Case Study – 4

ETL Pipeline

Name: Aathirainathan P

Date: 04-12-2024

---

1. Create an ETL pipeline of ingestion & transform and load queries on any data set and initiate the pipeline from workflow using notebook.

Step 1: Ingest the Data:

10:18 PM (4s) 1 Python

```
# Ingestion
customer_df = spark.read.format("delta").load("dbfs:/databricks-datasets/tpch/delta-001/customer")
orders_df = spark.read.format("delta").load("dbfs:/databricks-datasets/tpch/delta-001/orders")

# Display the data
print("Customer Data:")
customer_df.show(5)

print("Orders Data:")
orders_df.show(5)
```

(3) Spark Jobs

customer_df: pyspark.sql.dataframe.DataFrame = [c_custkey: long, c_name: string ... 6 more fields]

orders_df: pyspark.sql.dataframe.DataFrame = [o_orderkey: long, o_custkey: long ... 7 more fields]

```
Customer Data:
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+
|c_custkey|            c_name|           c_address|c_nationkey|        c_phone|c_acctbal|c_mktsegment|           c_comment|
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+
|   412445|Customer#000412445|0QAB3OjYnbP6mA0B,kgf|         21|31-421-403-4333|  5358.33|    BUILDING|arefully blithely...|
|   412446|Customer#000412446|5u8MSbyiC7J,7PuY4...|         20|30-487-949-7942|  9441.59|   MACHINERY|sleep according t...|
|   412447|Customer#000412447|HC4ZT62gKPgrjr ce...|          7|17-797-466-6308|  7868.75|  AUTOMOBILE|aggle blithely am...|
|   412448|Customer#000412448|         hJok1MMrDgH|          6|16-541-510-4964|  6060.98|   MACHINERY|ly silent request...|
|   412449|Customer#000412449|zAt1nZNG01gOhIqgy...|         14|24-710-983-5536|  4973.84|   HOUSEHOLD|refully final the...|
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+
only showing top 5 rows
```

```
Orders Data:
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
|o_orderkey|o_custkey|o_orderstatus|o_totalprice|o_orderdate|o_orderpriority|        o_clerk|o_shippriority|           o_comment|
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
|  13710944|   227285|            O|   162169.66| 1995-10-11|       1-URGENT|Clerk#000000432|            0|accounts. ruthles...|
|  13710945|   225010|            O|   252273.67| 1997-09-29|          5-LOW|Clerk#000002337|            0|ironic platelets ...|
|  13710946|   238820|            O|   179947.16| 1997-10-31|         2-HIGH|Clerk#000004135|            0|ole requests. reg...|
|  13710947|   581233|            O|    33843.49| 1995-05-25|         2-HIGH|Clerk#000000138|            0|arefully final pl...|
|  13710948|    10033|            O|    42500.65| 1995-09-04|4-NOT SPECIFIED|Clerk#000003398|            0|regular requests ...|
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
only showing top 5 rows
```

Step 2: Data Transformation:

1. Clean Column Names (Replace " " with "_"):

10:19 PM (<1s) 3 Python

```
# Clean column names
cleaned_customer_df = customer_df.toDF(*(c.replace(" ", "_").lower() for c in customer_df.columns))
cleaned_customer_df.show(5)
```

(1) Spark Jobs

cleaned_customer_df: pyspark.sql.dataframe.DataFrame = [c_custkey: long, c_name: string ... 6 more fields]

```
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+
|c_custkey|            c_name|           c_address|c_nationkey|        c_phone|c_acctbal|c_mktsegment|           c_comment|
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+
|   412445|Customer#000412445|0QAB3OjYnbP6mA0B,kgf|         21|31-421-403-4333|  5358.33|    BUILDING|arefully blithely...|
|   412446|Customer#000412446|5u8MSbyiC7J,7PuY4...|         20|30-487-949-7942|  9441.59|   MACHINERY|sleep according t...|
|   412447|Customer#000412447|HC4ZT62gKPgrjr ce...|          7|17-797-466-6308|  7868.75|  AUTOMOBILE|aggle blithely am...|
|   412448|Customer#000412448|         hJok1MMrDgH|          6|16-541-510-4964|  6060.98|   MACHINERY|ly silent request...|
|   412449|Customer#000412449|zAt1nZNG01gOhIqgy...|         14|24-710-983-5536|  4973.84|   HOUSEHOLD|refully final the...|
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+
only showing top 5 rows
```

2. Remove rows with null values in 'o_totalprice' column:

10:20 PM (1s) 4

```
from pyspark.sql.functions import col

# Remove rows with null values in 'o_totalprice' column
cleaned_orders_df = orders_df.filter(col("o_totalprice").isNotNull())
cleaned_orders_df.show(5)
```

