

STATEMENT 4

Create a pipeline to read the Customer table data from SQL and Customer Address data from CSV, join both of them, and then save the result where customer id > 1000 & Customer id < 2000 in ascending order as a Parquet file.

In this statement/problem, we need to combine data from two different datasets, customer and customer address. Records where customer id is between 1000 and 2000 will be joined and copied into parquet file.

Create Datasets:

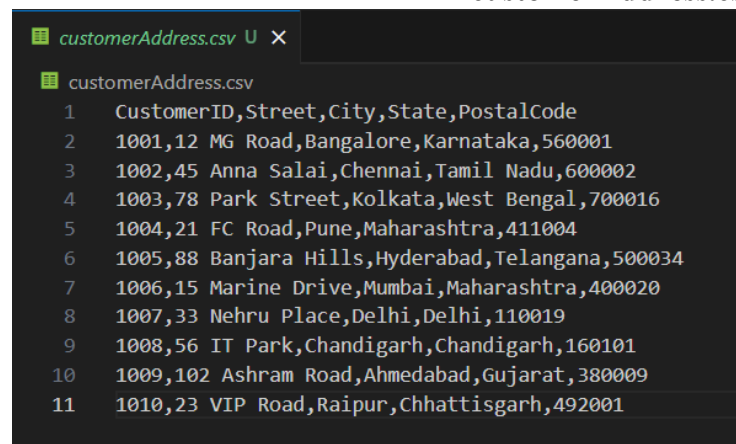
1. Create Customer Table

We had created customer table for statement 1 and statement 2. The same table will be used in this statement as well.

The SQL scripts for creating and inserting data into the table can be found in **customerTable.sql** and **insertCustomers.sql**.

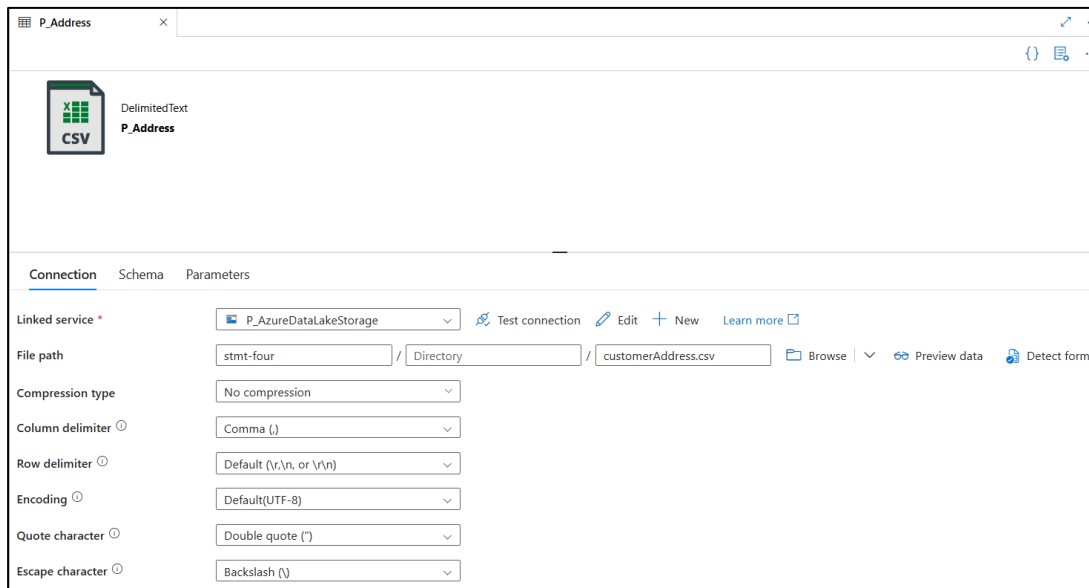
2. Create Customer Address CSV file

The Customer Address CSV file contains the address of some of the customers along with their customer IDs. The data can be found in **customerAddress.csv**

A screenshot of a code editor window titled 'customerAddress.csv'. The editor shows a CSV file with 11 lines of data. The first line is the header: 'CustomerID,Street,City,State,PostalCode'. The subsequent lines contain customer IDs from 1001 to 1010, their respective street addresses, cities, states, and postal codes.

```
customerAddress.csv
1 CustomerID,Street,City,State,PostalCode
2 1001,12 MG Road,Bangalore,Karnataka,560001
3 1002,45 Anna Salai,Chennai,Tamil Nadu,600002
4 1003,78 Park Street,Kolkata,West Bengal,700016
5 1004,21 FC Road,Pune,Maharashtra,411004
6 1005,88 Banjara Hills,Hyderabad,Telangana,500034
7 1006,15 Marine Drive,Mumbai,Maharashtra,400020
8 1007,33 Nehru Place,Delhi,Delhi,110019
9 1008,56 IT Park,Chandigarh,Chandigarh,160101
10 1009,102 Ashram Road,Ahmedabad,Gujarat,380009
11 1010,23 VIP Road,Raipur,Chhattisgarh,492001
```

Upload the CSV file in the storage container and create a new dataset for accessing the file.



P_Address

DelimitedText
P_Address

CSV

Connection Schema Parameters

Linked service * P_AzureDataLakeStorage Test connection Edit + New Learn more

File path stmt-four / Directory / customerAddress.csv Browse Preview data Detect format

Compression type No compression

Column delimiter ① Comma (,)

Row delimiter ① Default (\r,\n, or \r\n)

Encoding ① Default(UTF-8)

Quote character ① Double quote (")

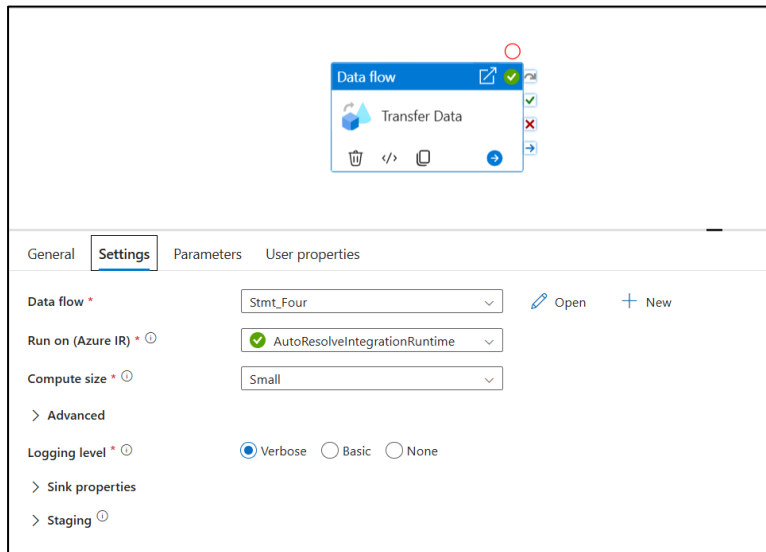
Escape character ① Backslash (\)

The dataset for CSV file has been imported from **stmt-four** directory present in the projectde storage account as shown in the above snapshot.

Step-Wise Guidelines:

