

# Incremental Data Load with UPSERT using Azure Data Factory

## 1. Objective

The objective of this implementation is to **load data incrementally into Azure SQL Database** by performing **UPSERT operations** using **Azure Data Factory**. The solution ensures that **new records are inserted** and **existing records are updated** based on key columns, avoiding full data reloads and maintaining data consistency.

## 2. What is UPSERT?

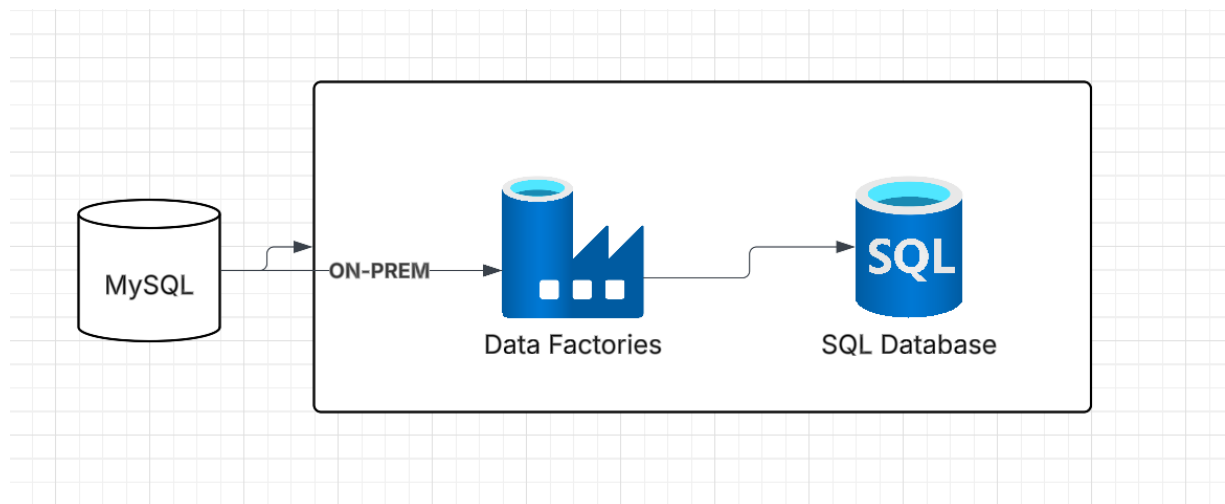
**UPSERT** is a combination of:

**INSERT** → when a record does not exist in the target

**UPDATE** → when a record already exists in the target

In Azure Data Factory, UPSERT is used in **Copy Activity** to efficiently handle **incremental data loads**.

## 3. PROJECT ARCHITECTURE



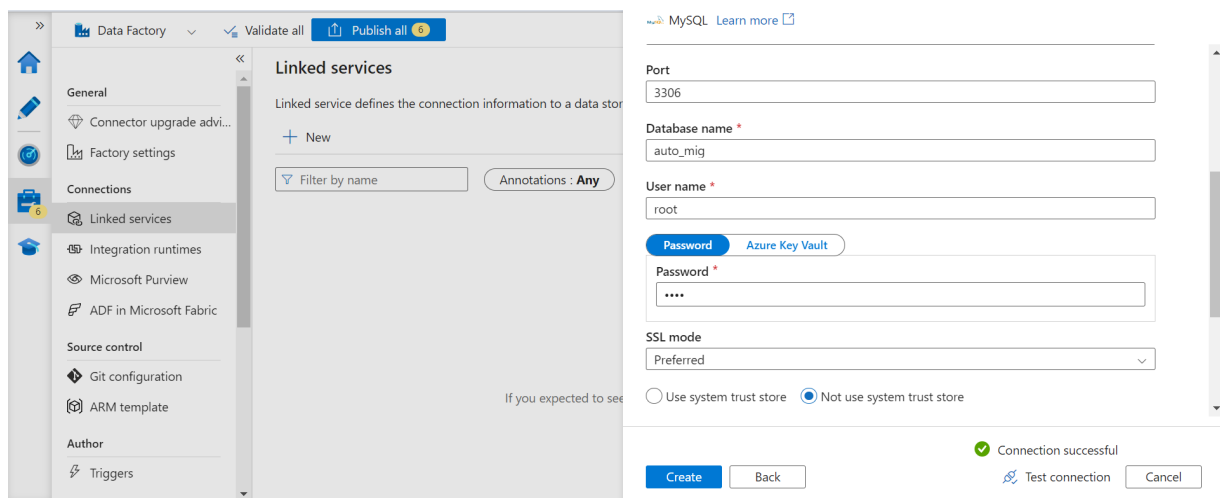
## 4. Resources Required

- Azure Data Factory
- Azure SQL Database
- Source system (MySQL / ADLS)

- Azure Integration Runtime / Self-hosted IR
- Primary / Business key columns

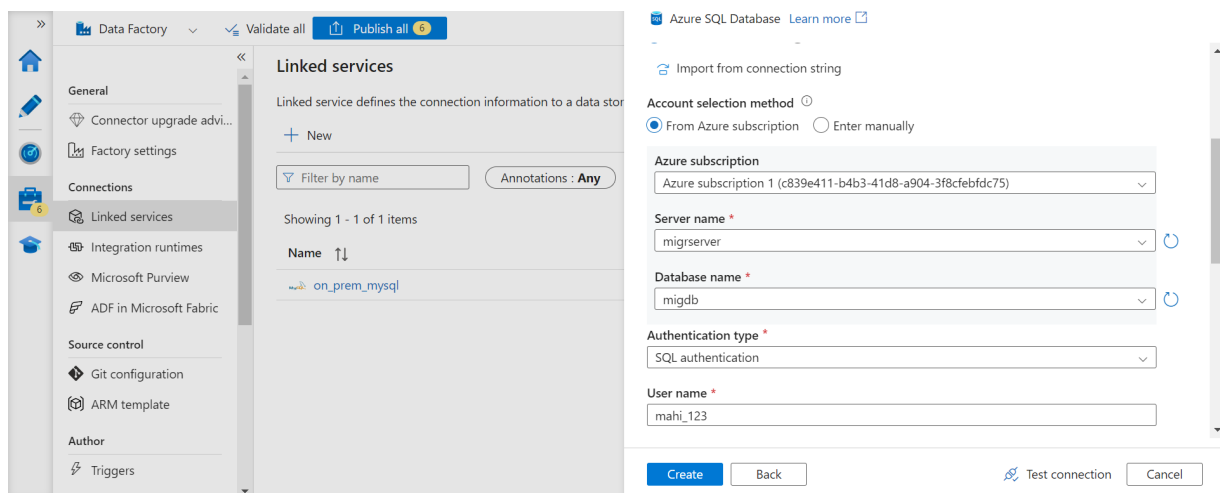
## 5. Creating Link Service

- Mysql to Azure link service
- Azure SQL Database to Azure Link Service



The screenshot shows the Azure Data Factory portal interface. On the left, the 'Connections' menu is open, and 'Linked services' is selected. The main pane displays the 'Linked services' section with a '+ New' button and a search filter. On the right, the 'MySQL' configuration form is shown. It includes fields for 'Port' (3306), 'Database name' (auto\_mig), 'User name' (root), and 'Password' (masked). There are tabs for 'Password' and 'Azure Key Vault'. The 'SSL mode' is set to 'Preferred'. At the bottom, there are buttons for 'Create', 'Back', 'Test connection', and 'Cancel'. A green checkmark indicates 'Connection successful'.

