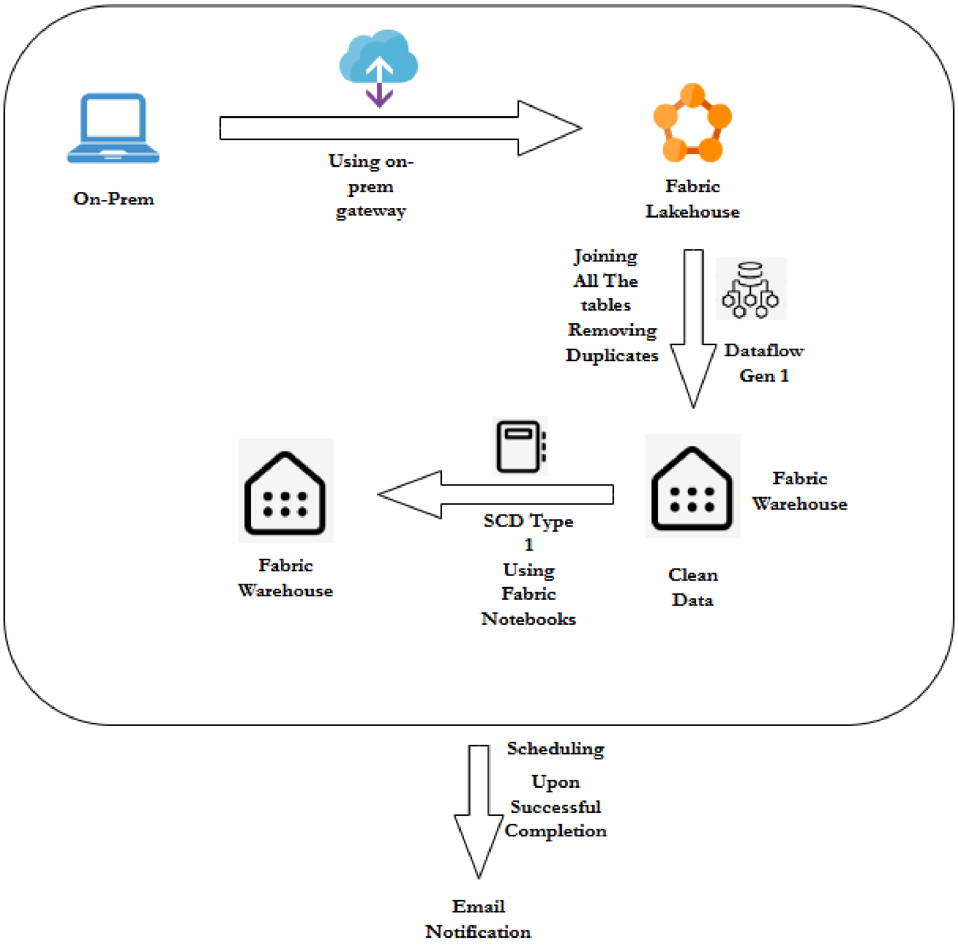
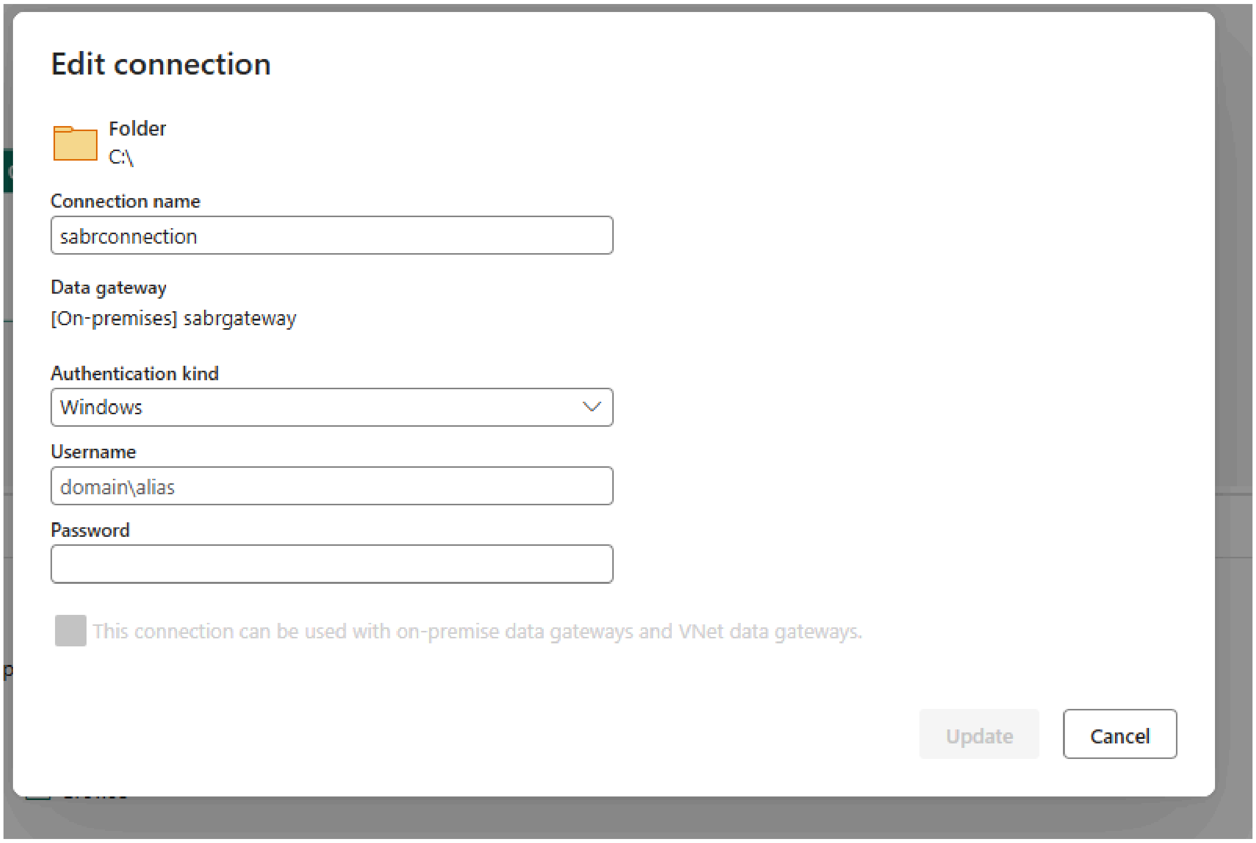
Bootcamp Project 4



The architecture for the project

Bronze Layer: For this layer I brought all the 5 raw files from on-premises to the Fabric Lakehouse using **On-Prem Gateway** that I installed.

The connection using On-Prem Gateway



Edit connection

Folder
C:\

Connection name
sabrconnection

Data gateway
[On-premises] sabrgateway

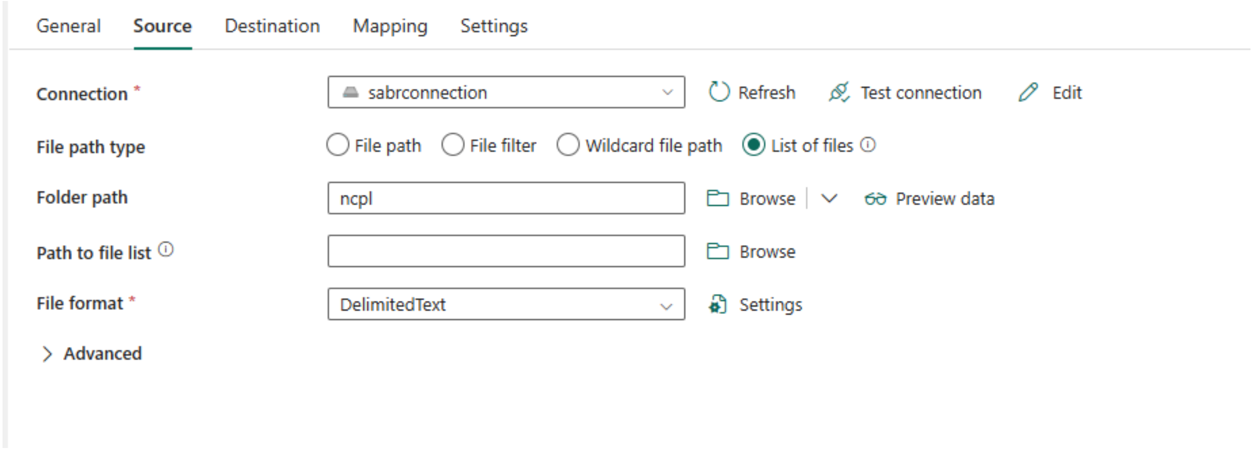
Authentication kind
Windows

Username
domain\alias

Password

□ This connection can be used with on-premise data gateways and VNet data gateways.

