Documentation Exam 01- ETL with Apache Hop

Carlo Scherrer, Fabian Del Conte & Shallau Muhamad

Modern Data Engineering - 03.06.2025

Table of content

Introduction 3

Data Analysis & Architecture 3

ETL-Process with Apache HOP 4

Data extraction & Transformation - Stage 1+2 4

Loading - Stage 3 7

Results & Visualization 8

Weekly number of open jobs 8

Distribution across sectors 9

Conclusion 10

Challenges 10

Appendix 11

Dashboard 11

SQL statements stage 1 - initial table creation 12

SQL statements stage 2 – cleansed data 13

SQL statements stage 3 – final table 13

Database Connection in PowerBi 13

Pipeline stage 1&2 - confirmation successful run 14

Pipeline stage 3 - confirmation successful run 14

Table of figures

Figure 1 - Database schema 3
Figure 2 - metadata stage 1 pipeline 4
Figure 3 - XML stage 1 pipeline 4
Figure 4 - X28_functions stage 2 pipeline 5
Figure 5 - tp_companies_sector stage 2 pipeline 5
Figure 6 - X28_companies stage 2 pipeline 5
Figure 7 - Operational XML data stage 2 pipeline 5
Figure 8 - ETL process Apache Hop stage 1-2 for Metadata & XML 6
Figure 9 - Stage 3 process Apache Hop 7
Figure 10 - Final table x_28_jobs_store 7
Figure 11 - Tracking the number of weeks jobs remain open 8
Figure 12 - Distribution of vacancies across different sectors 9

Introduction

In this case study, an ETL solution for processing XML-based job data was developed. The aim was to extract structured and cleansed data from partially unstructured XML files and metadata tables, transform it into a standardized form and store it in an Azure SQL database. The solution was implemented with Apache HOP. The processed data serves as the basis for two central analyses: determining how long job vacancies have been online and the distribution of open positions across different industries. These results are visualized in Power BI.

For the implementation, the project group used a clean staging concept that manages raw data (stage 1), cleansed data (stage 2) and final enriched data (stage 3) in separate tables. The three-stage staging concept was selected to ensure a clear separation between the individual processing phases, to increase traceability and error tolerance in ETL processing. In our view, this is an important aspect for data-driven projects with diverse/different sources.

Data Analysis & Architecture

The XML data of the startup that publishes the job offers uses its own function and organization names. Furthermore, a large amount of data is provided in these datasets that is not relevant to answer our research questions. There are also some fields in the metadata files that are not necessarily relevant for us (could be used as further metadata if required). Therefore, to approach the project efficiently, the XML structure and the metadata were analyzed at the beginning and a schematic structure was created using a sample data set, which shows the relationships and fields. After verification using another example, the following schema was the result and the basis for the first tables in the database (stage 1).

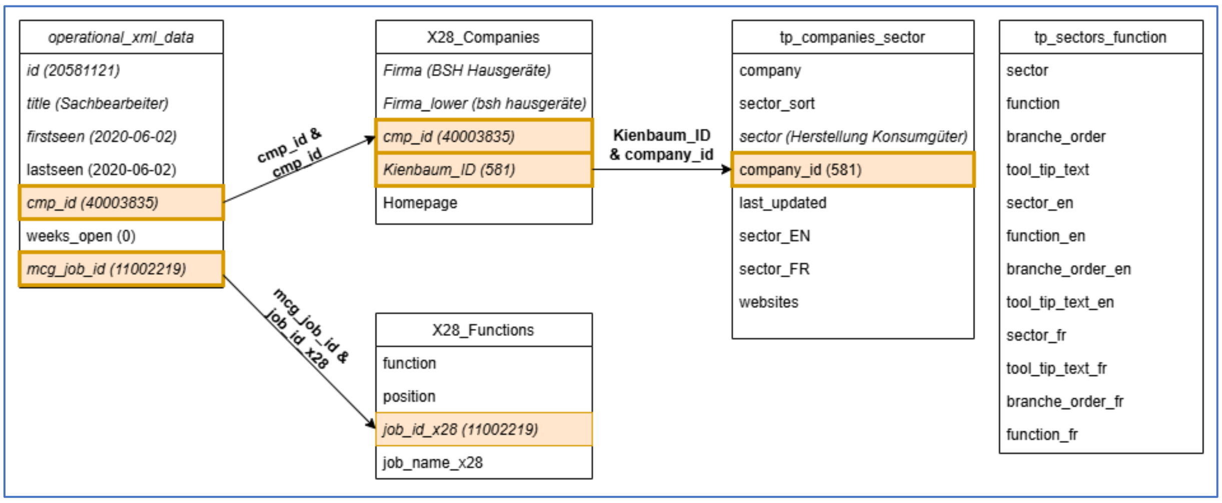


Figure 1 - Database schema

The corresponding SQL statements can be found in the appendix in chapter SQL statements stage 1 - initial table creation. The following chapter explains which steps were necessary in the process to create the schema initial for the stage 1 tables up to the final stage 3 table.

ETL-Process with Apache HOP

Data extraction & Transformation - Stage 1+2

In the 1st phase, a different approach was chosen between the metadata & operational XML data:

- **The metadata** files were loaded completely and without pre-processing into dedicated stage 1 tables in order to fully preserve their original structure and enable flexible further processing at a later stage.

