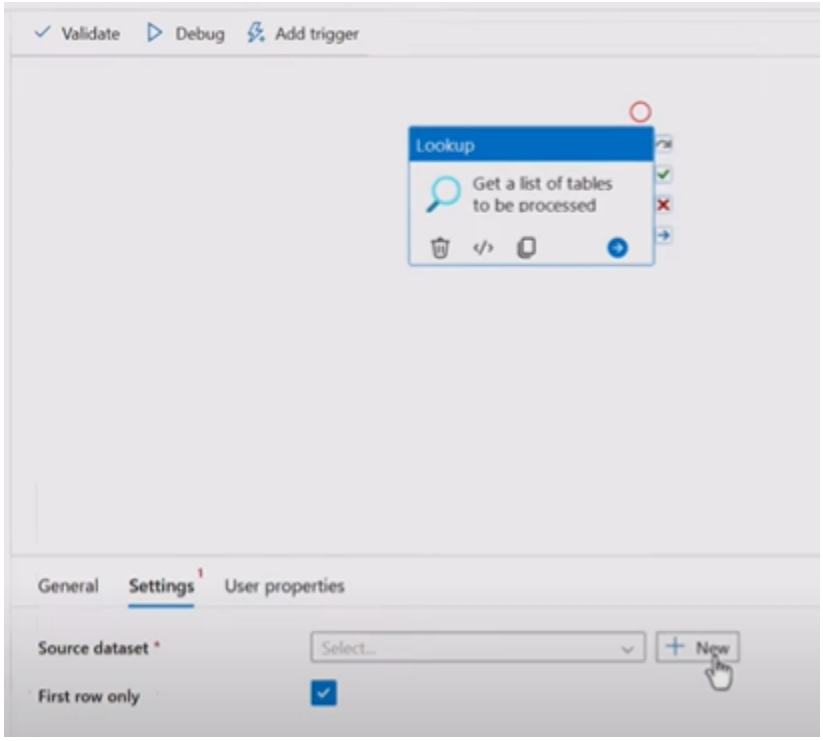
1. We'll start from pipeline

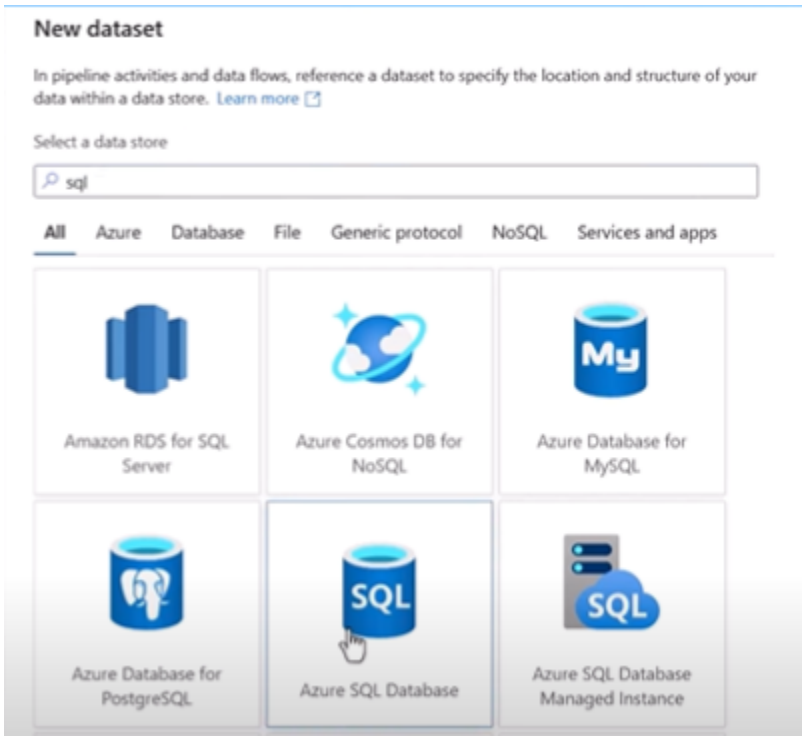
2. Next we create new pipeline and drag lookup activity