(1) Spark Jobs

cleaned_orders_df: pyspark.sql.dataframe.DataFrame = [o_orderkey: long, o_custkey: long ... 7 more fields]

```
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
|o_orderkey|o_custkey|o_orderstatus|o_totalprice|o_orderdate|o_orderpriority|        o_clerk|o_shippriority|           o_comment|
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
|  13710944|   227285|            O|   162169.66| 1995-10-11|       1-URGENT|Clerk#000000432|            0|accounts. ruthles...|
|  13710945|   225010|            O|   252273.67| 1997-09-29|          5-LOW|Clerk#000002337|            0|ironic platelets ...|
|  13710946|   238820|            O|   179947.16| 1997-10-31|         2-HIGH|Clerk#000004135|            0|ole requests. reg...|
|  13710947|   581233|            O|    33843.49| 1995-05-25|         2-HIGH|Clerk#000000138|            0|arefully final pl...|
|  13710948|    10033|            O|    42500.65| 1995-09-04|4-NOT SPECIFIED|Clerk#000003398|            0|regular requests ...|
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
only showing top 5 rows
```

3. Convert order date column to Date type:

10:23 PM (1s) 5 Python

```
from pyspark.sql.functions import col, to_date

# Convert order date column to Date type
orders_df = orders_df.withColumn("o_orderdate", to_date(col("o_orderdate"), "yyyy-MM-dd"))
orders_df.show(5)
```

(1) Spark Jobs

orders_df: pyspark.sql.dataframe.DataFrame = [o_orderkey: long, o_custkey: long ... 7 more fields]

o_orderkey	o_custkey	o_orderstatus	o_totalprice	o_orderdate	o_orderpriority	o_clerk	o_shippriority	o_comment
13710944	227285	O	162169.66	1995-10-11	1-URGENT	Clerk#000000432	0	accounts. ruthles...
13710945	225010	O	252273.67	1997-09-29	5-LOW	Clerk#000002337	0	ironic platelets ...
13710946	238820	O	179947.16	1997-10-31	2-HIGH	Clerk#000004135	0	ole requests. reg...
13710947	581233	O	33843.49	1995-05-25	2-HIGH	Clerk#000000138	0	arefully final pl...
13710948	10033	O	42500.65	1995-09-04	4-NOT SPECIFIED	Clerk#000003398	0	regular requests ...

only showing top 5 rows

4. Remove duplicates based on customer key:

10:23 PM (3s) 6 Python

```
# Remove duplicates based on customer key
unique_customers = customer_df.dropDuplicates(["c_custkey"])
unique_customers.show(5)
```

(2) Spark Jobs

unique_customers: pyspark.sql.dataframe.DataFrame = [c_custkey: long, c_name: string ... 6 more fields]

c_custkey	c_name	c_address	c_nationkey	c_phone	c_acctbal	c_mktsegment	c_comment
6	Customer#000000006	sKZz0CsnMD7mp4Xd0...	20	30-114-968-4951	7638.57	AUTOMOBILE	tions. even depos...
7	Customer#000000007	TcGe5gaZNgVePxU5k...	18	28-190-982-9759	9561.95	AUTOMOBILE	ainst the ironic,...
19	Customer#000000019	uc,3bHIx84H,wdrmL...	18	28-396-526-5053	8914.71	HOUSEHOLD	nag. furiously c...
22	Customer#000000022	QI6p41,FNs5k7RZoC...	3	13-806-545-9701	591.98	MACHINERY	s nod furiously a...
25	Customer#000000025	Hp8GyFQgGHFYSilH5...	12	22-603-468-3533	7133.70	FURNITURE	y. accounts sleep...

only showing top 5 rows

5. Filter orders for the year 1995:

10:21 PM (1s) 7 Python

```
# Filter orders for the year 1995
orders_1995 = orders_df.filter(col("o_orderdate").like("1995%"))
orders_1995.show(5)
```

(1) Spark Jobs

orders_1995: pyspark.sql.dataframe.DataFrame = [o_orderkey: long, o_custkey: long ... 7 more fields]

o_orderkey	o_custkey	o_orderstatus	o_totalprice	o_orderdate	o_orderpriority	o_clerk	o_shippriority	o_comment
13710944	227285	O	162169.66	1995-10-11	1-URGENT	Clerk#000000432	0	accounts. ruthles...
13710947	581233	O	33843.49	1995-05-25	2-HIGH	Clerk#000000138	0	arefully final pl...
13710948	10033	O	42500.65	1995-09-04	4-NOT SPECIFIED	Clerk#000003398	0	regular requests ...
13710949	615502	O	48225.35	1995-07-13	3-MEDIUM	Clerk#000004639	0	ate quickly along...
13711044	254206	O	243977.92	1995-11-07	5-LOW	Clerk#000001680	0	ial, ironic pinto...