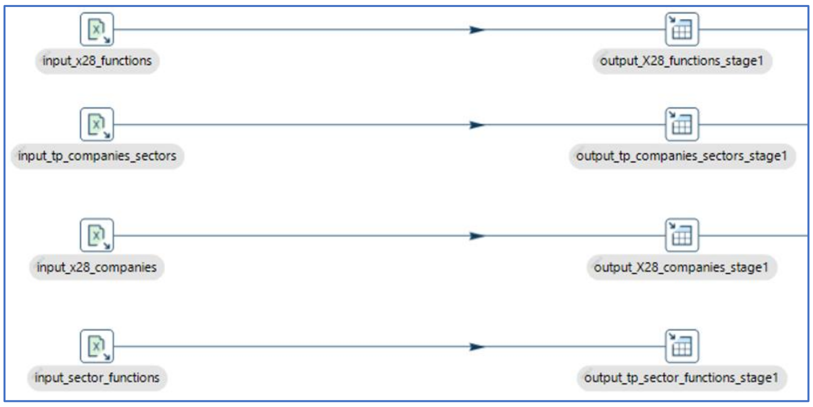

input_x28_functions
output_X28_functions_stage1
input_tp_companies_sectors
output_tp_companies_sectors_stage1
input_x28_companies
output_X28_companies_stage1
input_sector_functions
output_tp_sector_functions_stage1

Figure 2 - metadata stage 1 pipeline

- **The XML data**, on the other hand, was filtered down to the fields that were actually relevant before loading. The idea was to avoid unnecessary strain on database resources and only store the information required for analysis and transformation (reducing storage space and processing costs). This contains the following fields and a stream lookup was necessary due to the nested structure for the *mcg_job_id*:
  - *job_id*
  - *title*
  - *cmp_id*
  - *mcg_job_id*
  - *first_clean*
  - *last_clean*

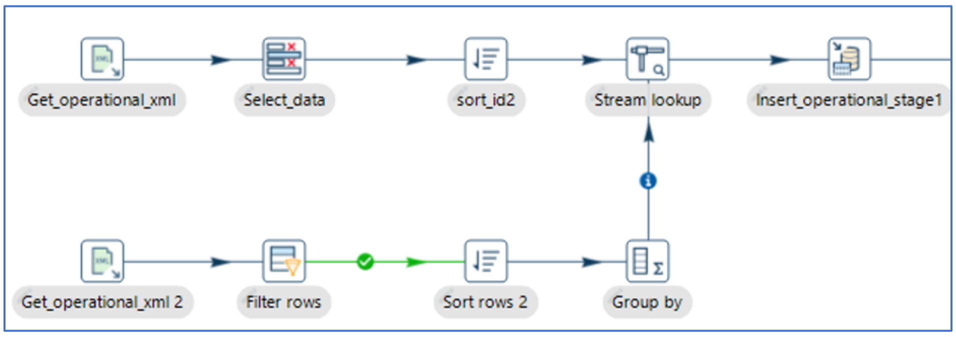


Figure 3 - XML stage 1 pipeline

In the 2nd phase, the process for the cleaned & transformed data for the stage 2 tables corresponds to the following procedure and the result is visualized in Figure 8, which includes all steps for stage 1 and 2:

- **X28_functions:**
  - Only the fields *function, job_id_x28, job_name_x28* were required for the further procedure; these were selected with a Select step at the beginning.
  - Sorting & filtering according to the unique *X28_job_id* to avoid duplicates.

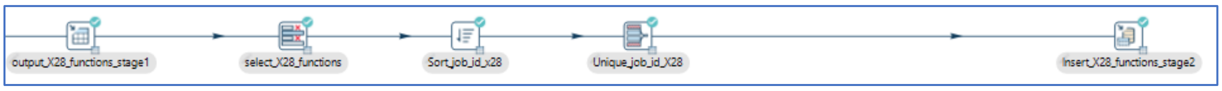


Figure 4 - X28_functions stage 2 pipeline

- **tp_companies_sectors:**
  - Only the fields *company, sector, company_id* were required for the further procedure; these were selected with a Select step at the beginning.
  - Removing entries with invalid *company_id*
  - Database lookup to import company names (field *Firma*) according to matching *company_id* and *Kienbaum_ID*.

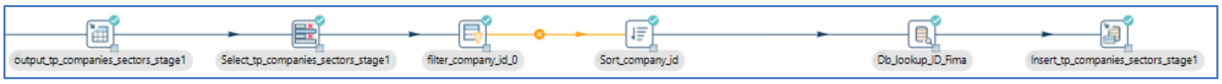

output_tp_companies_sectors_stage1
Select_tp_companies_sectors_stage1
filter_company_id_0
Sort_company_id
Db_lookup_ID_Firma
Insert_tp_companies_sectors_stage1

Figure 5 - tp_companies_sector stage 2 pipeline

- **X28_companies:**
  - Only the fields *Firma, cmp_id, Kienbaum_ID* were required for the further procedure; these were selected with a Select step at the beginning.
  - Removing entries with invalid *company_id*

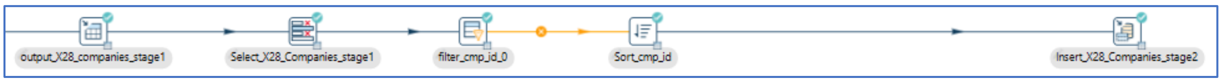

output_X28_companies_stage1
Select_X28_Companies_stage1
filter_cmp_id_0
Sort_cmp_id
Insert_X28_Companies_stage2

Figure 6 - X28_companies stage 2 pipeline

- **tp_sector_functions:** Not processed further as not relevant to the research questions.
- **Operational XML data:**
  - Clean up the date format (remove time zone & additional character "*T*")
  - Calculate the duration of how long an offer was public:
    - by calculating the number of days between "*Lastseen*" and "*Created*"
    - a constant of 7 for the calculation of the duration in weeks.

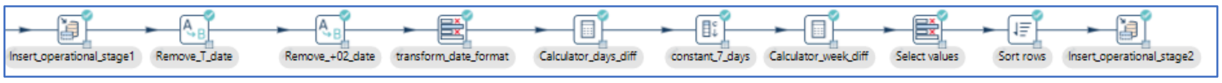


Figure 7 - Operational XML data stage 2 pipeline

Initially, separate pipelines were created for each stage, but after some adjustments we decided to merge stage 1 and stage 2 into one pipeline. The result is the Figure 8 below. In our opinion, the reasons for using only one pipeline are:

- **Avoidance of redundant database operations:**
  By processing the raw data (stage 1) directly in the same pipeline, there was no need for additional saving and subsequent re-reading (e.g. via table input). This reduces read/write operations in the database and saves time.

