HOCHSCHULE FÜR WIRTSCHAFT UND RECHTS BERLIN

Berlin School of Economics and Law

Department of Business and Economics

Master's Program Business Intelligence and Process Management

Winter Semester 2021/2022

SUPER-X Data Mart Design

Sales Department

Data Warehousing

Group 1

Minh Anh Hoang - 77211887950

Soere Valentin Werner - 77211892217

Roger Pujol Grau - 77211909676

TABLE OF CONTENTS

1. KPIs AND BUSINESS REQUIREMENTS

1.1 KPI

High importance KPIs				
#	**KPI**	**Description**	**Outcome vs Driver**	**Internal vs External**
1	Overall **Sales**	The total sales revenues and net sales in the years from 2010 to 2017 and per country	Outcome	External
2	**Net Sales** per retailer	The total net sales for each of the retailers throughout the years, and identify 3 retailers with the most/the least net sales.	Outcome	External
3	**Net Sales** development	The total net sales revenue in the 12 months of the calendar year	Outcome	External
4	**Retailer** movement	The total number of retailers in the last 12 months	Outcome	External
5	Cross-Selling rate	The number of different material types ordered by the retailers in the last 12 months, and identify the retailers with the highest/lowest rate.	Outcome	External
6	**Order** movement	The total number of orders in the last 12 months	Outcome	External
7	Fulfillment rate per **Retailer**	The percentage of shipped orders in relation to all orders placed per retailer	Outcome	External
8	How well are we **fulfilling** our **Sales forecasts**?	The percentage of Sales generated in relation to Sales forecasted per retailer per year	Outcome	Internal
9	**Employee Net Sales generation** per employee	The total revenue generated per employee over all time	Outcome	Internal
10	**Material revenue generation** per month	The revenue generated per material per month	Outcome	External
Medium importance KPIs				
	KPI objective	**KPI description**	**Outcome**	**Internal vs**

			vs Driver	External
1	Average **Customer** Lifetime Value (CLV)	The average accumulated value generated by a retailer over the time since first placing an order	Outcome	External
2	Average **selling** price	The average selling price of products in Euro, individually for finished, semifinished and OEM		
3	Average monthly hours per **employee**	The average number of employees working hours per month	Driver	Internal

Table 1: Identified high & medium importance KPIs

1.2 Business Requirements

#	Business Requirements	Importance	High Level Entities	Measures
1	What is the average monthly net sales?	High	month	Sales Revenues
2	Which 3 retailers have the highest/lowest sales net sales?	High	retailers	Sales Revenues
3	What does our sales development look like throughout the months / year?	High	month, year	Sales Revenues
4	How did the monthly number of retailers change over the last 12 months?	High	retailers, month	
5	Which 3 retailers have the most/the least cross-selling rate?	High	retailers, material, month	
6	How did the monthly number of orders change over the last 12 months?	High	month	
7	What is our fulfillment rate?	High		
8	How well are we fulfilling our Sales forecasts per retailer?	High	retailer	Sales Quantity, Forecast Quantity

9	Which Sales employee generates the most net sales?	High	employees	Sales Revenues
10	What is the most demanded material (based on net sales) by our retailers?	High	material	Sales Revenues

Table 2: Business requirements and high level entities

1.3 Information Packages

Entities	**Hierarchies in entities**
Material (Material Name, Material Description, Material Type	Material Type > Material Description > Material Name
Retailer (Name, Description, Category, Address, State, Contact Person, Phone, Email, Fax, Currency)	Category > Description > Name Country > Address
Employee (Name, Gender, Street, Zip Code, City, Country, Department, Phone, Email	Country > City > Zip Code > Street
Date (Year, Month, Week, Day)	Year > Month Year > Week
Measures	Price in Euro, Sales Quantity, Sales Revenues, Forecast Quantity

Table 3: Information packages

2. CONCEPTUAL MULTI-DIMENSIONAL DESIGN

Below is our ME/R Diagram based on the data from the Information Packages and from the raw database from Super X.

We decided on a **multi-star schema**, where we have 2 fact tables for Sales and Forecast measures. The fact table Sales is connected to dimension tables Date, Material, Employee and Retailer, while the fact table Forecast is only connected to Date, Retailer and Material.

In the original database, there was a separate table for Order Item, but we decided to break down that table, combining price and quantity into the fact table Sales.

For the dimension table Retailer, we added a new attribute called Country, whose data was extracted from the Address. This would be meaningful later on for looking at sales growth by country.

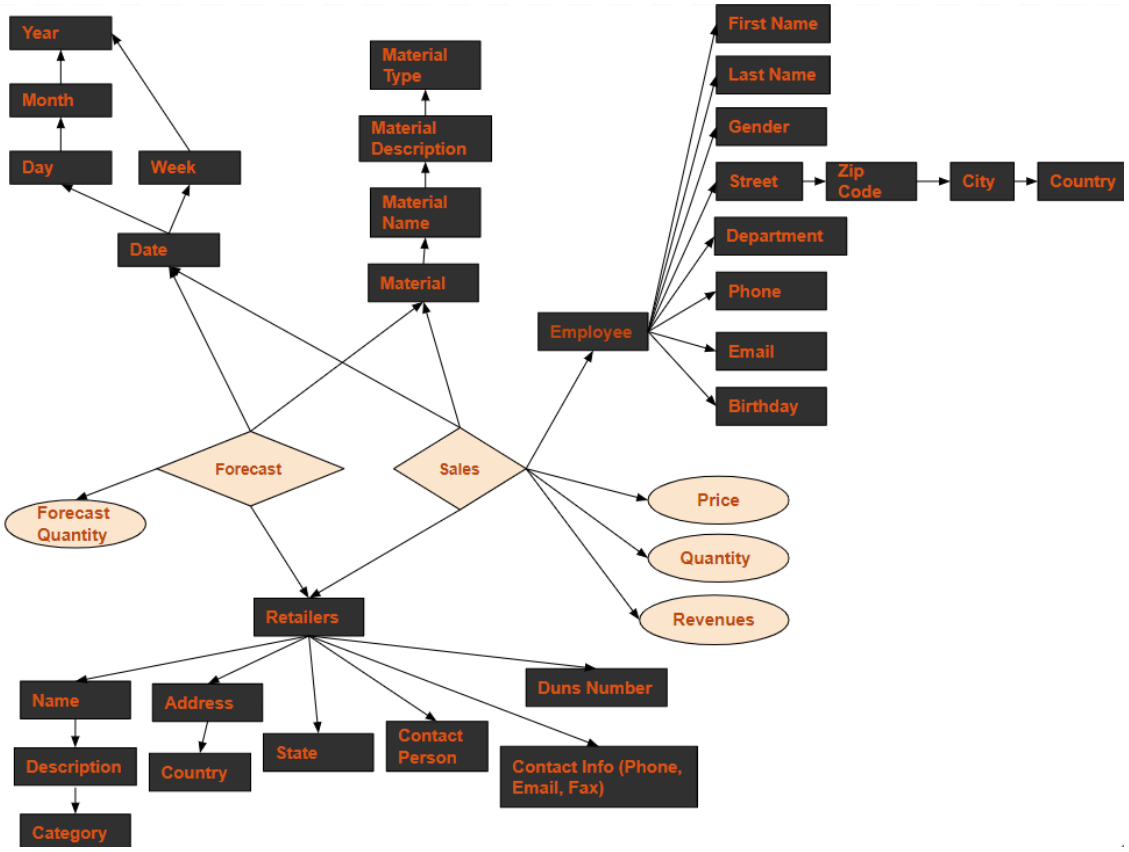


Figure 1: ME/R Diagram[1]

[1] https://docs.google.com/drawings/d/1A-zXo4cE0RDgWckY59NK1_usl5KzdDR9iQSwlU5nSwo/edit?usp=sharing

3. DATA PROFILING: Talend

We conducted a first data profiling at the start of the project, however, many more problems in the data quality appeared later on in the project, such as in the ETL & Dashboard phase. These data issues will not be described here, but mostly in the ETL part of the documentation and in the business recommendation part of the documentation.

The schema has 39 tables, with 679.058 rows. From these tables, there are two without key: schema_migrations and wrk_materials.

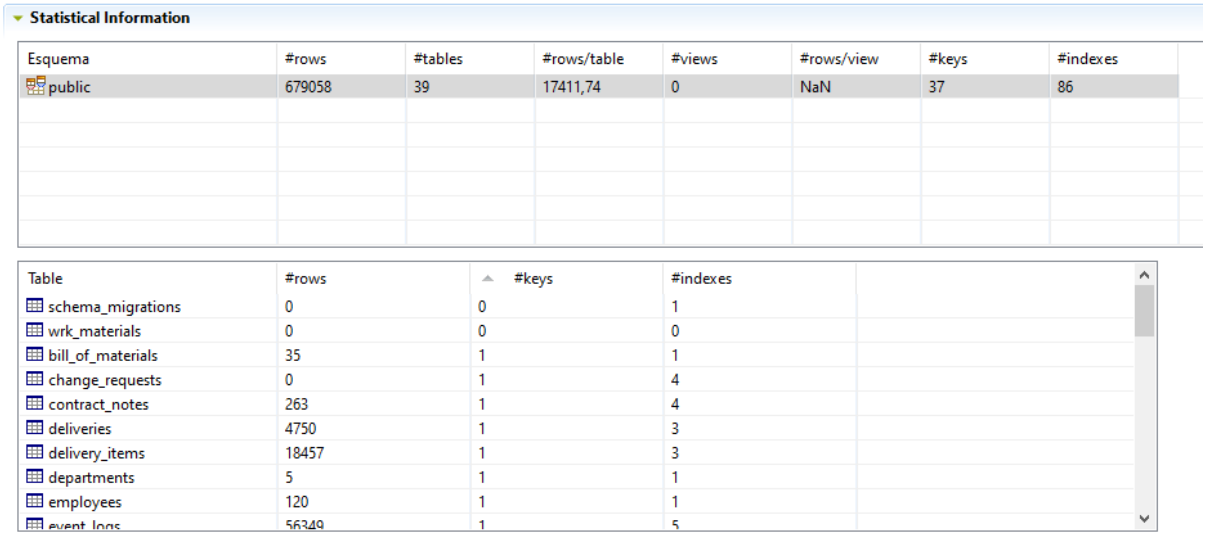


Figure 2. Talend General Overview

3.1 Retailer's Data Overview

The most usual categories of retailers are *offline* and *offline retailer*. Both of them seem to have the same meaning. They could be merged into a single category.

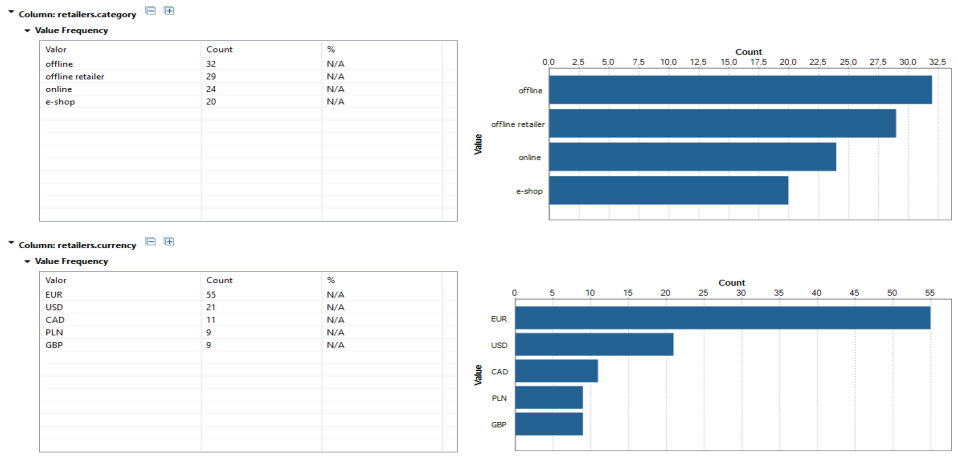


Figure 3. Retailer's Data Overview 1

The most common currency among our retailers is Euro, followed by USD. For comparing and aggregating revenues, we need to convert currencies into one uniform currency.