only showing top 5 rows

6. Calculate total revenue per order :

Just now (1s) 8 Python

```
# Calculate total revenue per order
from pyspark.sql.functions import when, col

orders_with_revenue = orders_df.withColumn(
    "revenue",
    when(col("o_orderstatus") == "O", col("o_totalprice")).otherwise(None)
)

# Show the result
orders_with_revenue.show(10)
```

(1) Spark Jobs

orders_with_revenue: pyspark.sql.dataframe.DataFrame = [o_orderkey: long, o_custkey: long ... 8 more fields]

o_orderkey	o_custkey	o_orderstatus	o_totalprice	o_orderdate	o_orderpriority	o_clerk	o_shippriority	o_comment	revenue
13710944	227285	O	162169.66	1995-10-11	1-URGENT	Clerk#000000432	0	accounts. ruthles...	162169.66
13710945	225010	O	252273.67	1997-09-29	5-LOW	Clerk#000002337	0	ironic platelets ...	252273.67
13710946	238820	O	179947.16	1997-10-31	2-HIGH	Clerk#000004135	0	ole requests. reg...	179947.16
13710947	581233	O	33843.49	1995-05-25	2-HIGH	Clerk#000000138	0	arefully final pl...	33843.49
13710948	10033	O	42500.65	1995-09-04	4-NOT SPECIFIED	Clerk#000003398	0	regular requests ...	42500.65
13710949	615502	O	48225.35	1995-07-13	3-MEDIUM	Clerk#000004639	0	ate quickly along...	48225.35
13710950	710665	F	265761.00	1992-11-29	2-HIGH	Clerk#000000735	0	, sly ideas among...	NULL
13710951	382528	F	137666.86	1993-05-21	5-LOW	Clerk#000000777	0	. blithely pendin...	NULL
13710976	122618	O	158725.42	1998-03-06	4-NOT SPECIFIED	Clerk#000001281	0	ages. final packa...	158725.42
13710977	575623	O	178703.66	1998-05-04	5-LOW	Clerk#000003371	0	, final requests ...	178703.66

7. Aggregate data to calculate total orders per customer:

10:22 PM (3s) 9 Python

```
from pyspark.sql.functions import count

# Aggregate data to calculate total orders per customer
customer_order_count = orders_df.groupBy("o_custkey").agg(count("o_orderkey").alias("total_orders"))
customer_order_count.show(5)
```

(2) Spark Jobs

customer_order_count: pyspark.sql.dataframe.DataFrame = [o_custkey: long, total_orders: long]

```
+---------+------------+
|o_custkey|total_orders|
+---------+------------+
|   105784|          18|
|   215485|          27|
|    51418|          13|
|   212203|          18|
|   295565|          13|
+---------+------------+
only showing top 5 rows
```

8. Join customer and orders data:

10:22 PM (8s) 10 Python

```
# Join customer and orders data
customer_orders = customer_df.join(orders_df, customer_df["c_custkey"] == orders_df["o_custkey"], "inner")
customer_orders.show(5)
```

(3) Spark Jobs

customer_orders: pyspark.sql.dataframe.DataFrame = [c_custkey: long, c_name: string ... 15 more fields]

```
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+----------+---------+---
----------+------------+-----------+---------------+---------------+--------------+--------------------+
|c_custkey|            c_name|           c_address|c_nationkey|        c_phone|c_acctbal|c_mktsegment|           c_comment|o_orderkey|o_custkey|o_o
rderstatus|o_totalprice|o_orderdate|o_orderpriority|        o_clerk|o_shippriority|           o_comment|
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+----------+---------+---
----------+------------+-----------+---------------+---------------+--------------+--------------------+
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|  13949350|        7|
O|   217862.05| 1997-12-05|          5-LOW|Clerk#000001887|             0|uthless requests....|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|   7325473|        7|
F|   111957.12| 1992-05-30|       3-MEDIUM|Clerk#000000446|             0|tes. furiously ir...|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|  25815556|        7|
F|   212906.99| 1992-10-03|4-NOT SPECIFIED|Clerk#000004413|             0|refully pinto bea...|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|  13668358|        7|
O|   173327.17| 1997-08-23|4-NOT SPECIFIED|Clerk#000004723|             0|d waters sleep ev...|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|  29797575|        7|
F|    71906.44| 1994-02-06|       3-MEDIUM|Clerk#000004415|             0|sts-- deposits al...|
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+----------+---------+---
----------+------------+-----------+---------------+---------------+--------------+--------------------+
only showing top 5 rows
```

9. Group orders by year and month to calculate total sales:

10:25 PM (3s) 11 Python

```
from pyspark.sql.functions import col, year, month, sum

# Group orders by year and month to calculate total sales
orders_by_month = orders_df.groupBy(year(col("o_orderdate")).alias("year"), month(col("o_orderdate")).alias("month")) \
                           .agg(sum("o_totalprice").alias("total_sales"))
orders_by_month.show(5)
```

(2) Spark Jobs

orders_by_month: pyspark.sql.dataframe.DataFrame = [year: integer, month: integer ... 1 more field]

```
+----+-----+--------------+
|year|month|   total_sales|
+----+-----+--------------+
|1997|   11|14008155122.62|
|1998|    2|13231342086.42|
|1995|   12|14569536356.53|
|1998|    7|14615808096.95|
|1994|    3|14584304371.16|
+----+-----+--------------+
only showing top 5 rows
```

10. Filter orders with total price greater than a specific value and sort by order date:

10:25 PM (3s) 12 Python

```
from pyspark.sql.functions import col

# Filter orders with total price greater than a specific value and sort by order date
filtered_orders_df = orders_df.filter(col("o_totalprice") > 1000) \
                              .orderBy(col("o_orderdate"))

filtered_orders_df.show(5)
```

(1) Spark Jobs

filtered_orders_df: pyspark.sql.dataframe.DataFrame = [o_orderkey: long, o_custkey: long ... 7 more fields]

```
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
|o_orderkey|o_custkey|o_orderstatus|o_totalprice|o_orderdate|o_orderpriority|        o_clerk|o_shippriority|           o_comment|
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
|   5640354|   292904|            F|    82036.60| 1992-01-01|4-NOT SPECIFIED|Clerk#000000575|            0|xes. ironic, spec...|
|  13743521|   228632|            F|   186961.35| 1992-01-01|       3-MEDIUM|Clerk#000000594|            0|s boost boldly bo...|
|  11463840|   207124|            F|   215760.02| 1992-01-01|       3-MEDIUM|Clerk#000004807|            0| along the blithe...|
|  13792930|   716917|            F|   192238.45| 1992-01-01|         2-HIGH|Clerk#000003787|            0|ily against the b...|
|   5663939|   249938|            F|   196815.58| 1992-01-01|         2-HIGH|Clerk#000004884|            0| deposits nag bli...|
+----------+---------+-------------+------------+-----------+---------------+---------------+-------------+--------------------+
only showing top 5 rows
```

Step 3: Load the data:

1. Write the filtered orders data in Delta format:

10:33 PM (20s) 14 Python

```
# Specify the output path for the transformed data
output_path = "dbfs:/mnt/mount/tpch_filtered_orders_delta"

# Write the filtered orders data in Delta format
filtered_orders_df.write.format("delta").mode("overwrite").save(output_path)

# To confirm, read the data back
filtered_orders_df_loaded = spark.read.format("delta").load(output_path)
filtered_orders_df_loaded.show(5)
```

(10) Spark Jobs

filtered_orders_df_loaded: pyspark.sql.dataframe.DataFrame = [o_orderkey: long, o_custkey: long ... 7 more fields]

```
+----------+---------+-------------+------------+-----------+---------------+---------------+--------------+--------------------+
|o_orderkey|o_custkey|o_orderstatus|o_totalprice|o_orderdate|o_orderpriority|        o_clerk|o_shippriority|           o_comment|
+----------+---------+-------------+------------+-----------+---------------+---------------+--------------+--------------------+
|  13716320|   428330|            F|   116124.79| 1992-01-01|       3-MEDIUM|Clerk#000004084|             0| foxes. slyly reg...|
|  13716672|   717422|            F|    56846.82| 1992-01-01|          5-LOW|Clerk#000003115|             0|fily after the ca...|
|  13717345|     3418|            F|    34274.24| 1992-01-01|          5-LOW|Clerk#000003832|             0|s. even theodolit...|
|  13723650|   101623|            F|   120543.82| 1992-01-01|          5-LOW|Clerk#000004494|             0|ar, bold pearls i...|
|  13742627|   540439|            F|    20629.17| 1992-01-01|       1-URGENT|Clerk#000003682|             0|y special instruc...|
+----------+---------+-------------+------------+-----------+---------------+---------------+--------------+--------------------+
only showing top 5 rows
```

2. Write the customer orders data in Delta format:

10:35 PM (32s) 15 Python

```
# Specify the output path for the transformed data
output_path = "dbfs:/mnt/mount/customer_orders"

# Write the customer orders data in Delta format
customer_orders.write.format("delta").mode("overwrite").save(output_path)

# To confirm, read the data back
customer_orders_loaded = spark.read.format("delta").load(output_path)

# Show the first 5 rows of the loaded Delta data
customer_orders_loaded.show(5)
```