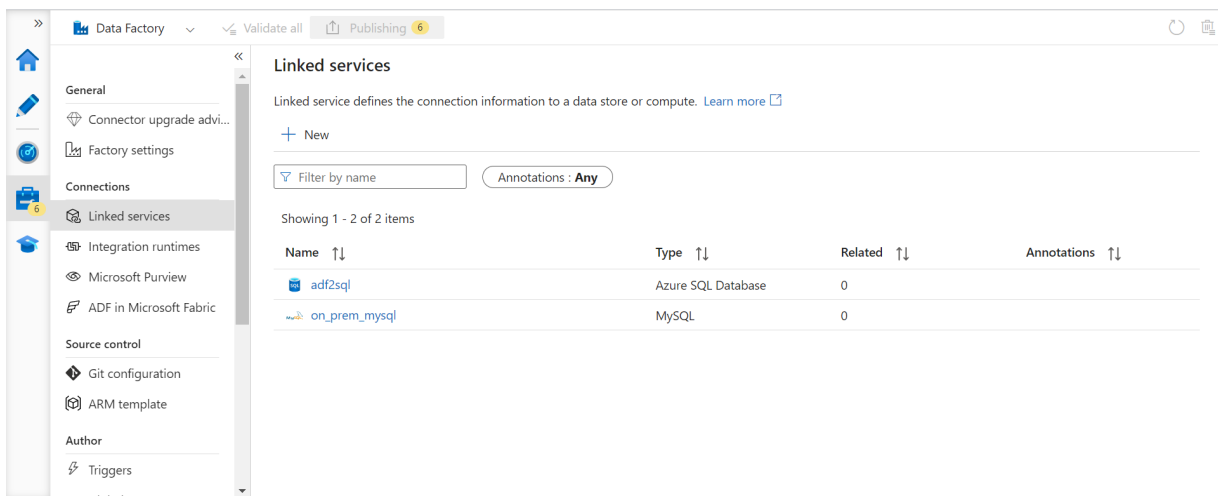
Creating Mysql2adf link service



The screenshot shows the Azure Data Factory portal interface. On the left, the 'Connections' menu is open, and 'Linked services' is selected. The main pane displays the 'Linked services' section with a '+ New' button and a search filter. On the right, the 'Azure SQL Database' configuration form is shown. It includes a section for 'Account selection method' with options 'From Azure subscription' (selected) and 'Enter manually'. Below this, there is a dropdown for 'Azure subscription' (Azure subscription 1 (c839e411-b4b3-41d8-a904-3f8cfebfcd75)). Other fields include 'Server name' (migrserver), 'Database name' (migdb), 'Authentication type' (SQL authentication), and 'User name' (mahi\_123). At the bottom, there are buttons for 'Create', 'Back', 'Test connection', and 'Cancel'.

Creating azure sql to adf link service

Link services



## 6. INTEGRATION RUNTIME SETUP

- In the process of creating the link service for Mysql to azure
- We need to create the self-host integration runtime

## Integration runtime setup

### Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:



#### Azure

Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.



#### Self-Hosted

Use this for running activities in an on-premises / private network


[View more](#) ▾

### External Resources:

You can use an existing self-hosted integration runtime that exists in another resource. This way you can reuse your existing infrastructure where self-hosted integration runtime is setup.



#### Linked Self-Hosted

[Learn more](#) 

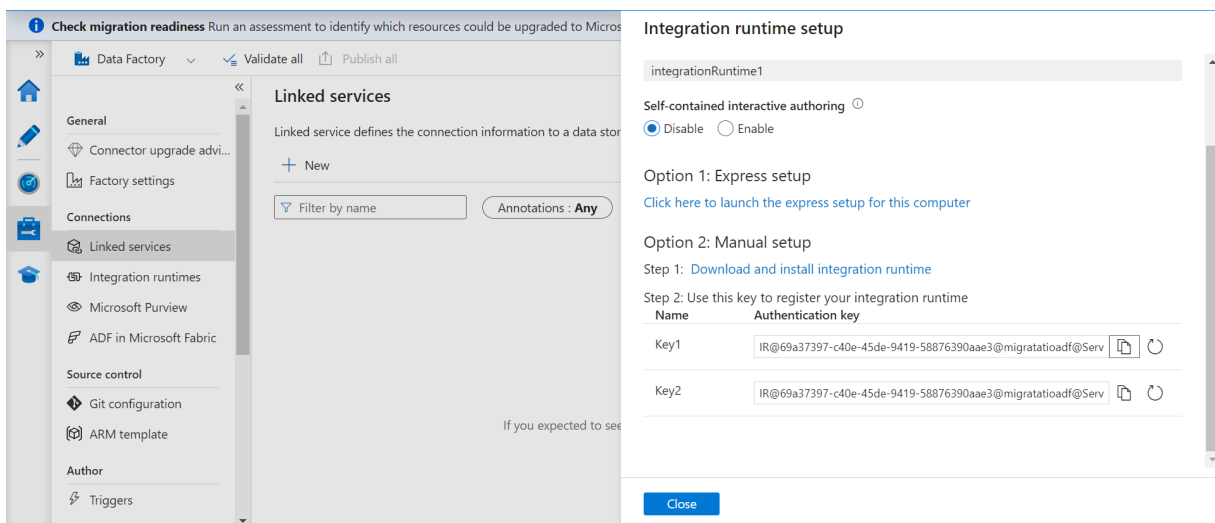
[Continue](#)

[Back](#)

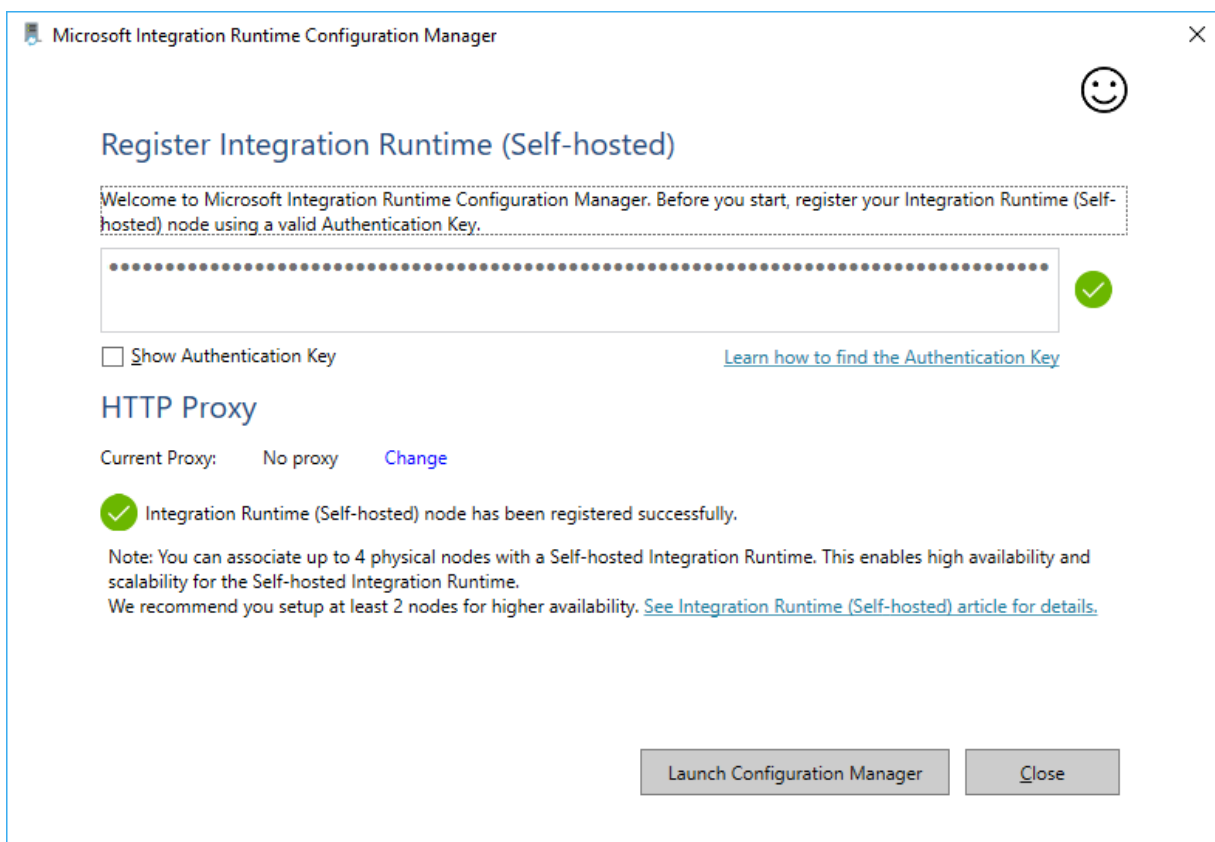
[Cancel](#)

SELF HOST INTEGRATION RUNTIME

➤ We need to download the integration runtime



- After installation of IR copy the key from azure and paste in checkbox of IR

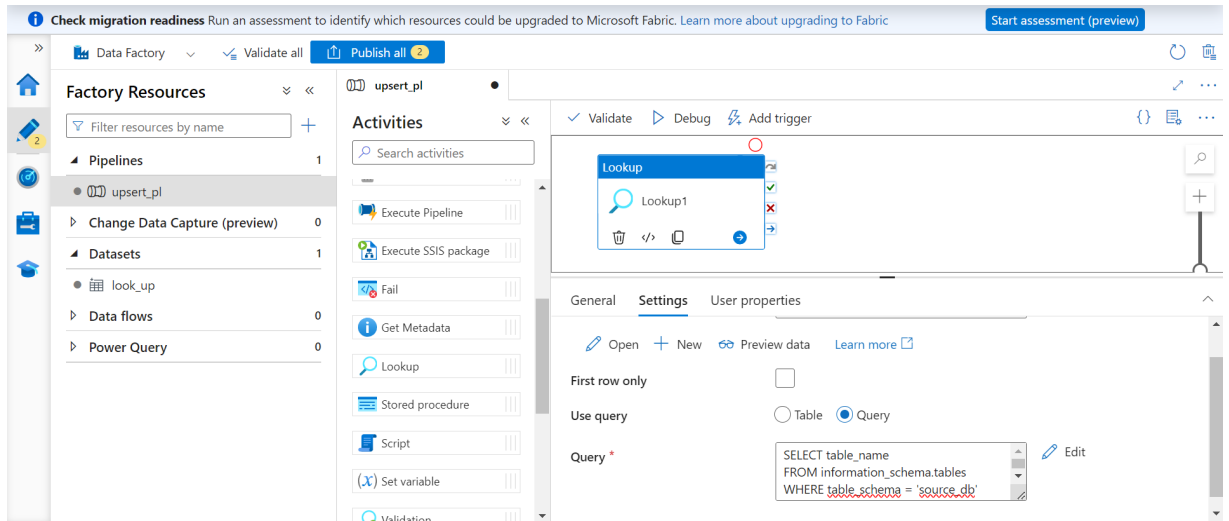


COPY ANY OF THE KEY AND PASTE

## 7. Implementation Steps

## ➤ COPY ACTIVITY

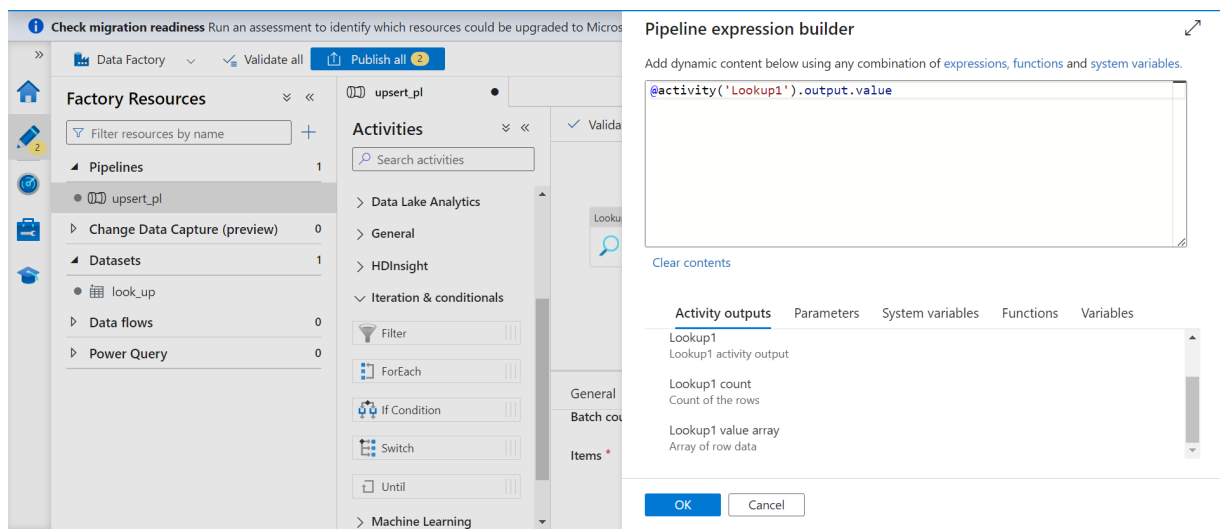
- Drag the copy activity and fill the expression with the query
- The query should display the table\_name in the database



## LOOKUP ACTIVITY

## ➤ FOREACH ACTIVITY

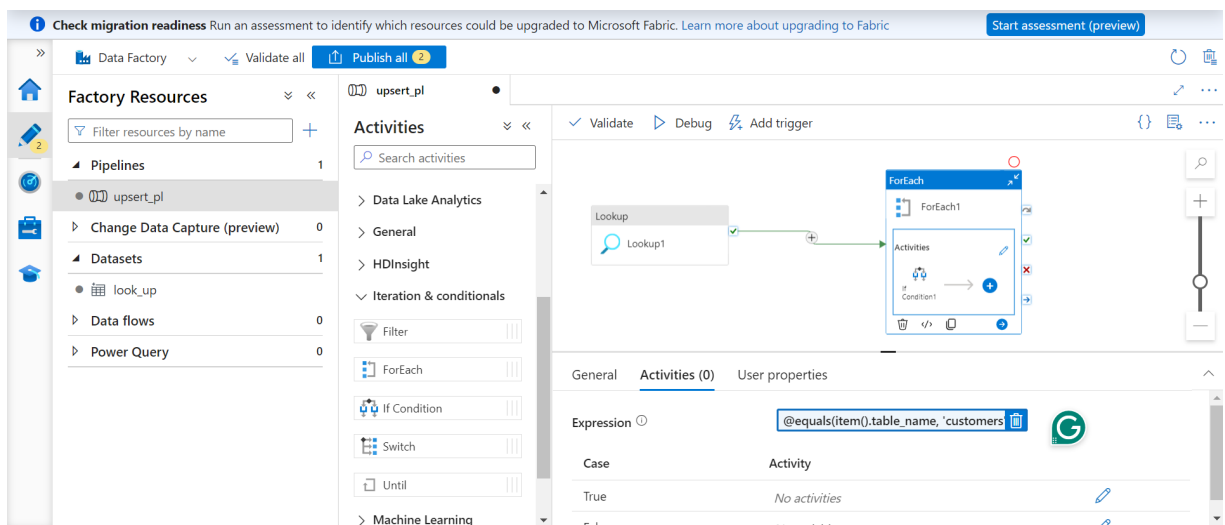
- Write a query to extract the values from the json file created by the lookup



