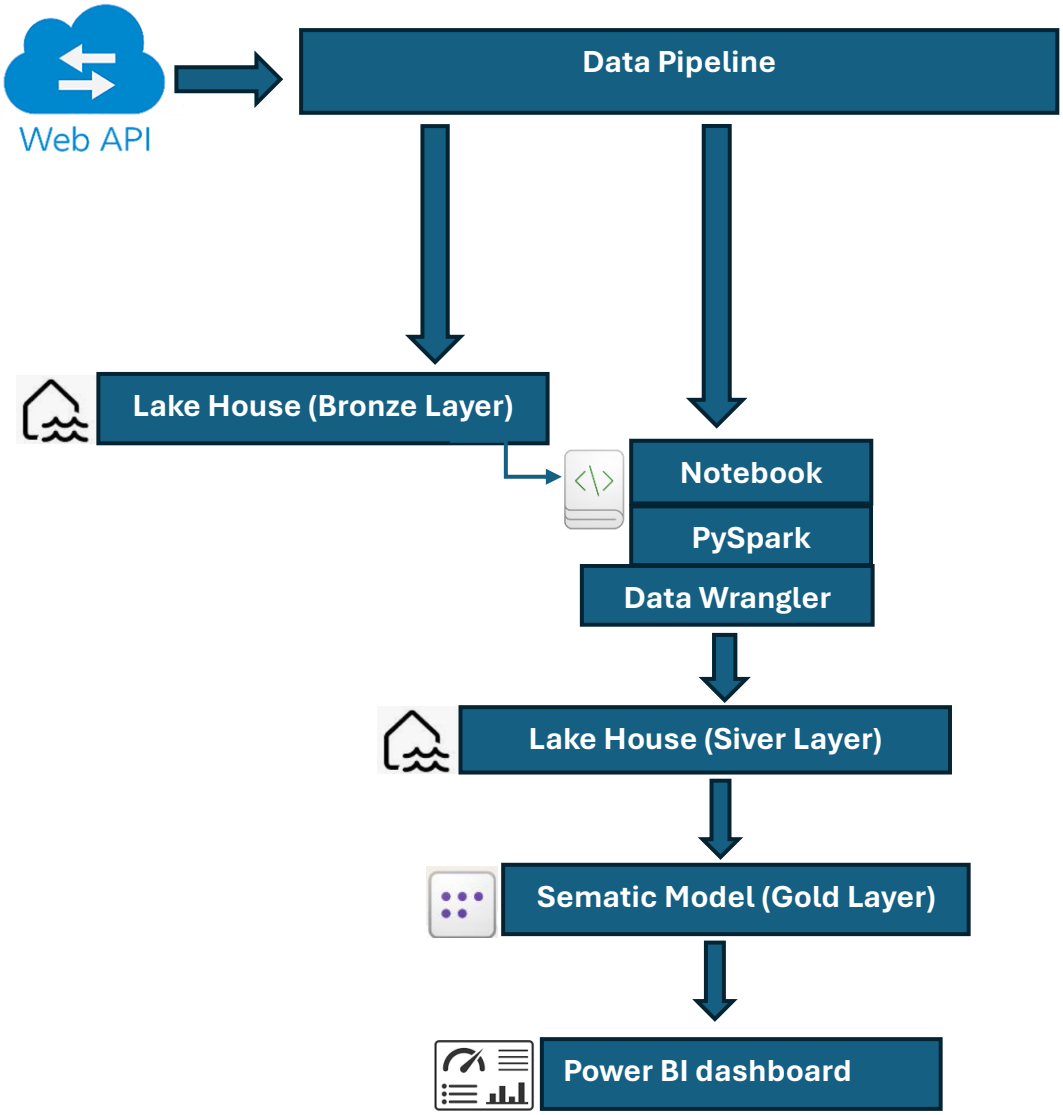
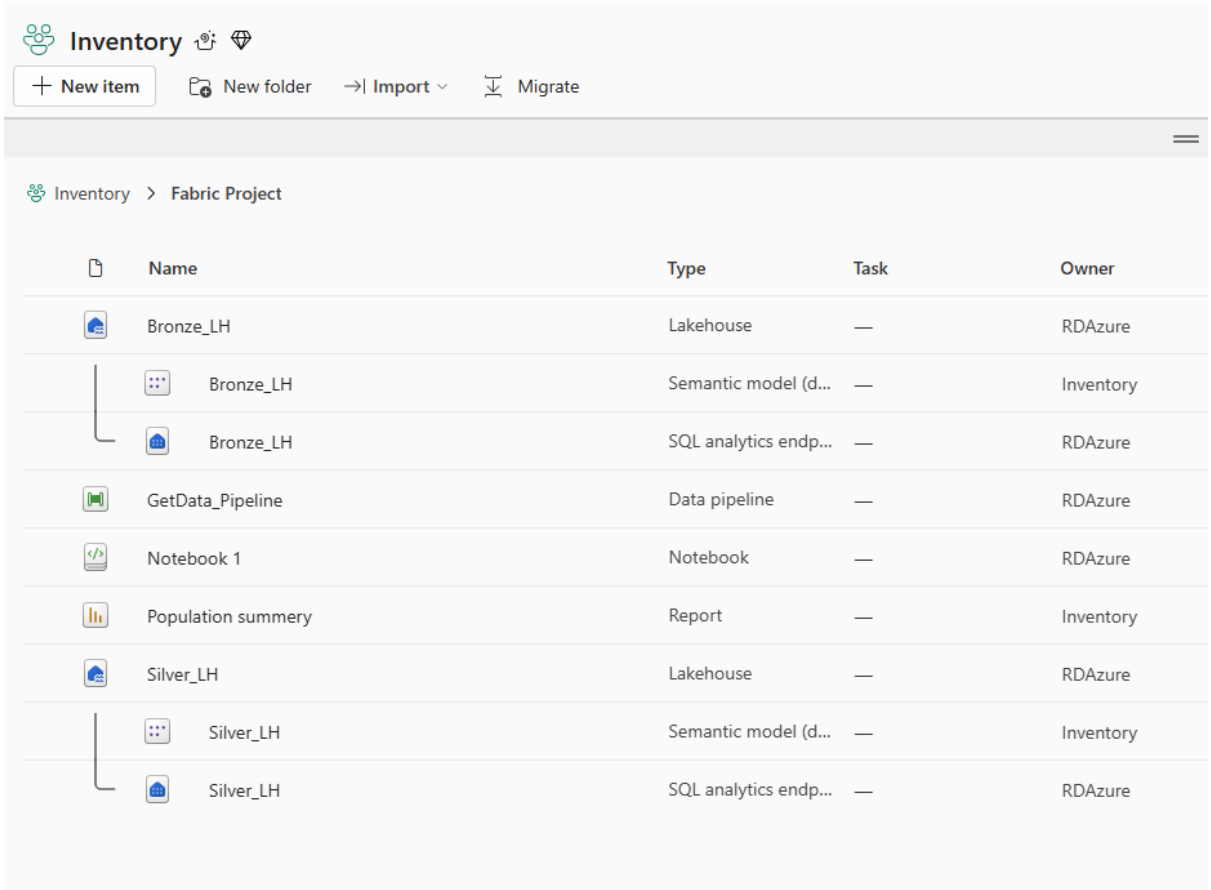
World Population Data analysis with Data Engineering and Data Analytics in Microsoft fabric.