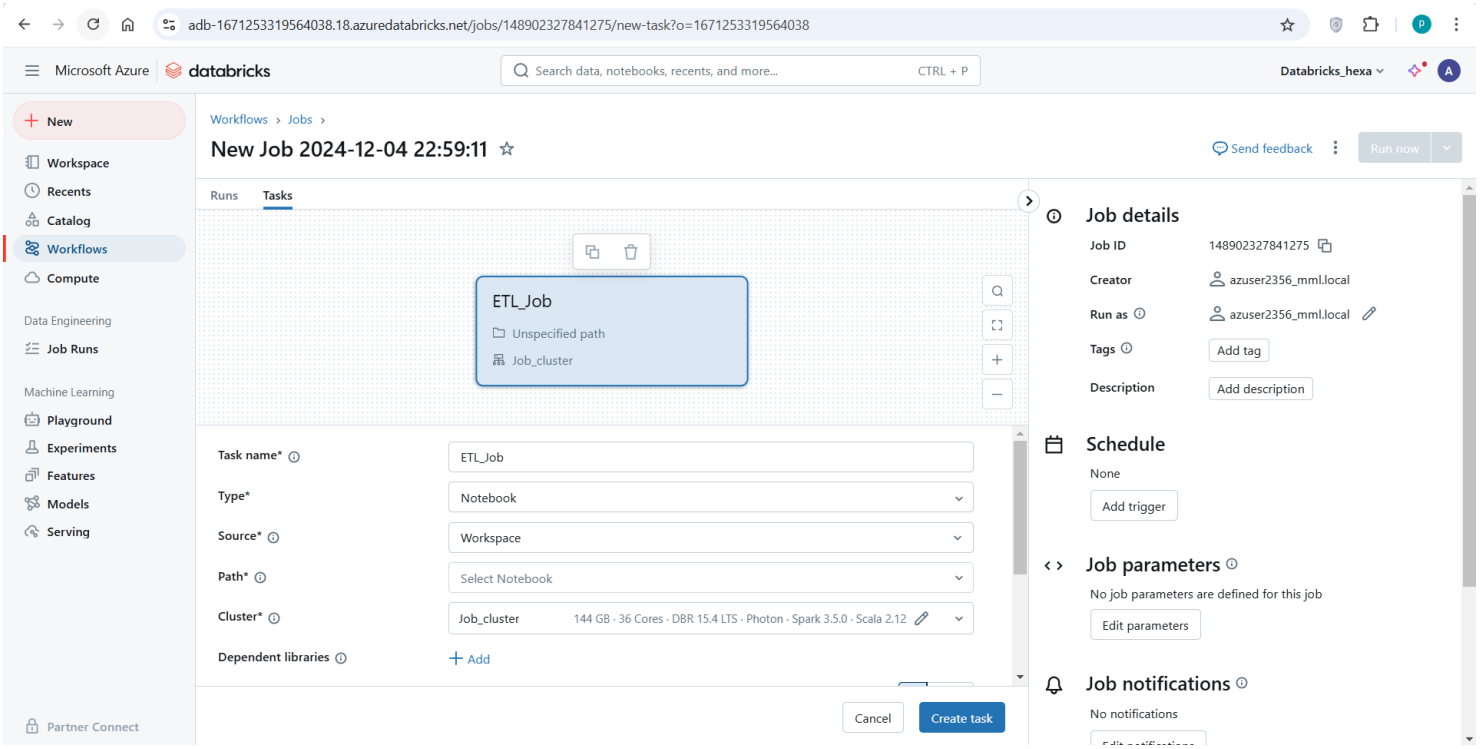
(10) Spark Jobs

customer_orders_loaded: pyspark.sql.dataframe.DataFrame = [c_custkey: long, c_name: string ... 15 more fields]

