Modern Data Engineering in the Cloud

Dr. Christian Dollfus

Dr. Pavlin Mavrodiev

Installation of ETL Software and Exam 1

INSTALLATION

1) Install PDI v 9.0 or newest 9.1 one from
   https://sourceforge.net/projects/pentaho/files/Data%20Integration/
   There is also the official Homepage: https://www.hitachivantara.com/en-us/products/data-management-analytics/pentaho/download-pentaho.html
   Keep in Mind that we only use one component of that Suite with the Name PDI (Pentaho Data Integration) and not the whole Lumada/Pentaho Suite

2) Installation JAVA: download jre-8XXX for all Op-Systems
   on http://java.com/en/download/manual.jsp
   (if its not starting: follow the instructions in this video:
   https://www.youtube.com/watch?v=PH6mWe3YVCQ)

3) Install Anaconda

4) Set PYTHONPATH to the root directory of Anaconda:

In order to use DB connection with PDI you need install the drivers:
PDI drivers : add them to the directory
D:\Programme\pdi-ce-9.0.0.0.0-4323\data-integration\lib

mysql-connector-java-5.1.25-bin.jar → for MySQL
jtds-1.2.5.jar → for the Azure DB
mssql-jdbc-7.4.1.jre8.jar → MSSQL-Server

Exam 1

Context: You are a data engineer in a startup that has to deliver actual jobdata daily into a **graphical Dashbaord**. The data is delivered by the company x28 in the xml-Format.
There is a special interest in

- **the number of weeks the jobs are open**
- **the distribution of open jobs in the different sectors**
- **show it graphically (use excel or PowerBI)**

The startup uses own identifiers (Kienbaum_ID) and x28 delivers another id. Mapping tables are present and should be linked

Sources: XML-Files von x28 im Verzeichnis «».
Method: Engineer a PDI Data Pipeline
Target System of the results : **MySQL Database or other DB**

Info:

- x28 delivers XML-Files with 200 jobs each. The startup is using own Functions and own Company Names → have to be mapped.

- You can use Metatables:

- tm_tp_companies_sectors → mapping from company to sector
- tm_tp_sectors_functions → mapping from company to function
- tm_X28_Companies → mapping comp_id to Kienbaum_ID and companyname
- tm_X28_Functions → mapping of the 19 MCG functions to 1200 job_name_x28

Exam 1

Procedure:

Part 1: Data Analysis:

Analyze the data you got:

- What is operational data
- What is Medtadata
- Which fields do i see there ?
- Do i need all the Metadata given ?
- What fields do i need in the solution ? Where are they ?
- Which kind of identifier do i have in which file in order to match them together ?

Part 2: Architecture:

What do i want engineer in one transformation ?

Tip:

- do small clear transformations that can be glued togeter in a Job
- Do not mix Metadata with operational data → they have another usage pattern in time
- Do distinguish the levels in the DB : stage, store
- Define where to match the data and where to calculate measures (i.e. weeks)

Exam 1

Part 3: Implementation:

Steps:

- Preparation: load all metatables into the DB with PDI-Pentaho
- Read all XML files with ist needed fields
- (Denormalize the first 4 metadatafields)
- Field «conpany_id» cleansing as it can be used for DB-Lookup
- DB-Lookup for the new fields «Kienbaum_ID, company_mcg_id
- Filter the fields
- Lookup with Kienbaum_ID and MCG companyname (Field Firma from tm_X28_Companies)
- Lookup with new function of MCG (Field «function» from tm_X28_Functions)
- Sort and clean fields
- Put everything into the DB with «truncate table» option with the tablename «t_x28_jobs_store»

- Calculate how long the job was open (mögl. mit SQL: ceil(datediff(date(curdate()), date(firstseen))/7) als SQL statement)