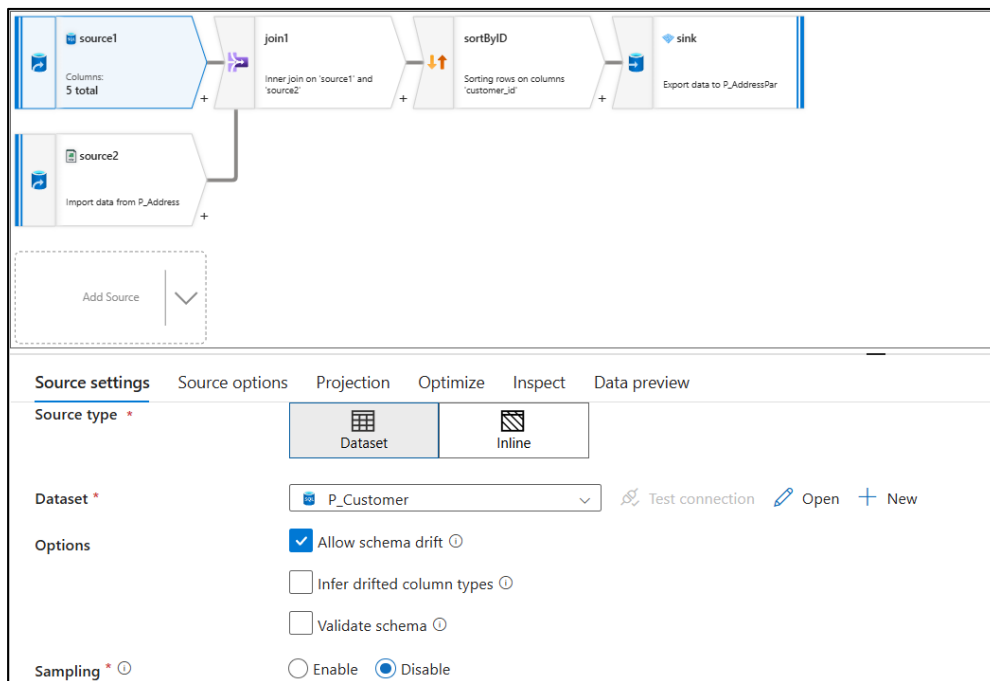
1. Create a **Data Flow** activity

Create new Data Flow



Data Flow:

1. Add Source1



The customer data is present in the customer table, which has been used in statement 1 and statement 2.

The screenshot shows a data pipeline in the Source options tab. The pipeline consists of source1 (Columns: 5 total), join1 (Inner join on 'source1' and 'source2'), sortByID (Sorting rows on columns 'customer_id'), and sink (Export data to P_AddressPar). The Source options tab is selected, showing the following configuration:

- Input:** Query (selected), Table, Stored procedure
- Query * ①:**

```
SELECT *
FROM customer
WHERE customer_id > 1000 and
customer_id < 2000
```
- Batch size ①:** (empty text field)
- Incremental column ①:** (checkbox, unchecked)
- Isolation level ①:** Read uncommitted (dropdown menu)

Go to **Source Options** tab and insert the following query to select record where customer id is between 1000 and 2000:

```
SELECT *
FROM customer
WHERE customer_id > 1000 and customer_id < 2000
```

The query can be found in **query.sql** script.

Go to **Projection** tab and import projection.

2. Add Source2

The screenshot shows a data pipeline in the Source settings tab. The pipeline consists of source1 (Import data from P_Customer), source2 (Columns: 5 total), join1 (Inner join on 'source1' and 'source2'), sortByID (Sorting rows on columns 'customer_id'), and sink (Export data to P_AddressPar). The Source settings tab is selected, showing the following configuration:

- Dataset *:** P_Address (dropdown menu)
- Options:**
 - ☒ Allow schema drift ①
 - ☐ Infer drifted column types ①
 - ☐ Validate schema ①
- Skip line count:** (empty text field)
- Sampling * ①:** Enable, **Disable** (selected)

Source2 is the Customer Address CSV file.

Go to **Projection** tab and import the schema.

3. Click on + to add a join between the two datasets.

Reference 1
Columns: 5 total

source2
Import data from P_Address

join1
Columns: 10 total

sortByID
Sorting rows on columns: customer_id

sink
Export data to P_AddressPar

Join settings | Optimize | Inspect | Data preview

Left stream *
source1

Right stream *
source2

Join type *
Full outer | Inner | Left outer | Right outer | Custom (cross)

Use fuzzy matching ☐

Join conditions *
Left: source1's column
Right: source2's column
123 customer_id == 123 CustomerID

Insert the Left Stream, i.e, left table and Right Stream, i.e, right table from dropdown.