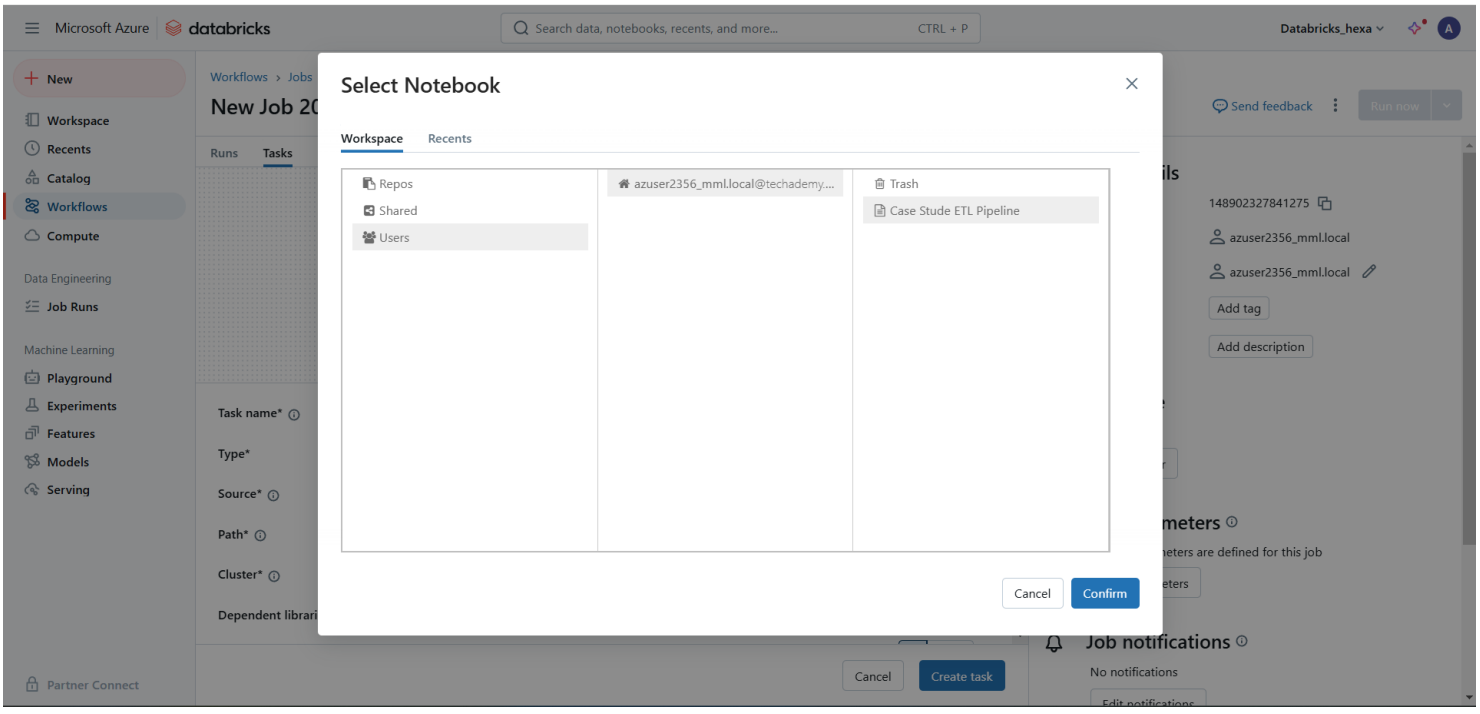
```
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+----------+---------+---
----------+------------+-----------+---------------+---------------+--------------+--------------------+
|c_custkey|            c_name|           c_address|c_nationkey|        c_phone|c_acctbal|c_mktsegment|           c_comment|o_orderkey|o_custkey|o_o
rderstatus|o_totalprice|o_orderdate|o_orderpriority|        o_clerk|o_shippriority|           o_comment|
+---------+------------------+--------------------+-----------+---------------+---------+------------+--------------------+----------+---------+---
----------+------------+-----------+---------------+---------------+--------------+--------------------+
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|  13949350|        7|
O|   217862.05| 1997-12-05|          5-LOW|Clerk#000001887|             0|uthless requests....|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|   7325473|        7|
F|   111957.12| 1992-05-30|       3-MEDIUM|Clerk#000000446|             0|tes. furiously ir...|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|  25815556|        7|
F|   212906.99| 1992-10-03|4-NOT SPECIFIED|Clerk#000004413|             0|refully pinto bea...|
|        7|Customer#000000007|TcGe5gaZNgVePxU5k...|         18|28-190-982-9759|  9561.95|  AUTOMOBILE|ainst the ironic,...|  13668358|        7|
O|   173327.17| 1997-08-23|4-NOT SPECIFIED|Clerk#000004723|             0|d waters sleep ev...|
```

Step 4 : Creating a Workflow:

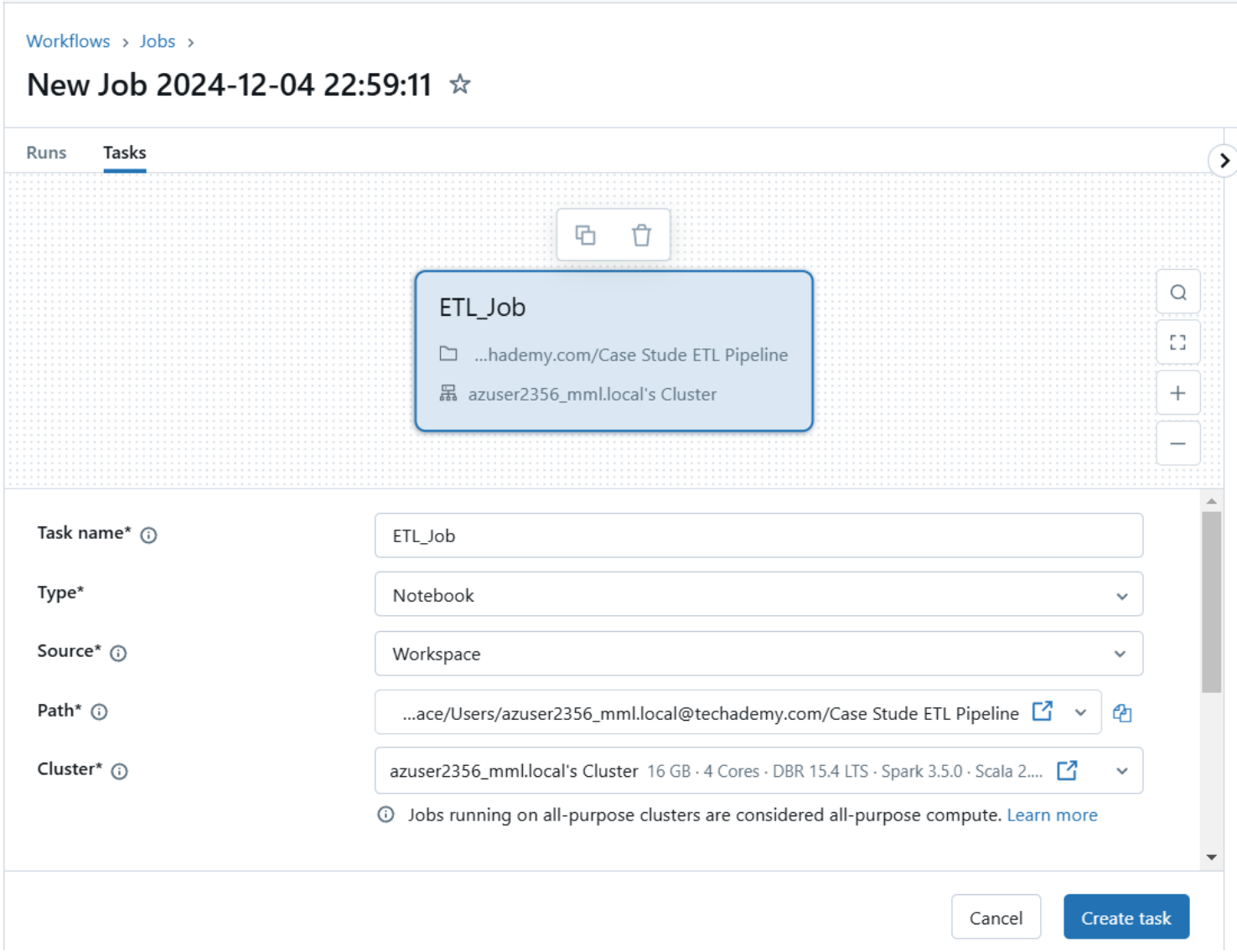
Define a Task name:



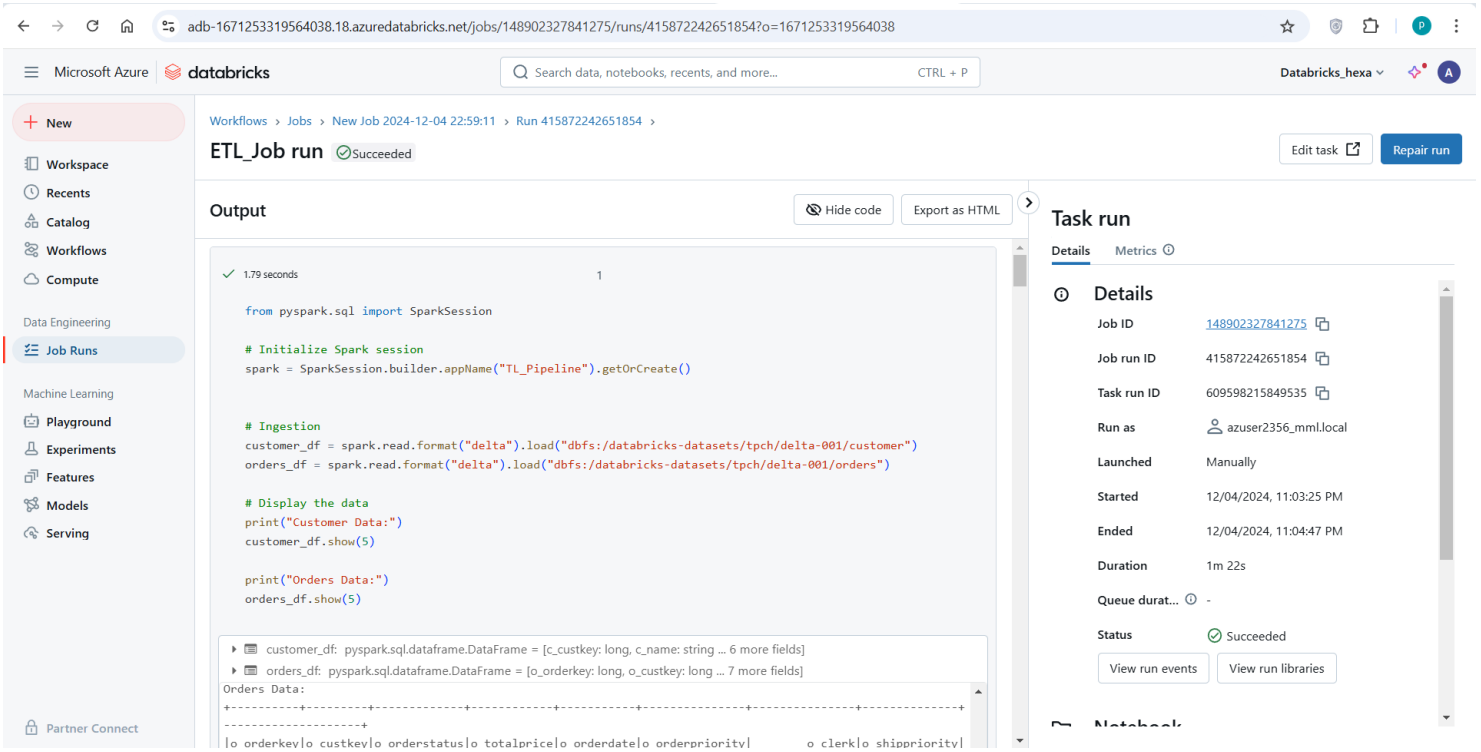
Select the notebook that has ingestion, transformation and loading:



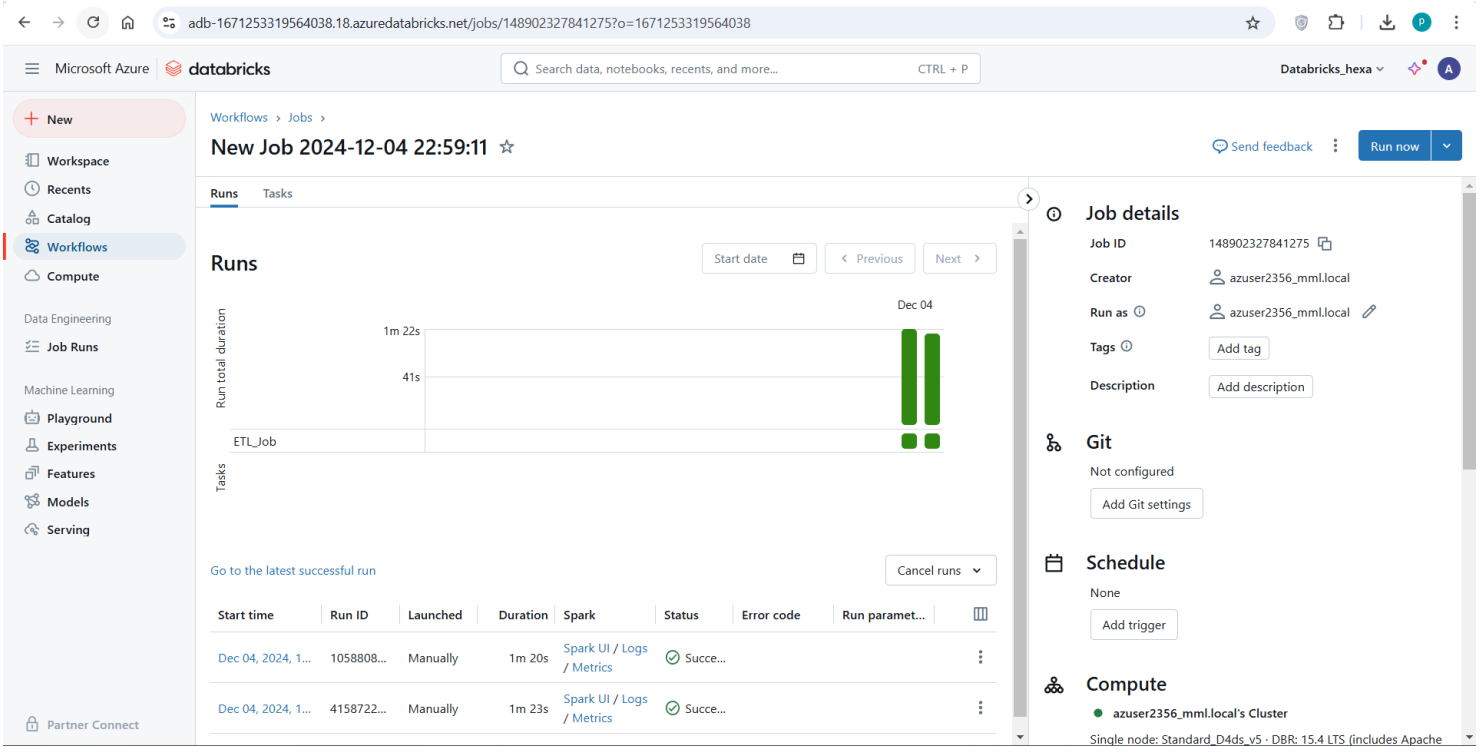
Choose a cluster:



Run:



View Job Runs:



Submitted By:
Aathirainathan P