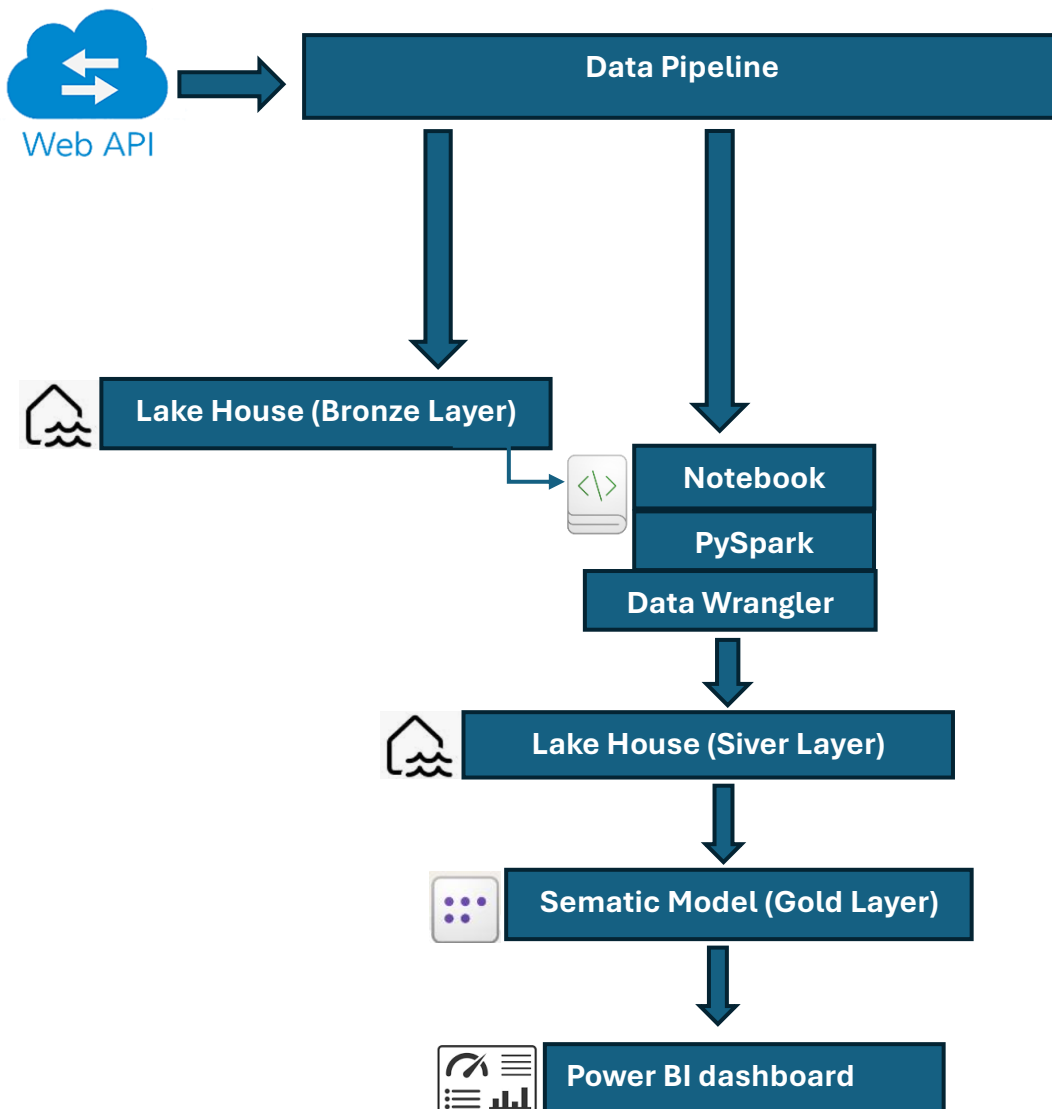


# World Population Data analysis with Data Engineering and Data Analytics in Microsoft fabric.

Components used:

- Web API
- Data Pipeline
- Lake house
- Notebook
- Data wrangler
- PySpark
- Semantic model
- Power BI report



**Inventory**

+ New item
New folder
→ Import
Migrate

Inventory > Fabric Project

	Name	Type	Task	Owner
	Bronze_LH	Lakehouse	—	RDAzure
	Bronze_LH	Semantic model (d...	—	Inventory
	Bronze_LH	SQL analytics endp...	—	RDAzure
	GetData_Pipeline	Data pipeline	—	RDAzure
	Notebook 1	Notebook	—	RDAzure
	Population summary	Report	—	Inventory
	Silver_LH	Lakehouse	—	RDAzure
	Silver_LH	Semantic model (d...	—	Inventory
	Silver_LH	SQL analytics endp...	—	RDAzure