3. Next we'll create the dataset





4. Now actually dataset must be connected to linked service…so first we have to create linked service
5. While creating a linkedservice…we will not choose any DB..instead we will add parameters

Authentication type *

SQL authentication

User name *

Password | Azure Key Vault

Password *

Always encrypted

Additional connection properties

+ New

Annotations

+ New

Parameters

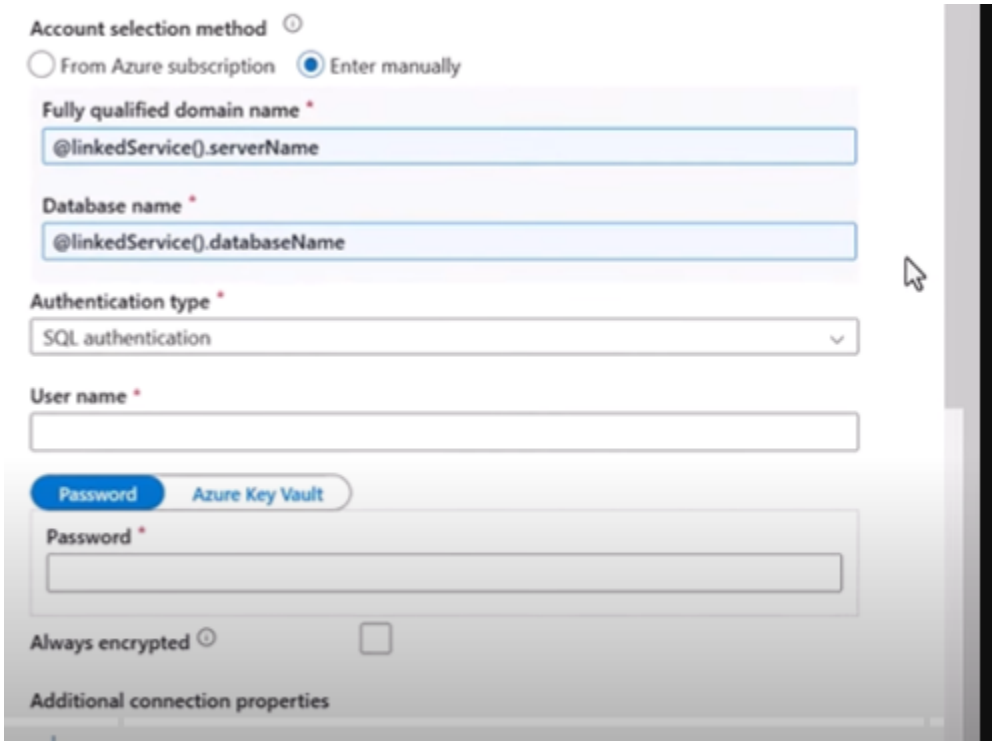
Advanced

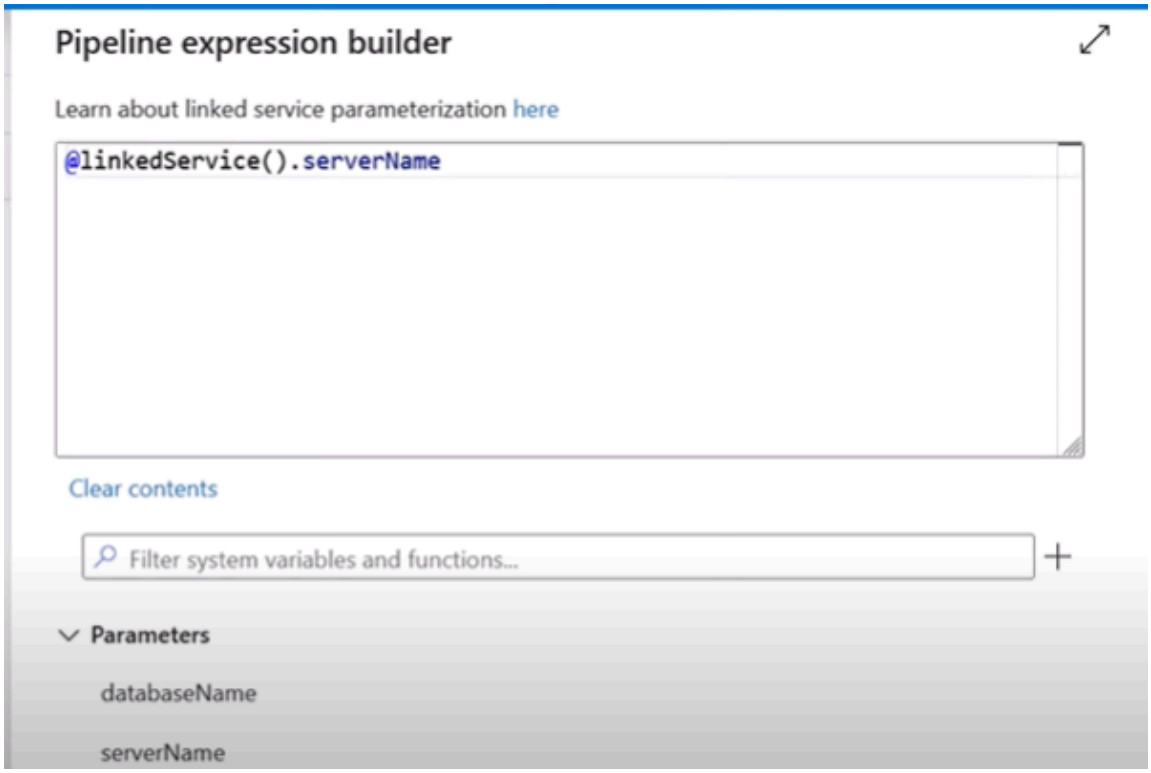
Parameters

+ New | Delete

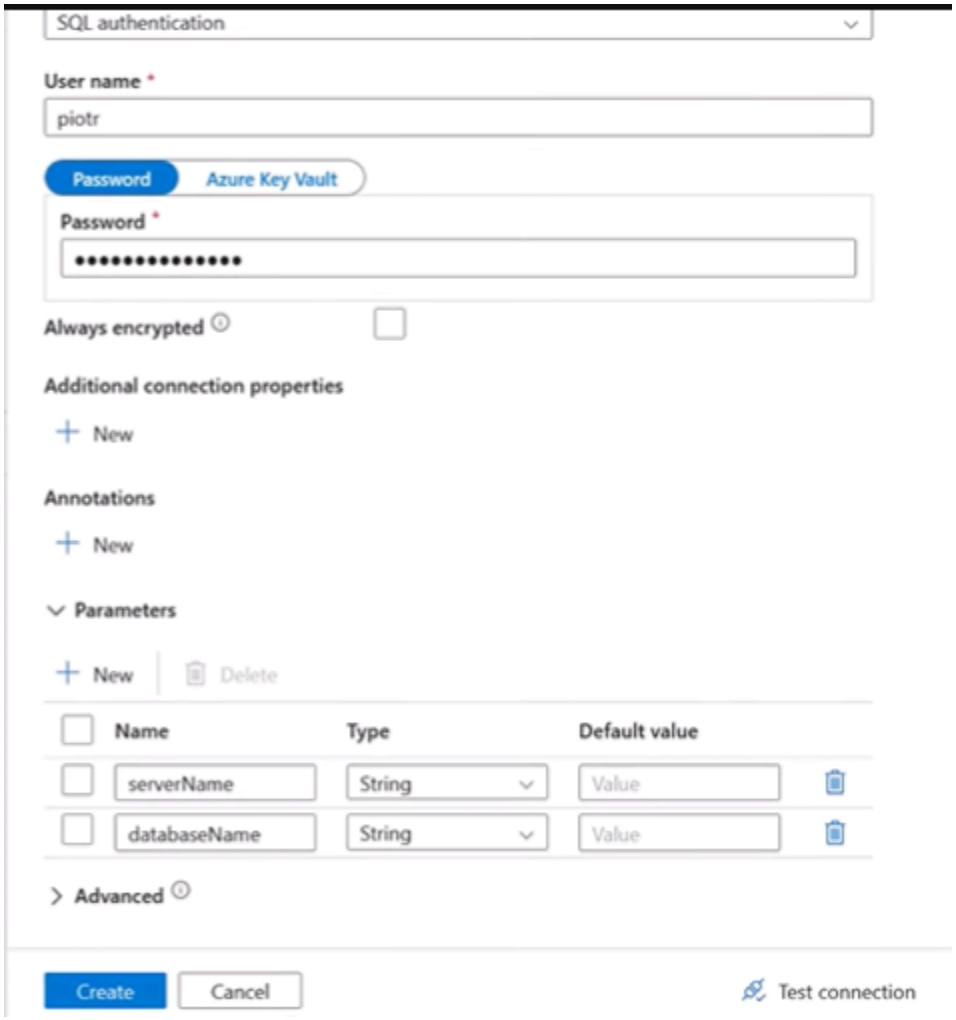
Name	Type	Default value
serverName	String	Value
databaseName	String	Value