Components used:

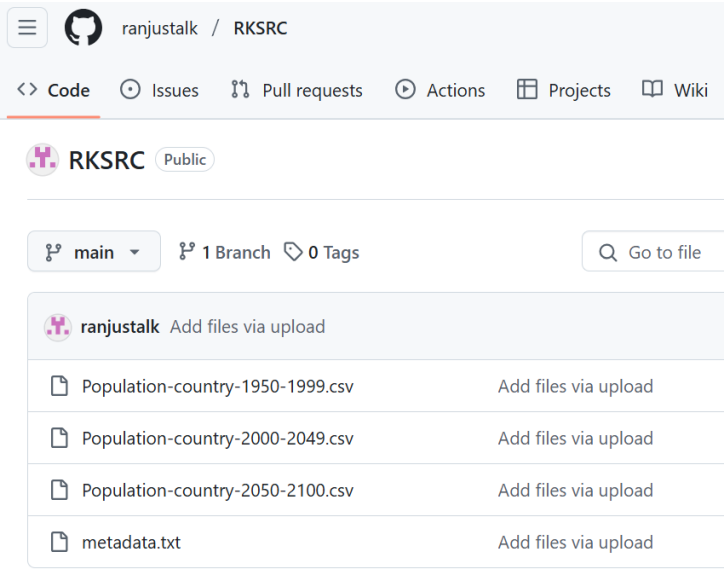
- Web API
- Data Pipeline
- Lake house
- Notebook
- Data wrangler
- PySpark
- Semantic model
- Power BI report