- **More efficient data flow - better traceability and maintainability:**
  The pipeline is processed as a continuous data stream without any intermediate stops. This makes the ETL process more performant and logical in structure - especially if the steps are clearly sequenced. Furthermore, a combined pipeline with clearly named and structured steps enables a more direct understanding of data processing, from raw data to cleansed data, without having to constantly switch between multiple files and flows.

- **Faster debugging:**
  Previous errors in the data flow are easier to detect and correct as all transformation steps are centrally visible and the option to include previews in several places.

- **Project context & scope:**
  An integrated pipeline was sufficient and pragmatic for the requirements of this project (manageable data volumes, clearly defined structure). Too much modularization would have created unnecessary complexity.

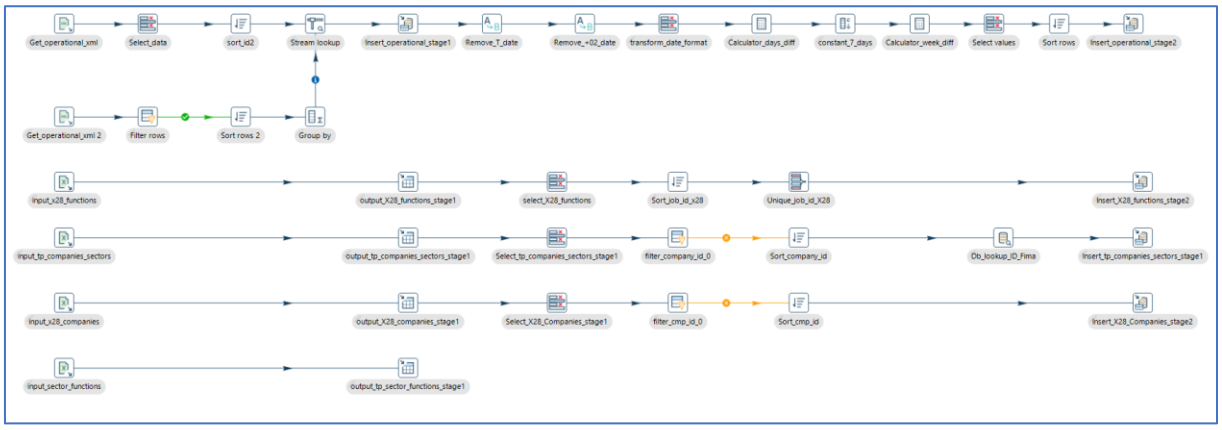


Figure 8 - ETL process Apache Hop stage 1-2 for Metadata & XML

Loading- Stage 3

For the final x28_jobs_store table, several database lookups are used which use the relevant fields from the stage 2 tables. This process includes:

- In a first Select-step the relevant fields from the operational XML job data are selected, including:
  - *mcg_job_id* (Primary Key in stage 3)
  - *weeks_open* (calculated in stage 2)
  - *job_id* (renamed to job_id_X28, *later dropped)
  - *cmp_id* (renamed to cmp_id_X28, *later dropped)
- In a first database lookup with the corresponding *cmp_id* & *cmp_id_X28* several fields from the *X28_companies* table are selected:
  - *Firma* (standard name for the company in stage 3)
  - *Cmp_id* (*later dropped)
  - *Kienbaum_ID* (*later dropped)
- The second database lookup is done with the *Kienbaum_ID* and the *company_id* in the table *tp_companies_sectors_stage2*, to get the relevant sectors for each job:
  - *Company* (*later dropped)
  - *Sector* (sector for the jobs in stage 3 table)
  - *Company_id* (*later dropped)
- In a third database lookup the field function for each *mcg_job_id* are selected due to the lookup with the field *mcg_job_id* and the field *job_id_X28* from the table *X28_Functions_stage2*.
- In the last step of the pipeline, the fields not required for the stage 3 table are removed (indicated above with *later dropped), resulting in the table in Figure 10.

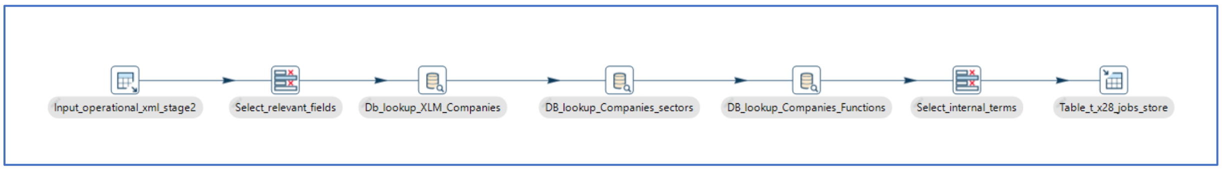
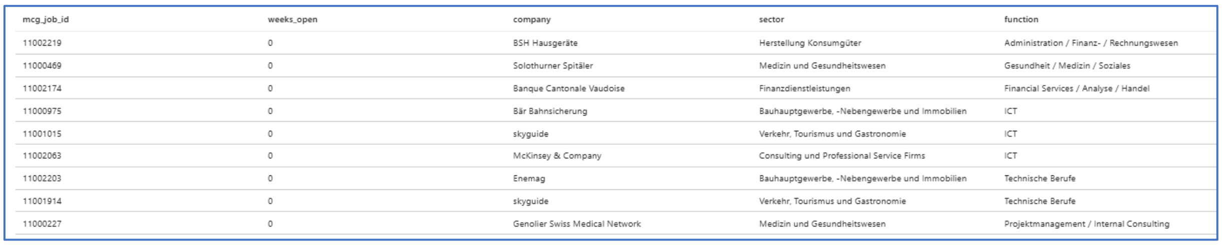


