1: Data Ingestion & Storage

Task 1: Download a Real-world Dataset

🔷 Dataset: **New York Taxi Trips Data**

📥 Download: NYC Taxi Data (Parquet format)

🔷 Alternative: **Kaggle Datasets** (Download CSV datasets)

Task 2: Load Data into a Local Database

- Install and Use **PostgreSQL** (or SQLite) as a database.
- Write a **Python script to load data** into the database.

Resources:

- PostgreSQL Installation Guide
- Pandas to PostgreSQL (Tutorial)
- SQLite Quickstart

📌 **Practice Steps:**

✅ Install **PostgreSQL** or **SQLite**.

✅ Use **Pandas** to **read the dataset**.

✅ Write a **Python script** to insert data into the database.

python ▶ Run Copy

```python
import pandas as pd
from sqlalchemy import create_engine

# Load dataset
df = pd.read_parquet("yellow_tripdata_2024-01.parquet")

# Connect to PostgreSQL
engine = create_engine("postgresql://user:password@localhost:5432/nyc_taxi")

# Load data into DB
df.to_sql('taxi_data', engine, if_exists='replace', index=False)
```

2: Data Processing & Transformation

Task 3: Transform Data Using Pandas & SQL

- **Filter out invalid data** (e.g., negative trip distances).
- **Convert datetime columns** into proper formats.
- **Aggregate data** (e.g., average fare per trip).

Resources:

- SQL Basics (W3Schools)
- Pandas Data Transformations

📌 **Practice Steps:**

✅ Write **SQL queries** to clean the data.

✅ Perform **aggregations** using Pandas.

sql Copy

```sql
SELECT
    pickup_datetime,
    dropoff_datetime,
    trip_distance,
    fare_amount
FROM taxi_data
WHERE trip_distance > 0;
```

python ▶ Run Copy

```python
# Remove rows with negative distances
df_cleaned = df[df["trip_distance"] > 0]

# Convert datetime columns
df_cleaned["pickup_datetime"] = pd.to_datetime(df_cleaned["pickup_datetime"])
df_cleaned["dropoff_datetime"] = pd.to_datetime(df_cleaned["dropoff_datetime"])

# Aggregate data
df_summary = df_cleaned.groupby(df_cleaned["pickup_datetime"].dt.date)["fare_amount"].mean()
print(df_summary.head())
```

3: Data Orchestration with Apache Airflow

Task 4: Automate Data Processing with Airflow

- Install **Apache Airflow** (pip install apache-airflow).
- Create an **Airflow DAG** (Directed Acyclic Graph) to automate:
    - **Ingesting data** from the dataset.
    - **Transforming data** using SQL.
    - **Storing results** in a database.

Resources:

- Airflow Quickstart Guide
- Airflow DAGs Tutorial

📌 **Practice Steps:**

✅ Install **Airflow** and configure it.

✅ Write a **DAG to automate data ingestion & transformation**.

✅ Schedule the DAG to run **every fixed interval e.g.: 5 minute** or every hour:

python ▶ Run Copy

```python
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from datetime import datetime
import pandas as pd
from sqlalchemy import create_engine

def ingest_data():
    df = pd.read_parquet("yellow_tripdata_2024-01.parquet")
    engine = create_engine("postgresql://user:password@localhost:5432/nyc_taxi")
    df.to_sql('taxi_data', engine, if_exists='replace', index=False)

default_args = {
    'owner': 'airflow',
    'start_date': datetime(2025, 2, 12),
    'schedule_interval': '@hourly'
}

dag = DAG('nyc_taxi_pipeline', default_args=default_args, schedule_interval='@hourly')

task1 = PythonOperator(task_id='ingest_data', python_callable=ingest_data, dag=dag)
task1
```

Additional Resources for Downloading Notebooks & Datasets

Open Datasets

1. **Kaggle** – https://www.kaggle.com/datasets
2. **Google Dataset Search** – https://datasetsearch.research.google.com/
3. **AWS Open Data** – https://registry.opendata.aws/
4. **NYC Taxi Data** – https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Jupyter Notebooks & Tutorials

1. **DataTalksClub Data Engineering Zoomcamp** – https://github.com/DataTalksClub/data-engineering-zoomcamp
2. **Data Engineering Notebooks (GitHub)** – https://github.com/awesomedata/awesome-public-datasets
3. **Pandas & SQL Practice Notebooks** – https://github.com/jakevdp/Pandas-Tutorial
4. **Apache Airflow Examples** – https://github.com/apache/airflow/tree/main/airflow/example_dags

---

🚀 **What You Will Have Built in 3 Labs Above:**

✅ **Ingested a real dataset** into a database (PostgreSQL).
✅ **Transformed & cleaned data** using Pandas & SQL.
✅ **Automated data processing** with Apache Airflow.
✅ **Created a reproducible data pipeline** for ML.

📌 What's Next?

If you have more time, try these:
🔥 **Deploy your pipeline on the cloud** (AWS/GCP/Azure).
🔥 **Use Kafka for real-time data ingestion**.
🔥 **Implement a Feature Store with Feast**.