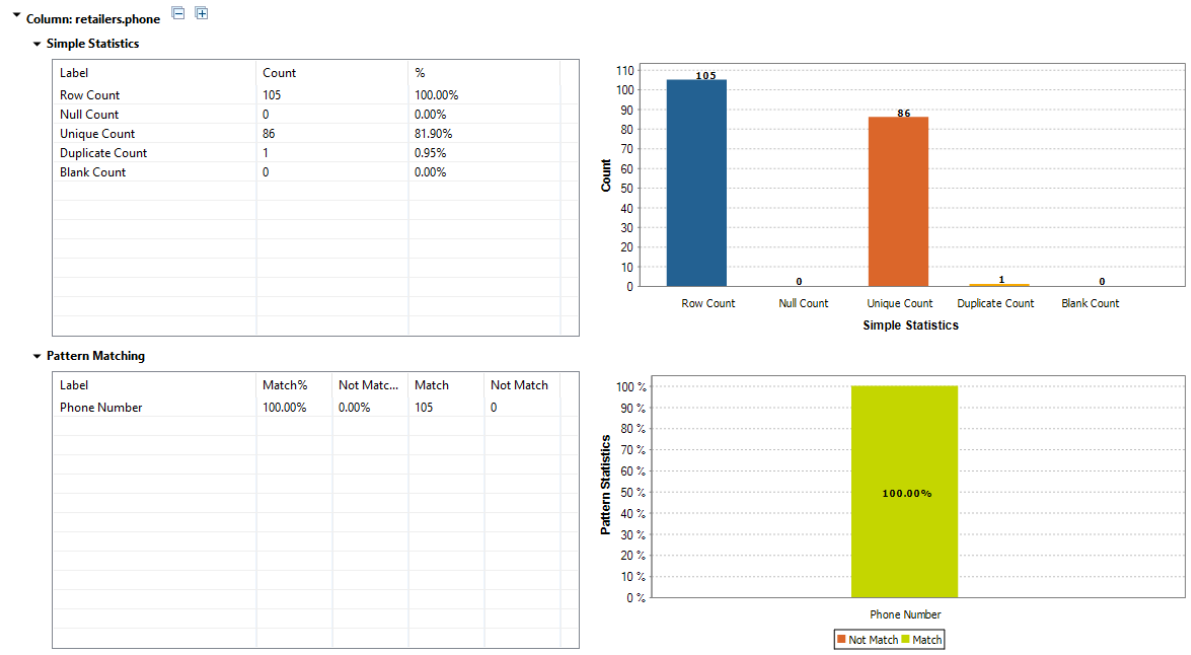


Figure 4. Retailer's Data Overview 2

For the phone numbers of the retailers, we observe that we have one phone value, the *(999) 999-999*, duplicated 19 times. It might be a default value if no phone is added. This should be corrected, being "NaN" or an empty field. Both of which would be a better approach for missing phone values.

1 [SELECT * FROM "public"...] | Messages

id	name	categ...	description	address	contactp...	phone	fax	email	dun...	state	currency	timestamp
5	Dekker V.O.F.	offlin...	Switchabl...	Broeklaan...	Luuk Boe...	(999) 999-999	06 3458 0442	iii.luuk.boer@jacobs.net	58-...	active	EUR	2009-12-31...
13	Hardy UG	online	Innovativ...	Im Winke...	Ana Sch...	(999) 999-999	(0103) 202343071	schwarthoff.ana@farbersaumweber.de	20-...	active	EUR	2009-12-31...
21	Wiśniewski, Wysocki and Grzegorczyk	offlin...	Progressi...	al. Drabik ...	Dariusz C...	(999) 999-999	58-212-75-66	czechowski_dariusz@ostrowski.com.pl	32-...	active	PLN	2009-12-31...
23	Sanford and Sons	offline	Persistent...	65614 An...	Ebony Sc...	(999) 999-999	172-977-9400 x1169	ebony.schaefer@harveytremblay com	36-...	active	CAD	2009-12-31...
34	Rath, Schuppe and Runte	e-shop	Optimize...	8577 Carri...	Pierre Be...	(999) 999-999	586-541-3422 x9594	pierre.beahan@wuckert.net	65-...	active	USD	2009-12-31...
35	Gabler OHG	offlin...	Front-line...	Brandenb...	Annie Ob...	(999) 999-999	+49-268-3573900	obergf.ll.annie@gellinghartlieb.org	76-...	active	EUR	2009-12-31...
36	Kozey Group	offline	Ameliorat...	474 Mack...	Kole Strei...	(999) 999-999	939.415.5601	streich.kole@terry.ca	25-...	active	CAD	2009-12-31...
37	Krohn Gruppe	offlin...	Virtual sol...	Im Jücher...	Victor Lei...	(999) 999-999	+49-7754-49210068	victor_leiteritz@khnke.com	69-...	active	EUR	2009-12-31...
39	Stürmer, Grotke und Gutowicz	offline	Switchabl...	Karl-Arno...	Saskia En...	(999) 999-999	+49-315-9662824	saskia_engelen@walter.ch	89-...	active	EUR	2009-12-31...
50	Daugherty-Effertz	online	Program...	55167 Har...	Stefan H...	(999) 999-999	(999) 999-999	stefan_homenick@olson.name	25-...	active	GBP	2009-12-31...
51	Villa-Ferretti SPA	online	Migrazio...	Strada M...	Carlo De ...	(999) 999-999	311 030 741	de.luca.carlo@marinograo.org	81-...	active	EUR	2009-12-31...
58	Weimer, Scheuring und Schönball	offlin...	Open-arc...	Berliner P...	Sena Götz	(999) 999-999	(05497) 0826090	sena.g.tz@noack.info	85-...	active	EUR	2009-12-31...
62	Schuppe, Feil and Flatley	offline	Open-so...	1396 Rhet...	Alfonzo ...	(999) 999-999	956.518.4463 x9408	alfonzo_beier@trompfritsch.net	30-...	active	USD	2009-12-31...
63	Ferretti, Monti e Fabbri e figli	online	Portale si...	Borgo M...	Rosalba ...	(999) 999-999	+79 989 18648742	gallo.rosalba@basile.it	08-...	active	EUR	2009-12-31...
67	Bieler-Ochs	offlin...	Reactive ...	Monheim...	Ryan vo...	(999) 999-999	(04587) 3466430	vom_ryan_dietrich@salow.com	91-...	active	EUR	2009-12-31...
68	Frączek Group	offlin...	Ergonomi...	al. Piłat 8...	Maksym ...	(999) 999-999	(999) 999-999	maksym.ra.g@komorowski.org	00-...	active	PLN	2009-12-31...
69	Lehner Group	offline	Profound...	982 Ziem...	Gabriel R...	(999) 999-999	108-726-4018 x8249	jr_gabriel_raynor@wisoky.net	67-...	active	CAD	2009-12-31...
77	Kunde, Welch and Pfeffer	online	Front-line...	99302 Ch...	Derick M...	(999) 999-999	344-152-7940 x707	mclau hlin_derick@ullrichwest.com	69-...	active	CAD	2009-12-31...
98	Charles et Lucas	offline	Open-so...	97 Place ...	Alexis Oli...	(999) 999-999	0322176609	olivier.alexis@richa d.net	54-...	active	EUR	2009-12-31...

Figure 5: Retailer's Phone Number Overview

All the retailer's emails are unique, which means that we have a unique communication channel for each retailer. However, there are 21.90% of email addresses that don't match the appropriate email pattern, making them invalid.