Figure 9 - Stage 3 process Apache Hop

The result of the process above is the following table x_28_jobs_store in the database:



mcg_job_id	weeks_open	company	sector	function
11002219	0	BSH Hausgeräte	Herstellung Konsumgüter	Administration / Finanz- / Rechnungswesen
11000469	0	Solothurner Spitäler	Medizin und Gesundheitswesen	Gesundheit / Medizin / Soziales
11002174	0	Banque Cantonale Vaudoise	Finanzdienstleistungen	Financial Services / Analyse / Handel
11000975	0	Bär Bahnsicherung	Bauhauptgewerbe, -Nebengewerbe und Immobilien	ICT
11001015	0	skyguide	Verkehr, Tourismus und Gastronomie	ICT
11002063	0	McKinsey & Company	Consulting und Professional Service Firms	ICT
11002203	0	Enemag	Bauhauptgewerbe, -Nebengewerbe und Immobilien	Technische Berufe
11001914	0	skyguide	Verkehr, Tourismus und Gastronomie	Technische Berufe
11000227	0	Genolier Swiss Medical Network	Medizin und Gesundheitswesen	Projektmanagement / Internal Consulting

Figure 10 - Final table x_28_jobs_store

Results & Visualization

The final *table x_28_jobs_store* was then used to answer the two questions “the number of weeks in which jobs remain open” and “the distribution of vacancies across the various sectors”. The business intelligence tool Power BI was used for this purpose. The connection to the database is created via the “Connection to SQL Server database” function, as shown in in the appendix in chapter Database Connection in PowerBi, using the credentials stored in the Azure Key vault.

Weekly number of open jobs

Regarding the number of weeks in which jobs remain open, it became clear that most jobs are only published for a few days and then disappear again. These are therefore displayed as 0 weeks. The distribution of the number of weeks ranges from 0 to 292 (mcg_job_id 11000066 – in sector consulting) weeks.

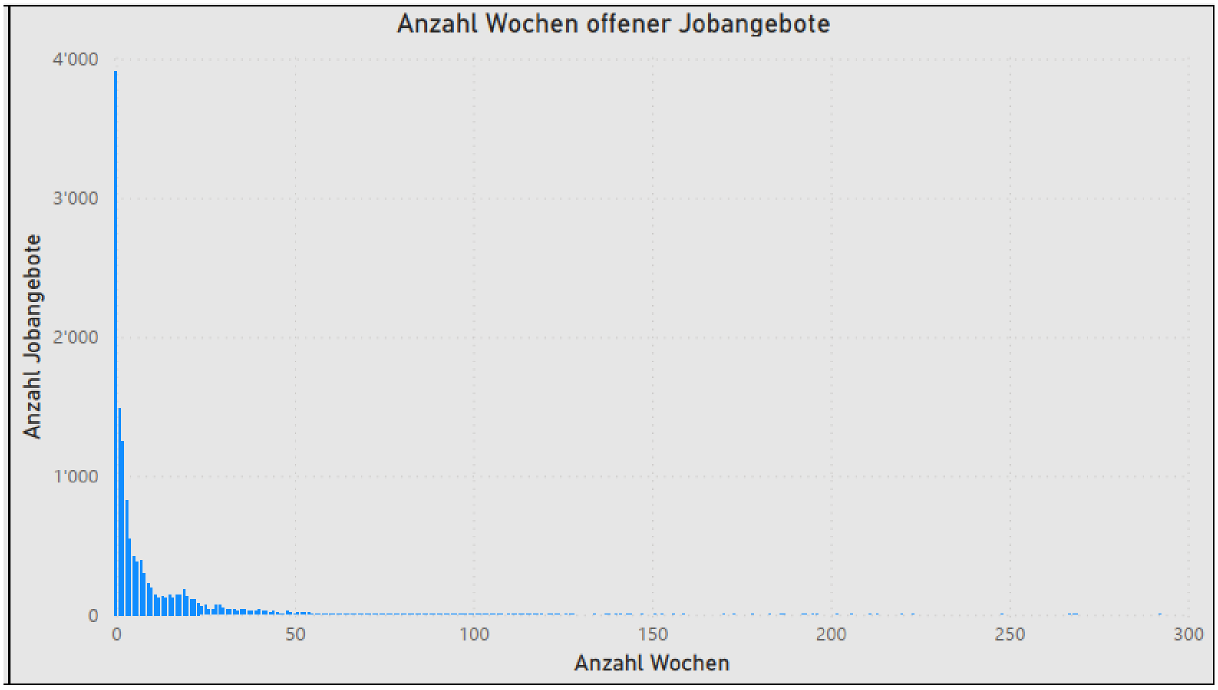


Figure 11 - Tracking the number of weeks jobs remain open

Distribution across sectors

In terms of the distribution of vacancies across the various sectors, most vacancies are in the medical and healthcare, financial services and retail sectors.