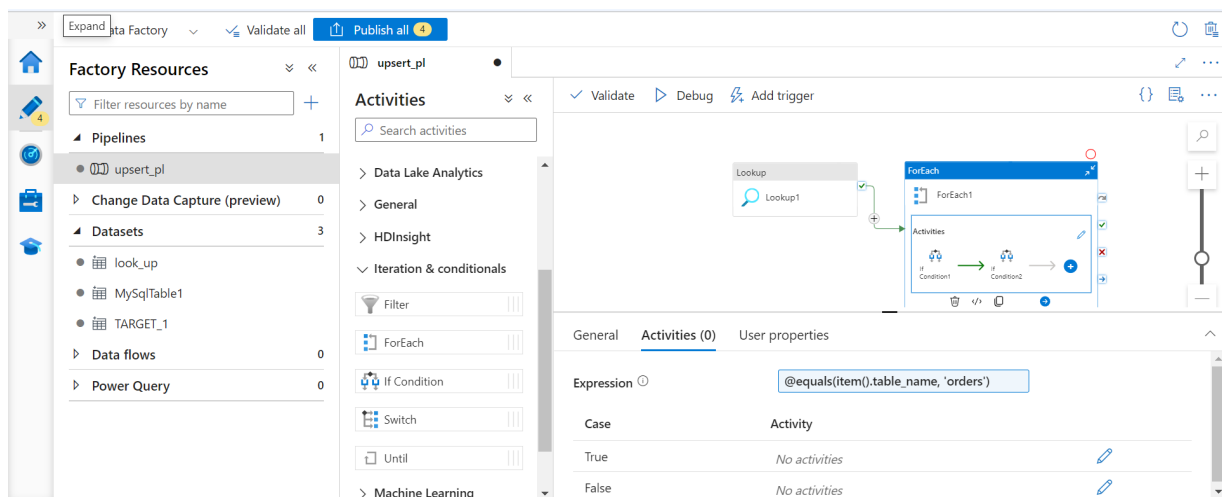
- Inside each for each activity we are going to add if condition activity
- Here we need to have copy activity inside every if condition
- Here the problem is if there are more tables then we need more if conditions and copy for each copy activity it creates two datasets

## ➤ IF CONDITION

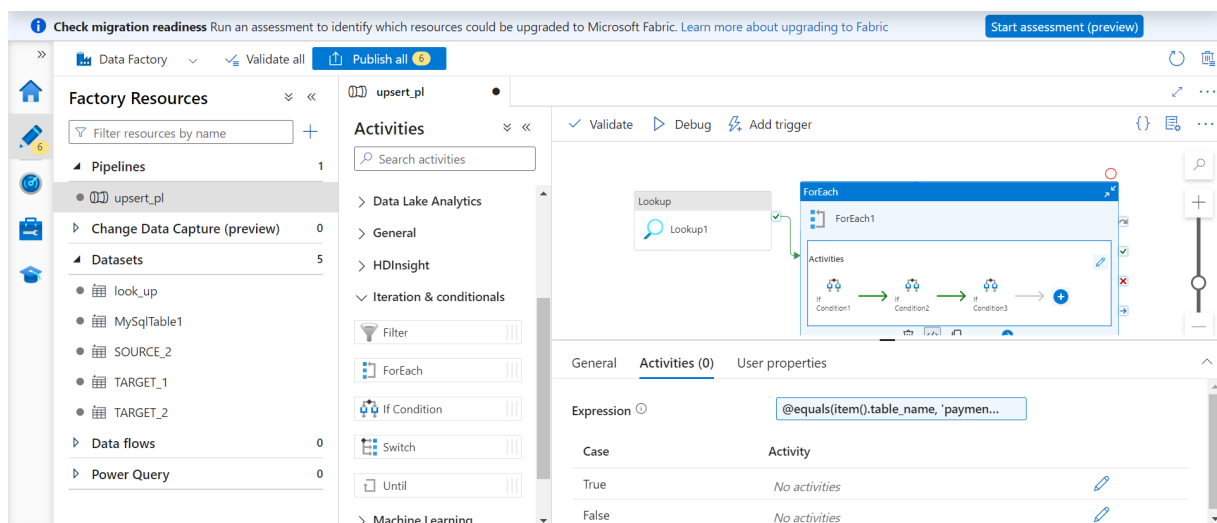
- In your project there are three tables so we need have three if conditions
- For each if condition we are going to use equal method if true then the copy activity is going execute
- The three tables are customers,orders,payments.



FIRST IF CONDITION FOR CUSTOMERS



SECOND IF CONDITION FOR ORDERS



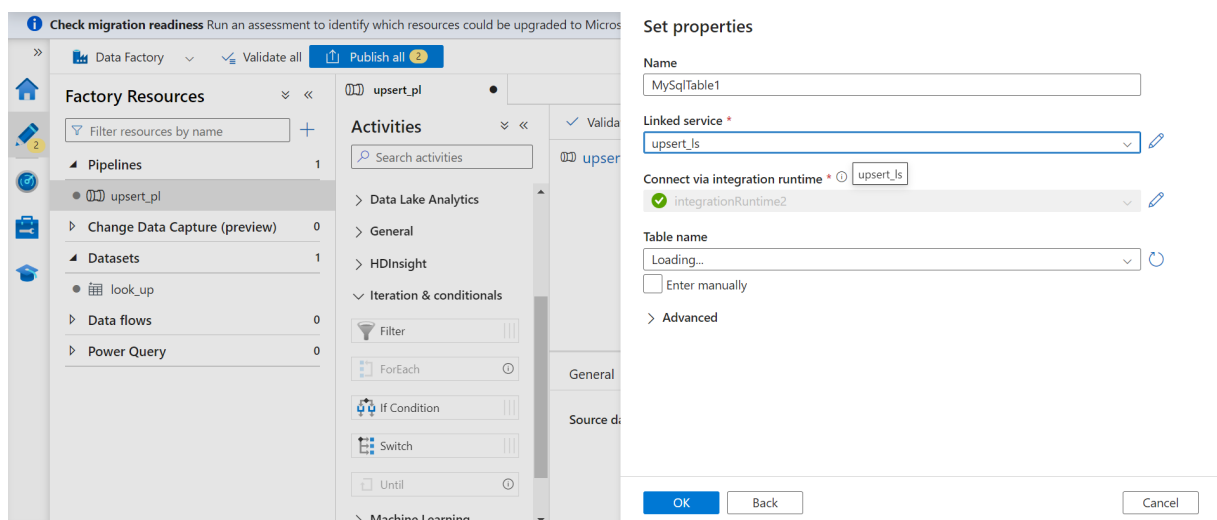
### THIRD IF CONDITION FOR PAYMENTS

## ➤ COPY ACTIVITY

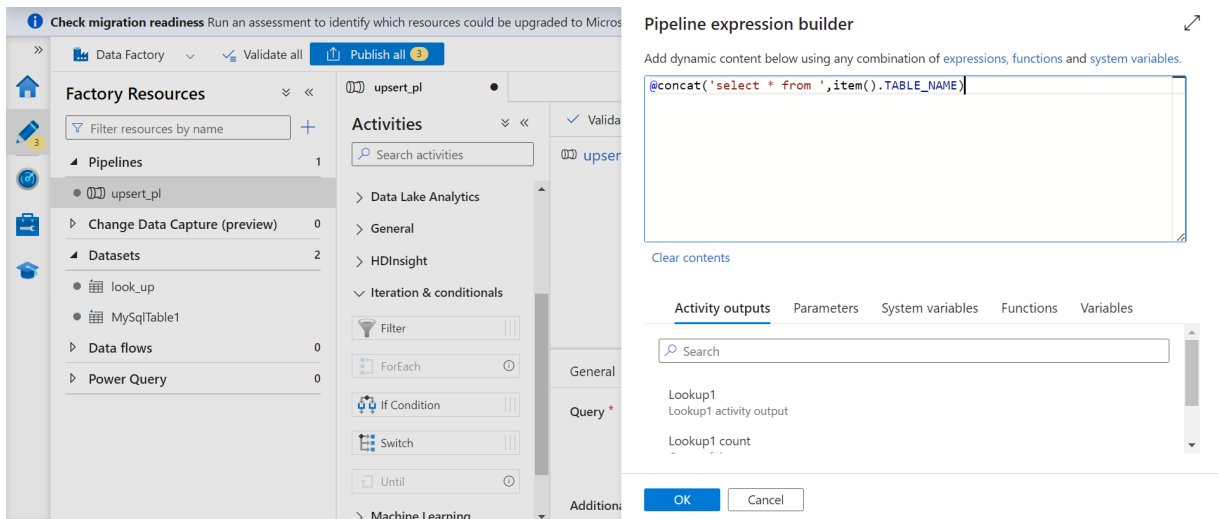
- In each copy activity I am going to create two datasets on for source and other for sink

### ❖ SOURCE-1

- In the source we are going to write code by using the method concat to write a query
- For every source in copy activity we are going to perform same



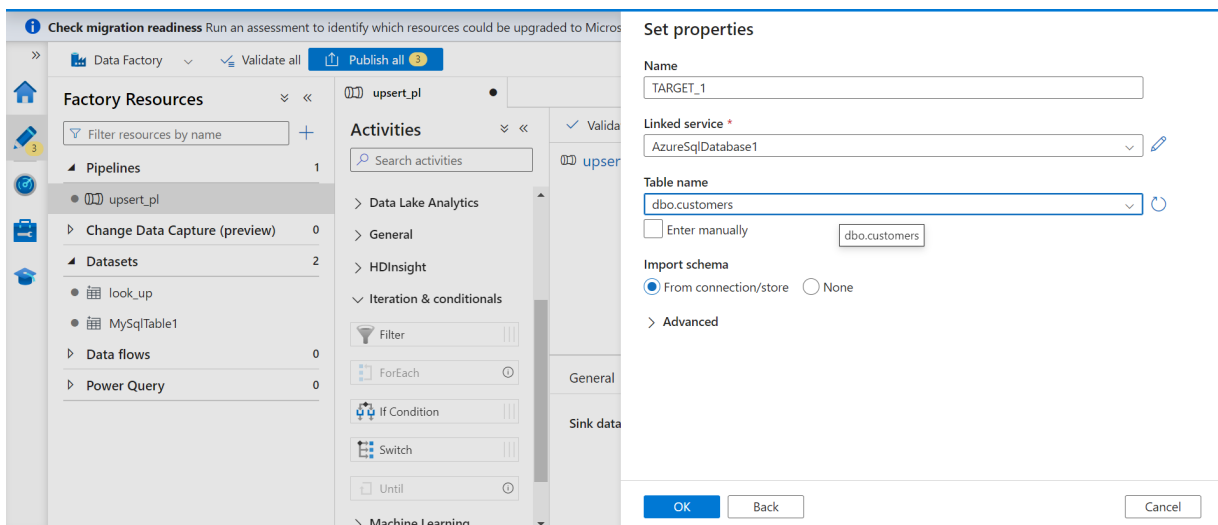
### SOURCE DATASET



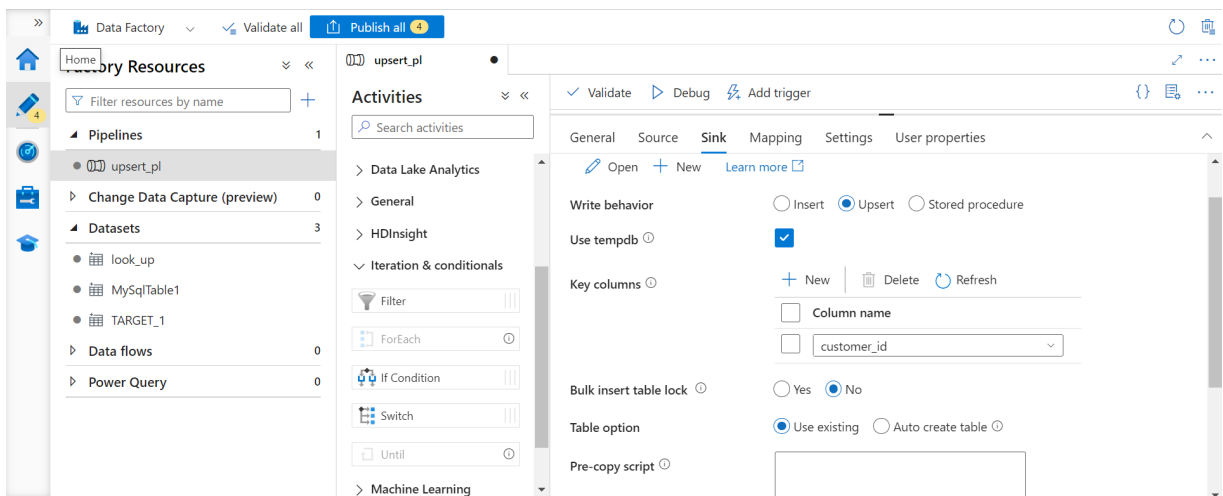
## QUERY TO CALL THE TABLE

### ❖ COPY-1

- In the copy we need to follow some rules and this same for other copy activity also
- First you need to create tables in the azure sql database and need assign primary key
- During creation of sink dataset we need to select the table
- In the sink part we need to select the upsert and use existing table
- And we need enter the primary key in the key column area