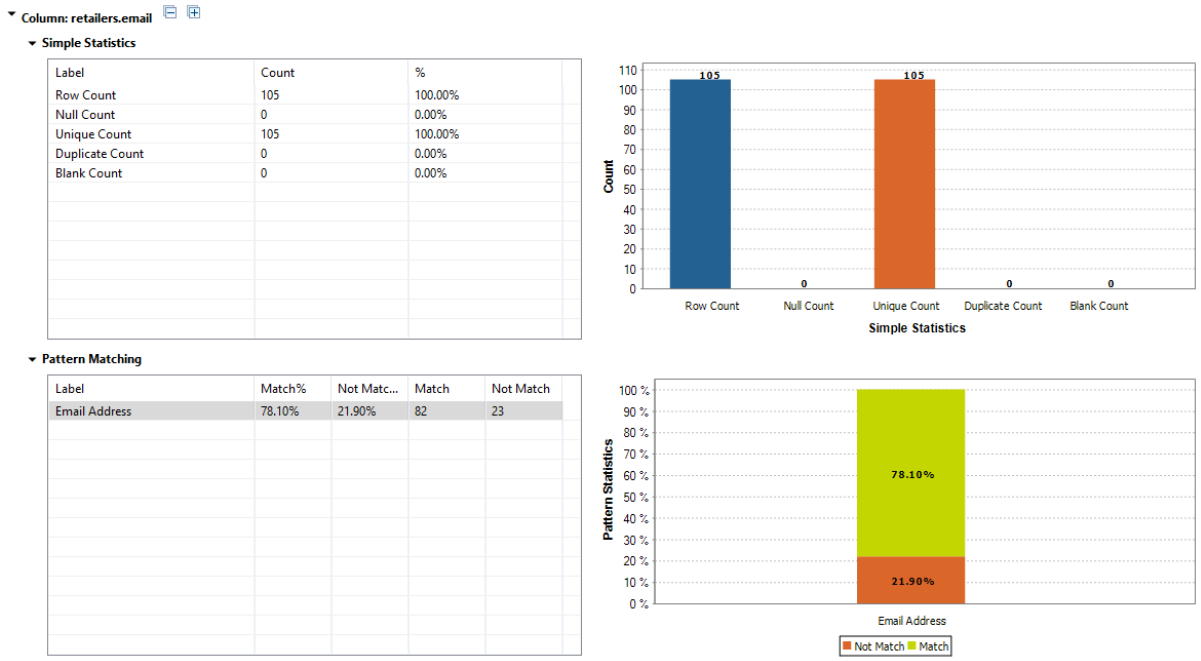


Figure 6: Retailer's Email Address Overview 1

If we take a closer look, we'll be able to identify typographic errors in the writing of the emails, where some letters are replaced with empty spaces. These errors should be corrected, verifying the right address for each contact.

1 [SELECT * FROM "public"...] | Messages

id	name	cat...	description	address	contactperson	phone	fax	email	duns_number	state	currency	timestamp
1	Klein, Willems and M...	e-s...	Business-f...	Kevinstraa...	Thijs van Janss...	5083327...	0653093140	van_janssen_thijs@wal nl	97-274-2311	active	EUR	2009-12-3...
6	Bétancourt Murillo e ...	offl...	metodolo...	Lado Marí...	Sr. Nicolás Ullo...	946-698...	993649906	nicol_sr_ulloa_s_lebr_n@alvarez.e	55-822-1619	active	EUR	2009-12-3...
7	Morar, Corwin and Lo...	offl...	Decentrali...	730 Johns...	Loyce Schmeler	484.814....	863-440-54...	sch eler.loyce@mann.org	08-344-2042	active	USD	2009-12-3...
8	Kautzer, Jast and Borer	e-s...	Business-f...	14218 Sch...	Demond Rau	563-158...	(999) 999-9...	rau.demon @johns.info	60-131-1972	active	CAD	2009-12-3...
10	Martinez et Vidal	offl...	Multi-late...	80 Rue de...	Mme Romane ...	0469642...	+33 596146...	mme_baro _romane@gonzalez.com	54-077-0309	active	EUR	2009-12-3...
15	Schouten, Ven and Bos	offl...	Sharable c...	Smitlaan ...	Eva Brink	1578708...	(8743) 8273...	eva.brin @vriesbrouwer.org	66-283-4274	active	EUR	2009-12-3...
20	Kutch Inc	onl...	Streamlin...	38356 Nin...	Beth Berge	890-099...	829-476-46...	berge_beth@f hey.org	26-485-9099	active	CAD	2009-12-3...
23	Sanford and Sons	offl...	Persistent ...	65614 Ana...	Ebony Schaefer	(999) 99...	172-977-94...	ebony.schaefer@harveytremblay com	36-060-8038	active	CAD	2009-12-3...
24	Maillard SEM	offl...	Networke...	9103 Impa...	Prof Juliette C...	+33 665...	0589986994	prof_juliette_colin@ irard.fr	49-446-8532	active	EUR	2009-12-3...
25	Barreto S.A.	e-s...	éxito terci...	Parque A...	Lucia Almonte...	9448968...	997-927-756	pri to.lucia.almonte@vega.es	49-542-6231	active	EUR	2009-12-3...
26	McClure, Gibson and ...	offl...	Fully-conf...	3198 Ima ...	Jewel Osinski	056 828...	(999) 999-9...	jewel.osinski@rayno kirlin.info	94-415-4414	active	GBP	2009-12-3...
28	Jast-Runolfsdottir	offl...	Cloned co...	977 Borer ...	Elna Stehr	01830 4...	0998 628 5...	elna_stehr@stantonn colas.biz	41-327-5663	active	GBP	2009-12-3...
40	Goyette and Sons	offl...	Right-size...	5178 Tony...	<null>	374-952...	357.831.43...	jermaine.rutherford@wit ing.ca	18-890-1200	active	CAD	2009-12-3...
44	Testa-Palumbo e figli	offl...	Forza lavo...	Borgo Avi...	Adriano Mazza	+60 976...	+39 019 14...	mazza_adriano@milan .org	02-416-6299	active	EUR	2009-12-3...
46	Dumont et Brunet	offl...	Synergisti...	823 Quai ...	M Mathis Lecl...	+33 667...	0620836878	mathis.m. eclerc@robert.org	56-620-3075	active	EUR	2009-12-3...
47	Szyszka, Zajączkowski...	offl...	Cross-plat...	ul. Karpiń...	Oleg Leśniak	41-416-...	(999) 999-9...	niak.oleg.le@decfalkows i.pl	38-687-8121	active	PLN	2009-12-3...
52	Daniel-Marvin	offl...	Vision-ori...	8811 Vand...	Kobe Goodwin	011916 ...	055 4572 0...	kobe. oodwin@towne.co.uk	03-718-3625	active	GBP	2009-12-3...
54	Koj-Flore	onl...	Enhanced...	Hannah-...	Laura Urbansky	(0177) 0...	(0518) 0838...	urbansky_la ra@hommel.com	38-235-1823	active	EUR	2009-12-3...
77	Kunde, Welch and Pfe...	onl...	Front-line...	99302 Cha...	Derick McLaug...	(999) 99...	344-152-79...	mclau hlin_derick@ullrichwest.com	69-369-5520	active	CAD	2009-12-3...
78	Rohan, Bogan and Co...	offl...	Distribute...	965 Charl...	Darian Wiza	0858 52...	055 6809 4...	wiza.darian@s orer.name	12-794-1917	active	GBP	2009-12-3...
90	Honz, Schima und Eff...	onl...	Organic l...	Geibelstr. ...	Raik Birkemeyer	(03236) ...	+49-902-9...	raik.birkemeyer@g eithanner.org	97-545-6397	active	EUR	2009-12-3...
94	Lockman and Sons	onl...	Re-engine...	584 Lonny...	Sydnie Bradtke	1-370-4...	1-785-510-...	bradtke.sydnie@bergstr m.biz	73-256-4926	active	CAD	2009-12-3...
98	Charles et Lucas	offl...	Open-sou...	97 Place d...	Alexis Olivier	(999) 99...	0322176609	olivier.alexis@richa d.net	54-109-3412	active	EUR	2009-12-3...