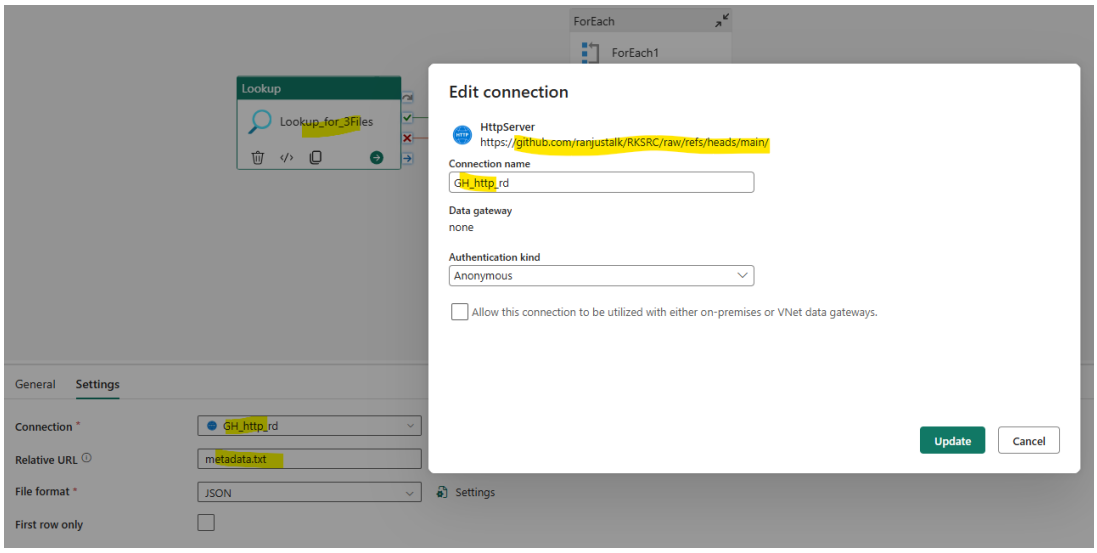


Source:

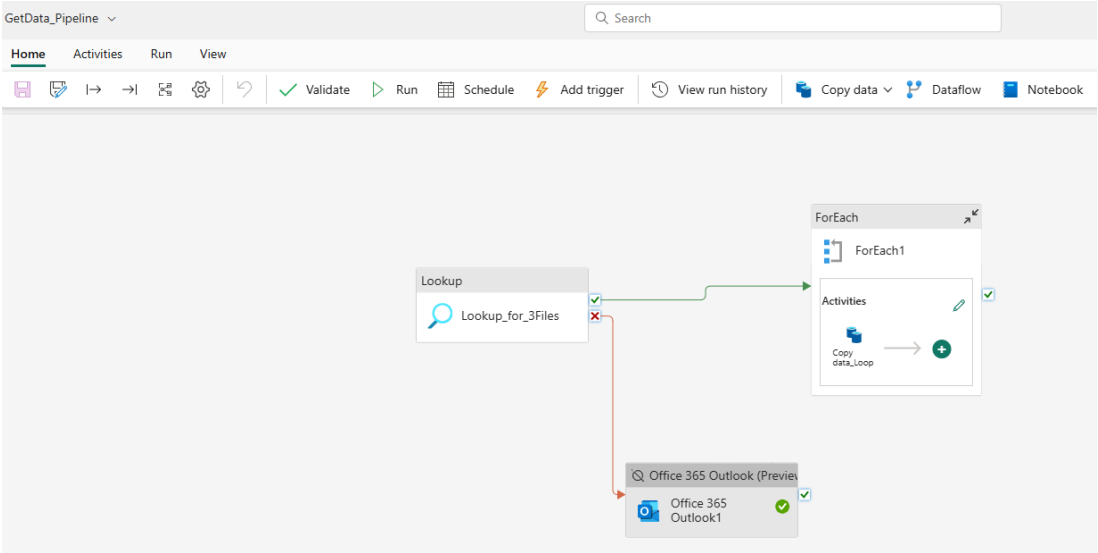
https://github.com/ranjustalk/RKSRC/

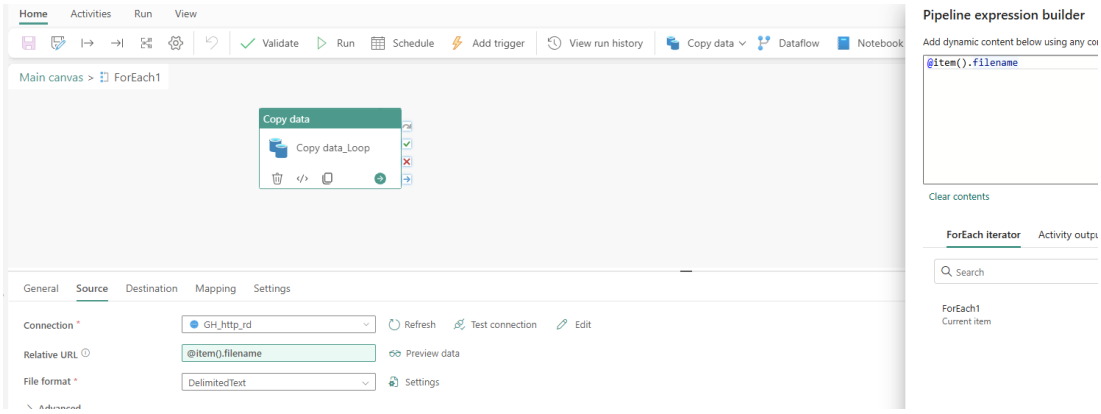


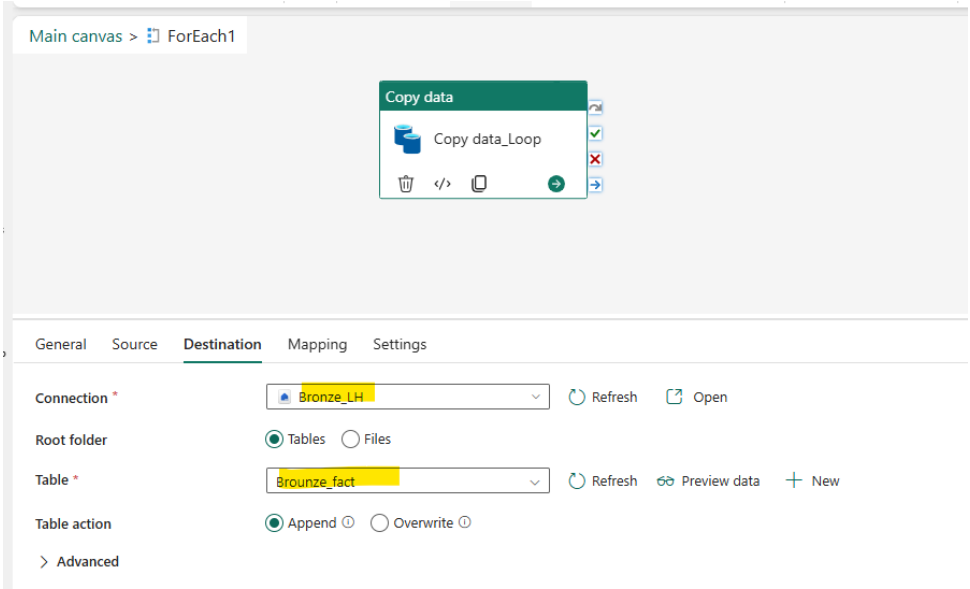
Connection to Source:



Bronze layer (Data pipeline):

