ETL Basics with Pandas + Power Query

EU-First Framework

What is ETL?

ETL stands for:

- **Extract**: Pull data from various sources (databases, APIs, files)
- **Transform**: Clean, reshape, and enrich the data
- **Load**: Store the processed data in a destination system

Real-World Example:
Collecting customer orders from multiple EU stores → cleaning/standardizing → loading into central EU database

EU-First Considerations

When working with data in the EU:

✓ **Data Residency**: Use EU-based cloud services

- AWS Frankfurt, Azure West Europe, Google Belgium

✓ **GDPR Compliance**: Ensure personal data handling follows GDPR

✓ **Data Sovereignty**: Keep sensitive data within EU borders

✓ **EU Cloud Providers**: OVHcloud, Hetzner, Scaleway

Recommended EU Platforms

Databases:

- PostgreSQL on EU servers
- MySQL on Hetzner

Data Lakes:

- MinIO on EU infrastructure
- AWS S3 EU regions

Processing:

- Python/pandas on EU-hosted VMs or containers

Part 1: ETL with Pandas

Pandas Installation

```
pip install pandas openpyxl sqlalchemy psycopg2-binary
```

Pandas is a powerful Python library for data manipulation, perfect for ETL workflows.

Extract: Reading Data Sources

```
import pandas as pd

# Extract from CSV
df_csv = pd.read_csv('eu_customers.csv')

# Extract from Excel
df_excel = pd.read_excel('orders.xlsx', sheet_name='Q1_2025')

# Extract from PostgreSQL (EU server)
from sqlalchemy import create_engine
engine = create_engine('postgresql://user:pass@eu-server.example.com:5432/dbname')
df_db = pd.read_sql(
    'SELECT * FROM transactions WHERE country IN (\'DE\', \'FR\', \'IT\')',
    engine
)
```

Transform: Data Cleaning

```
# Remove duplicates
df = df.drop_duplicates()

# Handle missing values
df['email'] = df['email'].fillna('no-email@example.com')

# Standardize column names
df.columns = df.columns.str.lower().str.replace(' ', '_')

# Filter EU countries only
eu_countries = ['DE', 'FR', 'IT', 'ES', 'NL', 'BE', 'AT', 'PL', 'SE', 'DK']
df = df[df['country'].isin(eu_countries)]
```

Transform: Calculations & Conversions

```
# Add calculated columns
df['total_with_vat'] = df['amount'] * 1.19  # German VAT example

# Data type conversions
df['order_date'] = pd.to_datetime(df['order_date'])
df['amount'] = pd.to_numeric(df['amount'], errors='coerce')
```

Load: Output Destinations

```
# Load to CSV
df.to_csv('cleaned_eu_data.csv', index=False)

# Load to PostgreSQL (EU server)
df.to_sql('clean_transactions', engine, if_exists='replace', index=False)

# Load to Excel
with pd.ExcelWriter('output.xlsx', engine='openpyxl') as writer:
    df.to_excel(writer, sheet_name='Cleaned_Data', index=False)
```

Complete ETL Script Example

```
import pandas as pd
from sqlalchemy import create_engine

def etl_pipeline():
    # EXTRACT
    print("Extracting data...")
    df = pd.read_csv('raw_data.csv')

    # TRANSFORM
    print("Transforming data...")
    df = df.drop_duplicates()
    df = df.dropna(subset=['customer_id', 'amount'])
    df['country'] = df['country'].str.upper()
    df = df[df['country'].isin(['DE', 'FR', 'IT', 'ES', 'NL'])]
    df['processed_date'] = pd.Timestamp.now()
```

Complete ETL Script (continued)

```
    # LOAD
    print("Loading data...")
    engine = create_engine('postgresql://user:pass@eu-db.example.com/warehouse')
    df.to_sql('processed_orders', engine, if_exists='append', index=False)

    print(f"✅ Loaded {len(df)} records to EU database")

if __name__ == "__main__":
    etl_pipeline()
```

Part 2: Power Query

Power Query Overview

Power Query is a visual ETL tool integrated into Excel and Power BI

Great for business users who prefer UI over code

EU-First Setup:

- Use Power BI Premium in EU regions (West Europe, North Europe)
- Configure data residency in Power BI admin settings
- Use on-premises data gateway for sensitive EU data

Accessing Power Query

Excel:

Data tab → Get Data → From File/From Database

Power BI Desktop:

Home → Get Data

Common Transformations (UI)

Operation	Location
Remove Duplicates	Home → Remove Rows → Remove Duplicates
Filter Rows	Click column dropdown → Filter by values
Change Data Types	Right-click column → Change Type
Replace Values	Right-click column → Replace Values
Add Columns	Add Column → Custom Column
Merge Queries	Home → Merge Queries (SQL JOIN)
Append Queries	Home → Append Queries (UNION)

M Language: Custom Columns

Add VAT calculation:

```
= Table.AddColumn(#"Previous Step", "VAT_Amount",
    each [Amount] * 0.19)
```

Filter EU countries:

```
= Table.SelectRows(#"Changed Type",
    each List.Contains({"DE", "FR", "IT", "ES", "NL"}, [Country]))
```

Power Query ETL Workflow

Scenario: Consolidate sales data from multiple EU Excel files

1. **Extract**
    - Get Data → From Folder
    - Select folder with multiple Excel files
    - Combine & Transform Data
2. **Transform**
    - Remove unnecessary columns
    - Filter: Keep only EU countries
    - Add Custom Column for calculations
    - Change data types, Group By for aggregation

M Language Script Example

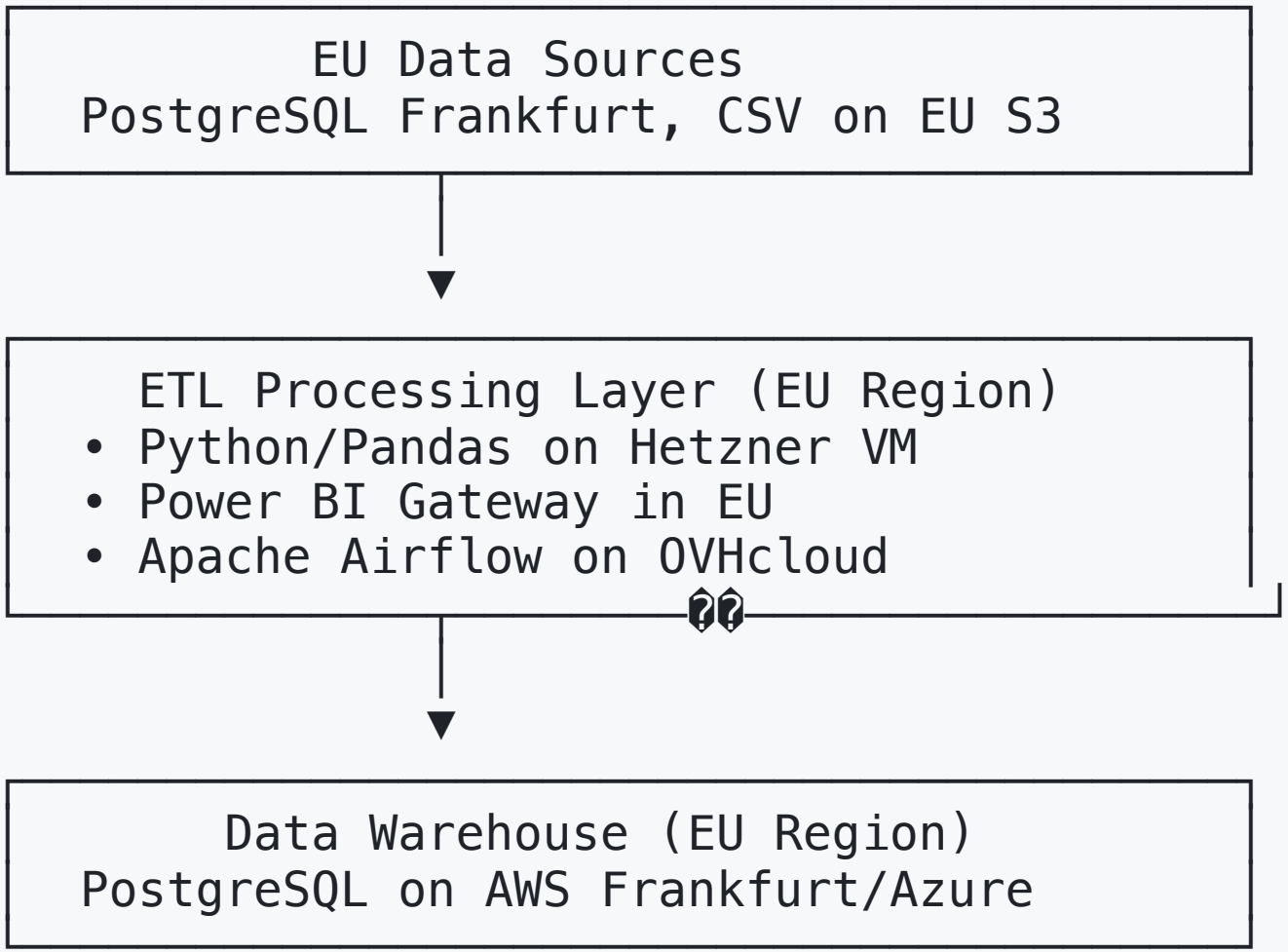
```
let
    // Extract
    Source = Csv.Document(File.Contents("C:\Data\eu_sales.csv")),
    PromotedHeaders = Table.PromoteHeaders(Source),

    // Transform
    ChangedType = Table.TransformColumnTypes(PromotedHeaders, {
        {"OrderDate", type date},
        {"Amount", type number},
        {"Country", type text}
    }),
    FilteredEU = Table.SelectRows(ChangedType,
        each List.Contains({"DE", "FR", "IT", "ES", "NL", "BE"}, [Country])),
    AddedVAT = Table.AddColumn(FilteredEU, "AmountWithVAT",
        each [Amount] * 1.19, type number),
    RemovedDuplicates = Table.Distinct(AddedVAT, {"OrderID"})
in
    RemovedDuplicates
```