In the source, I selected List of Files as it brings all the files in the selected folder:



General | **Source** | Destination | Mapping | Settings

Connection * sabrconnection — Refresh · Test connection · Edit

File path type ○ File path ○ File filter ○ Wildcard file path ● List of files ⓘ

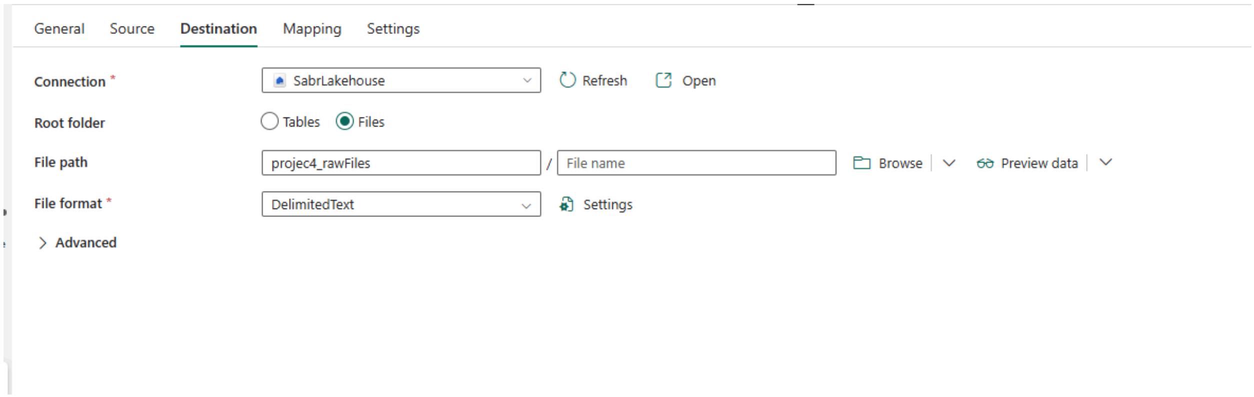
Folder path ncpl — Browse · Preview data

Path to file list ⓘ — Browse

File format * DelimitedText — Settings

> Advanced

Destination:



General | Source | **Destination** | Mapping | Settings

Connection * SabrLakehouse — Refresh · Open

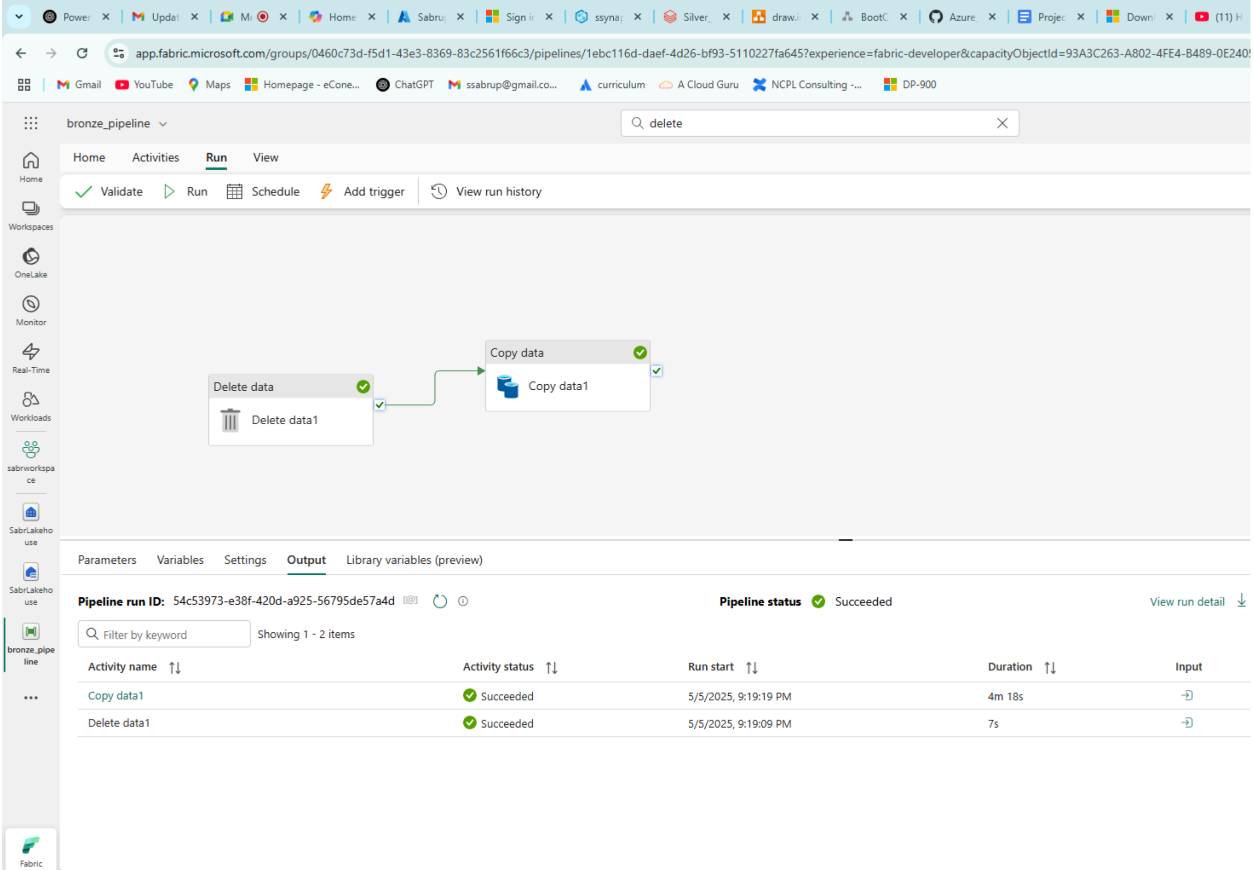
Root folder ○ Tables ● Files

File path projec4_rawFiles / File name — Browse · Preview data

File format * DelimitedText — Settings

> Advanced

Here the destination is Lakehouse,**SabrLakehouse.**


app.fabric.microsoft.com/groups/0460c73d-f5d1-43e3-8369-83c2561f66c3/pipelines/1ebc116d-daef-4d26-bf93-5110227fa645?experience=fabric-developer&capacityObjectId=93A3C263-A802-4FE4-B489-0E240
bronze_pipeline
delete
Home
Activities
Run
View
Validate
Run
Schedule
Add trigger
View run history
Delete data
Delete data1
Copy data
Copy data1
Parameters
Variables
Settings
Output
Library variables (preview)
View run detail

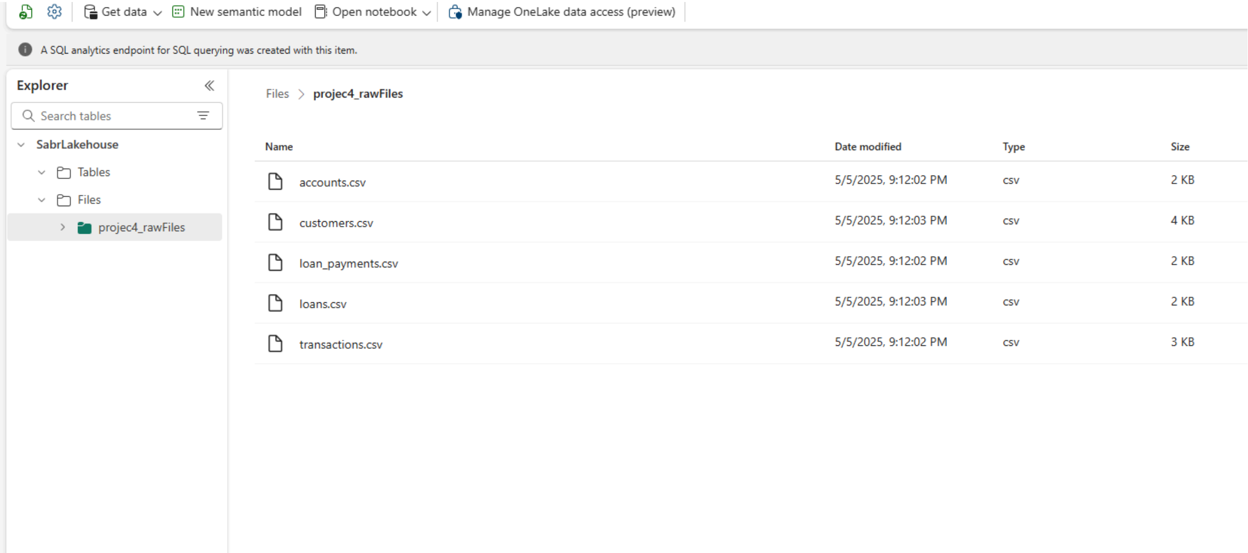
Pipeline run ID: 54c53973-e38f-420d-a925-56795de57a4d

Pipeline status ✓ Succeeded

Showing 1 - 2 items

Activity name	Activity status	Run start	Duration	Input
Copy data1	✓ Succeeded	5/5/2025, 9:19:19 PM	4m 18s	
Delete data1	✓ Succeeded	5/5/2025, 9:19:09 PM	7s	

The above SC shows the successfully running copy data activity.

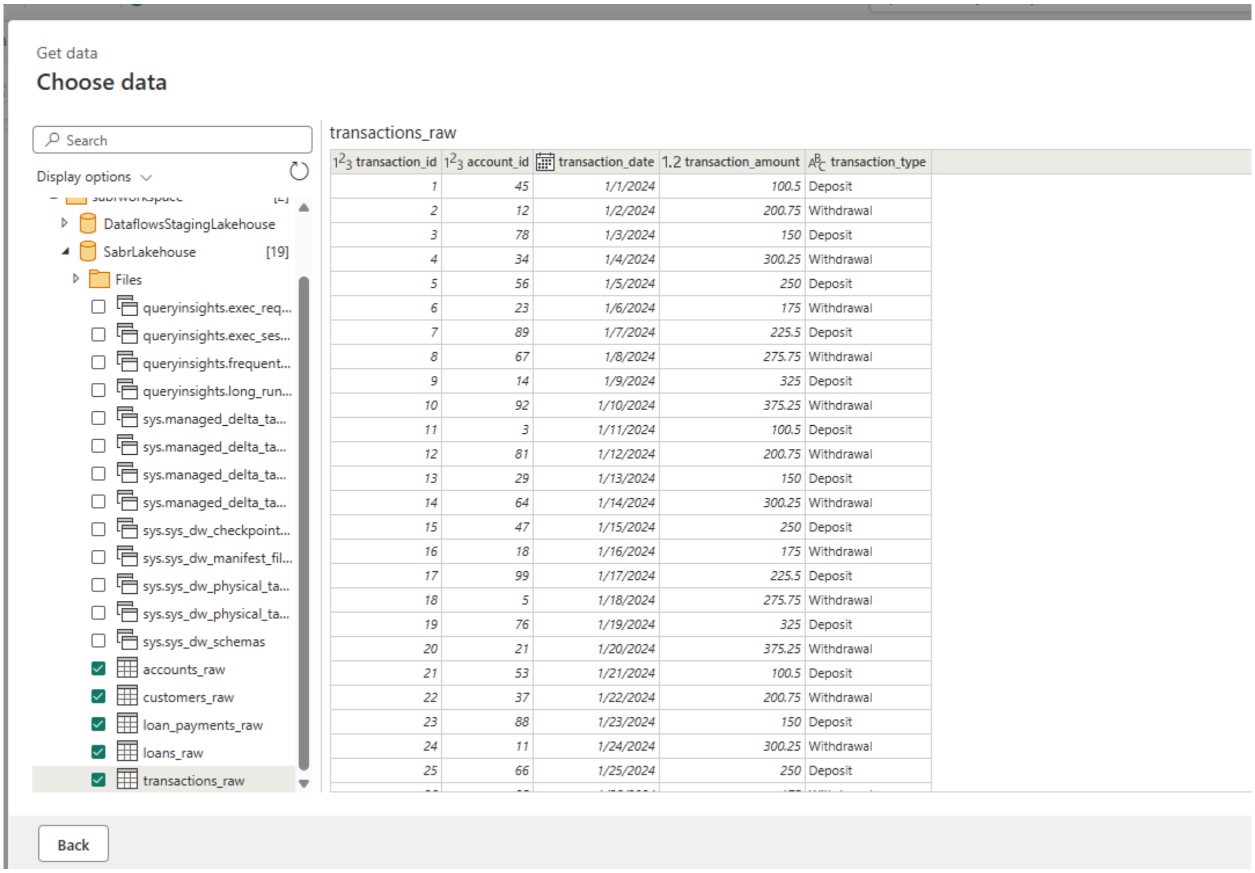


A SQL analytics endpoint for SQL querying was created with this item.

Name	Date modified	Type	Size
accounts.csv	5/5/2025, 9:12:02 PM	csv	2 KB
customers.csv	5/5/2025, 9:12:03 PM	csv	4 KB
loan_payments.csv	5/5/2025, 9:12:02 PM	csv	2 KB
loans.csv	5/5/2025, 9:12:03 PM	csv	2 KB
transactions.csv	5/5/2025, 9:12:02 PM	csv	3 KB

The above are the files in Lakehouse.

For silver layer:



Get data

Choose data

Search

Display options

- DataflowsStagingLakehouse
- SabrLakehouse [19]
  - Files
  - □ queryinsights.exec_req...
  - □ queryinsights.exec_ses...
  - □ queryinsights.frequent...
  - □ queryinsights.long_run...
  - □ sys.managed_delta_ta...
  - □ sys.managed_delta_ta...
  - □ sys.managed_delta_ta...
  - □ sys.managed_delta_ta...
  - □ sys.sys_dw_checkpoint...
  - □ sys.sys_dw_manifest_fil...
  - □ sys.sys_dw_physical_ta...
  - □ sys.sys_dw_physical_ta...
  - □ sys.sys_dw_schemas
  - ☑ accounts_raw
  - ☑ customers_raw
  - ☑ loan_payments_raw
  - ☑ loans_raw
  - ☑ transactions_raw

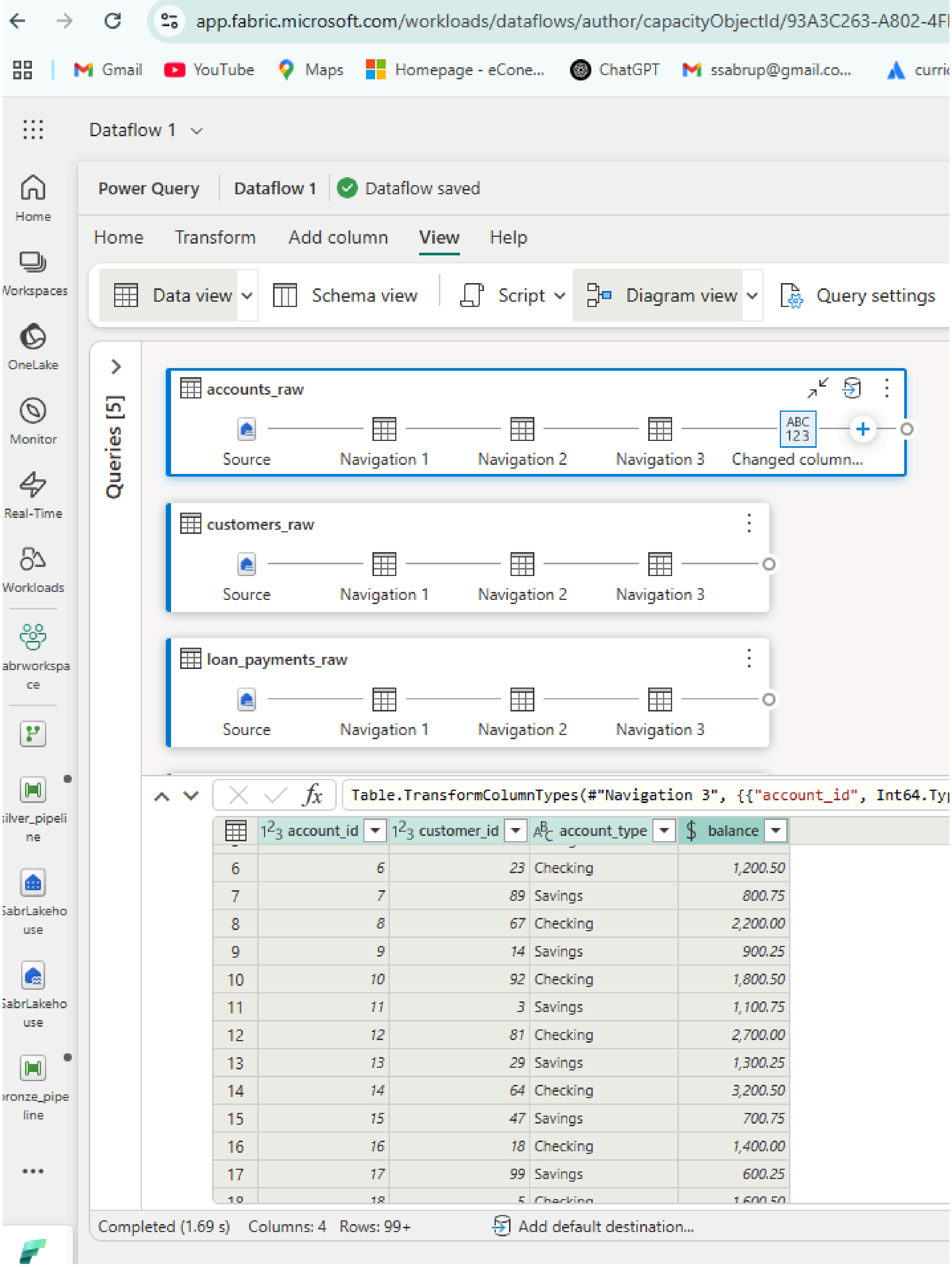
transactions_raw

transaction_id	account_id	transaction_date	transaction_amount	transaction_type
1	45	1/1/2024	100.5	Deposit
2	12	1/2/2024	200.75	Withdrawal
3	78	1/3/2024	150	Deposit
4	34	1/4/2024	300.25	Withdrawal
5	56	1/5/2024	250	Deposit
6	23	1/6/2024	175	Withdrawal
7	89	1/7/2024	225.5	Deposit
8	67	1/8/2024	275.75	Withdrawal
9	14	1/9/2024	325	Deposit
10	92	1/10/2024	375.25	Withdrawal
11	3	1/11/2024	100.5	Deposit
12	81	1/12/2024	200.75	Withdrawal
13	29	1/13/2024	150	Deposit
14	64	1/14/2024	300.25	Withdrawal
15	47	1/15/2024	250	Deposit
16	18	1/16/2024	175	Withdrawal
17	99	1/17/2024	225.5	Deposit
18	5	1/18/2024	275.75	Withdrawal
19	76	1/19/2024	325	Deposit
20	21	1/20/2024	375.25	Withdrawal
21	53	1/21/2024	100.5	Deposit
22	37	1/22/2024	200.75	Withdrawal
23	88	1/23/2024	150	Deposit
24	11	1/24/2024	300.25	Withdrawal
25	66	1/25/2024	250	Deposit

Back

Selecting the sources for Dataflow

For the accounts table, the SC shows the changed datatypes:

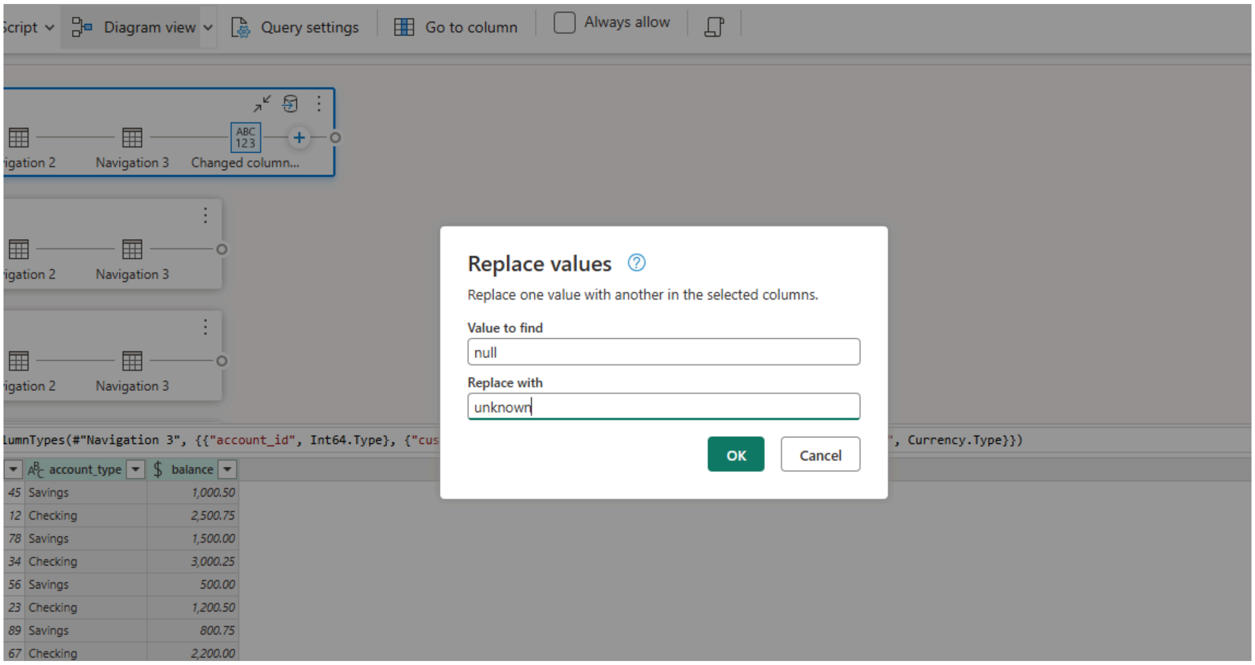


```
Table.TransformColumnTypes(#"Navigation 3", {{"account_id", Int64.Ty
```

	account_id	customer_id	account_type	balance
6	6	23	Checking	1,200.50
7	7	89	Savings	800.75
8	8	67	Checking	2,200.00
9	9	14	Savings	900.25
10	10	92	Checking	1,800.50
11	11	3	Savings	1,100.75
12	12	81	Checking	2,700.00
13	13	29	Savings	1,300.25
14	14	64	Checking	3,200.50
15	15	47	Savings	700.75
16	16	18	Checking	1,400.00
17	17	99	Savings	600.25
18	18	5	Checking	1,600.50

Completed (1.69 s)   Columns: 4   Rows: 99+   Add default destination...

Replacing null with ‘unknown’:



Replace values

Replace one value with another in the selected columns.

Value to find: null

Replace with: unknown

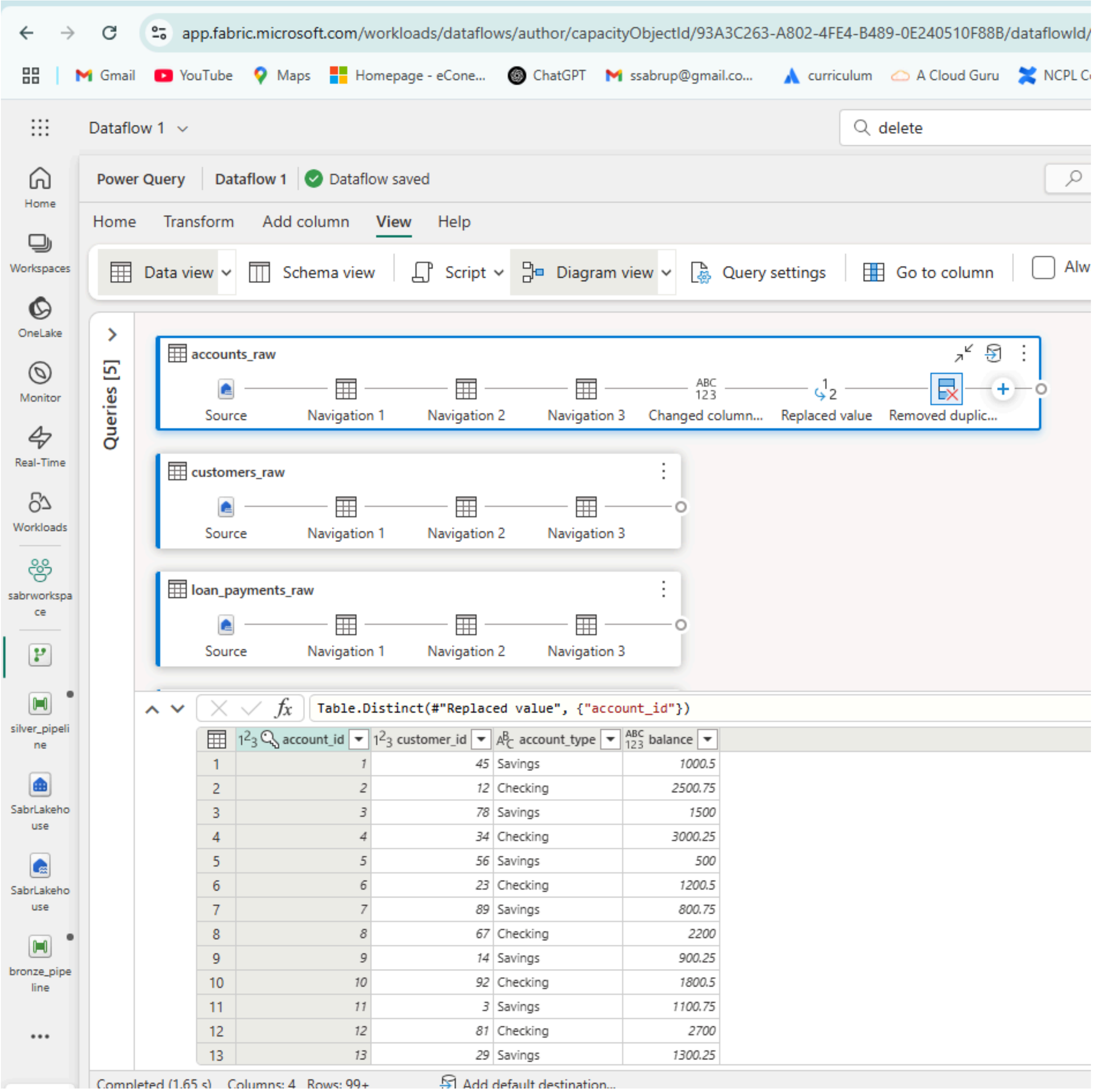
OK

Cancel

```
lumnTypes(#"Navigation 3", {{"account_id", Int64.Type}, {"cus
", Currency.Type}})
```

	account_type	balance
45	Savings	1,000.50
12	Checking	2,500.75
78	Savings	1,500.00
34	Checking	3,000.25
56	Savings	500.00
23	Checking	1,200.50
89	Savings	800.75
67	Checking	2,200.00

Removing the duplicates:

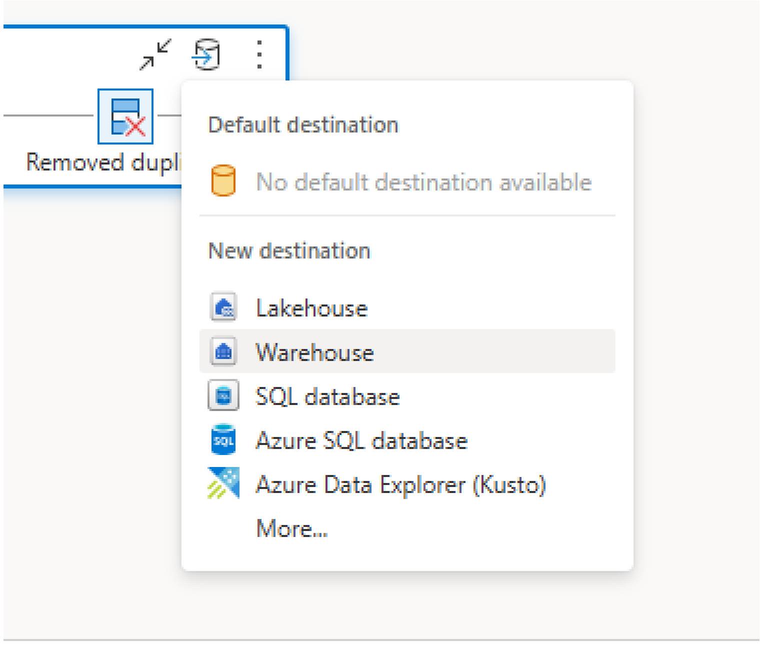


```
Table.Distinct(#"Replaced value", {"account_id"})
```

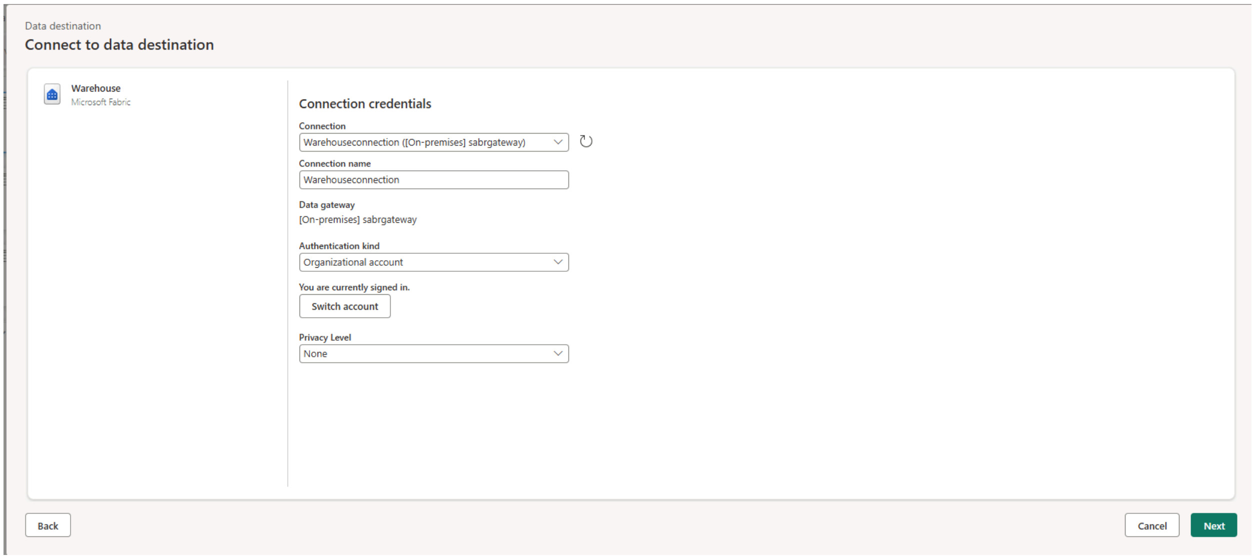
	account_id	customer_id	account_type	balance
1	1	45	Savings	1000.5
2	2	12	Checking	2500.75
3	3	78	Savings	1500
4	4	34	Checking	3000.25
5	5	56	Savings	500
6	6	23	Checking	1200.5
7	7	89	Savings	800.75
8	8	67	Checking	2200
9	9	14	Savings	900.25
10	10	92	Checking	1800.5
11	11	3	Savings	1100.75
12	12	81	Checking	2700
13	13	29	Savings	1300.25

Completed (1.65 s)   Columns: 4   Rows: 99+   Add default destination...

Adding Destination which is Data Warehouse:



Nextly, need to make a connection for the warehouse:


Data destination

Connect to data destination

Warehouse
Microsoft Fabric

Connection credentials

Connection: Warehouseconnection ([On-premises] sabrgateway)

Connection name: Warehouseconnection

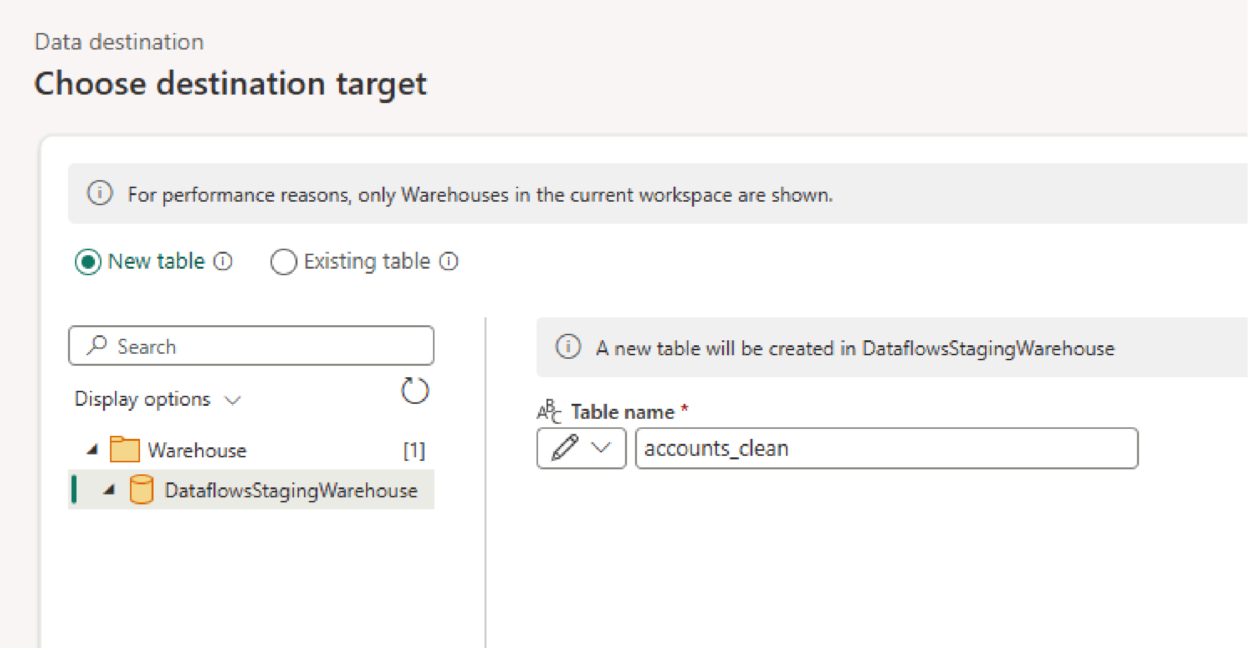
Data gateway: [On-premises] sabrgateway

Authentication kind: Organizational account

You are currently signed in.

Privacy Level: None

The target would table name would be “accounts_clean”



Data destination

Choose destination target

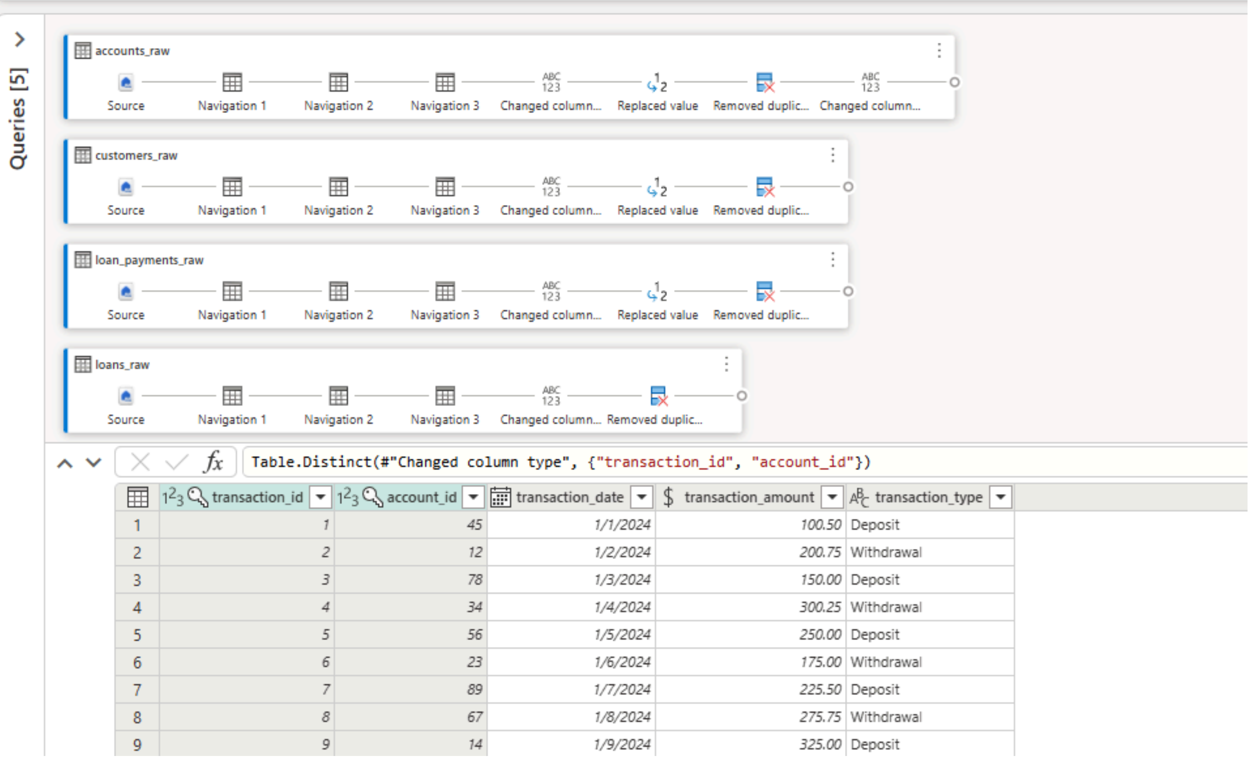
ⓘ For performance reasons, only Warehouses in the current workspace are shown.