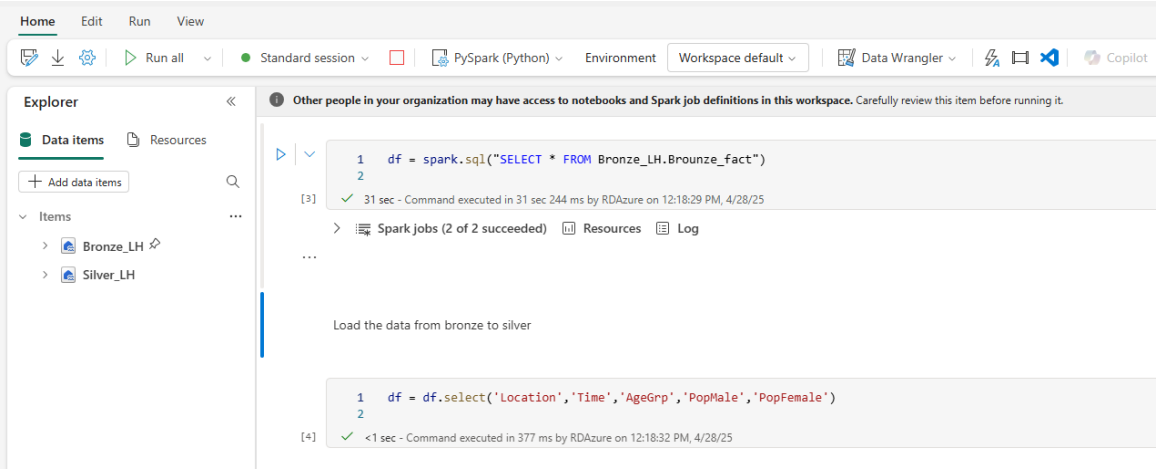






Silver layer (Notebook):

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



```python
# Code generated by Data Wrangler for PySpark DataFrame

from pyspark.sql import functions as F

def clean_data(df):
    # Replace all instances of "9-May" with "05-09" in column: 'AgeGrp'
    df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)9-May", "05-09"))
    # Replace all instances of "005-09" with "05-09" in column: 'AgeGrp'
    df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)005-09", "05-09"))
    # Replace all instances of "#N/A" with "N/A" in column: 'AgeGrp'
    df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)#N/A", "N/A"))
    # Replace all instances of "NA" with "N/A" in column: 'AgeGrp'
    df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)NA", "N/A"))
    # Replace all instances of "14-Oct" with "10-14" in column: 'AgeGrp'
    df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)14-Oct", "10-14"))
    # Replace all instances of "0-4" with "00-04" in column: 'AgeGrp'
    df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)0-4", "00-04"))
    # Replace all instances of "400-044" with "40-44" in column: 'AgeGrp'
    df = df.withColumn('AgeGrp', F.regexp_replace('AgeGrp', "(?i)400-044", "40-44"))
    return df

df_clean = clean_data(df)
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

Table + New chart

Table view

	Location	Time	AgeGrp	PopMale	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

```python
# Code generated by Data Wrangler for PySpark DataFrame

from pyspark.sql import types as T

def clean_data(df):
    # Change column type to int64 for column: 'Time'
    df = df.withColumn('Time', df['Time'].cast(T.LongType()))
    # Change column type to float64 for column: 'PopMale'
    df = df.withColumn('PopMale', df['PopMale'].cast(T.DoubleType()))
    # Change column type to float64 for column: 'PopFemale'
    df = df.withColumn('PopFemale', df['PopFemale'].cast(T.DoubleType()))
    return df

df_clean = clean_data(df)
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

	Location	Time	AgeGrp	PopMale	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

```python
# Code generated by Data Wrangler for PySpark DataFrame

def clean_data(df_clean):
    # Rename column 'PopMale' to 'Male'
    df_clean = df_clean.withColumnRenamed('PopMale', 'Male')
    # Rename column 'PopFemale' to 'Female'
    df_clean = df_clean.withColumnRenamed('PopFemale', 'Female')
    return df_clean

df_clean_1 = clean_data(df_clean)
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

	Location	Time	AgeGrp	Male	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

```
#upivot Male Female column
df_clean_2 = df_clean_1.unpivot(['Location','Time','AgeGrp'], ['Male', 'Female'], "Gender", "Population")


```

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

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

```

[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: