PROJECT- **DATA LOADING WITH INCREMENTAL PROCESSING**

NANDINI RATHORE

INDEX

1. Table Creation:

• Create 5 tables, where:

• Two tables should have a column with int data type that serves as a unique identifier.

• Three tables should have a datetime column to track the time of data insertion or update.

2. Watermark Table:

• Create a watermark table that will store the last processed value (int or datetime) for each table. This will be used to track the most recent value processed, ensuring that only new data is loaded during subsequent runs.

3. Stored Procedure for Incremental Data Load:

* Write a stored procedure that:

• Takes the table name and the last processed value (from the watermark table) as inputs.

• Fetches the new records based on the last processed value (delta or datetime).

• Updates the watermark table after successfully loading the new records.

4. Dynamic Pipeline Design:

• Design a dynamic data pipeline that performs the following:

• Extracts the latest data incrementally from the source tables.

• Uses the watermark table to track the last processed record for each table.

• Loads the extracted data to the target destination.

• Ensures that only new data is loaded into the target, preventing reprocessing of already loaded data.

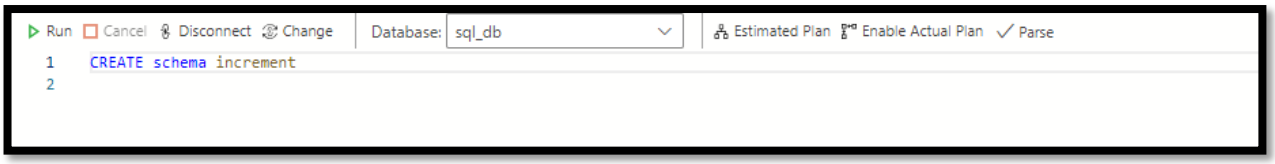
1. Table Creation:

• Create 5 tables, where:

• Two tables should have a column with int data type that serves as a unique identifier.

• Three tables should have a datetime column to track the time of data insertion or update.

1. INCREMENT .PATIENTS_RECORD




```
create table increment.patients_record (patient_id int IDENTITY(1,1),
name varchar(50),
gender varchar(50),
age int
)
insert into increment.patients_record values('ARZOO','F',18)
insert into increment.patients_record values('SUDESH','M',25)
insert into increment.patients_record values('ROHAN','M',28)
insert into increment.patients_record values('SHWETA','F',22)
SELECT * FROM increment.patients_record
```

Results | Messages

	patient_id	name	gender	age
1	1	ARZOO	F	18
2	2	SUDESH	M	25
3	3	ROHAN	M	28
4	4	SHWETA	F	22

PATIENTS_ID INCREMENT AUTOMATICALLY BECAUSE I WRITE IDENTITY(1,1).TABLE HAS PATIENT'S RECORD.

2.APPOINTMENT_LOG TABLE

```
CREATE TABLE increment.APPOINTMENT_LOG1 (APPOINTMENT_ID INT IDENTITY(1,1), patient_id INT ,DOCTOR_ID INT ,STATUS VARCHAR(20), APPOINTMENT_DATE DATETIME)
INSERT INTO  increment.APPOINTMENT_LOG1 VALUES(1,101,'COMPLETED','2025-06-27 10:15:00')
INSERT INTO  increment.APPOINTMENT_LOG1 VALUES(2,103,'COMPLETED','2025-06-26 12:18:00')
INSERT INTO  increment.APPOINTMENT_LOG1 VALUES(3,104,'PENDING','2025-06-30 01:15:00')
INSERT INTO  increment.APPOINTMENT_LOG1 VALUES(4,104,'COMPLETED','2025-06-22 08:45:00')
SELECT * FROM increment.APPOINTMENT_LOG1
```

TABLE HAVING APPOINTMENT_ID,PATIENT_ID,DOCTORS_ID ,STATUS AND APPOINTMENT_DATE.

Results Messages

	APPOINTMENT_ID	patient_id	DOCTOR_ID	STATUS	APPOINTMENT_DATE
1	1	1	101	COMPLETED	2025-06-27 10:15:00.000
2	2	2	103	COMPLETED	2025-06-26 12:18:00.000
3	3	3	104	PENDING	2025-06-30 01:15:00.000
4	4	4	104	COMPLETED	2025-06-22 08:45:00.000
5	5	5	105	COMPLETED	2025-06-25 11:18:00.000

I USE THIS STATEMENT BECAUSE I WANT TO START APPOINTMENT_ID FROM 1 ”DBCC CHECKIDENT('INCREMENT.APPOINTMENT_LOG',RESEED,1)”

3.REPORT TABLE

```
30    CREATE  TABLE INCREMENT.MEDICAL_REPORTS1(R_ID INT ,PATIENT_ID INT,R_TYPE VARCHAR(50),GENERATE_DATE DATETIME)
31    INSERT INTO  increment.MEDICAL_REPORTS1 VALUES(010,1,'SUGAR TEST','2025-06-28 12:00:00')
32    INSERT INTO  increment.MEDICAL_REPORTS1 VALUES(111,2,'THYROID','2025-06-29 12:00:00')
33    INSERT INTO  increment.MEDICAL_REPORTS1 VALUES(222,3,'VITB12','2025-07-10 12:00:00')
34    INSERT INTO  increment.MEDICAL_REPORTS1 VALUES(333,4,'BLOOD TEST','2025-06-29 12:00:00')
35    SELECT * FROM increment.MEDICAL_REPORTS1
```

Results Messages

	R_ID	PATIENT_ID	R_TYPE	GENERATE_DATE
1	10	1	SUGAR TEST	2025-06-28 12:00:00.000
2	111	2	THYROID	2025-06-29 12:00:00.000
3	222	3	VITB12	2025-07-10 12:00:00.000
4	333	4	BLOOD TEST	2025-06-29 12:00:00.000

Increment coloumn is “generate_date”.

4.STAFF_ACTIVITY

```
40  CREATE TABLE INCREMENT.STAFF_ACTIVITY1(ACTIVITY_ID INT ,STAFF_ID INT,ACTIVITY_TYPE VARCHAR(20),ACTIVITY_TIME DATETIME)
41  INSERT INTO  INCREMENT.STAFF_ACTIVITY1 VALUES(101,11,'LOGIN','2025-06-28 08:15:00')
42  INSERT INTO  INCREMENT.STAFF_ACTIVITY1 VALUES(102,12,'LOGOUT','2025-06-28 19:15:00')
43  INSERT INTO  INCREMENT.STAFF_ACTIVITY1 VALUES(103,13,'LOGOUT','2025-06-28 19:45:00')
44  INSERT INTO  INCREMENT.STAFF_ACTIVITY1 VALUES(104,14,'LOGIN','2025-06-28 08:00:00')
45  SELECT * FROM increment.STAFF_ACTIVITY1
```

Results Messages

	ACTIVITY_ID	STAFF_ID	ACTIVITY_TYPE	ACTIVITY_TIME
1	101	11	LOGIN	2025-06-28 08:15:00.000
2	102	12	LOGOUT	2025-06-28 19:15:00.000
3	103	13	LOGOUT	2025-06-28 19:45:00.000
4	104	14	LOGIN	2025-06-28 08:00:00.000

INCREMENT_COLOUMN=”ACTIVITY_TIME”

5.BILLING_TRANSACTIONS

```
49  CREATE TABLE INCREMENT.BILLING_TRANSACTION(TRANSACTION_ID INT ,PATIENT_ID INT,AMOUNT INT,PAYMENT_METHOD VARCHAR(100),TRANSACTION_DATE DATETIME)
50  INSERT INTO INCREMENT.BILLING_TRANSACTION VALUES (411,1,1200,'DEBIT CARD','2025-06-28 13:00:00')
51  INSERT INTO INCREMENT.BILLING_TRANSACTION VALUES (412,2,3500,'CREDIT CARD','2025-06-29 14:00:00')
52  INSERT INTO INCREMENT.BILLING_TRANSACTION VALUES (413,3,4200,'DEBIT CARD','2025-07-10 12:30:00')
53  INSERT INTO INCREMENT.BILLING_TRANSACTION  VALUES (414,4,2200,'GPAY','2025-06-29 13:30:00')
54  SELECT * FROM increment.BILLING_TRANSACTION
```

Results Messages

	TRANSACTION_ID	PATIENT_ID	AMOUNT	PAYMENT_METHOD	TRANSACTION_DATE
1	411	1	1200	DEBIT CARD	2025-06-28 13:00:00.000
2	412	2	3500	CREDIT CARD	2025-06-29 14:00:00.000
3	413	3	4200	DEBIT CARD	2025-07-10 12:30:00.000
4	414	4	2200	GPAY	2025-06-29 13:30:00.000

INCREMENT_COLOUMN=”TRANSACTION_DATE”

2. Watermark Table:

- Create a watermark table that will store the last processed value (int or datetime) for each table. This will be used to track the most recent value processed, ensuring that only new data is loaded during subsequent runs.