6. Now we'll choose enter manually instead of azure subcription

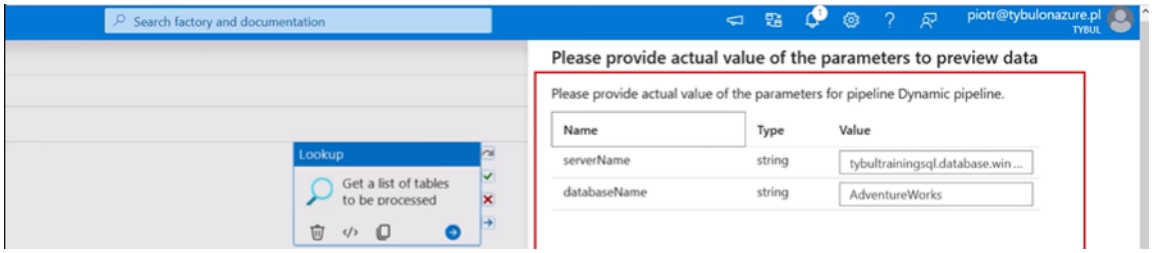




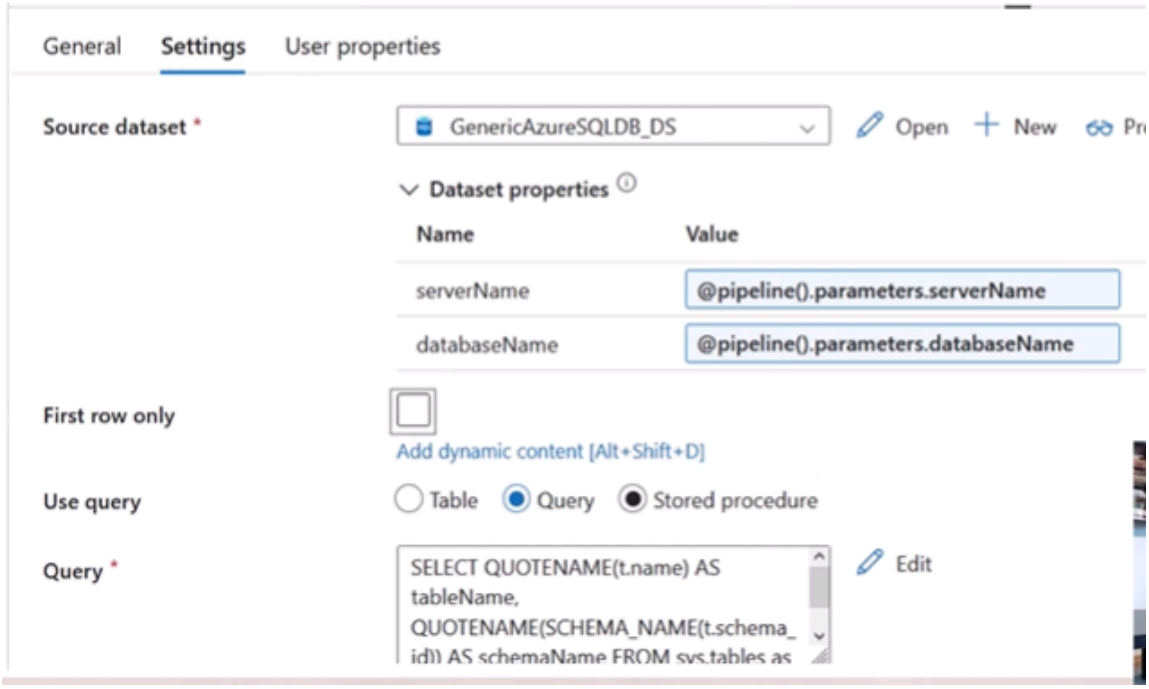
7. Next we create the pipeline



8. WHat we have done here is …after creating lookup table …we gave parameters of our azureSQL to our pipeline …and lookup activity got this parameters from pipeline and it gets connected to the pipeline