Pandas vs Power Query Comparison

Feature	Pandas	Power Query
User Level	Developers/Data Scientists	Business Analysts
Language	Python	M Language (or UI)
Flexibility	Very high	Moderate
Performance	Excellent (large datasets)	Good (medium datasets)
EU Hosting	Run anywhere	Power BI EU/Excel
Version Control	Easy (Git)	Harder (binary files)
Automation	Excellent (cron, Airflow)	Good (scheduled refresh)
Learning Curve	Steeper	Gentle (visual)

EU-Hosted ETL Architecture



Hands-On Exercise

Exercise: EU Sales ETL Pipeline

Sample Data (`eu_orders.csv`):

```
OrderID,CustomerName,Email,Country,Amount,OrderDate
1001,Hans Müller,hans@example.de,DE,150.00,2025-01-15
1002,Marie Dubois,marie@example.fr,FR,200.00,2025-01-16
1003,John Doe,john@example.us,US,300.00,2025-01-17
1004,Luigi Rossi,luigi@example.it,IT,175.50,2025-01-18
1005,Emma Schmidt,,DE,220.00,2025-01-19
```

Pandas Solution

```
import pandas as pd

# Extract
df = pd.read_csv('eu_orders.csv')

# Transform
df = df[df['Country'].isin(['DE', 'FR', 'IT', 'ES', 'NL'])]  # EU only
df['Email'] = df['Email'].fillna('noemail@example.com')
df['AmountWithVAT'] = df['Amount'] * 1.19
df['ProcessedDate'] = pd.Timestamp.now()

# Load
df.to_csv('eu_orders_clean.csv', index=False)
print(f"✅ Processed {len(df)} EU orders")
```

Power Query Solution

Steps:

1. Get Data → From Text/CSV → Select `eu_orders.csv`
2. Transform → Filter `Country` → Keep DE, FR, IT, ES, NL
3. Replace null values in `Email` with "noemail@example.com"
4. Add Column → Custom Column: `[Amount] * 1.19`
    - Name it `AmountWithVAT`
5. Close & Load

Best Practices for EU ETL

1. **Data Minimization**: Only extract what you need (GDPR)
2. **Encryption**: TLS/SSL in transit, encryption at rest
3. **Audit Logging**: Log all ETL operations for compliance
4. **Anonymization**: Pseudonymize personal data when possible
5. **EU Hosting**: Ensure all processing within EU
6. **Documentation**: Maintain data lineage and transformation logic
7. **Testing**: Validate data quality at each stage
8. **Incremental Loads**: Process only new/changed data

Next Steps

Practice:

- Run the hands-on exercise with your own data

Explore:

- Try combining pandas and Power Query in hybrid workflows

Scale:

- Look into Apache Airflow (EU-hosted) for production pipelines

Learn More:

- Pandas: https://pandas.pydata.org
- Power Query M: https://learn.microsoft.com/power-query
- EU cloud providers: OVHcloud, Hetzner, Scaleway

Resources

EU Cloud Providers:

- OVHcloud - France
- Hetzner - Germany
- Scaleway - France

Tools:

- Pandas: `pip install pandas`
- Power BI Desktop: Microsoft Download
- DBeaver: EU database management

GDPR Resources:

- Official GDPR text: https://gdpr-info.eu

Thank You!

🔲 ETL Basics with Pandas + Power Query

EU-First Framework

Questions?