● New table ⓘ ○ Existing table ⓘ

ⓘ A new table will be created in DataflowsStagingWarehouse

Table name *: accounts_clean

Applied changes to all the tables:

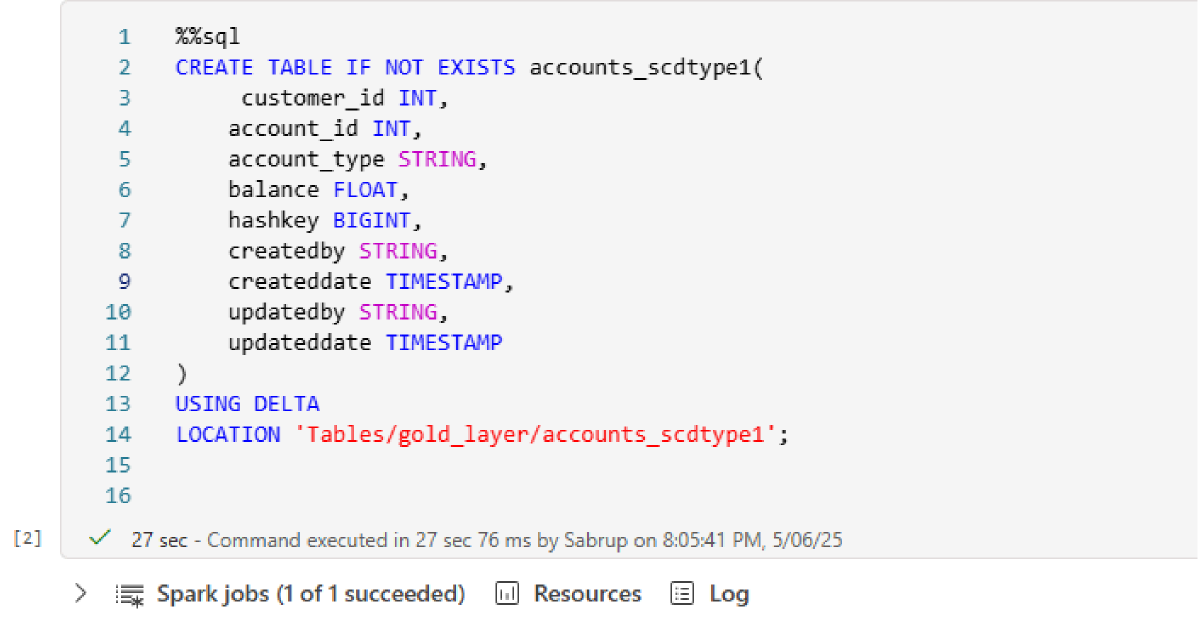


```
Table.Distinct(#"Changed column type", {"transaction_id", "account_id"})
```

	transaction_id	account_id	transaction_date	transaction_amount	transaction_type
1	1	45	1/1/2024	100.50	Deposit
2	2	12	1/2/2024	200.75	Withdrawal
3	3	78	1/3/2024	150.00	Deposit
4	4	34	1/4/2024	300.25	Withdrawal
5	5	56	1/5/2024	250.00	Deposit
6	6	23	1/6/2024	175.00	Withdrawal
7	7	89	1/7/2024	225.50	Deposit
8	8	67	1/8/2024	275.75	Withdrawal
9	9	14	1/9/2024	325.00	Deposit

For the gold layer, I have transformed the data into SCD Type-1 format.
I used Fabric Notebook to do this task.
The code for the same is below:

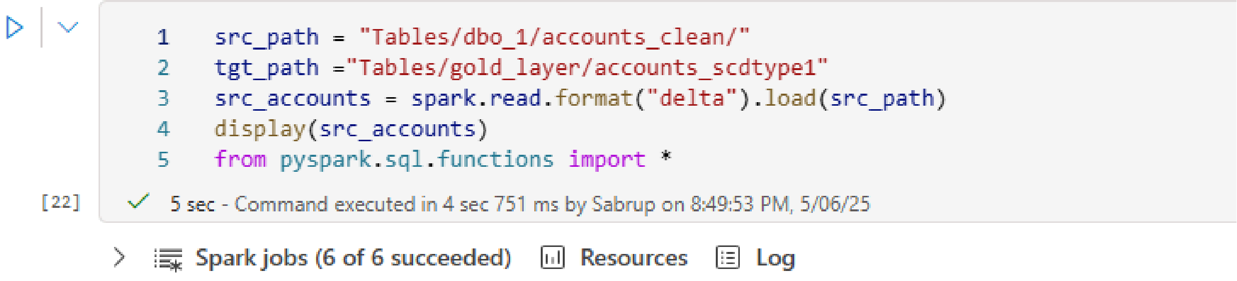
Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review



```
%%sql
CREATE TABLE IF NOT EXISTS accounts_scdtype1(
     customer_id INT,
    account_id INT,
    account_type STRING,
    balance FLOAT,
    hashkey BIGINT,
    createdby STRING,
    createddate TIMESTAMP,
    updatedby STRING,
    updateddate TIMESTAMP
)
USING DELTA
LOCATION 'Tables/gold_layer/accounts_scdtype1';


```

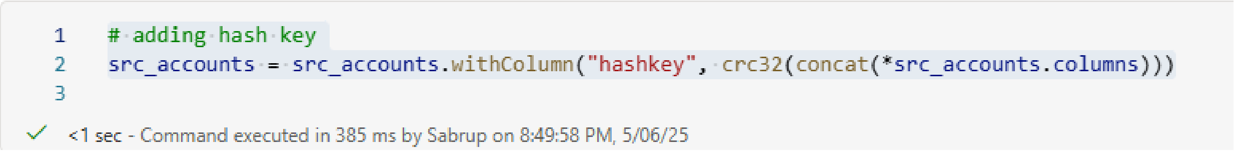
[2] ✓ 27 sec - Command executed in 27 sec 76 ms by Sabrup on 8:05:41 PM, 5/06/25



```
src_path = "Tables/dbo_1/accounts_clean/"
tgt_path ="Tables/gold_layer/accounts_scdtype1"
src_accounts = spark.read.format("delta").load(src_path)
display(src_accounts)
from pyspark.sql.functions import *
```

[22] ✓ 5 sec - Command executed in 4 sec 751 ms by Sabrup on 8:49:53 PM, 5/06/25

The SCD Type 1 has been created for the accounts table.



```
# adding hash key
src_accounts = src_accounts.withColumn("hashkey", crc32(concat(*src_accounts.columns)))

```

✓ <1 sec - Command executed in 385 ms by Sabrup on 8:49:58 PM, 5/06/25

The hashkey added.