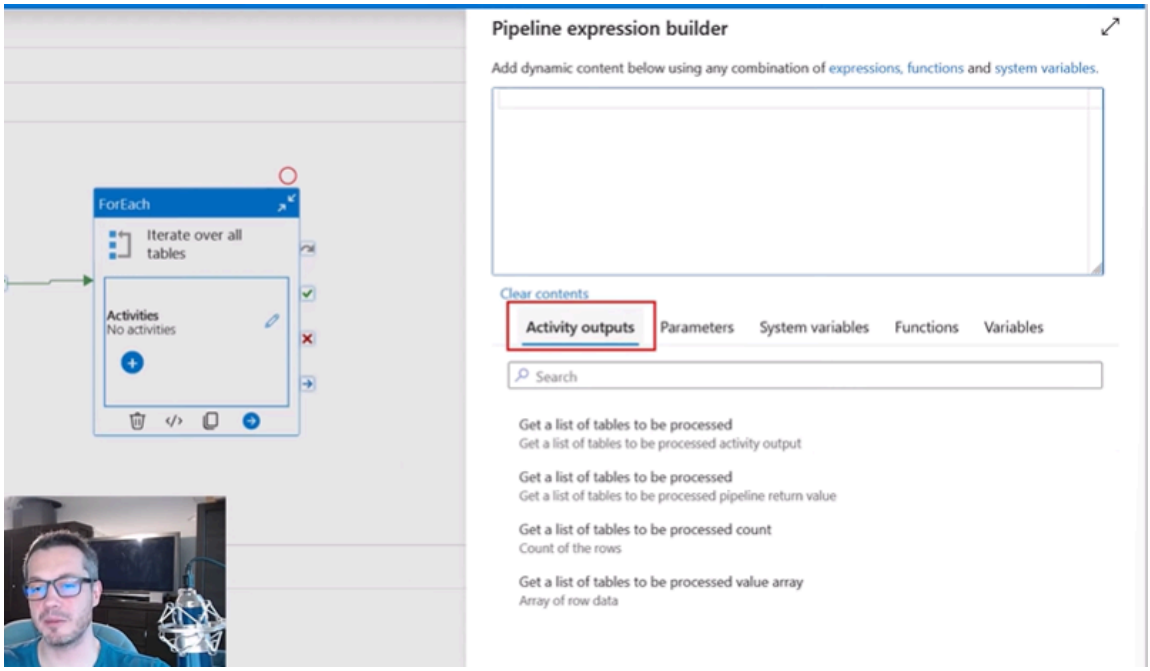


And we write a query to get all tables from DB

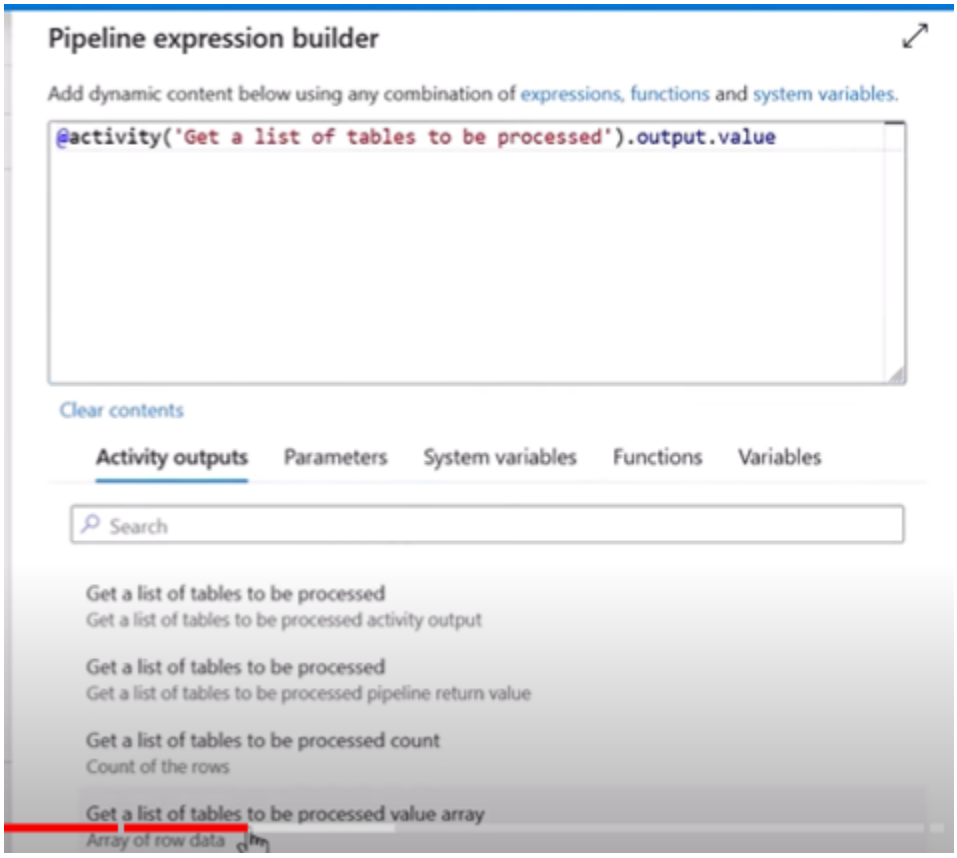


9. Later this output will