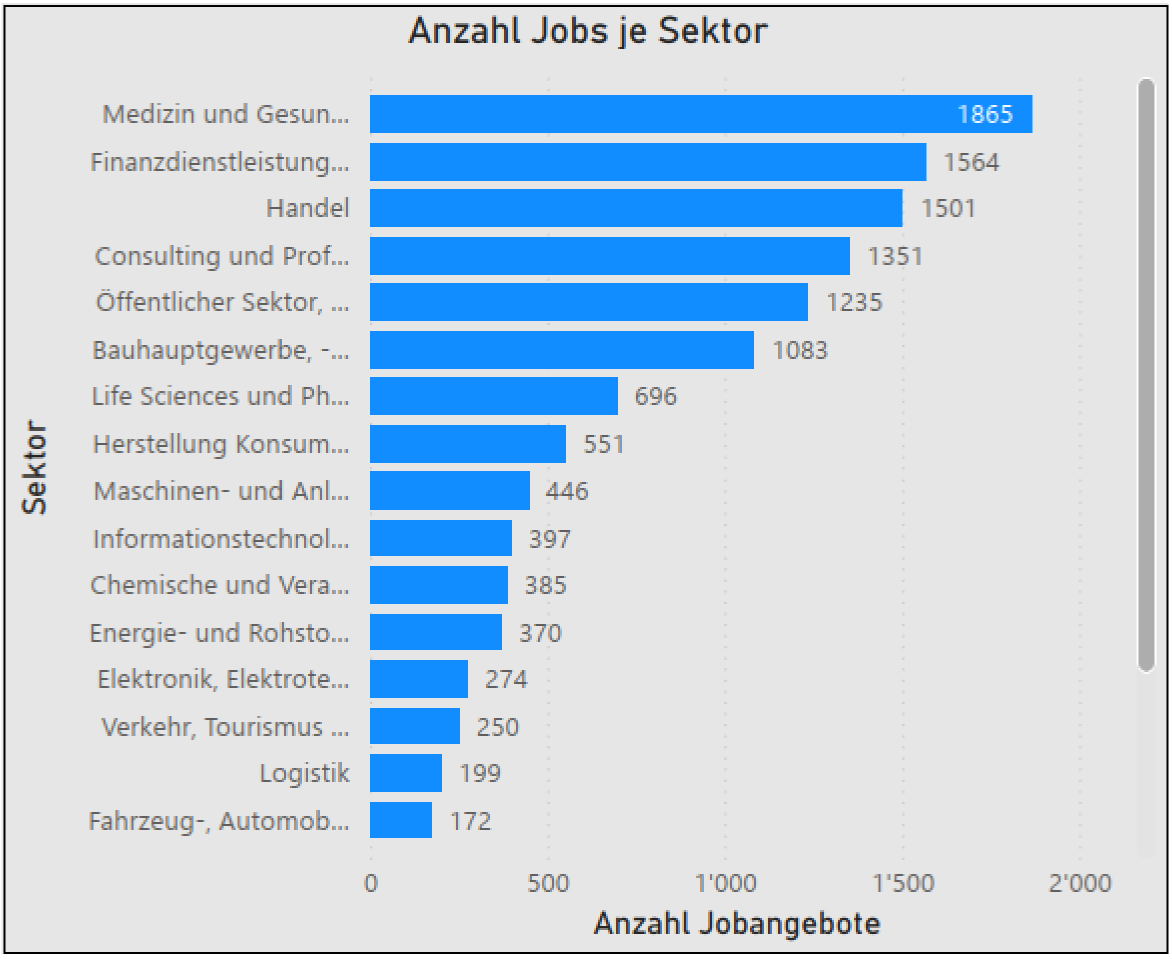


Figure 12 - Distribution of vacancies across different sectors

Conclusion

In summary, it can be said that we have learned the basics of Apache Hop through this task with a practical example and can now deepen our knowledge independently. Looking back, we would like to take a look at the biggest challenges in this task.

Challenges

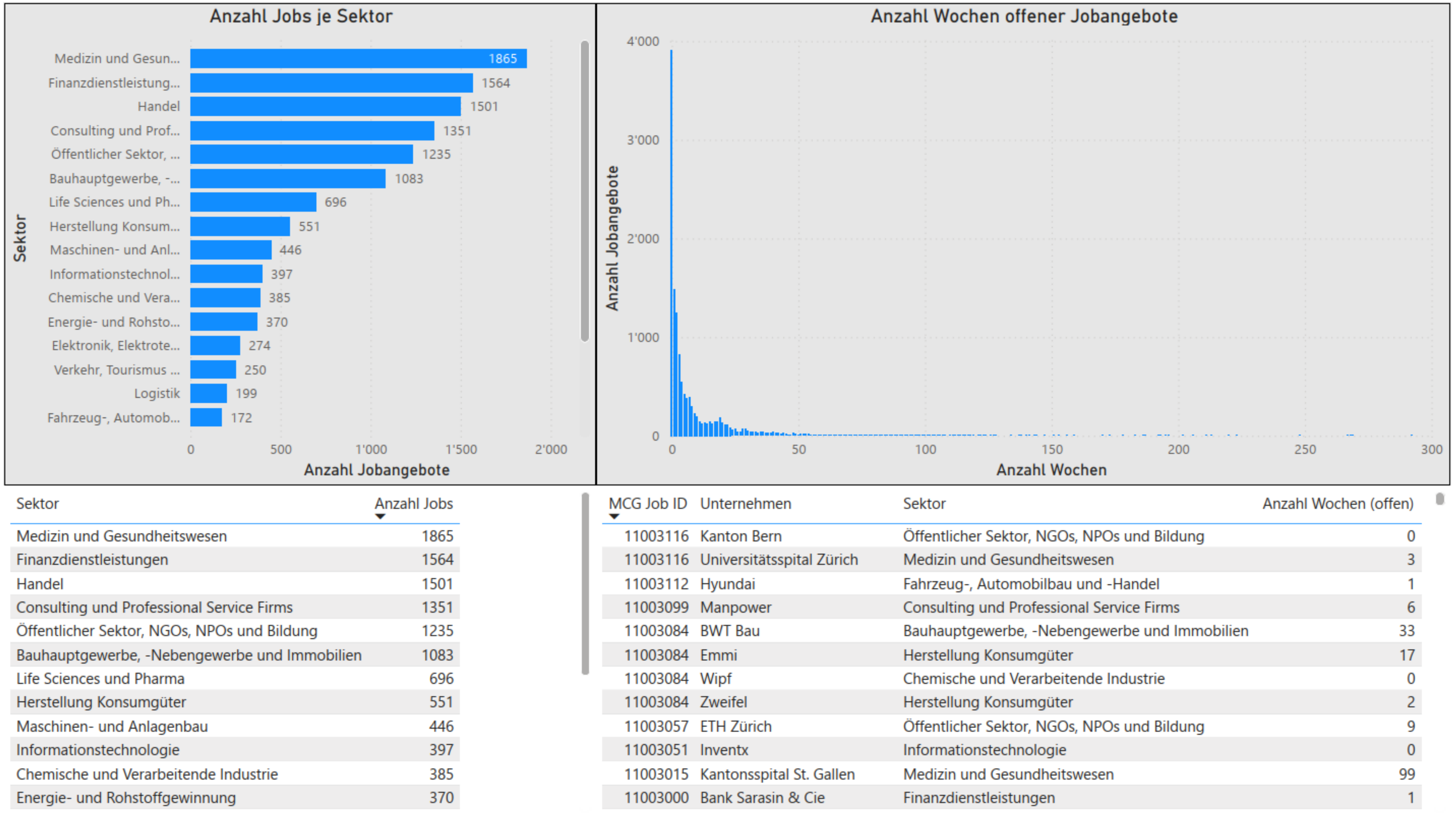
The biggest challenge in this project was understanding the relevant data fields in order to be able to answer the analysis questions in a well-founded manner and build the corresponding pipelines. As a large proportion of the metadata and fields provided in the XML files were not necessarily relevant, the relevant values had to be precisely identified in advance. A detailed initial analysis and the creation of a logical schema based on examples proved to be particularly helpful.