CREATE INCREMENT.WATERMARK TABLE:-

```
create table increment.watermark2 (tablename varchar(50),schemaname varchar(50),LPV nvarchar(100),increment_coloumn varchar(50),foldername varchar(50), filename varchar (50))
insert into increment.watermark2 values ('PATIENTS_RECORD','increment',0,'PATIENT_id','increment/patients_record','patients_record.csv')
insert into increment.watermark2 values ('APPOINTMENT_LOG1','increment',0,'APPOINTMENT_ID','increment/APPOINTMENT_LOG1','APPOINTMENT_LOG1.csv')
insert into increment.watermark2 values ('MEDICAL_REPORTS1','increment','1900-01-01 00:00:00','generate_date','increment/MEDICAL_REPORTS1','MEDICAL_REPORTS1.csv')
insert into increment.watermark2 values ('STAFF_ACTIVITY1','increment','1900-01-01 00:00:00','ACTIVITY_time','increment/STAFF_ACTIVITY1','STAFF_ACTIVITY1.csv')
insert into increment.watermark2 values ('BILLING_TRANSACTION','increment','1900-01-01 00:00:00','TRANSACTION_date','increment/BILLING_TRANSACTION','BILLING_TRANSACTION.csv')
SELECT * FROM INCREMENT.WATERMARK2
```

Results Messages

	tablename	schemaname	LPV	increment_coloumn	foldername	filename
1	PATIENTS_RECORD	increment	0	PATIENT_id	increment/patients_record	patients_record.csv
2	APPOINTMENT_LOG1	increment	0	APPOINTMENT_ID	increment/APPOINTMENT_LOG1	APPOINTMENT_LOG1.csv
3	MEDICAL_REPORTS1	increment	1900-01-01 00:00:00	generate_date	increment/MEDICAL_REPORTS1	MEDICAL_REPORTS1.csv
4	STAFF_ACTIVITY1	increment	1900-01-01 00:00:00	ACTIVITY_time	increment/STAFF_ACTIVITY1	STAFF_ACTIVITY1.csv
5	BILLING_TRANSACTION	increment	1900-01-01 00:00:00	TRANSACTION_date	increment/BILLING_TRANSACTION	BILLING_TRANSACTION.csv

- LPV=0 FOR “PATIENTS_RECORD” AND”APPOINTMENT_LOG1”TABLE BECAUSE INCREMENT_COLOUMN IS “PATIENTS_ID” AND “APPOINTMENT_ID”AND
- LPV=’1900-00-00 00:00:00’/’YYYY-MM-DD HH:MM:SS’ IT SHOULD NEVER BE 0 FOR DATETIME DATATYPE.
- DATATYPE OF LPV IS ‘NVARCHAR’ SO THAT IT CAN HANDLE BOTH INT AND VARCHAR.

3. Stored Procedure for Incremental Data Load:

* Write a stored procedure that:

• Takes the table name and the last processed value (from the watermark table) as inputs.

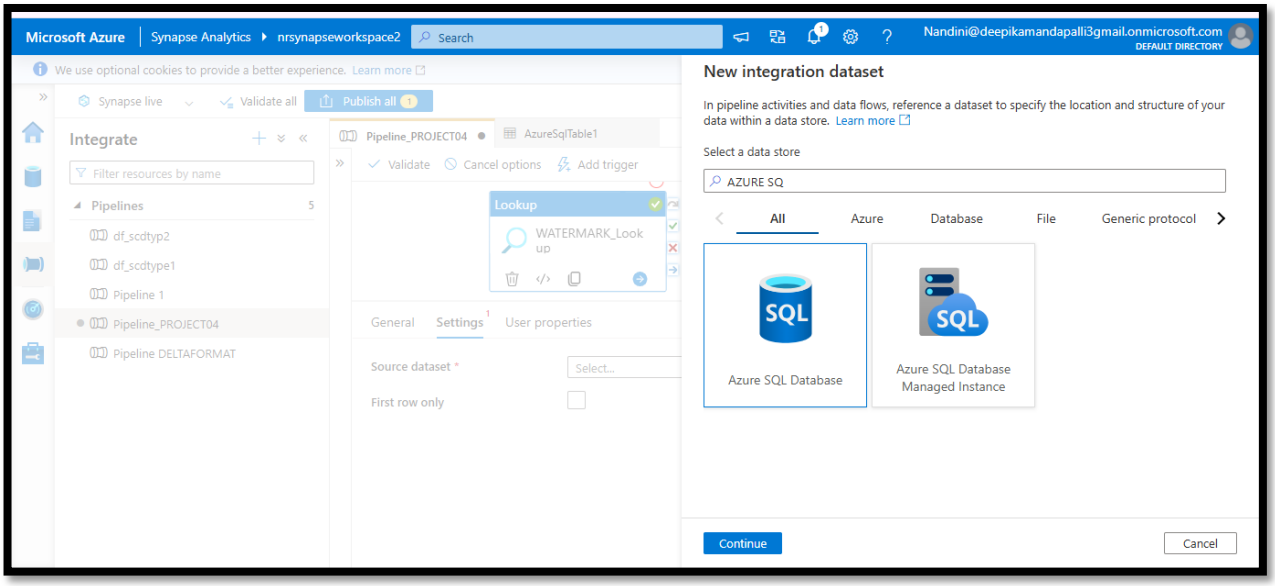
```
create proc increment.usp_watermark_update @tablename varchar(50), @lpv varchar(100) as update increment.watermark  set lpv=@lpv where  tablename =@tablename
```

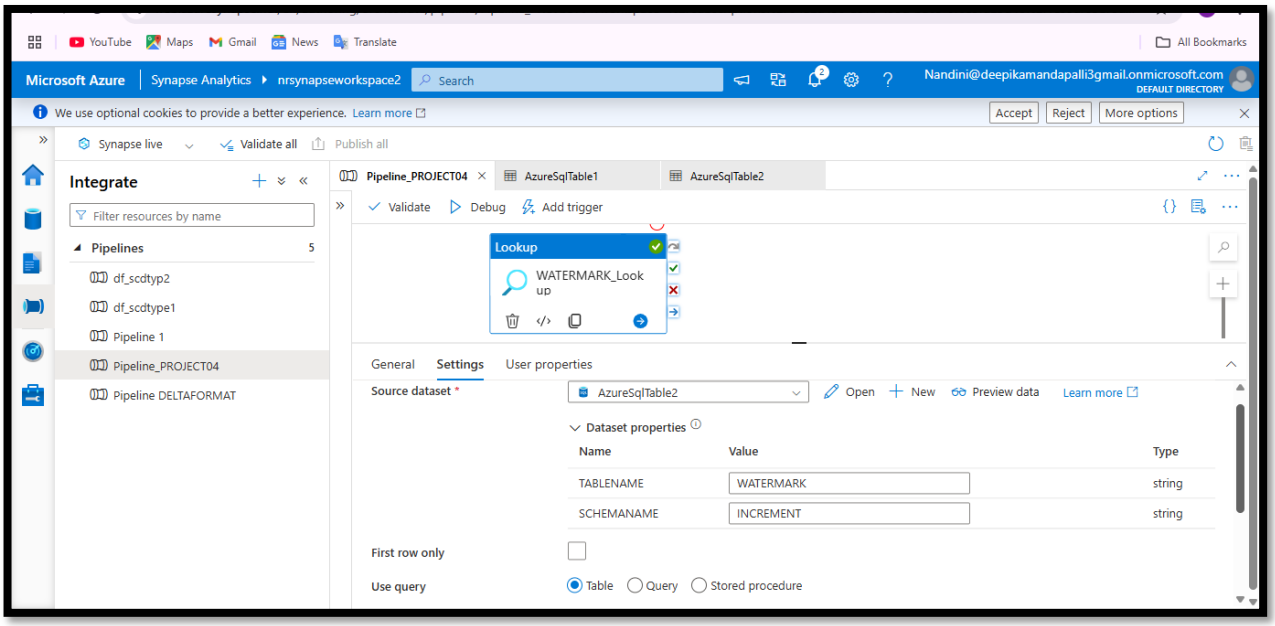
- WE CREATE A STORED PROCEDURE SO THAT AFTER EXECUTION OF PIPELINE IT SHOWS US NEW PROCESSED VALUE IN THE TABLE.
- FOR THE FIRST TIME WE ALWAYS GET FULL LOAD ,AFTER THAT BASED ON LPV VALUE AND INCREMENT_COLOUMN WE GET INCREMENT DATA.

4• Design a dynamic data pipeline that performs the following:

• Extracts the latest data incrementally from the source tables.