Figure 7: Retailer's Email Address Overview 2

If we proceed to do a matching analysis, we'll find out that there are five retailers which have been created twice. Two of these retailers include misspellings. The correct retailer needs to be verified, and the duplicate needs to be dealt with.

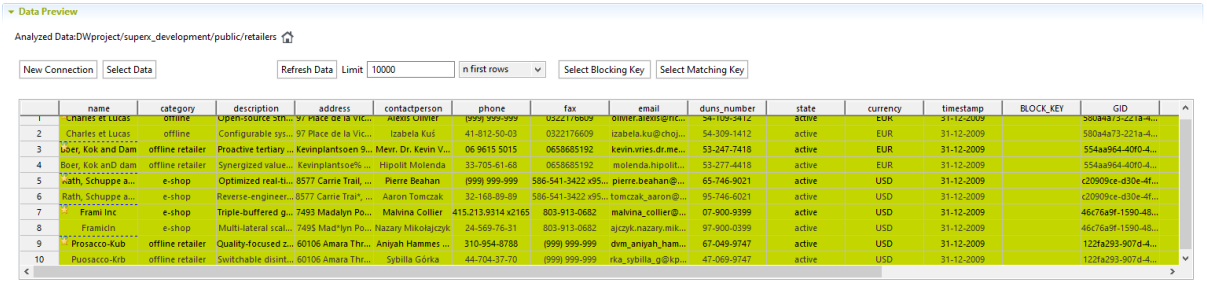


Figure 8: Retailer's Duplicates

To identify the correct retailers, they can be validated using addresses and company names. Apart from that, the field *description* should also be checked, as we have different descriptions for the same retailer *name*, which might or might not be correct.

To conclude, all retailers IDs are found in the orders. This means that all registered retailers are active and their appearance in the database is meaningful:

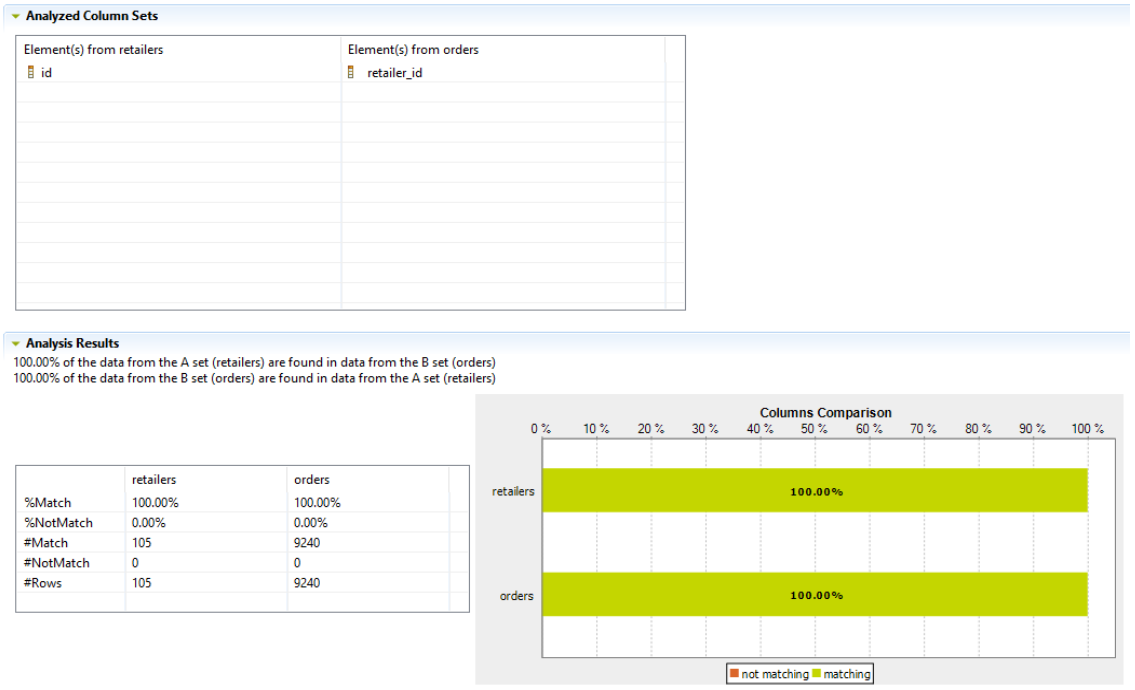
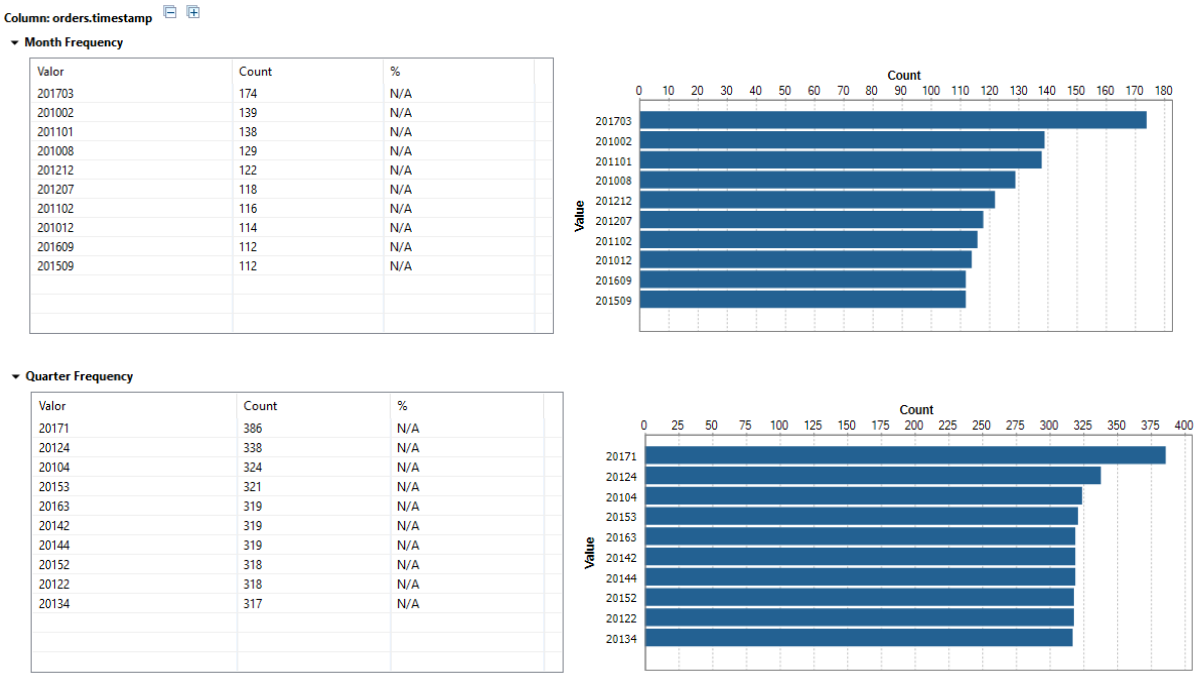