Specify the join condition, in this case, data has to matched according to Customer ID.

4. Add Sort Row Modifier (Optional)

source1
Import data from P_Customer

source2
Import data from P_Address

join1
Inner join on 'source1' and 'source2'

sortByID
Columns: 10 total
Sorting rows on columns: customer_id

sink
Export data to P_AddressPar

Sort settings | Optimize | Inspect | Data preview

Output stream name *
sortByID

Description
Sorting rows on columns: customer_id

Incoming stream *
join1

Options *
☐ Case insensitive
☐ Sort only within partition

Sort conditions *
join1's column
Order
Nulls first
123 customer_id Ascending

Specify the column according to which data has to be sorted, customer_id in this case and sorting order, i.e, ascending.

5. Add Sink

The screenshot shows the Databricks Data Flow Editor interface. At the top, there's a header with 'Stmt_Four' and 'P_StmtFour'. Below it, a 'Validate' button and a 'Data flow debug' toggle are visible. The main workspace displays a data flow diagram with the following components: 'source1' (Import data from P_Customer), 'join1' (Inner join on 'source1' and 'source2'), 'sortByID' (Sorting rows on columns 'customer_id'), and 'sink' (Columns: 10 total). Below the diagram, the 'Sink' configuration panel is open, showing the following settings:

- Output stream name: sink
- Description: Export data to P_AddressPar
- Incoming stream: sortByID
- Sink type: Dataset (selected), Inline, Cache
- Dataset: P_AddressPar
- Options: ☒ Allow schema drift, ☐ Validate schema

Create new dataset, using **New** option beside dataset.

Select Azure Data Lake Storage Gen2

Parquet file format, name the dataset, select the ADLS linked service and specify the file system(in this scenario, stmt-four) and file name(in this scenario, output.parquet)

The screenshot shows the 'P_AddressPar' dataset configuration dialog. The 'Connection' tab is selected, showing the following configuration:

- Linked service: P_AzureDataLakeStorage
- File path: stmt-four / Directory / output.parquet
- Compression type: snappy

Go to **Settings** tab and select the following options:

Sink

Settings

Errors

Mapping

Optimize

Inspect

Data preview

This sink currently has Single partition set in Optimize. This will make your data flow execution longer. The record

