# Data Foundations

## Hoofdstuk 4

## Lab: metadata management, data catalogus en data lineage visualisatie

Hassan Haddouchi

## Uitkomsten van dit lab

Na dit lab kan je:

- Metadata toewijzen en een data catalogus opbouwen in Python.
- Een data lineage volgen door het toepassen van transformaties en het bijhouden van wijzigingen.
- Data lineage visualiseren om inzicht te krijgen in de veranderingen en het traject van de data.

## Opdracht

Werk met de **Online Retail Dataset** om een data catalogus te creëren, metadata toe te wijzen aan variabelen, en de data lineage te visualiseren door transformaties en wijzigingen te volgen. De dataset bevat verkooptransacties van een Britse online winkel.

De dataset wordt in `.xlsx`-formaat geleverd.

## Dataset Inladen in Python

1. Download de **Online Retail Dataset**.
2. Laad de dataset in Python met `pandas`:

```
import pandas as pd
df = pd.read_excel("Online Retail.xlsx")
```

3. Controleer of de dataset correct is geladen door de eerste paar rijen weer te geven met df.head().

## Stap 1: Metadata verzamelen en toevoegen

Voeg voor elke kolom in de dataset de volgende metadata toe:

- Beschrijving: korte uitleg van wat elke kolom voorstelt.
- Datatype: noteer het datatype, bijvoorbeeld int, float, string.
- Belangrijkheidsniveau: ken een niveau toe (laag, medium, hoog) afhankelijk van de relevantie voor analyses.

Sla de metadata op in een JSON- of CSV-bestand.

# Voorbeeld van metadata

```
{
  "InvoiceNo": {"description": "Factuurnummer", "datatype": "string", "importance": "hoog"},
  "StockCode": {"description": "Productcode", "datatype": "string", "importance": "hoog"},
  "Quantity": {"description": "Aantal verkochte eenheden", "datatype": "int", "importance": "medium"}
}
```

Gebruik pandas of json om deze structuur aan te maken en op te slaan.

# Stap 2: bouw een data catalogus

1. Data catalogus maken

- Maak een overzicht in Python waarin de metadata van elke kolom wordt weergegeven.
- Gebruik pandas om een tabel te genereren met kolomnamen en hun respectieve metadata (beschrijving, datatype, belangrijkheidsniveau).

```
# Voorbeeld voor het tonen van de catalogus
metadata = pd.read_json("metadata.json")
print(metadata)
```

2. Interactie met de data catalogus

- Implementeer een zoekfunctie waarmee gebruikers kunnen zoeken op beschrijving of belangrijkheidsniveau.
- Voorbeeld: toon alle kolommen met het belangrijkheidsniveau hoog of die sales bevatten in hun beschrijving.

# Stap 3: data lineage logboek

1. Transformaties:
    - Verwijder duplicaten
    - Voeg een kolom *total_sales* toe door *UnitPrice * Quantity* te berekenen.
    - Filter records waar *Quantity > 10*
2. Logboek voor data transformaties: voor elke transformatie houd je bij:
    - Welke kolommen zijn gewijzigd, toegevoegd of gefilterd.
    - Hoeveel rijen zijn aangepast.

## Stap 4: data lineage visualiseren

Grafische weergave van data lineage:

- Maak een grafische weergave van de data lineage waarin elke stap (transformatie) als een knooppunt in het proces wordt weergegeven.
- Gebruik *matplotlib* of *networkx* om een grafiek te genereren die het traject van de data visualiseert.

## Eindrapportage en reflecite (in je portfolio)

1. Samenvatting van de metadata en data catalogus:

Toon de gegenereerde data catalogus en geef een overzicht van de metadata die bij elke kolom hoort.

2. Data lineage grafiek:

Bespreek welke transformaties de data heeft ondergaan.
Analyseer de impact van de transformaties op de dataset en leg uit waarom data lineage belangrijk is in data management.

## Tools en libraries

- Programmeertaal: Python
- Dataopslag: JSON of CSV voor metadata
- Data visualisatie: matplotlib of networkx voor data lineage
- Dataset import: pandas voor het laden en manipuleren van de dataset

## Vereisten voor inlevering opdracht

1. Code: werkend Python-script dat de metadata, data catalogus en data lineage bevat.
2. Metadata bestand: JSON- of CSV-bestand met de metadata van elke kolom.
3. Data lineage grafiek: een grafische weergave van de data lineage, bij voorkeur als afbeelding (screenshot).

Vergeet je portfolio niet voor een korte reflectie van dit labo.