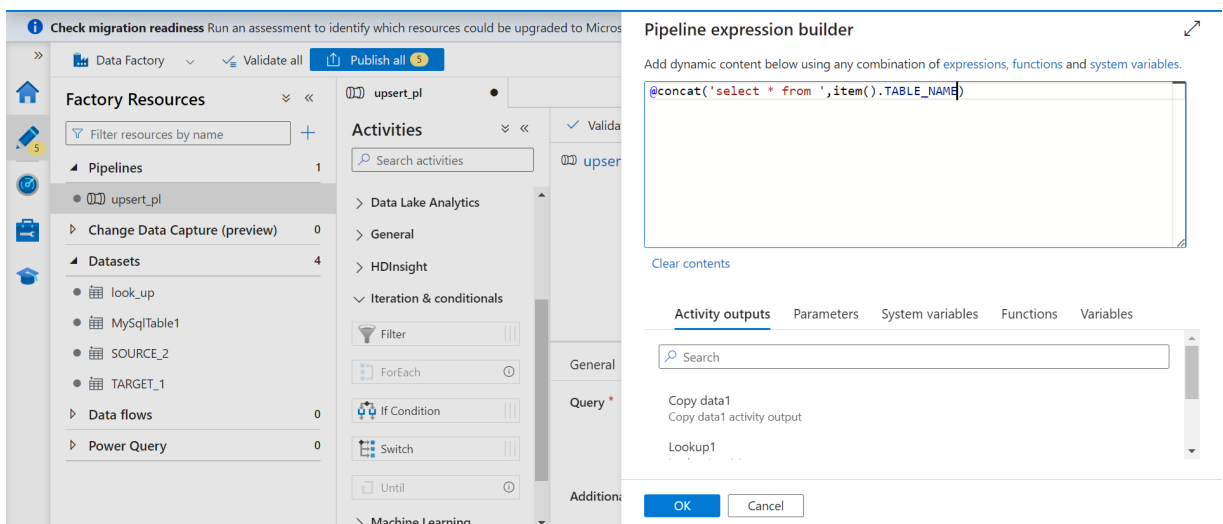
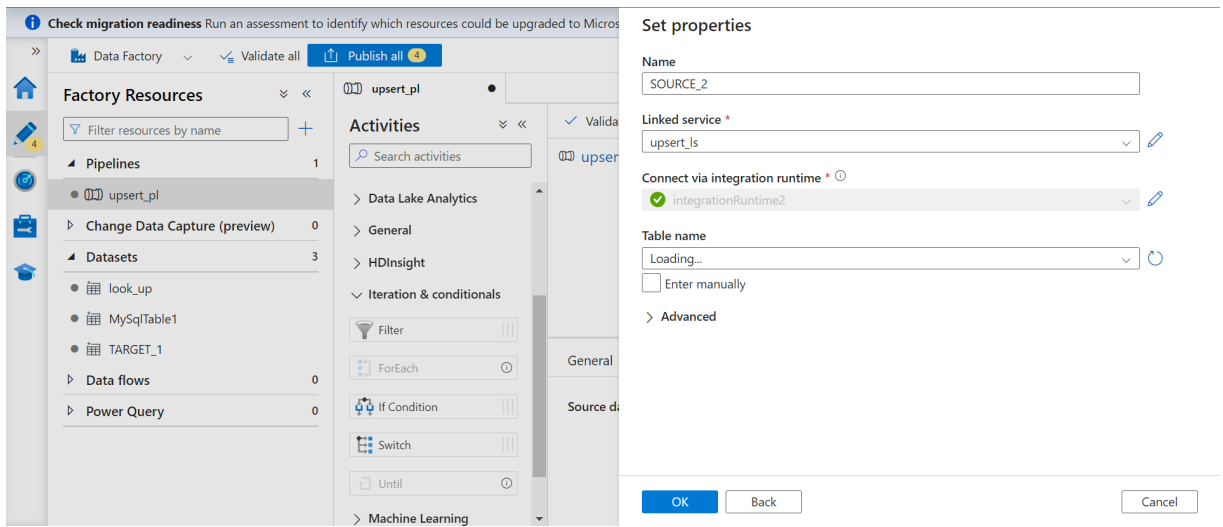


## SELECTING THE TABLE



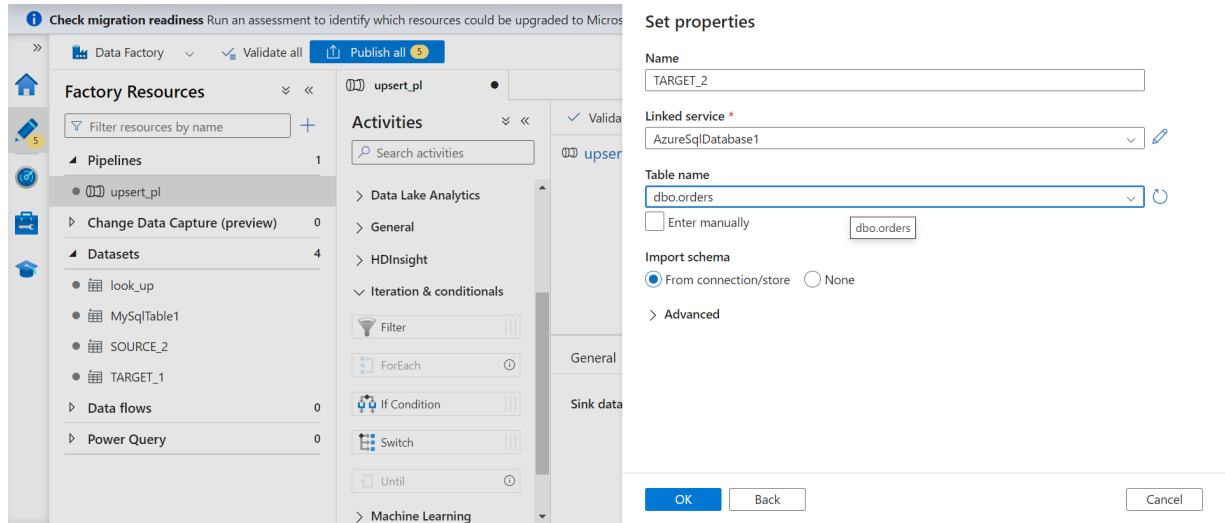
## SELECTING THE UPSERT

## SOURCE-2

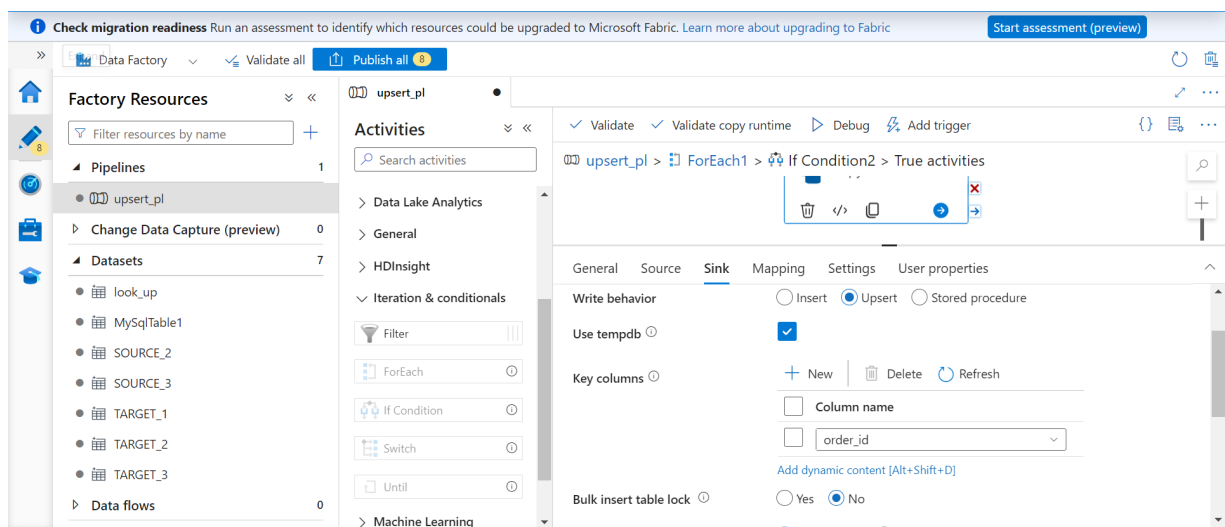


## SOURCE FOR ORDERS TABLE

## ➤ COPY-2



## SELECTING THE ORDERS TABLE



## SELECTING UPSERT, USE EXISTING

## ➤ SOURCE-3

**Factory Resources**

- Pipelines: 1
  - upsert\_pl
- Datasets: 5
  - look\_up
  - MySQLTable1
  - SOURCE\_2
  - TARGET\_1
  - TARGET\_2
- Data flows: 0
- Power Query: 0

**Activities**

- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
  - Filter
  - ForEach
  - If Condition
  - Switch
  - Until
- Machine Learning

**Set properties**

Name: SOURCE\_3

Linked service \*: upsert\_ls

Connect via integration runtime \*: integrationRuntime2

Table name: Loading...