## Source:

<https://github.com/ranjustalk/RKSRC/>

ranjustalk / RKSRC

<> Code
Issues
Pull requests
Actions
Projects
Wiki

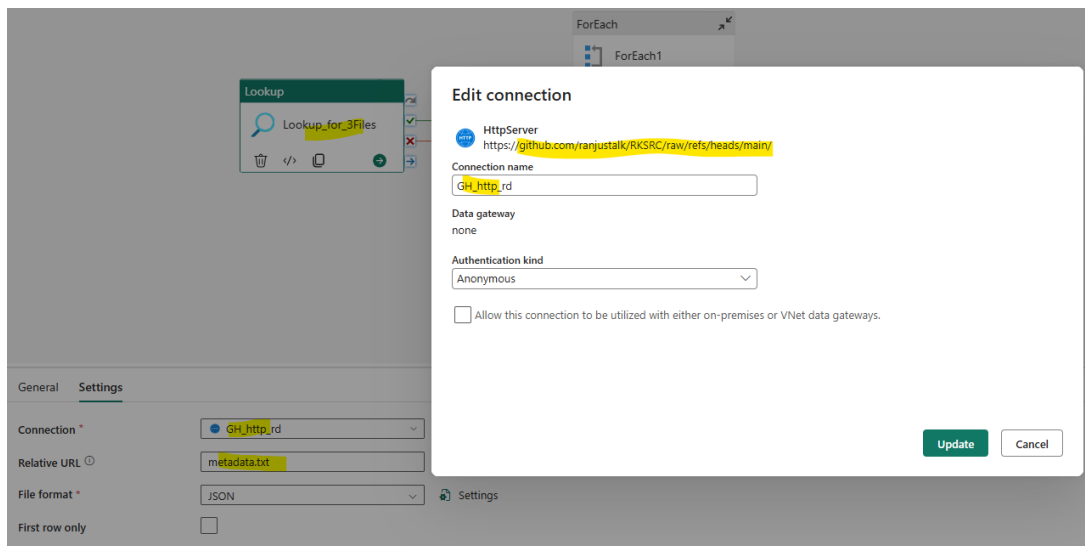
**RKSRC**
Public

main
1 Branch
0 Tags
Go to file

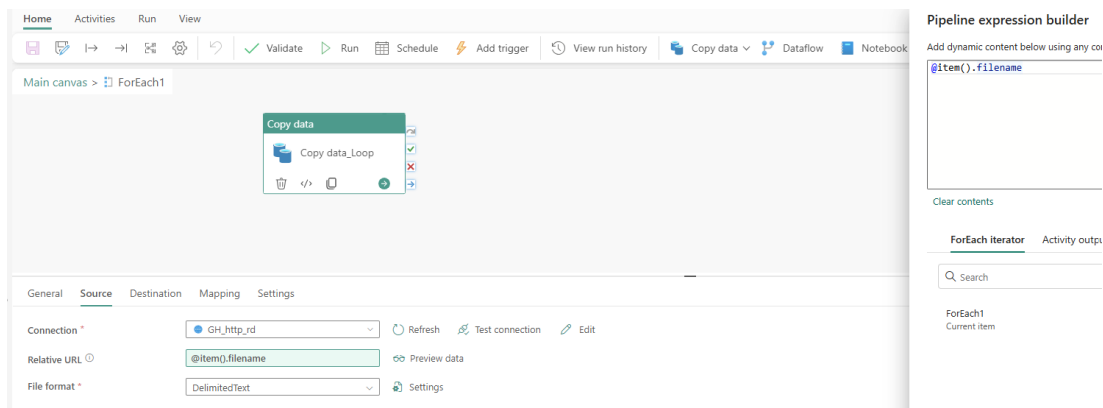
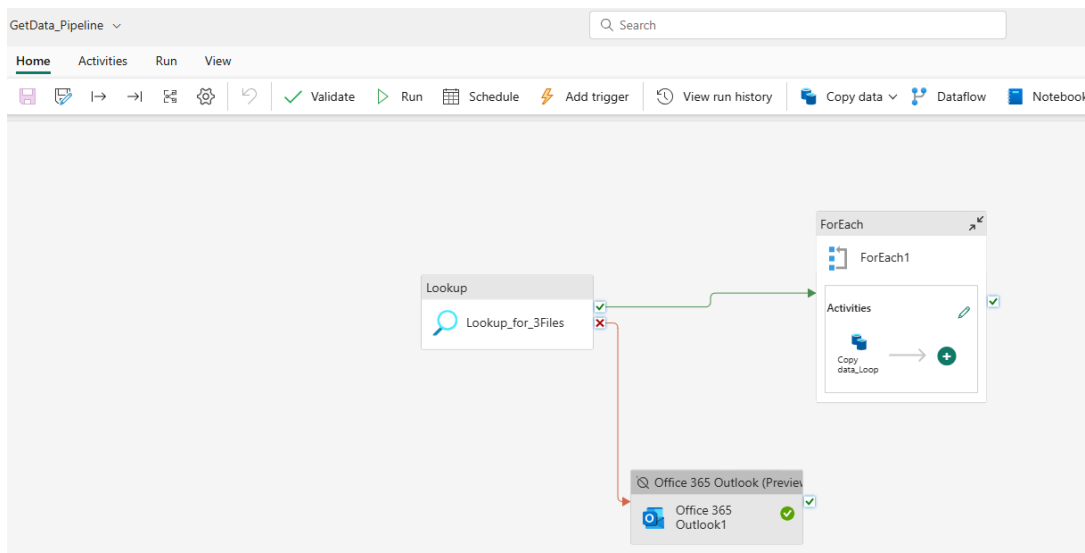
Add files via upload

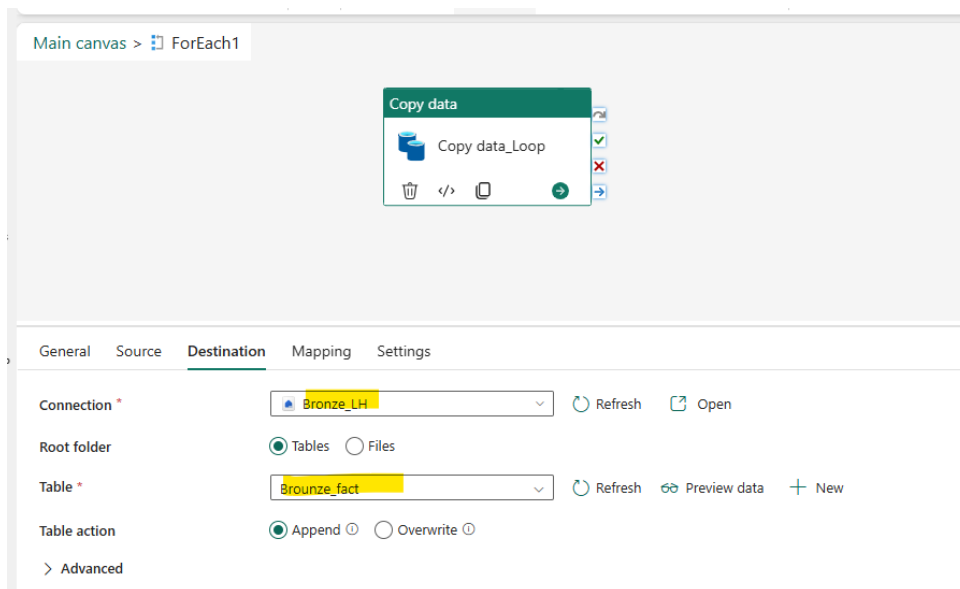
Population-country-1950-1999.csv	Add files via upload
Population-country-2000-2049.csv	Add files via upload
Population-country-2050-2100.csv	Add files via upload
metadata.txt	Add files via upload

## Connection to Source:



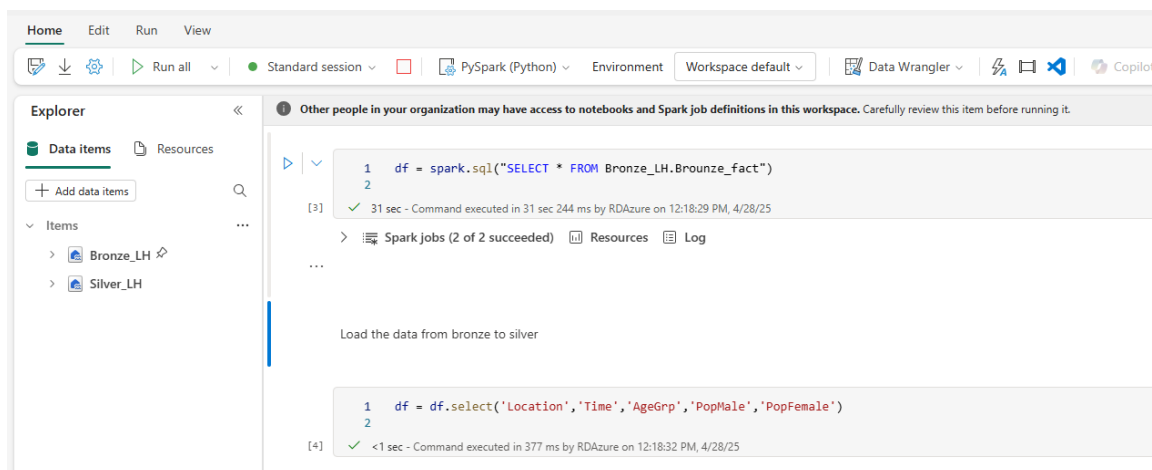
## Bronze layer (Data pipeline):





## Silver layer (Notebook):

Data cleaning and transformation. Using PySpark (Python) and Data Wrangler.









```

1  # Code generated by Data Wrangler for PySpark DataFrame
2
3  from pyspark.sql import functions as F
4
5  def clean_data(df):
6      # Replace all instances of "9-May" with "05-09" in column: 'AgeGrp'
7      df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)9-May", "05-09"))
8      # Replace all instances of "005-09" with "05-09" in column: 'AgeGrp'
9      df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)005-09", "05-09"))
10     # Replace all instances of "#N/A" with "N/A" in column: 'AgeGrp'
11     df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)#N/A", "N/A"))
12     # Replace all instances of "NA" with "N/A" in column: 'AgeGrp'
13     df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)NA", "N/A"))
14     # Replace all instances of "14-Oct" with "10-14" in column: 'AgeGrp'
15     df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)14-Oct", "10-14"))
16     # Replace all instances of "0-4" with "00-04" in column: 'AgeGrp'
17     df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)0-4", "00-04"))
18     # Replace all instances of "400-044" with "40-44" in column: 'AgeGrp'
19     df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)400-044", "40-44"))
20     return df
21
22 df_clean = clean_data(df)
23 display(df_clean)

```

✓ 7 sec - Command executed in 7 sec 821 ms by RDAzure on 12:19:00 PM, 4/28/25

>  Spark jobs (5 of 5 succeeded)  Resources  Log

<div> <div> Table</div> <div> New chart</div> </div>						
Table view						
	ABC Location	ABC Time	ABC AgeGrp	ABC PopMale	ABC PopFemale	
1	Botswana	2000	100+	0.001	0.002	
2	Ghana	2000	100+	0.001	0.002	
3	Botswana	2001	100+	0.001	0.002	

```
1 # Code generated by Data Wrangler for PySpark DataFrame
2
3 from pyspark.sql import types as T
4
5 def clean_data(df):
6     # Change column type to int64 for column: 'Time'
7     df = df.withColumn('Time', df['Time'].cast(T.LongType()))
8     # Change column type to float64 for column: 'PopMale'
9     df = df.withColumn('PopMale', df['PopMale'].cast(T.DoubleType()))
10    # Change column type to float64 for column: 'PopFemale'
11    df = df.withColumn('PopFemale', df['PopFemale'].cast(T.DoubleType()))
12    return df
13
14 df_clean = clean_data(df)
15 display(df_clean)
```

[6] ✓ 1 sec - Command executed in 1 sec 503 ms by RDAzure on 12:19:12 PM, 4/28/25

> Spark jobs (1 of 1 succeeded) Resources

Table + New chart

Table view

	ABC Location	12L Time	ABC AgeGrp	12 PopMale	12 PopFemale	
1	Botswana	2000	100+	0.001	0.002	
2	Ghana	2000	100+	0.001	0.002	
3	Botswana	2001	100+	0.001	0.002	
4	Ghana	2001	100+	0.001	0.002	
5	Botswana	2002	100+	0.001	0.002	

```
1 # Code generated by Data Wrangler for PySpark DataFrame
2
3 def clean_data(df_clean):
4     # Rename column 'PopMale' to 'Male'
5     df_clean = df_clean.withColumnRenamed('PopMale', 'Male')
6     # Rename column 'PopFemale' to 'Female'
7     df_clean = df_clean.withColumnRenamed('PopFemale', 'Female')
8     return df_clean
9
10 df_clean_1 = clean_data(df_clean)
11 display(df_clean_1)
```

[7] ✓ 1 sec - Command executed in 1 sec 462 ms by RDAzure on 12:19:23 PM, 4/28/25

> Spark jobs (1 of 1 succeeded) Resources

Table + New chart

Table view

	ABC Location	12L Time	ABC AgeGrp	12 Male	12 Female	
1	Botswana	2000	100+	0.001	0.002	
2	Ghana	2000	100+	0.001	0.002	
3	Botswana	2001	100+	0.001	0.002	
4	Ghana	2001	100+	0.001	0.002	
5	Botswana	2002	100+	0.001	0.002	

```

1 #pivot Male Female column
2 df_clean_2 = df_clean_1.unpivot(['Location', 'Time', 'AgeGrp'], ['Male', 'Female'], "Gender", "Population")
3
4

```

[8] ✓ <1 sec - Command executed in 345 ms by RDAzure on 12:19:34 PM, 4/28/25

```

1 #writting to Silver_LH
2 df_clean_2.write.format("delta").mode("overwrite").saveAsTable('Silver_LH.Silver_fact')
3

```

[9] ✓ 33 sec - Command executed in 33 sec 162 ms by RDAzure on 12:20:17 PM, 4/28/25

> Spark jobs (8 of 8 succeeded) Resources Log

## Gold layer reporting / Dashboarding:

