

Case Study-4

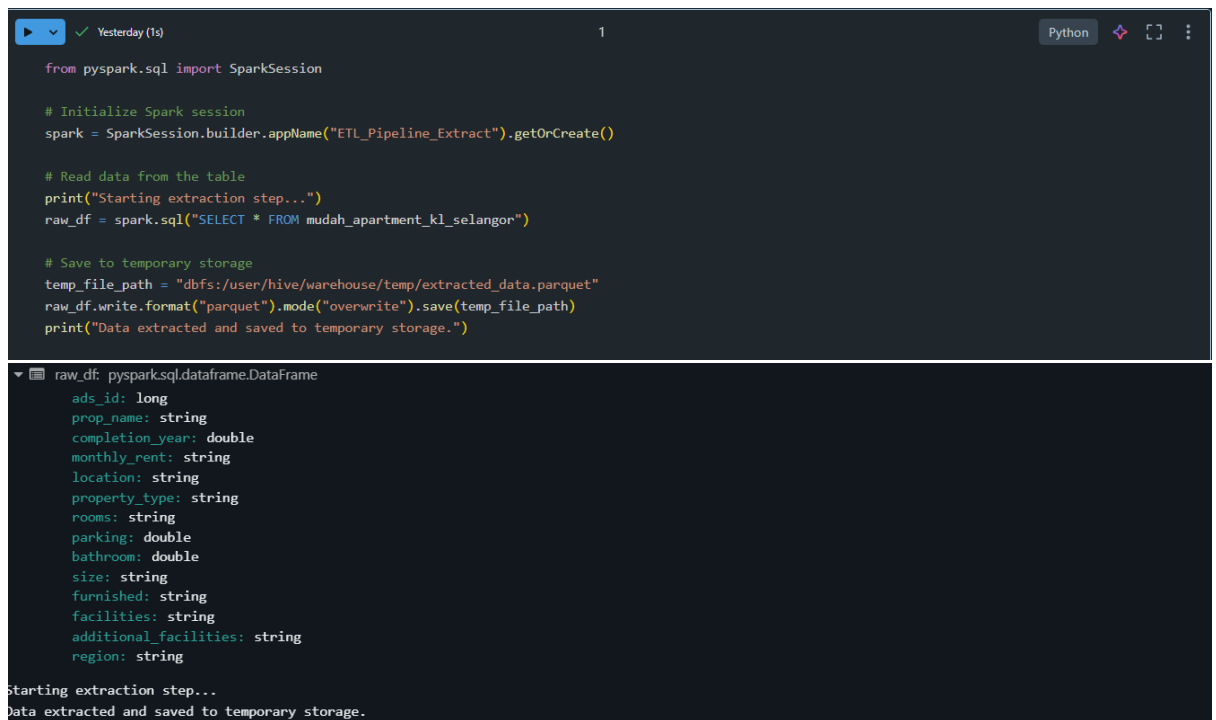
Azure Databricks Case Study

Name: **Harish E.R**

Q. create a ETL pipeline of ingestion & transform and load queries on any data set and initiate the pipeline from workflow using notebook.

1. Notebook for Ingestion (Extract)

This notebook will load the raw data into the Databricks environment, either from a file or a table.



```
from pyspark.sql import SparkSession

# Initialize Spark session
spark = SparkSession.builder.appName("ETL_Pipeline_Extract").getOrCreate()

# Read data from the table
print("Starting extraction step...")
raw_df = spark.sql("SELECT * FROM mudah_apartment_kl_selangor")

# Save to temporary storage
temp_file_path = "dbfs:/user/hive/warehouse/temp/extracted_data.parquet"
raw_df.write.format("parquet").mode("overwrite").save(temp_file_path)
print("Data extracted and saved to temporary storage.")
```

▼ raw_df: pyspark.sql.dataframe.DataFrame

```
ads_id: long
prop_name: string
completion_year: double
monthly_rent: string
location: string
property_type: string
rooms: string
parking: double
bathroom: double
size: string
furnished: string
facilities: string
additional_facilities: string
region: string
```

Starting extraction step...
Data extracted and saved to temporary storage.

2. Notebook for Transformation

This notebook will read the extracted data, apply transformations, and prepare it for loading.

```
▶ Yesterday (1s) 1 Python
```

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col

# Initialize Spark session
spark = SparkSession.builder.appName("ETL_Pipeline_Transform").getOrCreate()

# Paths
temp_file_path = "dbfs:/user/hive/warehouse/temp/extracted_data.parquet"
transformed_file_path = "dbfs:/user/hive/warehouse/temp/transformed_data.parquet"

# Load intermediate data
print("Starting transformation step...")
raw_df = spark.read.format("parquet").load(temp_file_path)

# Inspect schema
print("Schema of the dataset:")
raw_df.printSchema()

# Perform transformations
transformed_df = (
    raw_df
    .select("rooms", "region", "size")
    .filter(col("rooms").isNotNull())
    .withColumnRenamed("rooms", "new_rooms")
)

# Save transformed data
transformed_df.write.format("parquet").mode("overwrite").save(transformed_file_path)
print("Data transformed and saved to temporary storage.")
```

▶ (2) Spark Jobs

- ▶ raw_df: pyspark.sql.dataframe.DataFrame = [ads_id: long, prop_name: string ... 12 more fields]
- ▶ transformed_df: pyspark.sql.dataframe.DataFrame = [new_rooms: string, region: string ... 1 more field]

Starting transformation step...

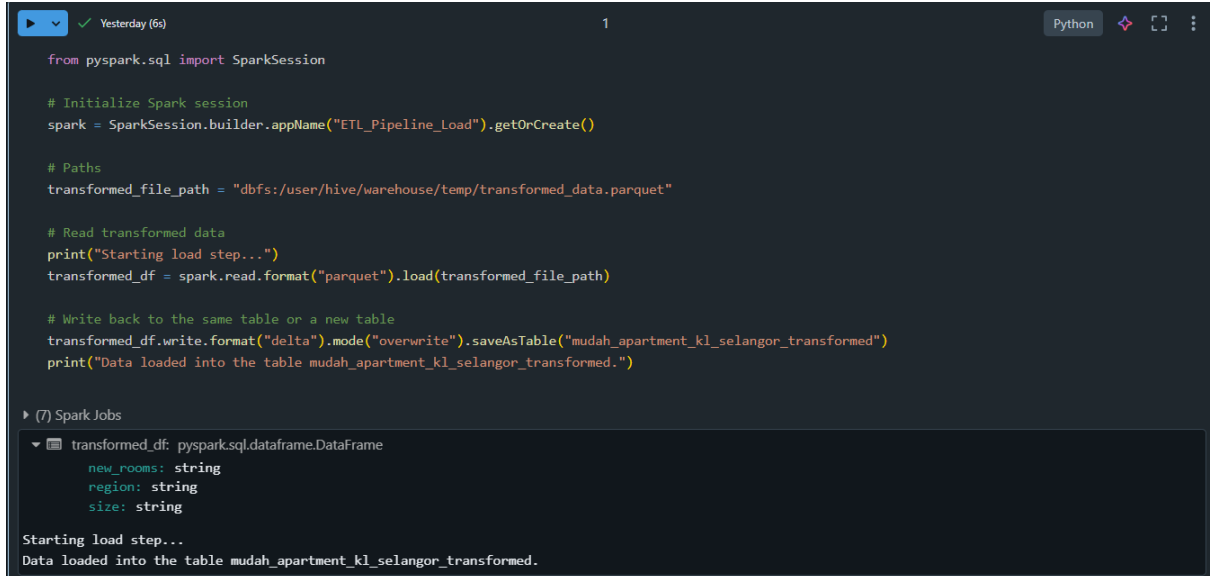
Schema of the dataset:

```
root
|-- ads_id: long (nullable = true)
|-- prop_name: string (nullable = true)
|-- completion_year: double (nullable = true)
|-- monthly_rent: string (nullable = true)
|-- location: string (nullable = true)
|-- property_type: string (nullable = true)
|-- rooms: string (nullable = true)
|-- parking: double (nullable = true)
|-- bathroom: double (nullable = true)
|-- size: string (nullable = true)
|-- furnished: string (nullable = true)
|-- facilities: string (nullable = true)
|-- additional_facilities: string (nullable = true)
|-- region: string (nullable = true)
```

Data transformed and saved to temporary storage.

3. Notebook for Loading

This notebook will load the transformed data into the final destination, such as a Delta table or another storage format.



```
from pyspark.sql import SparkSession

# Initialize Spark session
spark = SparkSession.builder.appName("ETL_Pipeline_Load").getOrCreate()

# Paths
transformed_file_path = "dbfs:/user/hive/warehouse/temp/transformed_data.parquet"

# Read transformed data
print("Starting load step...")
transformed_df = spark.read.format("parquet").load(transformed_file_path)

# Write back to the same table or a new table
transformed_df.write.format("delta").mode("overwrite").saveAsTable("mudah_apartment_kl_selangor_transformed")
print("Data loaded into the table mudah_apartment_kl_selangor_transformed.")
```

▶ (7) Spark Jobs

transformed_df: pyspark.sql.dataframe.DataFrame

```
new_rooms: string
region: string
size: string
```

Starting load step...
Data loaded into the table mudah_apartment_kl_selangor_transformed.

4. Create a Databricks Workflow

The image displays three sequential screenshots of the Databricks Workflow Editor, illustrating the configuration of a workflow named "ETL_Pipeline_Workflow".

First Screenshot: The workflow diagram shows three tasks: `case_study_notebook_E`, `case_study_notebook_T`, and `case_study_notebook_L`. The configuration panel for `case_study_notebook_E` is active, showing the following details:

- Task name: `case_study_notebook_E`
- Type: Notebook
- Source: Workspace
- Path: `...orkspace/Users/azuser2365_mml.local@techademy.com/casestudy_notebook`
- Cluster: `Job_cluster` (144 GB - 36 Cores - DBR 15.4 LTS - Photon - Spark 3.5.0 - Scala 2.12)
- Depends on: Select task dependencies...

Second Screenshot: The configuration panel for `case_study_notebook_T` is active, showing the following details:

- Task name: `case_study_notebook_T`
- Type: Notebook
- Source: Workspace
- Path: `...kspace/Users/azuser2365_mml.local@techademy.com/case_study_notebook_t`
- Cluster: `Job_cluster` (144 GB - 36 Cores - DBR 15.4 LTS - Photon - Spark 3.5.0 - Scala 2.12)
- Depends on: `case_study_notebook_E`
- Run if dependencies: All succeeded
- Dependent libraries: + Add

Third Screenshot: The configuration panel for `case_study_notebook_L` is active, showing the following details:

- Task name: `case_study_notebook_L`
- Type: Notebook
- Source: Workspace
- Path: `...kspace/Users/azuser2365_mml.local@techademy.com/case_study_notebook_L`
- Cluster: `Job_cluster` (144 GB - 36 Cores - DBR 15.4 LTS - Photon - Spark 3.5.0 - Scala 2.12)
- Depends on: `case_study_notebook_T`
- Run if dependencies: All succeeded
- Dependent libraries: + Add

The right-hand panel of each screenshot displays job details for the selected task, including Job ID (425198602460938), Creator (azuser2365_mml.local), Run as (azuser2365_mml.local), Tags, Description, Git settings, Schedule (None), and Compute settings (Job_cluster).

5. Run the Workflow

Workflows > Jobs >

ETL_Pipeline_Workflow ☆

Send feedback

Run now

RunsTasks

Runs

Start datePreviousNext

Run total duration

6m 28s3m 14s

Dec 07

case_study_notebook_E

case_study_notebook_T

case_study_notebook_L

Tasks

Go to the latest successful run

Cancel runs

Start time	Run ID	Launched	Duration	Status	Error code	Run parameters
Dec 07, 2024, 10:57 AM	25496517849943	Manually	6m 29s	Succeeded		

Job details

Job ID4251986024600938

Creatorazuser2365_mmjlocal

Run asazuser2365_mmjlocal

TagsAdd tag

DescriptionAdd description

GitNot configured

Add Git settings

ScheduleNone

Add trigger

ComputeJob_cluster

Driver: Standard_D4ds_v5 · Workers: Standard_D4ds_v5 · 8 workers · DBR: 15.4x-photon-scala2.12

ConfigureSwap