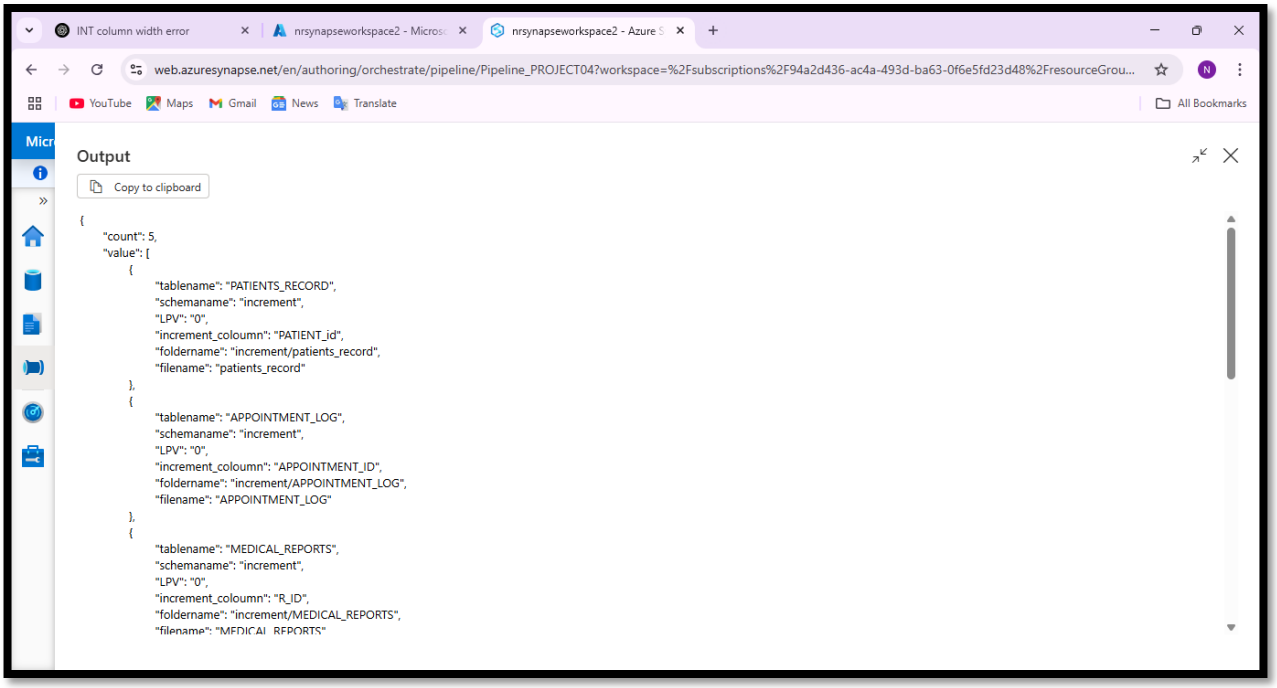
NOW GO TO SYNAPSE, DRAG LOOKUP ACTIVITY AND SELECT SOURCE DATASET=AZURE SQLDB.



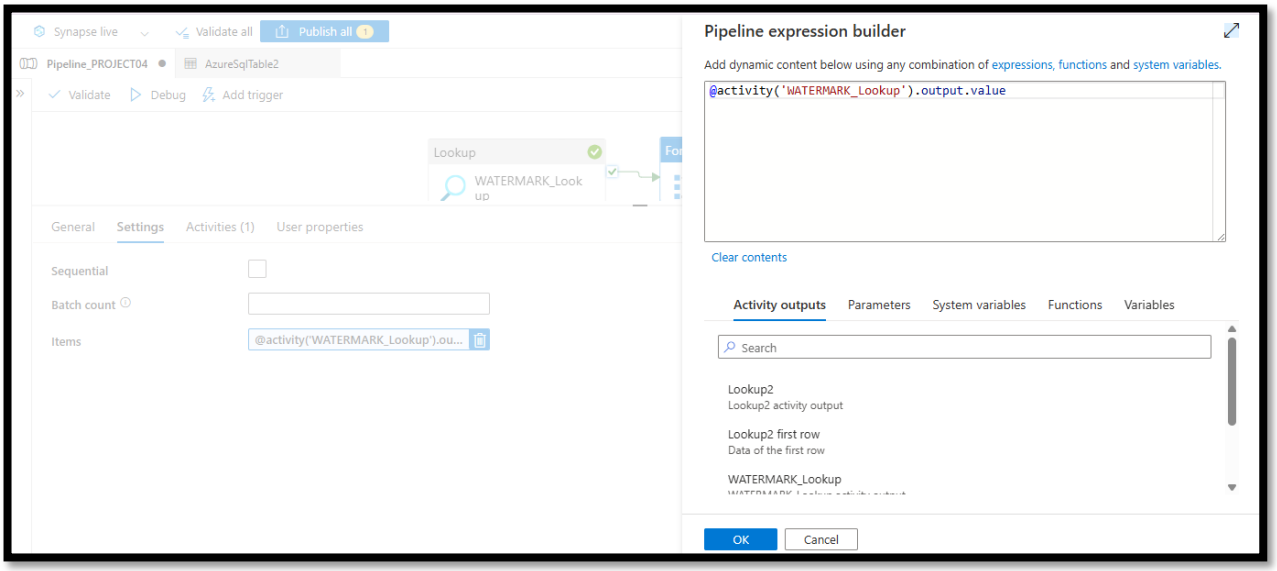


CREATE PARAMETERS SCHEMANAME AND TABLENAME ,WRITE TABLENAME TO READ ALL THE TABLES PRESENT IN WATERMARK TABLE BY USING LOOKUP ACTIVITY .

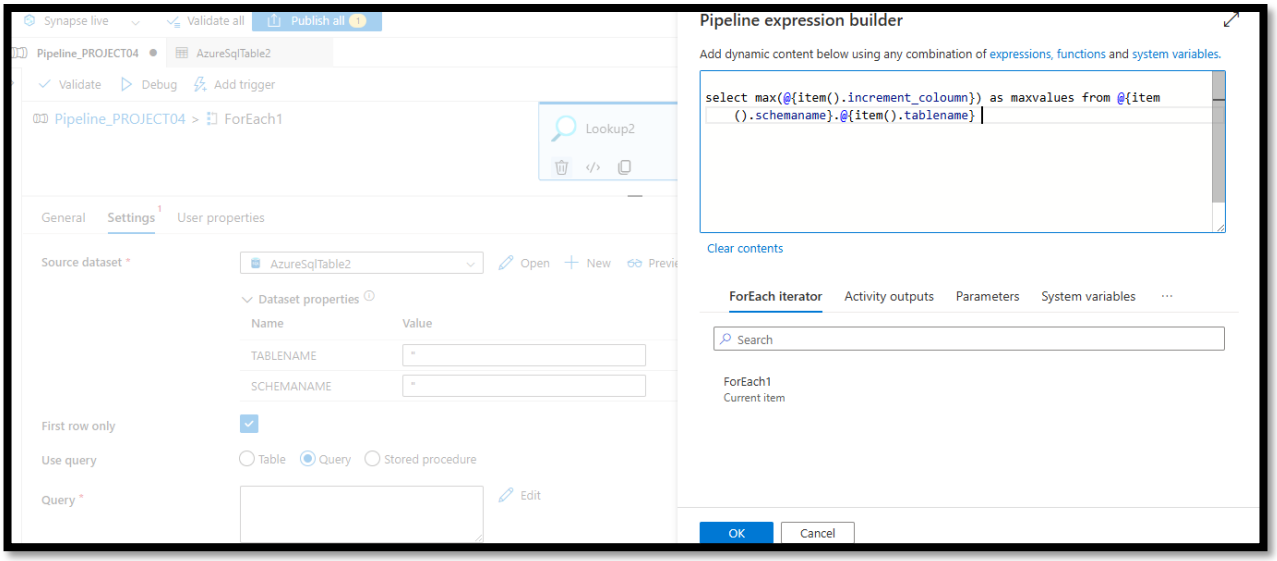
WATERMARK TABLE IS DIFFERENT FOR METADATA TABLE ,USED IN INCREMENT LOADING TO CHECK THE LPV,SIMPLE AND FOCUS ONLY DATA CHANGING TRACKINGS.



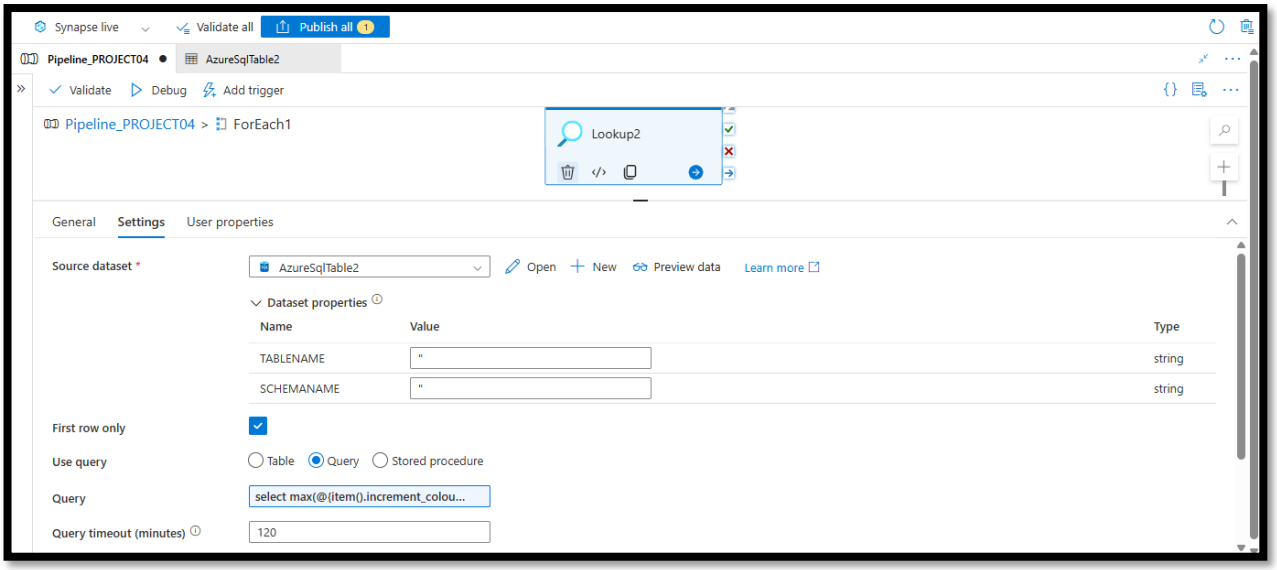
OUTPUT:-> COUNT=5 MEANS 5 TABLES ARE THERE IN WATERMARK TABLE.



NOW DRAG FOR EACH ACTIVITY, IT TAKES THE OUTPUT ARRAY FROM THE LOOKUP AND LOOP EVERY ROW OF THE TABLE.

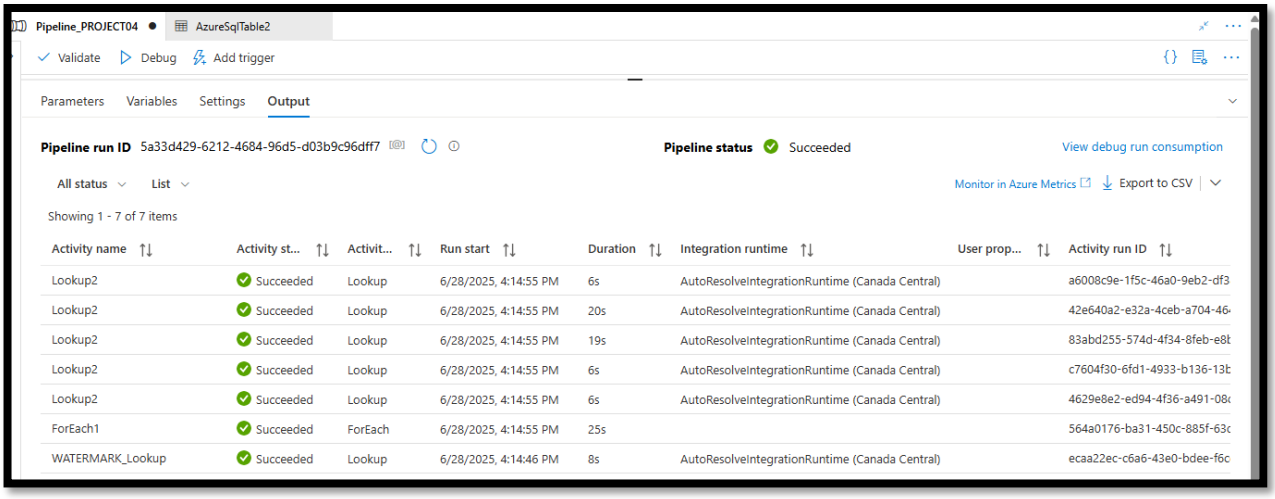


ADD LOOKUP ACTIVITY IN FOR EACH ACTIVITY TO GET SPECIFIC DETAILS FOR EACH ITEM WHILE LOOPING.(FROM WATERMARK TABLE LIKE LPV VALUE.)



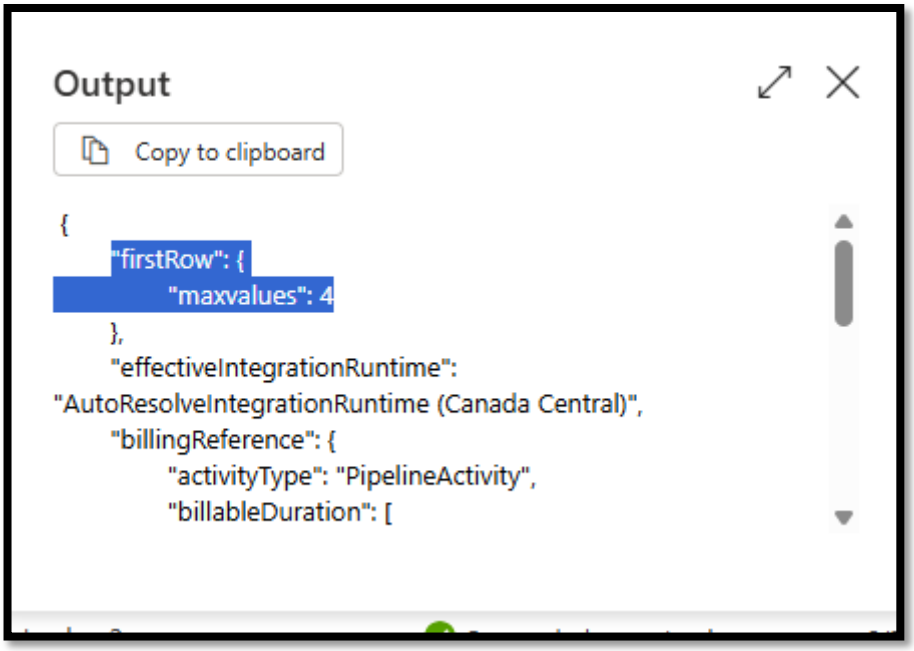
You must put "" (or some dummy value) here to save the pipeline DURING Design-Time (before pipeline runs).These will be replaced by actual table and schema values dynamically from ForEach during Run-Time (when ForEach runs). table name and schema will be taken from a control table or metadata list dynamically. These expressions override the design-time "" and pass real values during pipeline execution. tablename : @item().tablename, schemaname : @item().schemaname.

In query,TO select the max value "**SELECT MAX(@{ITEM().INCREMENT_COLOUMN}) AS MAXVALUES FROM @{ITEM().SCHEMANAME}.@{ITEM().TABLENAME}**"



PUBLISH PIPELINE AND DEBUG THIS.

OUTPUT OF ONE TABLE:> MAXVALUES=2025-06-28T19:45:00Z(T SEPARATE THE TIME FROM DATE ,Z MEANS TIME IS UTC TIME ZONE)



OUTPUT FOR MAX ID IS 4.

When you query the watermark table to get the **last processed value for a specific table**, we expect **only one record** per table.

If you don't limit it to **first row only**, and somehow the table has duplicate records (maybe due to error), it could:

- Cause **pipeline failures**.
- Or produce **ambiguous output** in the Lookup activity (Lookup can return array instead of scalar if >1 row).

That's why we select'first row only ' in second lookup inside for each activity.

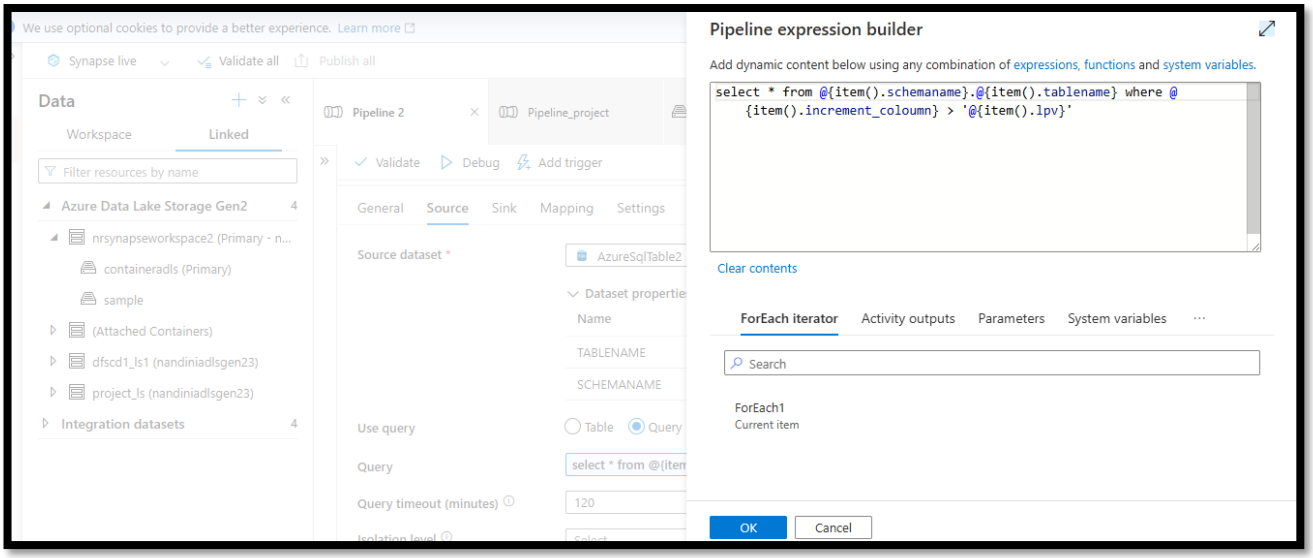
NOW ADD **COPY ACTIVITY** ,SELECT SOURCE DATASET IS SQLDB AND CREATE PARAMETERSAND WRITE A QUERY TO GET MAX INCREMENT CO;OUMN BASED ON LPV .

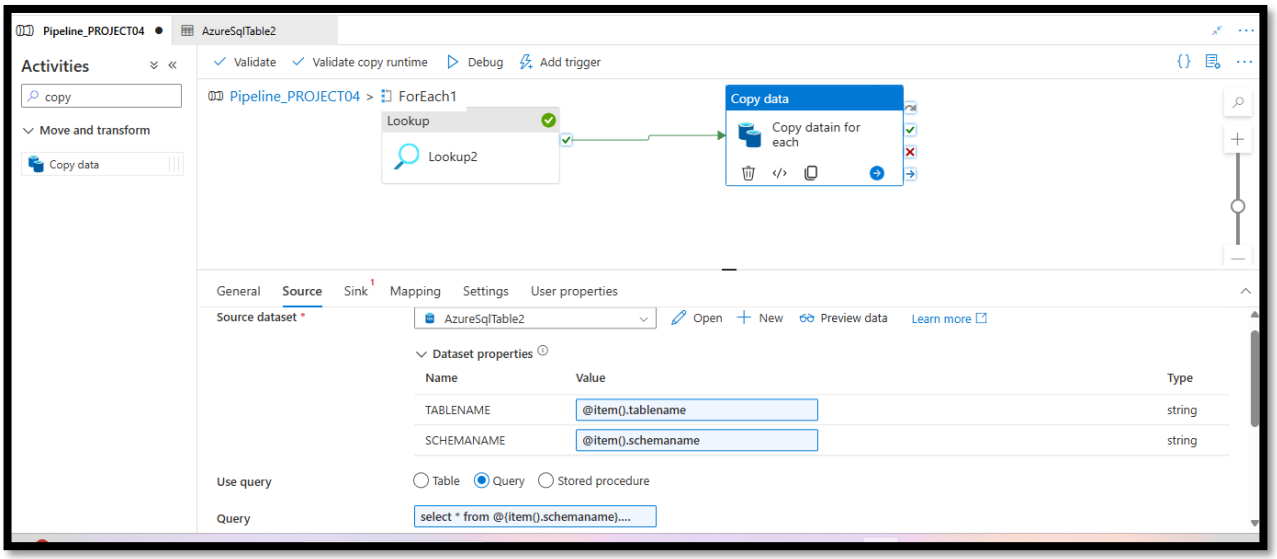
QUERY:"SELECT * FROM @ {ITEM().SCHEMANAME}.@{ITEM().TABLENAME} WHERE @{ITEM().INCREMENT_COLOUMN}>'@{ITEM().LPV}' "

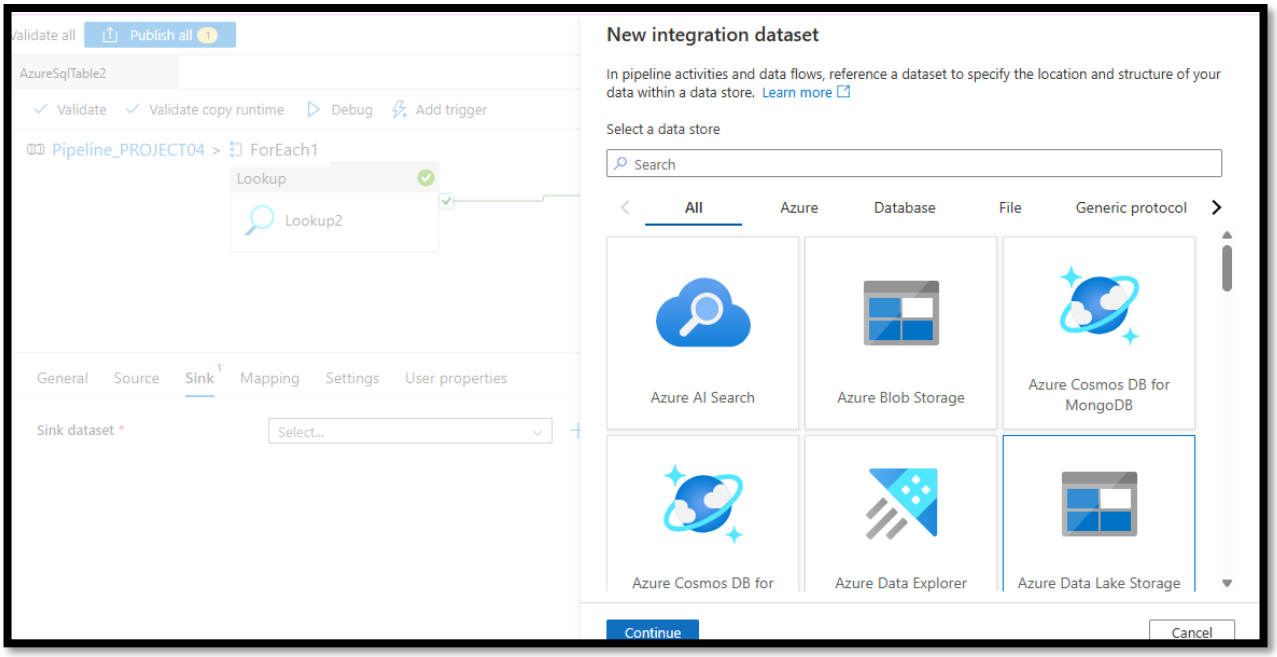
LPV IN 'LPV':> The single quotes **treat the LPV value as a string literal** in SQL.

If LPV is a **datetime or string type**, it **must be enclosed in quotes** for the SQL query to work correctly.

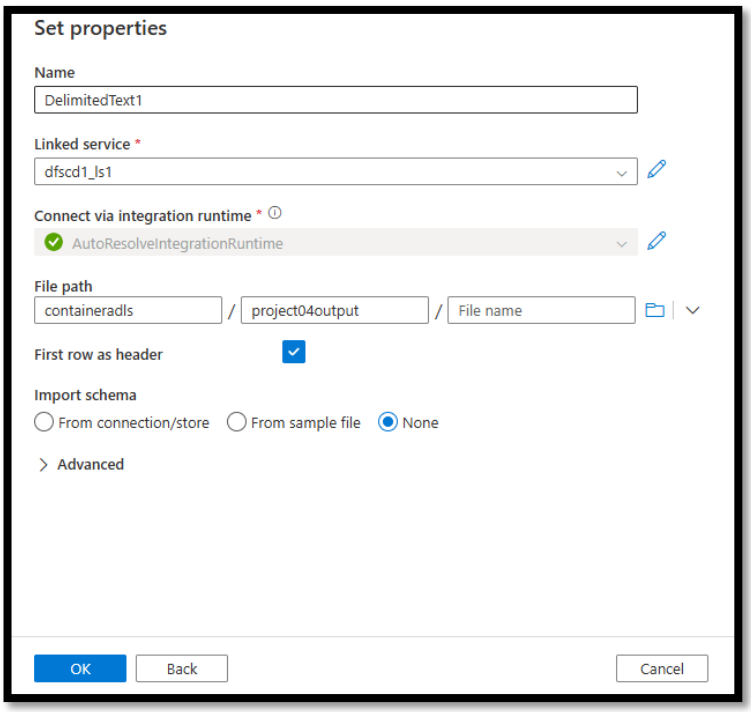
Without quotes, SQL would treat it like a column or keyword, causing errors.

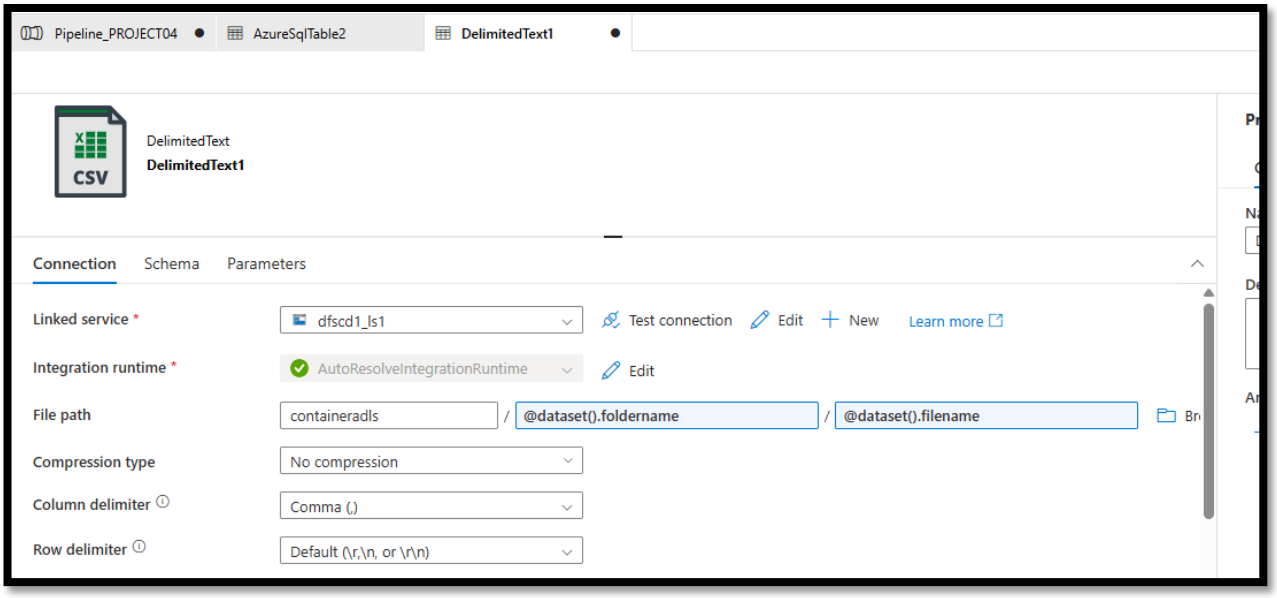

Pipeline expression builder
Add dynamic content below using any combination of expressions, functions and system variables.
select * from @{item().schemaname}.@{item().tablename} where @
{item().increment_coloumn} > '@{item().lpv}'
Clear contents
ForEach iterator
Activity outputs
Parameters
System variables
Search
ForEach1
Current item
OK
Cancel


Pipeline_PROJECT04
AzureSqlTable2
Activities
copy
Move and transform
Copy data
Validate
Validate copy runtime
Debug
Add trigger
Pipeline_PROJECT04 > ForEach1
Lookup
Lookup2
Copy data
Copy datain for each
General
Source
Sink
Mapping
Settings
User properties
Source dataset *
AzureSqlTable2
Open
New
Preview data
Learn more
Dataset properties
Name
Value
Type
TABLENAME
@item().tablename
string
SCHEMANAME
@item().schemaname
string
Use query
Table
Query
Stored procedure
Query
select * from @{item().schemaname}....

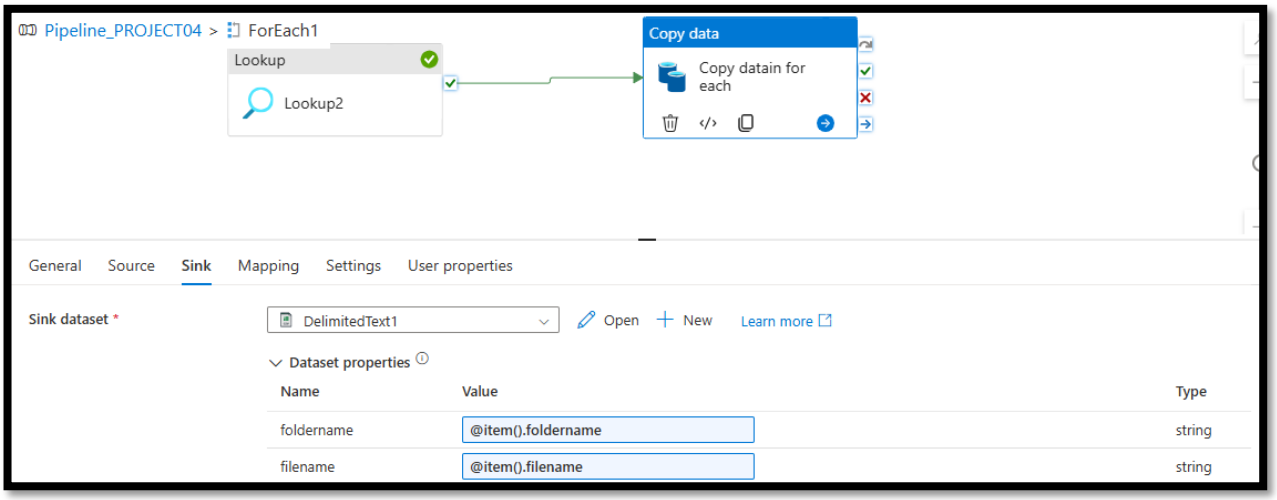


NOW SELECT SINK DATASET:>ADLSGEN2 IN CSV FILE FORMAT.

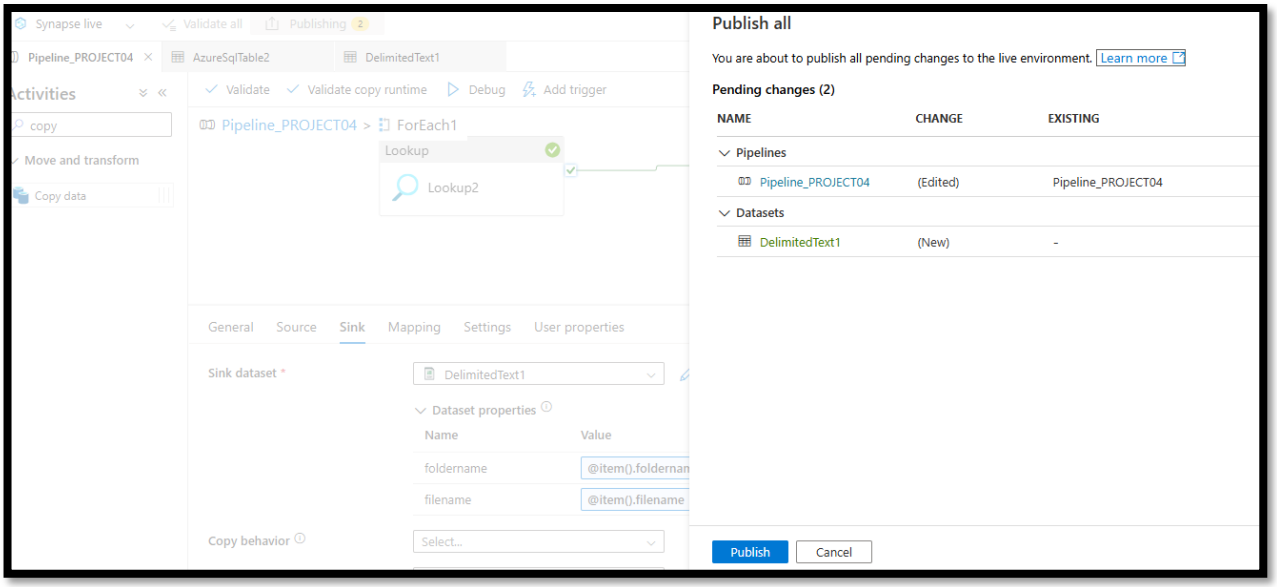




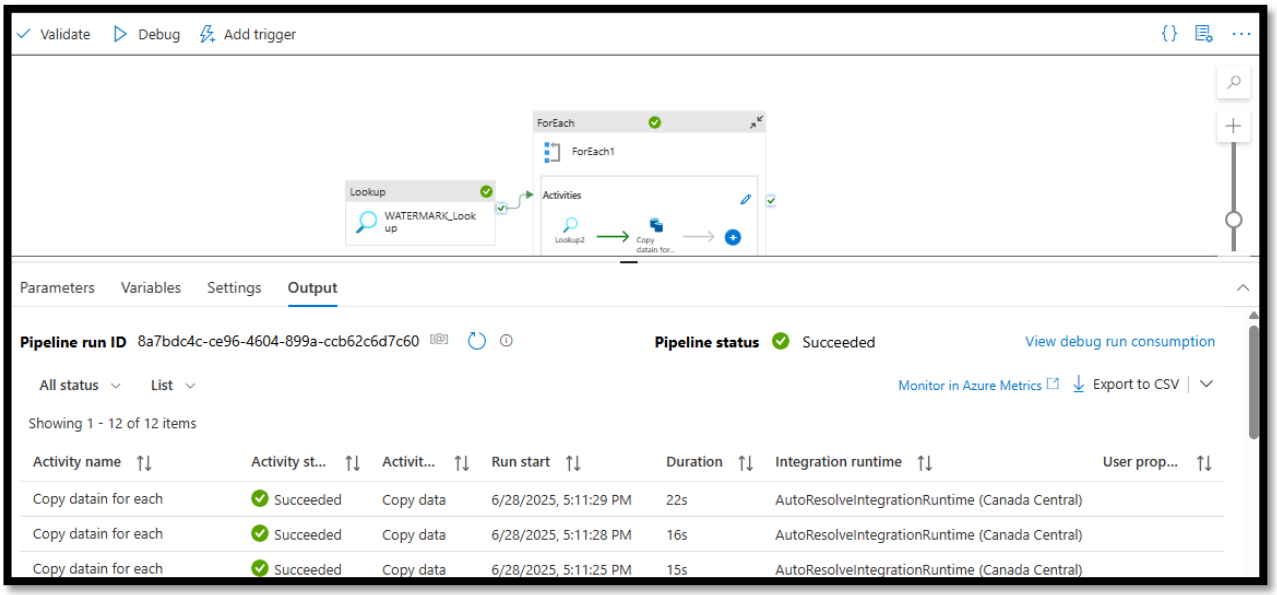
CREATE PARAMETRS, We use @dataset() to dynamically access the parameter values passed into a dataset during pipeline execution.



@item(). Function is used for iteration of rows one by one in an array.
@dataset(). Function is used for dynamically access the parameter values pass into datasets.

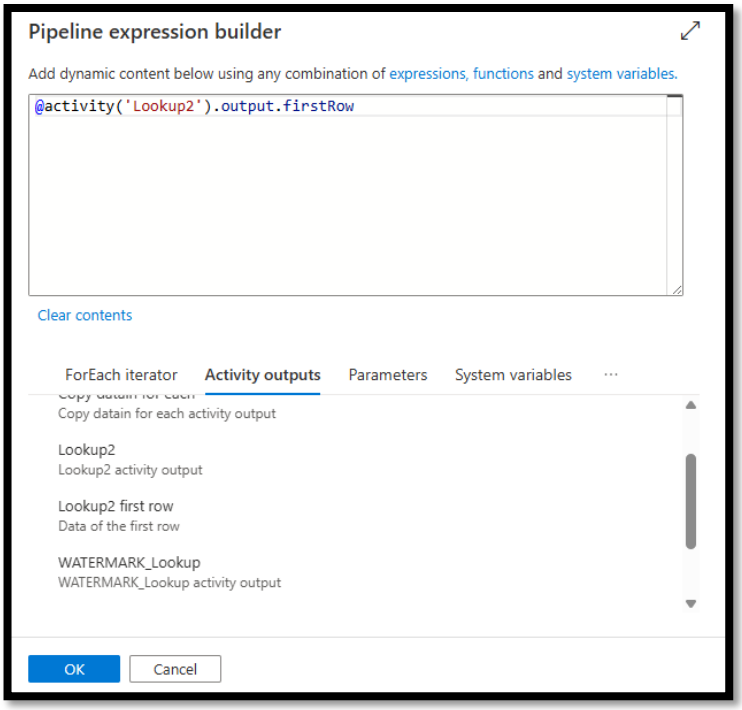


PUBLISH PIPELINE AND DEBUG IT.



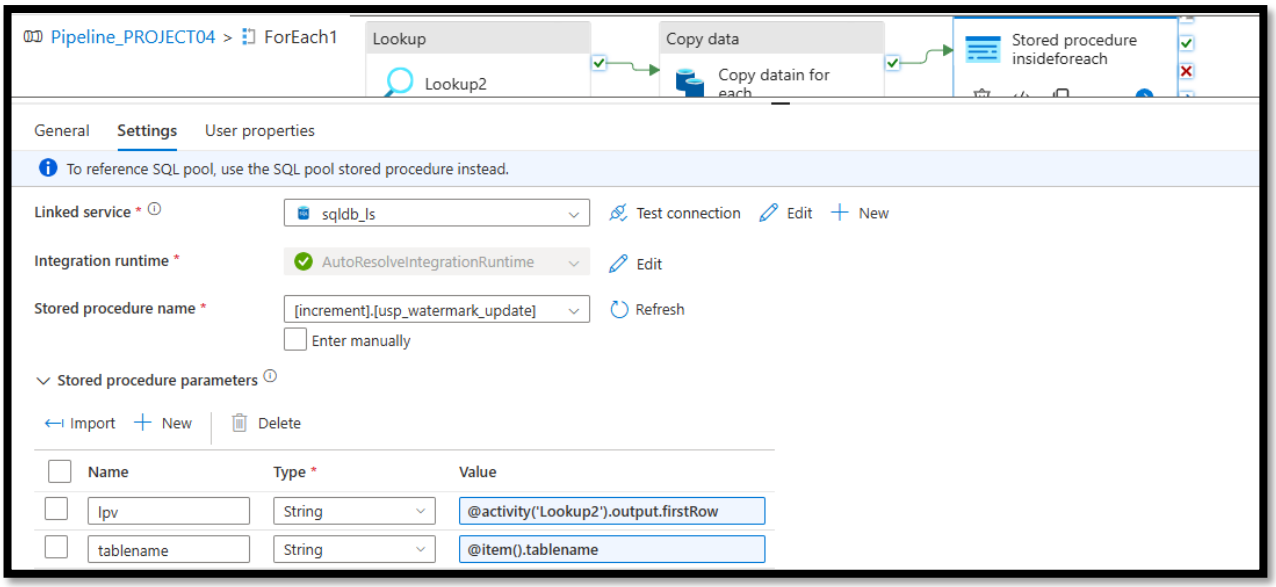
OUTPUT SHOWING NO.OF ROWS READ=NO. OF ROWS COPIED.

THIS IS THE DATA COPIED IN ADLSGEN2 IN CONTAINER'SAMPLE'.

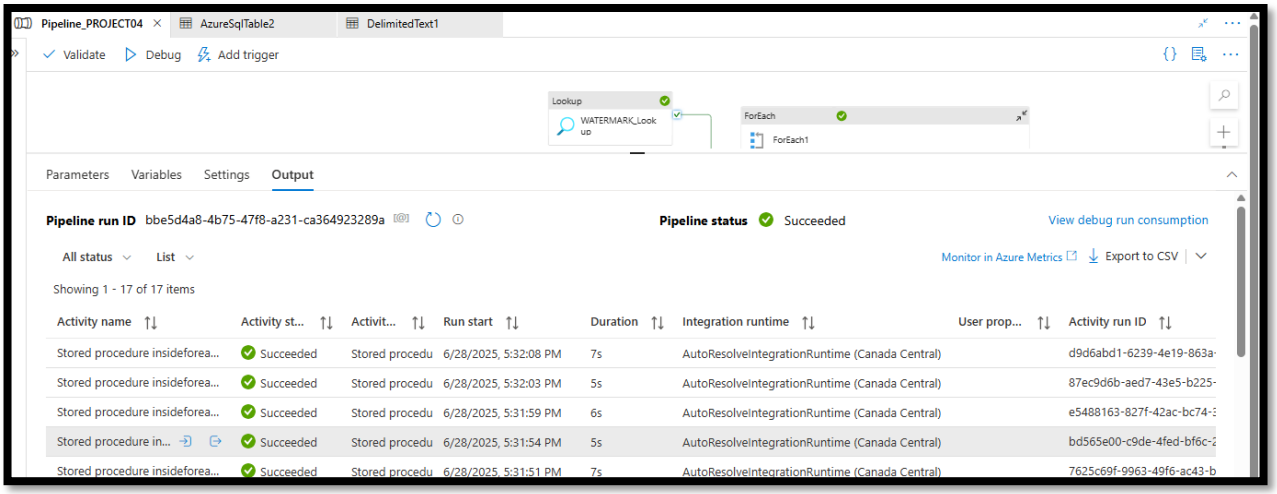


"@activity('Lookup2').output.firstRow"
This expression is used to **fetch the first row of data** returned by a Lookup activity named **Lookup2**.

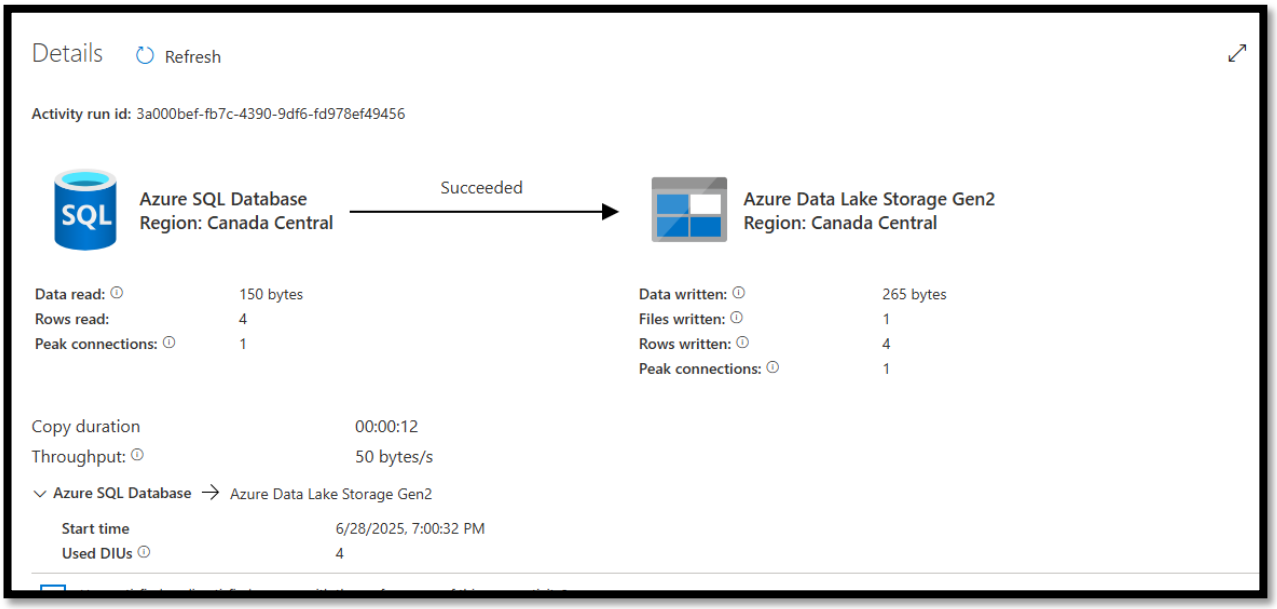
- **Uses the watermark table to track the last processed record for each table**

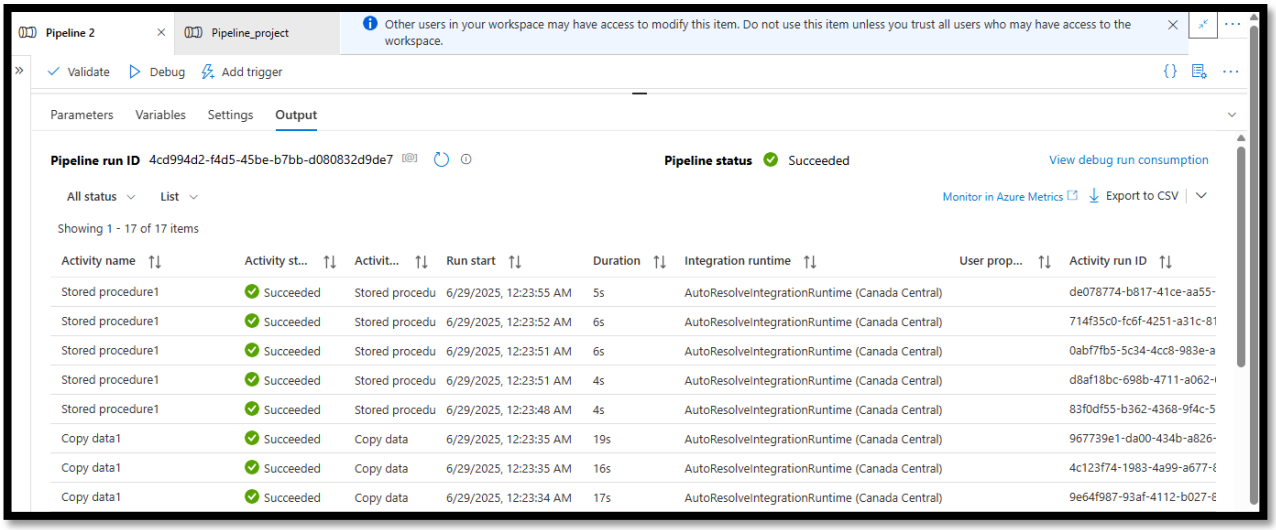


NOW AFTER THAT SELECT A STORED PROCEDURE ACTIVITY,SELECT THE NAME OF THE TABLE AND SELECT IMPORT ,IT imports the stored procedure's metadata and parameters from the SQL database so you can use and configure it in your pipeline easily.

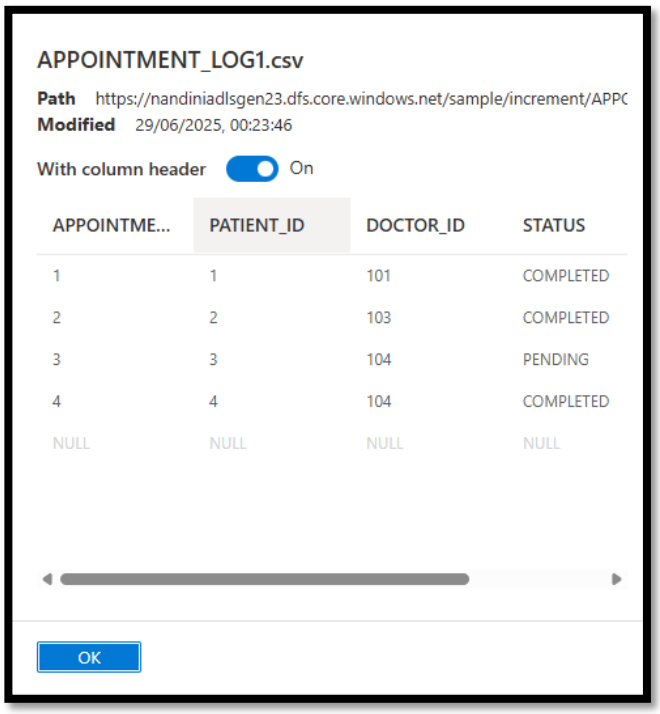


RUN PIPELINE.





- Loads the extracted data to the target destination.



THE FIRST OUTPUT IS ALWAYS FULL LOAD AFTER ANY CHANGE IN TABLE OR WATERMARK TABLE IT WILL SHOW AS AN INCREMENTAL LOAD.

- Ensures that only new data is loaded into the target, preventing reprocessing of already loaded data.

```
6
7   insert into patients_record VALUES('SIREE','F',25)
8   INSERT INTO  increment.APPOINTMENT_LOG1 VALUES(5,105,'COMPLETED','2025-06-25 11:18:00')
9   INSERT INTO INCREMENT.BILLING_TRANSACTION  VALUES (415,5,3200,'DEBIT CARD','2025-06-26 15:30:00')
10
```

I MANUALLY DIDI SOME CHANGES, TO GET AN INCREMENT LOAD.

- Updates the watermark table after successfully loading the new records.

```
57  create table increment.watermark2 (tablename varchar(50),schemaname varchar(50),LPV nvarchar(100),increment_coloumn varchar(50),foldername varchar(50), filename varchar (50))
58  insert into increment.watermark2 values ('PATIENTS_RECORD','increment',4,'PATIENT_id','increment/patients_record','patients_record.csv')
59  insert into increment.watermark2 values ('APPOINTMENT_LOG1','increment',4,'APPOINTMENT_ID','increment/APPOINTMENT_LOG1','APPOINTMENT_LOG1.csv')
60  insert into increment.watermark2 values ('MEDICAL_REPORTS1','increment','2025-07-10T12:00:00Z','generate_date','increment/MEDICAL_REPORTS1','MEDICAL_REPORTS1.csv')
61  insert into increment.watermark2 values ('STAFF_ACTIVITY1','increment','2025-06-28T19:45:00Z','ACTIVITY_time','increment/STAFF_ACTIVITY1','STAFF_ACTIVITY1.csv')
62  insert into increment.watermark2 values ('BILLING_TRANSACTION','increment','2025-07-10T12:30:00Z','TRANSACTION_date','increment/BILLING_TRANSACTION','BILLING_TRANSACTION.csv')
63  SELECT * FROM INCREMENT.WATERMARK2
```

Results | Messages

	tablename	schemaname	LPV	increment_coloumn	foldername	filename
1	PATIENTS_RECORD	increment	4	PATIENT_id	increment/patients_record	patients_record.csv
2	APPOINTMENT_LOG1	increment	4	APPOINTMENT_ID	increment/APPOINTMENT_LOG1	APPOINTMENT_LOG1.csv
3	MEDICAL_REPORTS1	increment	2025-07-10T12:00:00Z	generate_date	increment/MEDICAL_REPORTS1	MEDICAL_REPORTS1.csv
4	STAFF_ACTIVITY1	increment	2025-06-28T19:45:00Z	ACTIVITY_time	increment/STAFF_ACTIVITY1	STAFF_ACTIVITY1.csv
5	BILLING_TRANSACTION	increment	2025-07-10T12:30:00Z	TRANSACTION_date	increment/BILLING_TRANSACTION	BILLING_TRANSACTION.csv

AFTER FIRST PIPELINE RUN , I CHANGED THE LPV=4 WHICH IS MAX VALUE WHICH I GET DURING LOOKUP2 AND LPV OF AS DATETIME FOR OTHER TABLES.

- Fetches the new records based on the last processed value (delta or datetime).

AFTER THIS CHANGE IN WATERMARK TABLE ,RUN PIPELINE AGAIN TO GET INCREMENT LOAD.

OUTPUT :>INCREMENT LOAD OR NEW DATA IS ADDED IN CSV FILE.

APPOINTMENT_LOG1.csv

Path https://nandiniadlsgen23.dfs.core.windows.net/sample/increment/APPC
Modified 29/06/2025, 01:06:25

With column header On

DOCTOR_ID	STATUS	APPOINTMENT_DATE
105	COMPLETED	2025-06-25 11:18:00.0000000

```
66    select MAX(PATIENT_ID) as maxvalue from increment.PATIENTS_RECORD where PATIENT_id>4
67    select Min(PATIENT_ID) as minvalue from increment.PATIENTS_RECORD
68    select MAX(TRANSACTION_DATE) as maxvalue from increment.BILLING_TRANSACTIONS where TRANSACTION_date > '2025-07-10T12:30:00Z'
```

Results Messages

	maxvalue
1	5

IN ADS, WHEN WRITE A QUERY" `select MAX(PATIENT_ID) as maxvalue from increment.PATIENTS_RECORD where PATIENT_id>4" ,IT WILL GIVE ABOVE RESULT.`