During the implementation with Apache Hop, an attempt was made to build precise and robust pipelines in order to minimize sources of error and increase maintainability. We tried to follow the principle of “as simple as possible, as stable as necessary”.

Clear naming conventions in the transformation steps and a structured separation of the phases in stage tables made it possible to realize a transparent and comprehensible ETL architecture that is also easy for third parties to understand and extend.

Appendix

Dashboard


Anzahl Jobs je Sektor
Sektor
Medizin und Gesun... 1865
Finanzdienstleistung... 1564
Handel 1501
Consulting und Prof... 1351
Öffentlicher Sektor, ... 1235
Bauhauptgewerbe, -... 1083
Life Sciences und Ph... 696
Herstellung Konsum... 551
Maschinen- und Anl... 446
Informationstechnol... 397
Chemische und Vera... 385
Energie- und Rohsto... 370
Elektronik, Elektrote... 274
Verkehr, Tourismus ... 250
Logistik 199
Fahrzeug-, Automob... 172
0 500 1'000 1'500 2'000
Anzahl Jobangebote
Anzahl Wochen offener Jobangebote
Anzahl Jobangebote
4'000 3'000 2'000 1'000 0
0 50 100 150 200 250 300
Anzahl Wochen

Sektor	Anzahl Jobs
Medizin und Gesundheitswesen	1865
Finanzdienstleistungen	1564
Handel	1501
Consulting und Professional Service Firms	1351
Öffentlicher Sektor, NGOs, NPOs und Bildung	1235
Bauhauptgewerbe, -Nebengewerbe und Immobilien	1083
Life Sciences und Pharma	696
Herstellung Konsumgüter	551
Maschinen- und Anlagenbau	446
Informationstechnologie	397
Chemische und Verarbeitende Industrie	385
Energie- und Rohstoffgewinnung	370

MCG Job ID	Unternehmen	Sektor	Anzahl Wochen (offen)
11003116	Kanton Bern	Öffentlicher Sektor, NGOs, NPOs und Bildung	0
11003116	Universitätsspital Zürich	Medizin und Gesundheitswesen	3
11003112	Hyundai	Fahrzeug-, Automobilbau und -Handel	1
11003099	Manpower	Consulting und Professional Service Firms	6
11003084	BWT Bau	Bauhauptgewerbe, -Nebengewerbe und Immobilien	33
11003084	Emmi	Herstellung Konsumgüter	17
11003084	Wipf	Chemische und Verarbeitende Industrie	0
11003084	Zweifel	Herstellung Konsumgüter	2
11003057	ETH Zürich	Öffentlicher Sektor, NGOs, NPOs und Bildung	9
11003051	Inventx	Informationstechnologie	0
11003015	Kantonsspital St. Gallen	Medizin und Gesundheitswesen	99
11003000	Bank Sarasin & Cie	Finanzdienstleistungen	1

SQL statements stage 1- initial table creation

```
CREATE TABLE operational_xml_stage1 (
    job_id INT,
    title NVARCHAR(255),
    firstseen NVARCHAR(50),
    lastseen NVARCHAR(50),
    cmp_id INT,
    mcg_job_id INT
);


CREATE TABLE X28_Companies_stage1 (
    Firma NVARCHAR(255),
    Firma_lower NVARCHAR(255),
    cmp_id BIGINT,
    Kienbaum_ID INT,
    Homepage NVARCHAR(255)
);

CREATE TABLE X28_Functions_stage1 (
    [function] NVARCHAR(255),
    position NVARCHAR(255),
    job_id_x28 BIGINT,
    job_name_x28 NVARCHAR(255)
);

CREATE TABLE tp_companies_sectors_stage1 (
    company NVARCHAR(255),
    sector_sort INT,
    sector NVARCHAR(255),
    company_id INT,
    last_updated DATETIME,
    sector_EN NVARCHAR(255),
    sector_FR NVARCHAR(255),
    websites NVARCHAR(255)
);

CREATE TABLE tp_sectors_functions_stage1 (
    sector NVARCHAR(255),
    [function] NVARCHAR(255),
    branche_order NVARCHAR(255),
    tool_tip_text NVARCHAR(MAX),
    sector_en NVARCHAR(255),
    function_en NVARCHAR(255),
    branche_order_en NVARCHAR(255),
    tool_tip_text_en NVARCHAR(MAX),
    sector_fr NVARCHAR(255),
    function_fr NVARCHAR(255),
    branche_order_fr NVARCHAR(255),
    tool_tip_text_fr NVARCHAR(MAX)
);
```

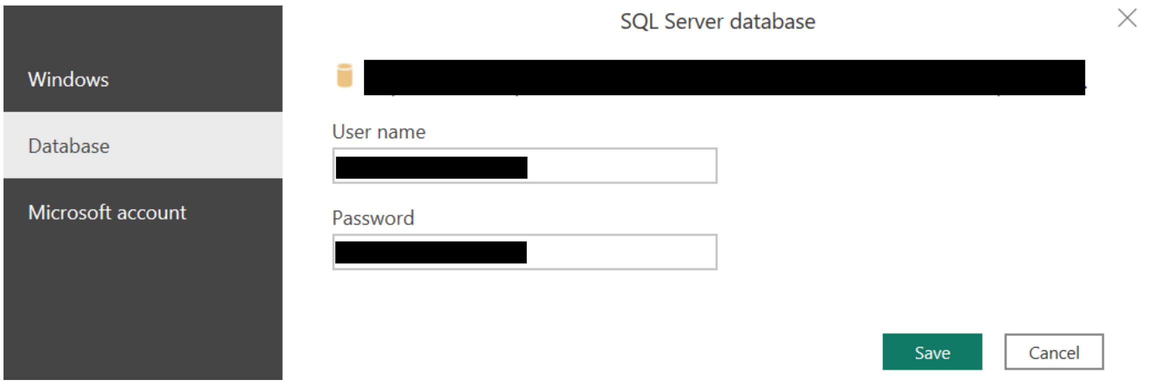
SQL statements stage 2 – cleansed data

```
CREATE TABLE dbo.operational_xml_stage2 (
    job_id INT,
    title NVARCHAR(255),
    cmp_id INT,
    mcg_job_id INT,
    first_clean DATE,
    last_clean DATE,
    days_open INT,
    weeks_open INT
);


CREATE TABLE [dbo].[X28_Companies_stage2] (
    Firma NVARCHAR(255),
    cmp_id BIGINT,
    Kienbaum_ID INT
);

CREATE TABLE tp_companies_sectors_stage2 (
    company NVARCHAR(255),
    sector NVARCHAR(255),
    company_id INT
);


CREATE TABLE dbo.X28_Functions_stage2 (
    [function] NVARCHAR(255),
    job_id_x28 INT,
    job_name_x28 NVARCHAR(255)
);
```