Clear the folder

☐

File name option *

Output to single file

Output to single file * ⓘ

output.parquet

Umask ⓘ

Owner

☐ R

☐ W

☐ X

Group

☐ R

☒ W

☐ X

Others

☐ R

☒ W

☐ X

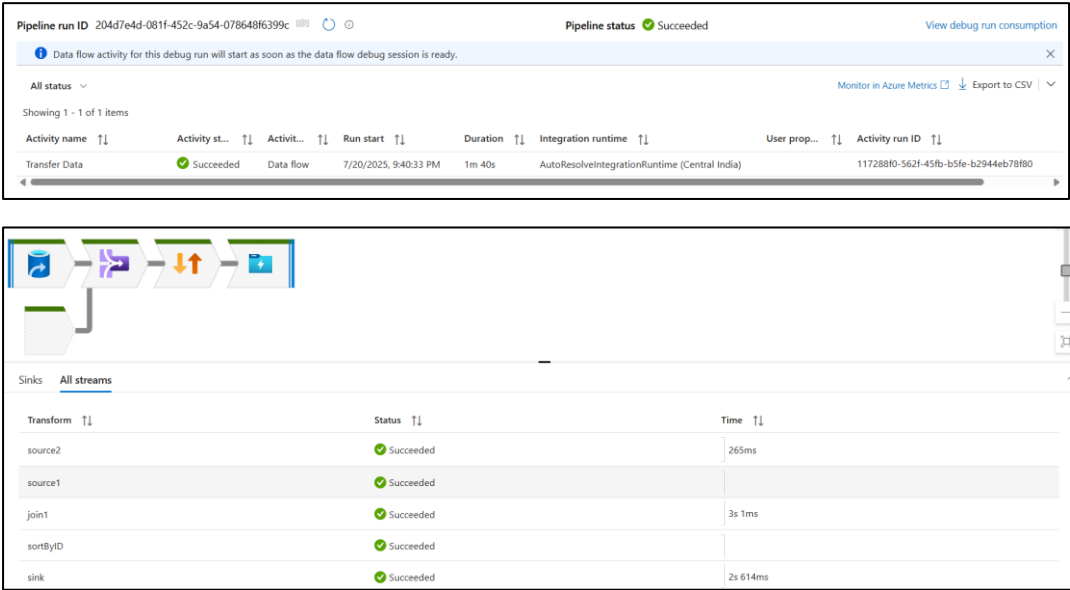
Octal

022

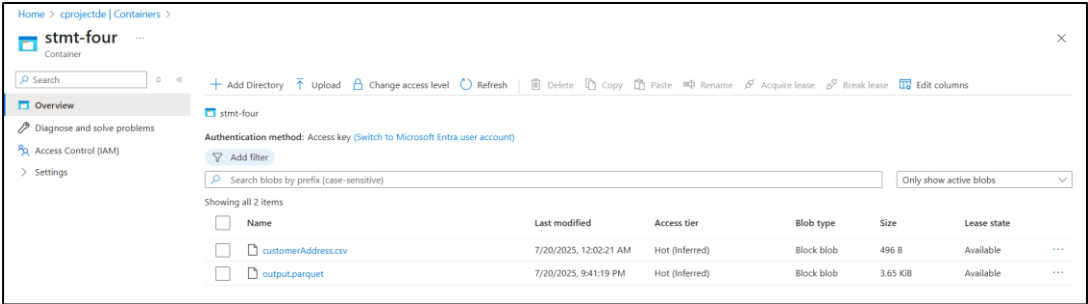
Pre/post commands ⓘ

Final Pipeline:

Pipeline Execution:



Parquet File



The file **output.parquet** has been created in **stmt-four** directory of cprojectde storage account..

Parquet File Data

Showing Records 1 to 10
output.parquet (3.6 KB)

[Edit](#) [Download](#)

Upgrade to download more than 50 records

#	customer_id	first_name	last_name	email	created_date	CustomerID	Street	City	State	PostalCode
1	1001	Ankit	Verma	ankit.verma@mail.in	2025-08-17	1001	12 MG Road	Bangalore	Karnataka	560001
2	1002	Priya	Sharma	priya.sharma@mail.in	2025-08-18	1002	45 Anna Salai	Chennai	Tamil Nadu	600002
3	1003	Rohit	Mehta	rohit.mehta@mail.in	2025-08-19	1003	78 Park Street	Kolkata	West Bengal	700016
4	1004	Sneha	Patel	sneha.patel@mail.in	2025-08-20	1004	21 FC Road	Pune	Maharashtra	411004
5	1005	Arjun	Kapoor	arjun.kapoor@mail.in	2025-08-21	1005	88 Banjara Hills	Hyderabad	Telangana	500034
6	1006	Neha	Reddy	neha.reddy@mail.in	2025-08-22	1006	15 Marine Drive	Mumbai	Maharashtra	400020
7	1007	Karan	Joshi	karan.joshi@mail.in	2025-08-23	1007	33 Nehru Place	Delhi	Delhi	110019
8	1008	Pooja	Mishra	pooja.mishra@mail.in	2025-08-24	1008	56 IT Park	Chandigarh	Chandigarh	160101
9	1009	Manish	Gupta	manish.gupta@mail.in	2025-08-25	1009	102 Ashram Road	Ahmedabad	Gujarat	380009
10	1010	Divya	Singh	divya.singh@mail.in	2025-08-26	1010	23 VIP Road	Raipur	Chhattisgarh	492001