The rest commands are below:

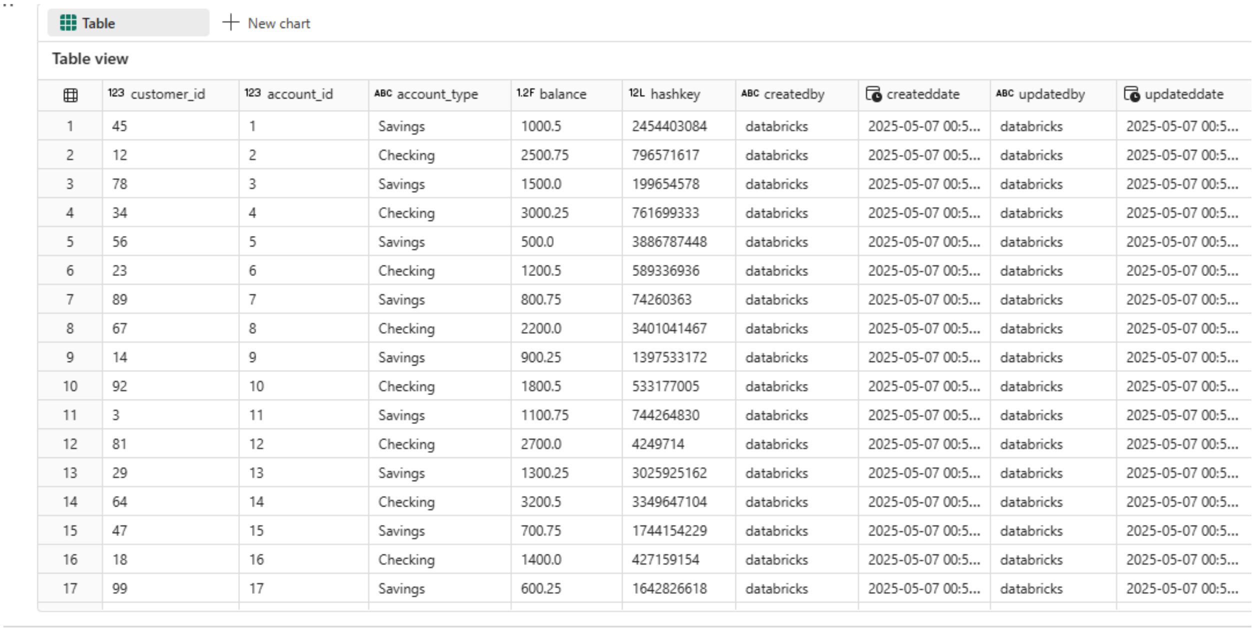
```
from delta.tables import DeltaTable
deltatable = DeltaTable.forPath(spark, tgt_path)
deltatable.toDF().show

from pyspark.sql.functions import col
src_accounts =
src_accounts.alias("account").join(deltatable.toDF().alias("tgt"),((col("a
ccount.account_id")==col("tgt.account_id")) &
(col("account.hashkey")==col("tgt.hashkey"))),"anti").select("account.*")

from pyspark.sql.functions import lit, current_timestamp
deltatable.alias("tgt").merge(
    src_accounts.alias("src"),
    "tgt.account_id = src.account_id")\
  .whenMatchedUpdate(set={"tgt.account_id": "src.account_id
","tgt.customer_id": "src.customer_id","tgt.account_type":
"src.account_type","tgt.balance": "src.balance","tgt.hashkey":
"src.hashkey", "tgt.updatedby":lit("databricks"),
"tgt.updateddate":current_timestamp()})\
  .whenNotMatchedInsert(values={"tgt.account_id": "src.account_id
","tgt.customer_id": "src.customer_id","tgt.account_type":
"src.account_type","tgt.balance": "src.balance","tgt.hashkey":
"src.hashkey", "tgt.createdby":lit("databricks"),
"tgt.createddate":current_timestamp(), "tgt.updatedby":lit("databricks"),
"tgt.updateddate":current_timestamp()})\
      .execute()
display(spark.read.format("delta").load(tgt_path))
```

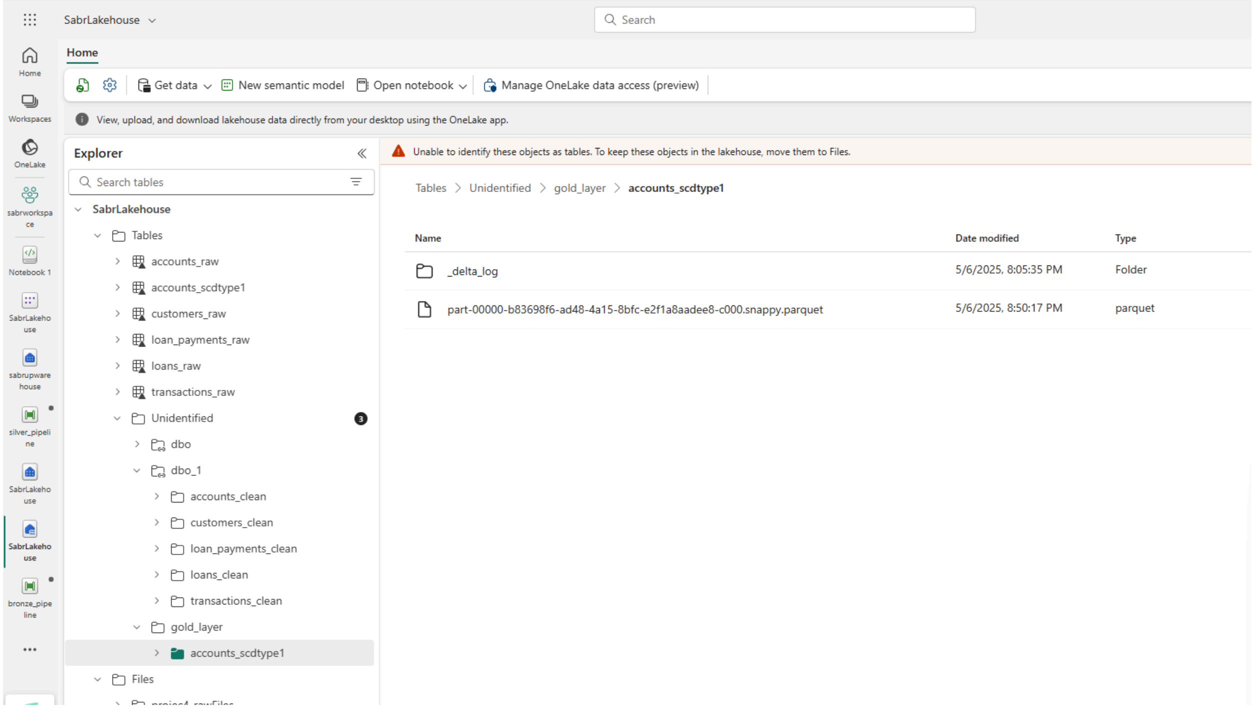
The SCD Type-1 table is below:

Accounts:



	customer_id	account_id	account_type	balance	hashkey	createdby	createddate	updatedby	updateddate
1	45	1	Savings	1000.5	2454403084	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
2	12	2	Checking	2500.75	796571617	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
3	78	3	Savings	1500.0	199654578	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
4	34	4	Checking	3000.25	761699333	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
5	56	5	Savings	500.0	3886787448	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
6	23	6	Checking	1200.5	589336936	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
7	89	7	Savings	800.75	74260363	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
8	67	8	Checking	2200.0	3401041467	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
9	14	9	Savings	900.25	1397533172	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
10	92	10	Checking	1800.5	533177005	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
11	3	11	Savings	1100.75	744264830	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
12	81	12	Checking	2700.0	4249714	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
13	29	13	Savings	1300.25	3025925162	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
14	64	14	Checking	3200.5	3349647104	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
15	47	15	Savings	700.75	1744154229	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
16	18	16	Checking	1400.0	427159154	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...
17	99	17	Savings	600.25	1642826618	databricks	2025-05-07 00:5...	databricks	2025-05-07 00:5...

The table stored in the shortcut:



View, upload, and download lakehouse data directly from your desktop using the OneLake app.

Unable to identify these objects as tables. To keep these objects in the lakehouse, move them to Files.

Tables > Unidentified > gold_layer > **accounts_scdtype1**

Name	Date modified	Type
_delta_log	5/6/2025, 8:05:35 PM	Folder
part-00000-b83698f6-ad48-4a15-8bfc-e2f1a8aadee8-c000.snappy.parquet	5/6/2025, 8:50:17 PM	parquet

Customers Table:

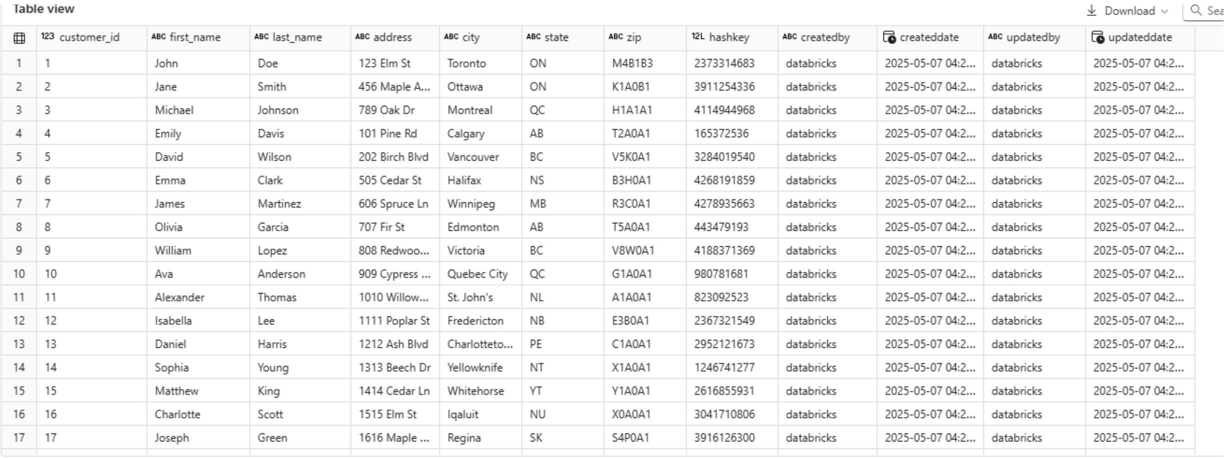


Table view

	customer_id	first_name	last_name	address	city	state	zip	hashkey	createdby	createddate	updatedby	updateddate
1	1	John	Doe	123 Elm St	Toronto	ON	M4B1B3	2373314683	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
2	2	Jane	Smith	456 Maple A...	Ottawa	ON	K1A0B1	3911254336	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
3	3	Michael	Johnson	789 Oak Dr	Montreal	QC	H1A1A1	4114944968	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
4	4	Emily	Davis	101 Pine Rd	Calgary	AB	T2A0A1	165372536	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
5	5	David	Wilson	202 Birch Blvd	Vancouver	BC	V5K0A1	3284019540	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
6	6	Emma	Clark	505 Cedar St	Halifax	NS	B3H0A1	4268191859	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
7	7	James	Martinez	606 Spruce Ln	Winnipeg	MB	R3C0A1	4278935663	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
8	8	Olivia	Garcia	707 Fir St	Edmonton	AB	T5A0A1	443479193	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
9	9	William	Lopez	808 Redwoo...	Victoria	BC	V8W0A1	4188371369	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
10	10	Ava	Anderson	909 Cypress ...	Quebec City	QC	G1A0A1	980781681	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
11	11	Alexander	Thomas	1010 Willow...	St. John's	NL	A1A0A1	823092523	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
12	12	Isabella	Lee	1111 Poplar St	Fredericton	NB	E3B0A1	2367321549	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
13	13	Daniel	Harris	1212 Ash Blvd	Charlotteto...	PE	C1A0A1	2952121673	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
14	14	Sophia	Young	1313 Beech Dr	Yellowknife	NT	X1A0A1	1246741277	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
15	15	Matthew	King	1414 Cedar Ln	Whitehorse	YT	Y1A0A1	2616855931	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
16	16	Charlotte	Scott	1515 Elm St	Iqaluit	NU	X0A0A1	3041710806	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...
17	17	Joseph	Green	1616 Maple ...	Regina	SK	S4P0A1	3916126300	databricks	2025-05-07 04:2...	databricks	2025-05-07 04:2...

Loan_payments:

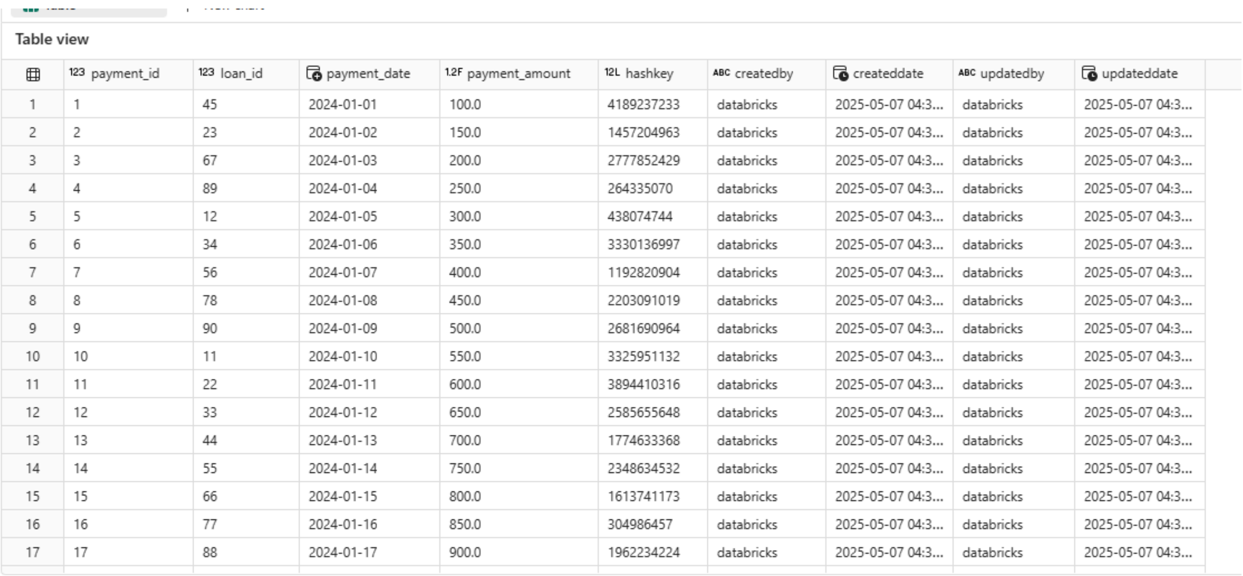


Table view

	payment_id	loan_id	payment_date	payment_amount	hashkey	createdby	createddate	updatedby	updateddate
1	1	45	2024-01-01	100.0	4189237233	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
2	2	23	2024-01-02	150.0	1457204963	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
3	3	67	2024-01-03	200.0	2777852429	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
4	4	89	2024-01-04	250.0	264335070	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
5	5	12	2024-01-05	300.0	438074744	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
6	6	34	2024-01-06	350.0	3330136997	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
7	7	56	2024-01-07	400.0	1192820904	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
8	8	78	2024-01-08	450.0	2203091019	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
9	9	90	2024-01-09	500.0	2681690964	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
10	10	11	2024-01-10	550.0	3325951132	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
11	11	22	2024-01-11	600.0	3894410316	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
12	12	33	2024-01-12	650.0	2585655648	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
13	13	44	2024-01-13	700.0	1774633368	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
14	14	55	2024-01-14	750.0	2348634532	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
15	15	66	2024-01-15	800.0	1613741173	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
16	16	77	2024-01-16	850.0	304986457	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...
17	17	88	2024-01-17	900.0	1962234224	databricks	2025-05-07 04:3...	databricks	2025-05-07 04:3...

Loans:

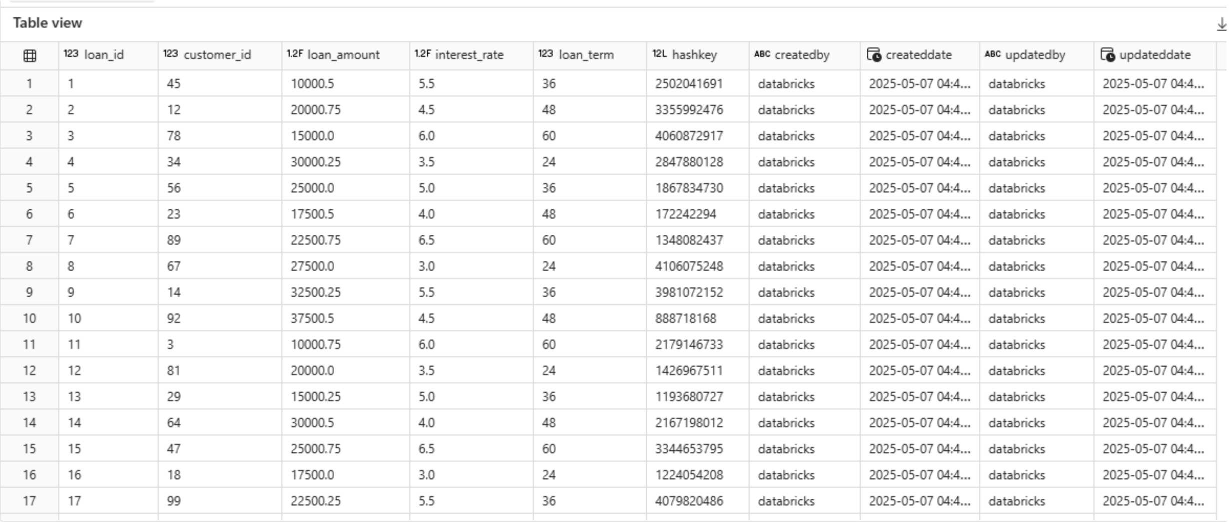


Table view

	123 loan_id	123 customer_id	1.2F loan_amount	1.2F interest_rate	123 loan_term	12L hashkey	ABC createdby	createddate	ABC updatedby	updateddate
1	1	45	10000.5	5.5	36	2502041691	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
2	2	12	20000.75	4.5	48	3355992476	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
3	3	78	15000.0	6.0	60	4060872917	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
4	4	34	30000.25	3.5	24	2847880128	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
5	5	56	25000.0	5.0	36	1867834730	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
6	6	23	17500.5	4.0	48	172242294	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
7	7	89	22500.75	6.5	60	1348082437	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
8	8	67	27500.0	3.0	24	4106075248	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
9	9	14	32500.25	5.5	36	3981072152	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
10	10	92	37500.5	4.5	48	888718168	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
11	11	3	10000.75	6.0	60	2179146733	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
12	12	81	20000.0	3.5	24	1426967511	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
13	13	29	15000.25	5.0	36	1193680727	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
14	14	64	30000.5	4.0	48	2167198012	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
15	15	47	25000.75	6.5	60	3344653795	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
16	16	18	17500.0	3.0	24	1224054208	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...
17	17	99	22500.25	5.5	36	4079820486	databricks	2025-05-07 04:4...	databricks	2025-05-07 04:4...