SQL statements stage 3 – final table

```
CREATE TABLE dbo.t_x28_jobs_store (
    mcg_job_id INT,
    weeks_open INT,
    company NVARCHAR(255),
    sector NVARCHAR(255),
    [function] NVARCHAR(255)
);
```

Database Connection in PowerBi



SQL Server database

Windows

Database

Microsoft account

[redacted]

User name

[redacted]

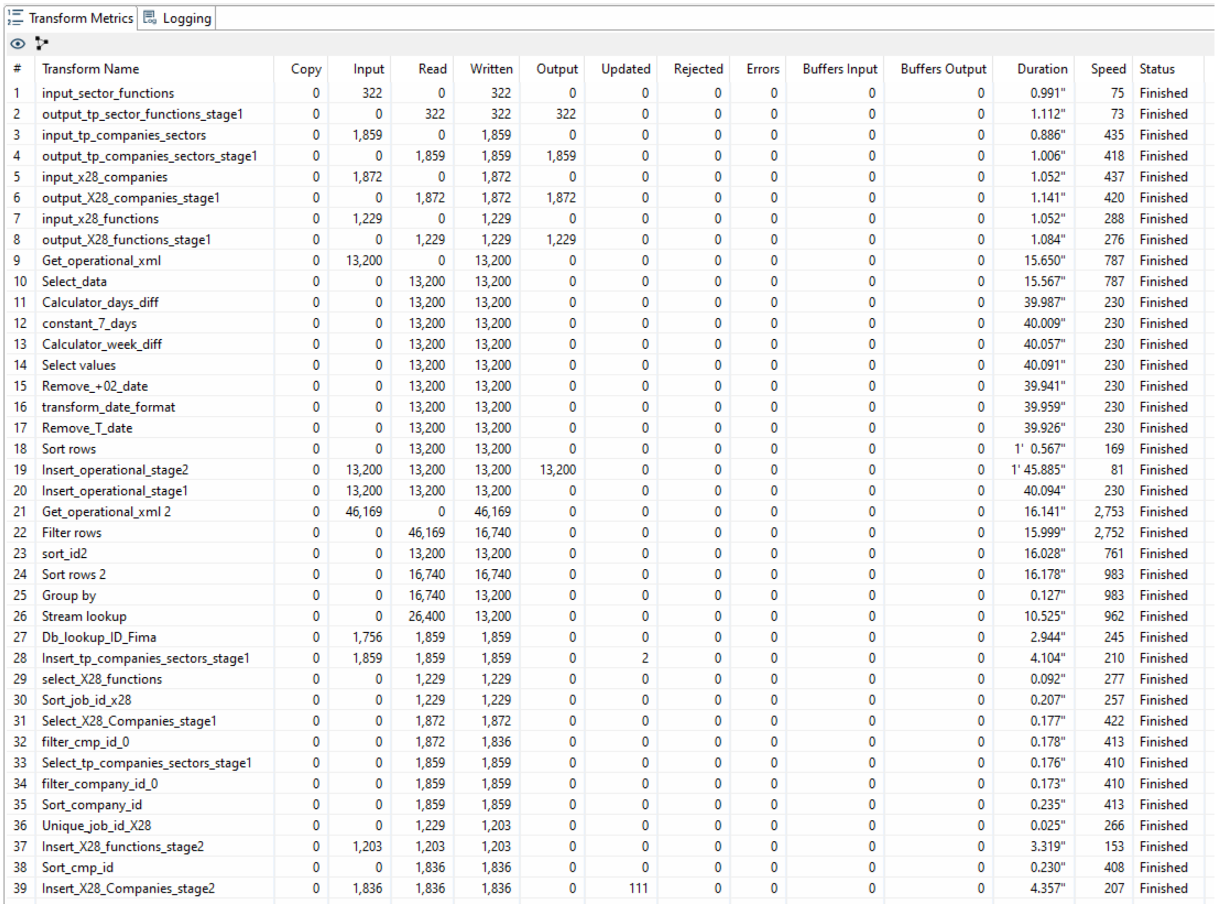
Password

[redacted]