be feed to foreach activity through dynamic content

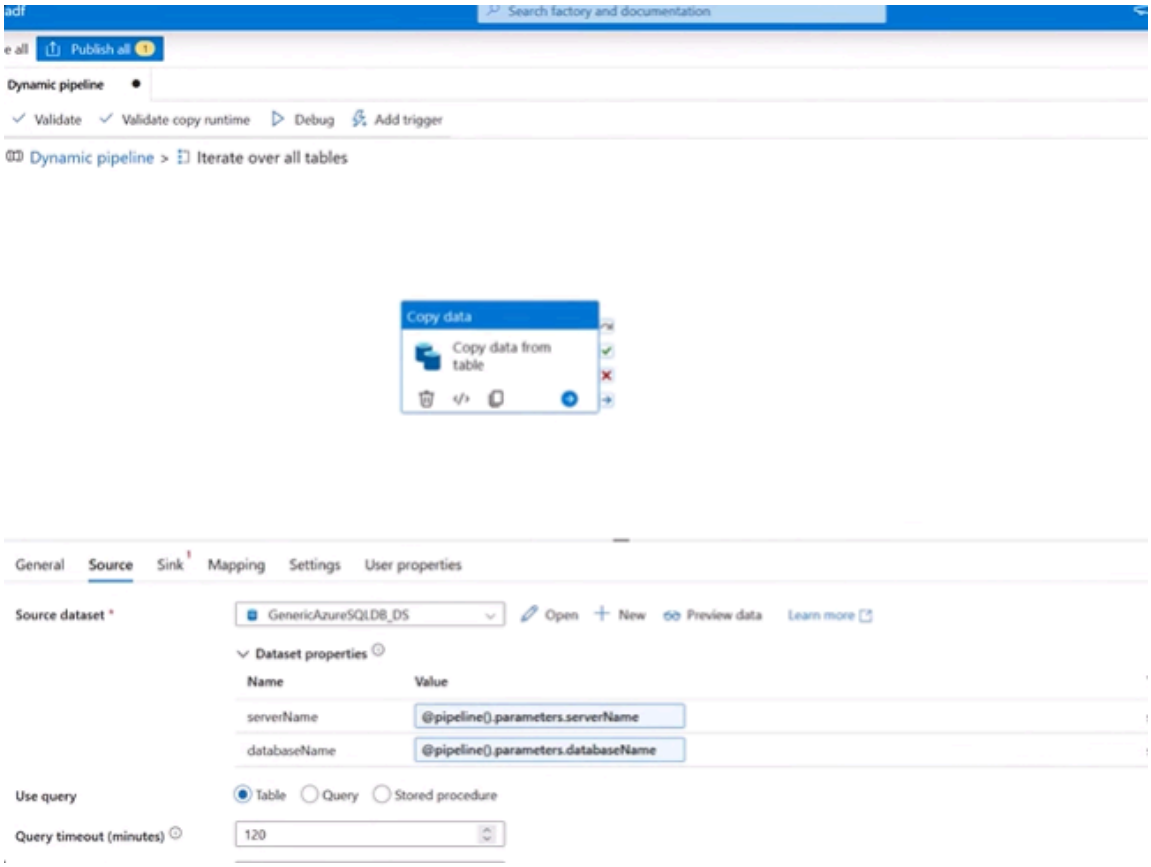
10. In ADF in any task…we can have output of previous activity



11. As lookup activity return output in array…we choose array of row data



12. Inside for each activity…we add copy activity
13. For copy activity we have mention source and sink..dynamically
14. For source …we add dynamic content of parameters(Pipeline parameters of AzureSQL DB)

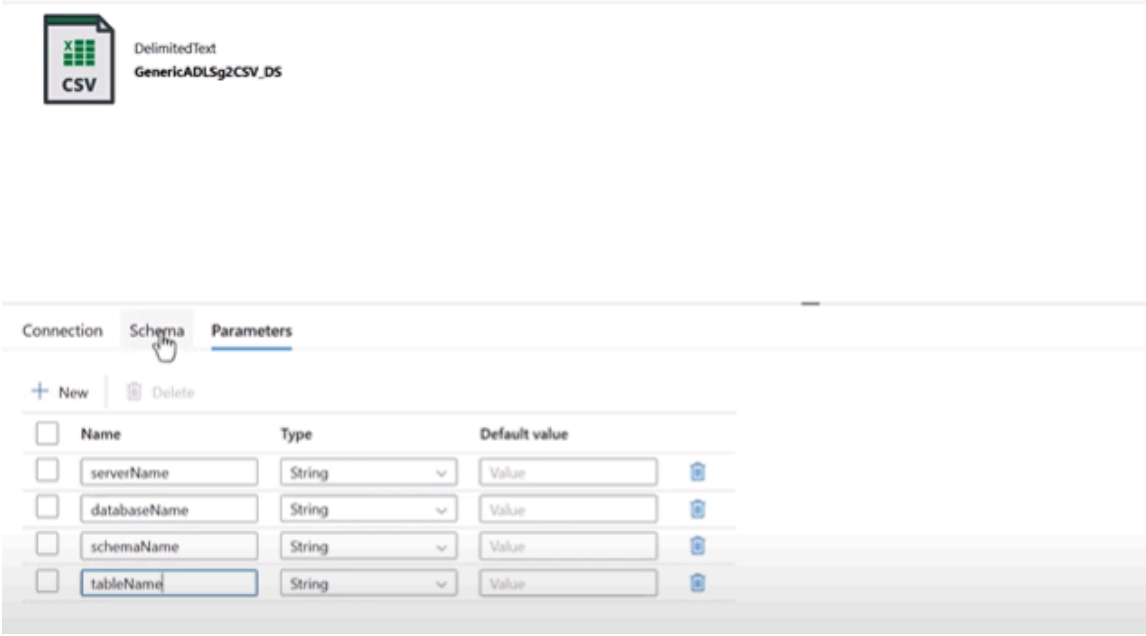


it is just connection string to our source DB

15. Now we will write a query which gets each table from name from lookup..and then it get table data from source

16. Coming to the sink

17. First we have to create a dataset for storing data inside ADLSg2 ..after creating a dataset we can open it up and add dynamic parameters



18.