☐ Enter manually

**Advanced**

OK Back Cancel

## COPY-3

**Factory Resources**

- Pipelines: 1
  - upsert\_pl
- Datasets: 6
  - look\_up
  - MySQLTable1
  - SOURCE\_2
  - SOURCE\_3
  - TARGET\_1
  - TARGET\_2
- Data flows: 0
- Power Query: 0

**Activities**

- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
  - Filter
  - ForEach
  - If Condition
  - Switch
  - Until
- Machine Learning

**Set properties**

Name: TARGET\_3

Linked service \*: AzureSqlDatabase1

Table name: dbo.payments

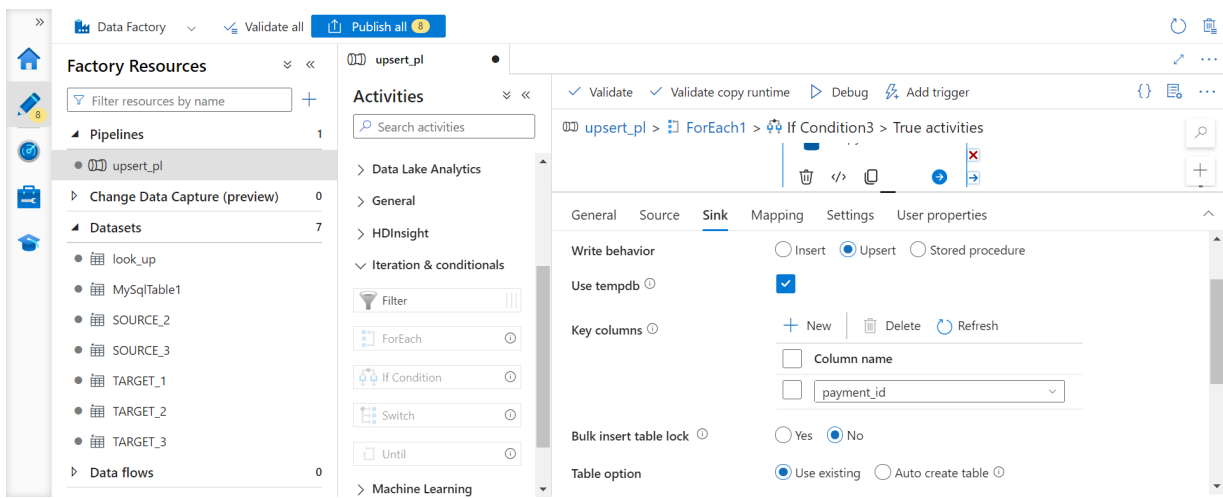
☐ Enter manually

Import schema: ☒ From connection/store ☐ None

**Advanced**

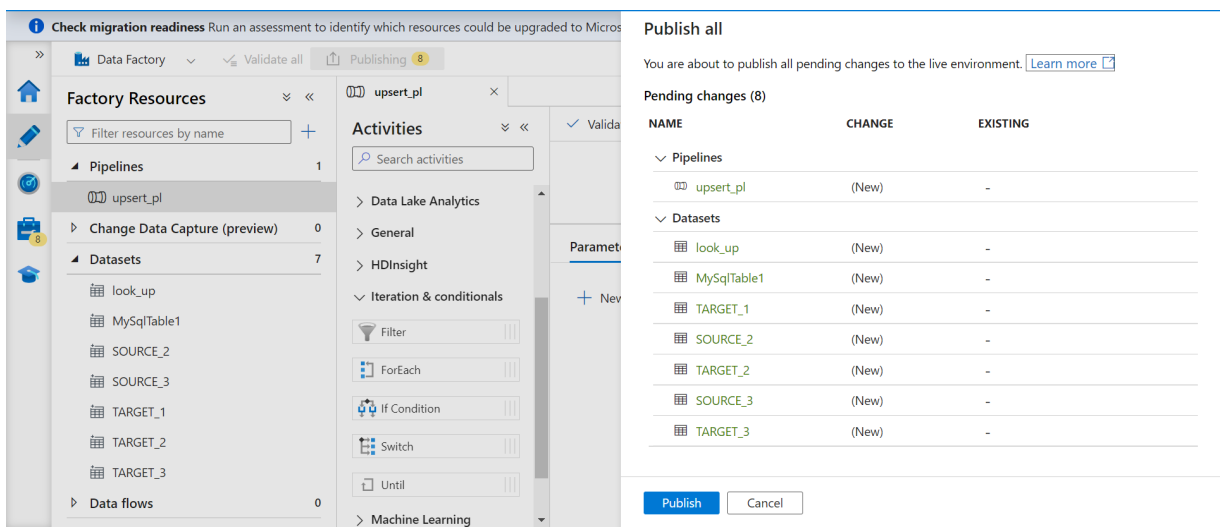
OK Back Cancel

SELECTING THE LAST TABLE PAYMENTS



## SELECTING THE UPSERT,USE EXISTING

## PUBLISH AND TRIGGER



Check migration readiness

Run an assessment to identify which resources could be upgraded to Microsoft Fabric.

Validate all

Publish all

Factory Resources

Filter resources by name

Pipelines1

Change Data Capture (preview)0

Datasets7

look\_up

MySQLTable1

SOURCE\_2

SOURCE\_3

TARGET\_1

TARGET\_2

TARGET\_3

Data flows0

Activities

Search activities

Data Lake Analytics

General

HDInsight

Iteration & conditionals

Filter

ForEach

If Condition

Switch

Until

Machine Learning

Parameters

+ New

Pipeline run

Trigger pipeline now using last published configuration.

Parameters

Name	Type	Value
No records found		

OK

Cancel

## EXCUTION

Check migration readiness

Run an assessment to identify which resources could be upgraded to Microsoft Fabric.

[Learn more about upgrading to Fabric](#)

Start assessment (preview)

Dashboards

Runs

Pipeline runs

Trigger runs

Change Data Capture (previ...

Runtimes & sessions

Integration runtimes

Data flow debug

Notifications

Alerts & metrics

All pipeline runs > upsert\_pl - Activity runs

Rerun

Cancel

Refresh

Update pipeline

List

Gantt

Lookup

Lookup1

ForEach

ForEach1

Activities

If Condition1

If Condition2

If Condition3

Activity runs

Pipeline run ID 6c67b782-608a-4a8c-a4b4-14be354876ac

All status

List

Monitor in Azure Metrics

Export to CSV

Check migration readiness

Run an assessment to identify which resources could be upgraded to Microsoft Fabric.

[Learn more about upgrading to Fabric](#)

Start assessment (preview)

Dashboards

Runs

Pipeline runs

Trigger runs

Change Data Capture (previ...

Runtimes & sessions

Integration runtimes

Data flow debug

Notifications

Alerts & metrics

All pipeline runs > upsert\_pl - Activity runs

Rerun

Cancel

Refresh

Update pipeline

List

Gantt

Activity runs

Pipeline run ID 6c67b782-608a-4a8c-a4b4-14be354876ac

All status

List

Monitor in Azure Metrics

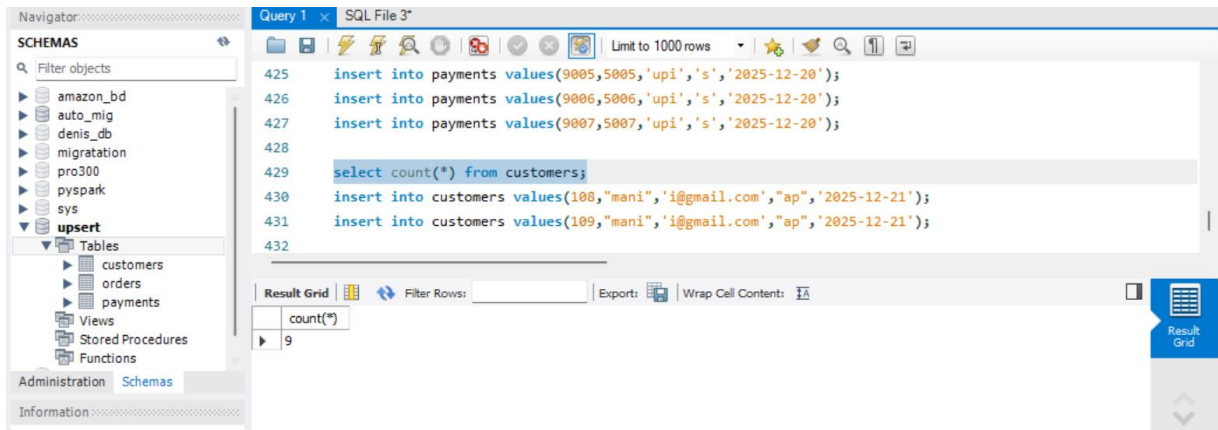
Export to CSV

Showing 1 - 14 items

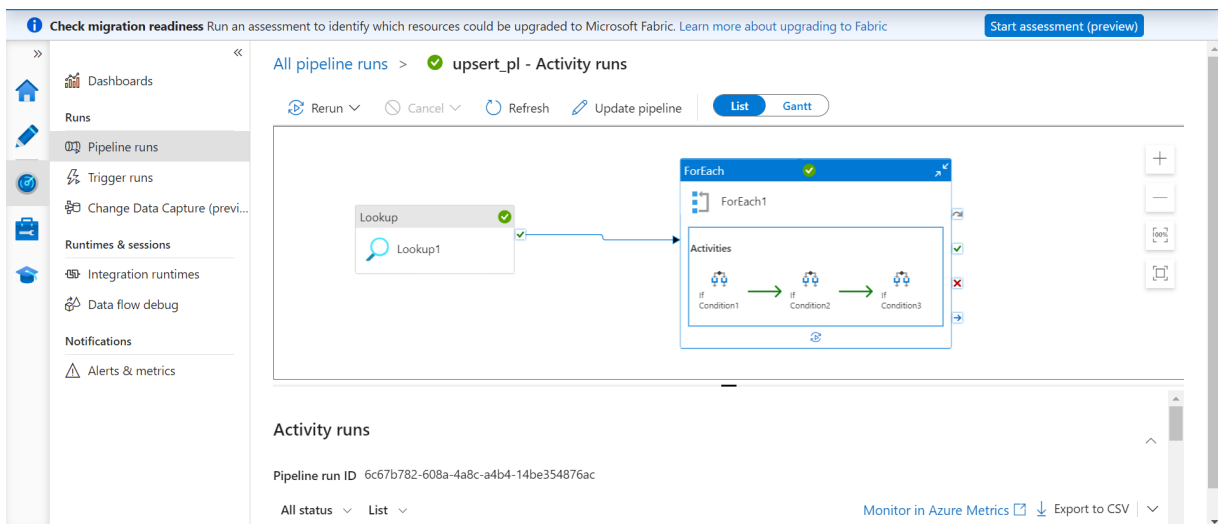
Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
Lookup1	Succeeded	Lookup	12/21/2025, 11:42:15 PM	29s	integrationRuntime2
ForEach1	Succeeded	ForEach	12/21/2025, 11:42:45 PM	1m 3s	
If Condition1	Succeeded	If Condition	12/21/2025, 11:42:45 PM	Less than 1s	
If Condition1	Succeeded	If Condition	12/21/2025, 11:42:45 PM	54s	
If Condition1	Succeeded	If Condition	12/21/2025, 11:42:45 PM	Less than 1s	
Copy data1	Succeeded	Copy data	12/21/2025, 11:42:46 PM	53s	integrationRuntime2

## ➤ INSERTION

- In Mysql I am going to insert some of the rows
- And I will rerun the pipeline
- If I look at the already existing table in my there 7 rows in the customers
- And I inserted two rows into customer and rerun the pipeline after the execution the number of columns should be 9



## INSERTION IN MYSQL



## RERUN THE PIPELINE

migdb (kalvapallisenapathireddy@g...)

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

Tables

- dbo.addresses
- dbo.copy1
- dbo.customers
- dbo.departments
- dbo.employees
- dbo.order\_items
- dbo.orders
- dbo.payments

Query 1 × Query 2 × Query 3 × Query 4 ×

Run Cancel query Save query Export data as Show only Editor

```
1 SELECT count(*) FROM [dbo].[customers]
2
3
```

Results Messages

Search to filter items...

9

Query succeeded | 1s

**TWO ROWS SUCCESSFULLY INSERTED**

## ➤ UPDATE

- In the customer table I want to update ap in city to amaravati
- I have performed update operation in mysql

```
select count(*) from customers;
insert into customers values(108,"mani",'i@gmail.com',"ap",'2025-12-21');
insert into customers values(109,"mani",'i@gmail.com',"ap",'2025-12-21');
```

```
update customers set city = 'Amaravati' where city = 'ap';
```

- Going to rerun the pipeline

Check migration readiness Run an assessment to identify which resources could be upgraded to Microsoft Fabric. [Learn more about upgrading to Fabric](#) Start assessment (preview)

Dashboards

Runs

- Pipeline runs
- Trigger runs
- Change Data Capture (previ...

Runtimes & sessions

- Integration runtimes
- Data flow debug

Notifications

- Alerts & metrics

All pipeline runs > upsert\_pl - Activity runs

Rerun Cancel Refresh Update pipeline List Gantt

Lookup

Lookup1

ForEach

ForEach1

Activities

If Condition1 → If Condition2 → If Condition3

Activity runs

Pipeline run ID 6c67b782-608a-4a8c-a4b4-14be354876ac

All status List Monitor in Azure Metrics Export to CSV

- In the Azure Sql database

The screenshot displays the Azure Data Studio interface. On the left, the 'Object Explorer' pane shows a list of tables under the 'dbo' schema: addresses, copy1, customers, departments, employees, order\_items, orders, and payments. A message box at the top left states: 'Showing limited object explorer here. For full capability please click here to open Azure Data Studio.' The central pane contains a SQL query editor with the following code:

```
1 SELECT count(*) FROM [dbo].[customers]
2 select * from [dbo].[customers]
3
```

Below the query editor, the 'Results' pane shows a table with 5 columns and 3 rows of data:

107	mani	i@gmail.com	Amaravti	2025-12
108	mani	i@gmail.com	Amaravti	2025-12
109	mani	i@gmail.com	Amaravti	2025-12

At the bottom, a status bar indicates 'Query succeeded | 1s'.

## 8. Conclusion

- ◆ UPSERT is suitable when the number of tables to migrate is **less** and **incremental loading** is required.
- ◆ It helps in **updating existing records** and **inserting new records** without full reload.
- ◆ Azure Data Factory does **not support dynamic UPSERT key columns**, which limits scalability.
- ◆ As the number of tables increases, the **number of datasets and If Conditions also increases**.
- ◆ This increases **pipeline complexity and maintenance effort** for large-scale migrations.
- ◆ In this project, UPSERT was used because the **table count was limited** and **incremental load was a business requirement**.