Save

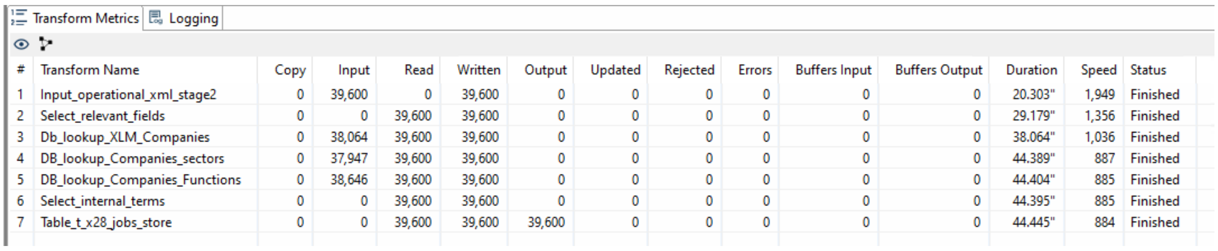
Cancel

Pipeline stage 1&2 - confirmation successful run


Transform Metrics | Logging

#	Transform Name	Copy	Input	Read	Written	Output	Updated	Rejected	Errors	Buffers Input	Buffers Output	Duration	Speed	Status
1	input_sector_functions	0	322	0	322	0	0	0	0	0	0	0.991"	75	Finished
2	output_tp_sector_functions_stage1	0	0	322	322	322	0	0	0	0	0	1.112"	73	Finished
3	input_tp_companies_sectors	0	1,859	0	1,859	0	0	0	0	0	0	0.886"	435	Finished
4	output_tp_companies_sectors_stage1	0	0	1,859	1,859	1,859	0	0	0	0	0	1.006"	418	Finished
5	input_x28_companies	0	1,872	0	1,872	0	0	0	0	0	0	1.052"	437	Finished
6	output_X28_companies_stage1	0	0	1,872	1,872	1,872	0	0	0	0	0	1.141"	420	Finished
7	input_x28_functions	0	1,229	0	1,229	0	0	0	0	0	0	1.052"	288	Finished
8	output_X28_functions_stage1	0	0	1,229	1,229	1,229	0	0	0	0	0	1.084"	276	Finished
9	Get_operational_xml	0	13,200	0	13,200	0	0	0	0	0	0	15.650"	787	Finished
10	Select_data	0	0	13,200	13,200	0	0	0	0	0	0	15.567"	787	Finished
11	Calculator_days_diff	0	0	13,200	13,200	0	0	0	0	0	0	39.987"	230	Finished
12	constant_7_days	0	0	13,200	13,200	0	0	0	0	0	0	40.009"	230	Finished
13	Calculator_week_diff	0	0	13,200	13,200	0	0	0	0	0	0	40.057"	230	Finished
14	Select values	0	0	13,200	13,200	0	0	0	0	0	0	40.091"	230	Finished
15	Remove_+02_date	0	0	13,200	13,200	0	0	0	0	0	0	39.941"	230	Finished
16	transform_date_format	0	0	13,200	13,200	0	0	0	0	0	0	39.959"	230	Finished
17	Remove_T_date	0	0	13,200	13,200	0	0	0	0	0	0	39.926"	230	Finished
18	Sort rows	0	0	13,200	13,200	0	0	0	0	0	0	1' 0.567"	169	Finished
19	Insert_operational_stage2	0	13,200	13,200	13,200	13,200	0	0	0	0	0	1' 45.885"	81	Finished
20	Insert_operational_stage1	0	13,200	13,200	13,200	0	0	0	0	0	0	40.094"	230	Finished
21	Get_operational_xml 2	0	46,169	0	46,169	0	0	0	0	0	0	16.141"	2,753	Finished
22	Filter rows	0	0	46,169	16,740	0	0	0	0	0	0	15.999"	2,752	Finished
23	sort_id2	0	0	13,200	13,200	0	0	0	0	0	0	16.028"	761	Finished
24	Sort rows 2	0	0	16,740	16,740	0	0	0	0	0	0	16.178"	983	Finished
25	Group by	0	0	16,740	13,200	0	0	0	0	0	0	0.127"	983	Finished
26	Stream lookup	0	0	26,400	13,200	0	0	0	0	0	0	10.525"	962	Finished
27	Db_lookup_ID_Fima	0	1,756	1,859	1,859	0	0	0	0	0	0	2.944"	245	Finished
28	Insert_tp_companies_sectors_stage1	0	1,859	1,859	1,859	0	2	0	0	0	0	4.104"	210	Finished
29	select_X28_functions	0	0	1,229	1,229	0	0	0	0	0	0	0.092"	277	Finished
30	Sort_job_id_x28	0	0	1,229	1,229	0	0	0	0	0	0	0.207"	257	Finished
31	Select_X28_Companies_stage1	0	0	1,872	1,872	0	0	0	0	0	0	0.177"	422	Finished
32	filter_cmp_id_0	0	0	1,872	1,836	0	0	0	0	0	0	0.178"	413	Finished
33	Select_tp_companies_sectors_stage1	0	0	1,859	1,859	0	0	0	0	0	0	0.176"	410	Finished
34	filter_company_id_0	0	0	1,859	1,859	0	0	0	0	0	0	0.173"	410	Finished
35	Sort_company_id	0	0	1,859	1,859	0	0	0	0	0	0	0.235"	413	Finished
36	Unique_job_id_X28	0	0	1,229	1,203	0	0	0	0	0	0	0.025"	266	Finished
37	Insert_X28_functions_stage2	0	1,203	1,203	1,203	0	0	0	0	0	0	3.319"	153	Finished
38	Sort_cmp_id	0	0	1,836	1,836	0	0	0	0	0	0	0.230"	408	Finished
39	Insert_X28_Companies_stage2	0	1,836	1,836	1,836	0	111	0	0	0	0	4.357"	207	Finished

Pipeline stage 3 - confirmation successful run


Transform Metrics | Logging

#	Transform Name	Copy	Input	Read	Written	Output	Updated	Rejected	Errors	Buffers Input	Buffers Output	Duration	Speed	Status
1	Input_operational_xml_stage2	0	39,600	0	39,600	0	0	0	0	0	0	20.303"	1,949	Finished
2	Select_relevant_fields	0	0	39,600	39,600	0	0	0	0	0	0	29.179"	1,356	Finished
3	Db_lookup_XLM_Companies	0	38,064	39,600	39,600	0	0	0	0	0	0	38.064"	1,036	Finished
4	DB_lookup_Companies_sectors	0	37,947	39,600	39,600	0	0	0	0	0	0	44.389"	887	Finished
5	DB_lookup_Companies_Functions	0	38,646	39,600	39,600	0	0	0	0	0	0	44.404"	885	Finished
6	Select_internal_terms	0	0	39,600	39,600	0	0	0	0	0	0	44.395"	885	Finished
7	Table_t_x28_jobs_store	0	0	39,600	39,600	39,600	0	0	0	0	0	44.445"	884	Finished