Figure 9: Retailer's ID Overview

3.2 Order's Timestamp Overview

If we analyze the timestamp of the orders, it's interesting to observe that 2017 is a promising year. The database contains orders until 2017-04-18 and, until that date, both the highest month frequency and quarter frequency of orders are led by a register of that year.



Column: orders.timestamp

Month Frequency

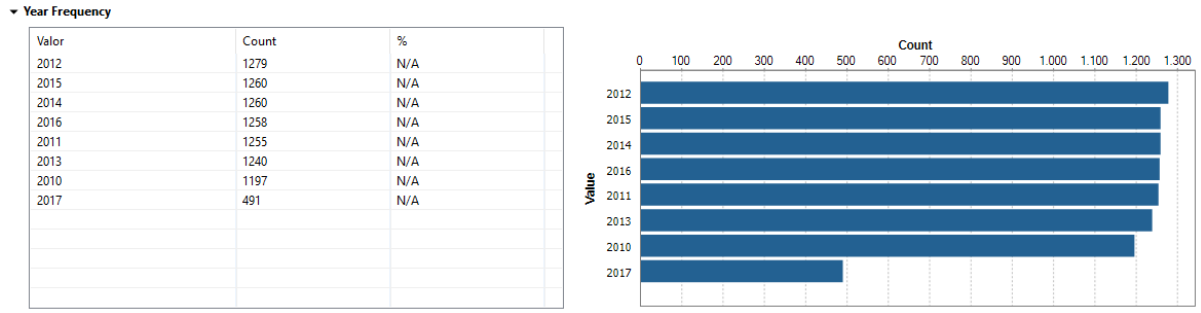
Valor	Count	%
201703	174	N/A
201002	139	N/A
201101	138	N/A
201008	129	N/A
201212	122	N/A
201207	118	N/A
201102	116	N/A
201012	114	N/A
201609	112	N/A
201509	112	N/A

Quarter Frequency

Valor	Count	%
20171	386	N/A
20124	338	N/A
20104	324	N/A
20153	321	N/A
20163	319	N/A
20142	319	N/A
20144	319	N/A
20152	318	N/A
20122	318	N/A
20134	317	N/A

Figure 10: Order Count Overview (Month & Quarter)

The important part for data quality is being aware that the order's registers reach until April 2017. So year frequency, for example, shouldn't be taken into account, because the classification might be misleading, as the year is not finished yet.



Year Frequency

Valor	Count	%
2012	1279	N/A
2015	1260	N/A
2014	1260	N/A
2016	1258	N/A
2011	1255	N/A
2013	1240	N/A
2010	1197	N/A
2017	491	N/A

Figure 11: Order Count Overview (Year)

3.3 Employees Overview

We have a valid phone and email for each employee, which is good to confirm.

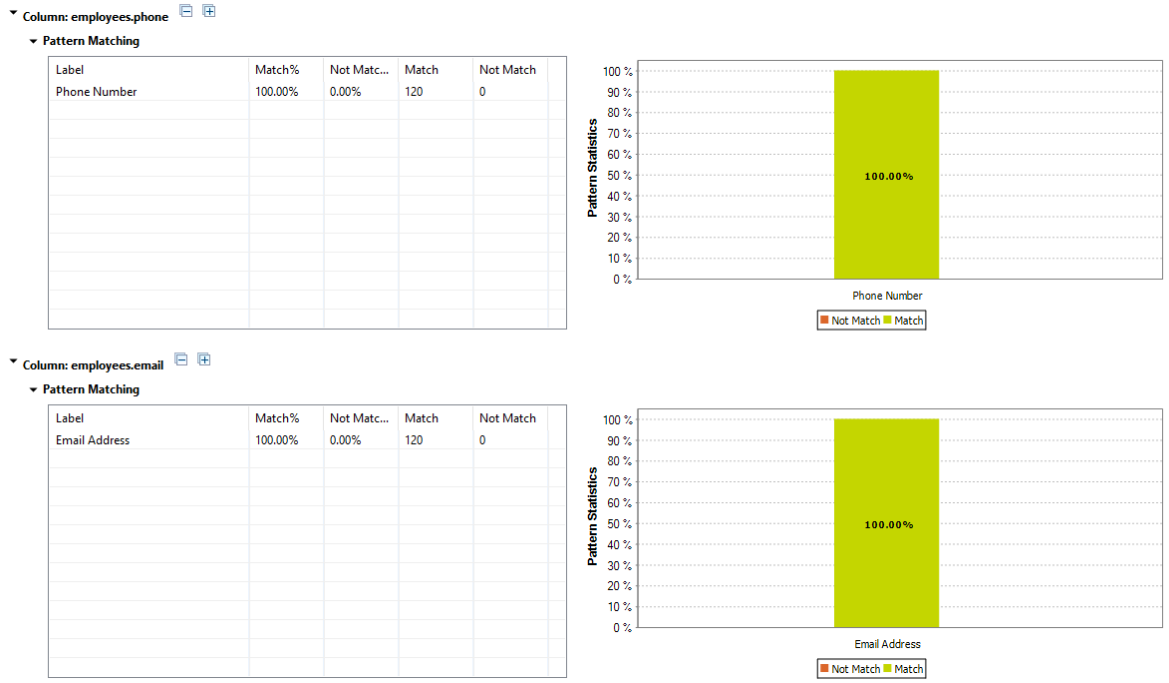


Figure 12: Employer's Phone and Email Overview

What should be solved in the employee database is the gender, as 100% of the workers are classified as males, even though it looks like there are some female names in the company.

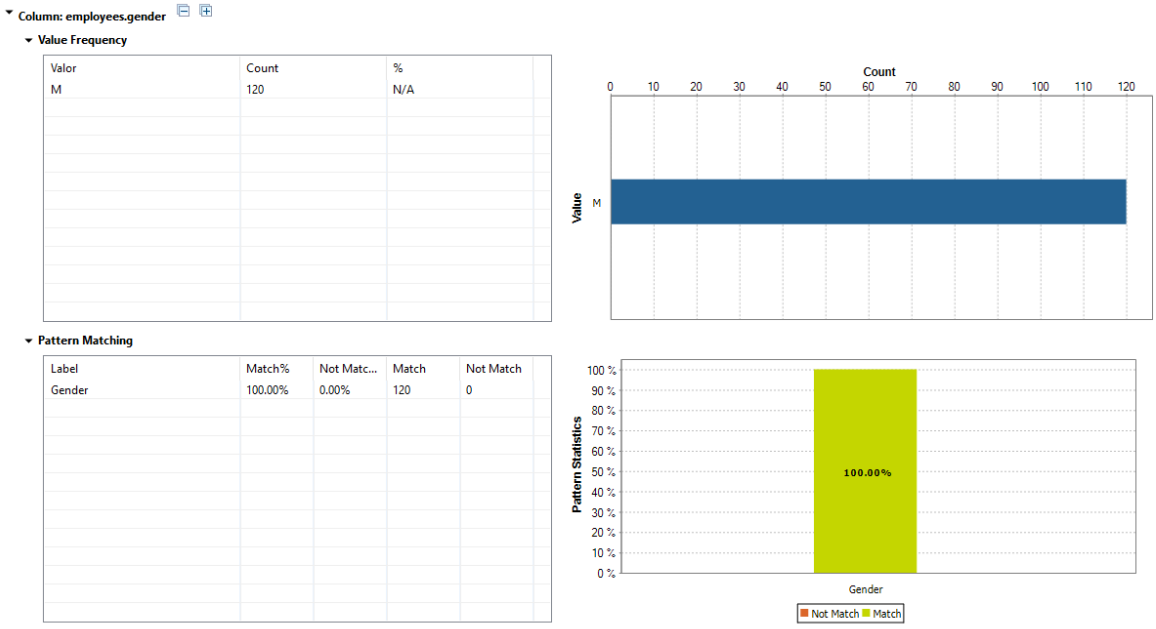


Figure 13: Employee's Gender Overview

1 [SELECT * FROM "public"...] Messages

id	firstname	lastname	street	zipcode	city	phone	email	birthday	country	department_id	timestamp	gender
1	Lyn	Steinert	Hufer Weg 47b	88996	Berlin	(08291) 1150004	lyn@heebrosch.info	1973-06-06 00:00:00.000	Germany	2	2009-12-31 00:00:00.000	M
2	Jamal	Ne	Am Benthal 50b	60452	Berlin	(07270) 1022065	jamal@marahrens.net	1993-04-24 00:00:00.000	Germany	5	2009-12-31 00:00:00.000	M
3	Selina	Ranftl	Max-Delbrück-Str. 93	59426	Berlin	(07932) 9126124	selina@lack.de	1998-12-19 00:00:00.000	Germany	5	2009-12-31 00:00:00.000	M
4	Jamie	Wyludda	Umlag 13c	40527	Berlin	(07407) 1601766	jamie@loogen.net	1970-01-04 00:00:00.000	Germany	4	2009-12-31 00:00:00.000	M
5	Inka	Rau	Hüscheider Str. 38b	28617	Berlin	+49-230-5447753	inka@schleymalucha.ch	1966-09-01 00:00:00.000	Germany	1	2009-12-31 00:00:00.000	M
6	Nisa	Schüppel	Domblick 94a	48985	Berlin	(04491) 9492708	nisa@dchertstppler.ch	1974-08-30 00:00:00.000	Germany	3	2009-12-31 00:00:00.000	M
7	Yara	Zimmermann	Philipp-Ott-Str. 92a	76169	Berlin	(04362) 8097968	yara@slotta.com	1997-08-17 00:00:00.000	Germany	3	2009-12-31 00:00:00.000	M
8	Annika	Jucken	A.-W.-v.-Hofmann-Str. 58c	51437	Berlin	(04522) 7237788	annika@walzdrre.ch	1978-08-02 00:00:00.000	Germany	4	2009-12-31 00:00:00.000	M
9	Linus	Peselman	Lingenfeld 83c	56647	Berlin	(05684) 5040071	linus@trautmann.de	1987-06-10 00:00:00.000	Germany	2	2009-12-31 00:00:00.000	M

Figure 14: Employee Name's Overview

4. LOGICAL MULTI-DIMENSIONAL DESIGN

Based on our ME/R diagram, we created the logical multidimensional design of our data mart with column specifications such as data types and name.

Here is the screenshot of our data mart in SQL Power Architect.

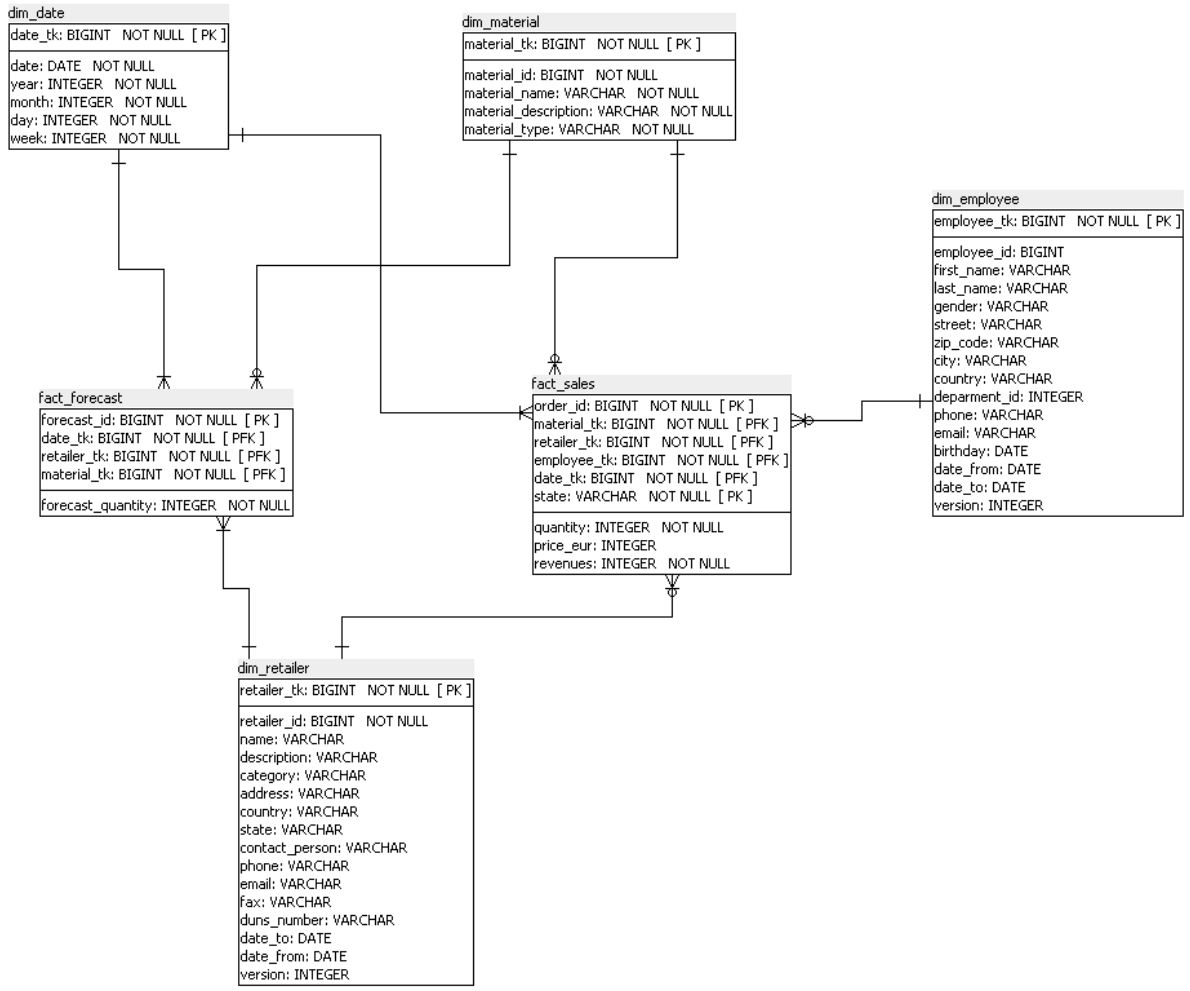


Figure 14: Logical Multi-Dimensional Design

For the ID number and technical keys, we chose BIGINT as the data type with the NOT NULL constraint. The rest of the attributes are either DATE (for dates), INTEGER (for numerical values) or VARCHAR (for name and text).

Since we will apply SCD 2 for dim_retailer and dim_employee, we added attribute date_from, date_to, and version to both tables.

5. ETL

5.1 Loading Dimensions

According to the star schema, we have to load 4 Dimensions: Dim_date, dim_retailer, dim_employee, and dim_material.

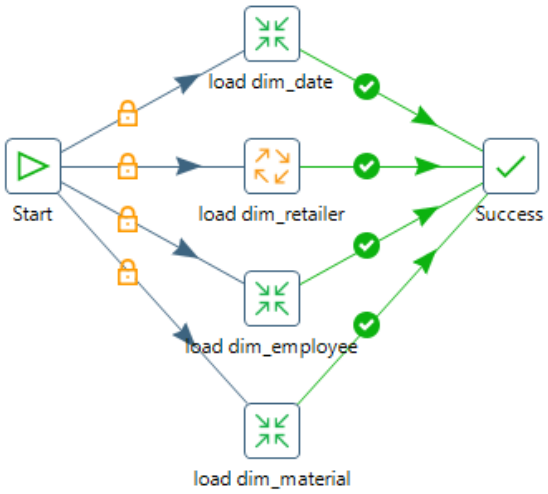


Figure 15: ETL Transformation Overview

5.1.1 Dim_date

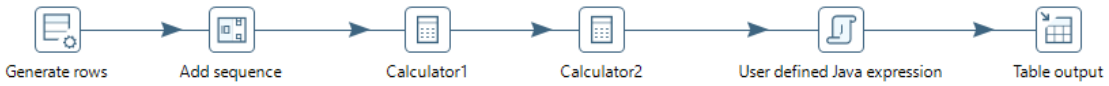


Figure 16: Transformation “dim_date”

For the Dim_Date table we first generate 5000 rows, which should be a good buffer for now, all having the start date 01-01-2010, which is when the first orders came in. Next we generate rows containing values from 0 to 4999 and add these values to the start date from step1. As such, row1 contains 01-01-2010, row2 contains 02-01-2010, and so on.

In a second calculator, we extract day, month, and year from the date, as this is needed for our Dim_Date table as separate attributes. The java expression “Integer.parseInt(new java.text.SimpleDateFormat("yyyyMMdd").format(Date))” creates the date_tk, which will be 20100101 for 01-01-2010. Now all the attributes have been created, and the table can be loaded into the database, this step also drops the start_date and the generated sequence, as those won’t be needed.

5.1.2 Dim_retailer

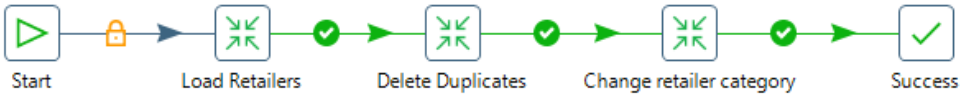


Figure 17: Transformation “dim_retailer” Overview

This is how the loading Dim_Retailer process looks:

Figure 18: Transformation to Load Retailers

To build the Dimension Retailer in the new database, we start by creating a Table Input which will take the raw data from the old retailer table.
After that, we insert a Dimension lookup/update, which will load the data we selected into the new dimension table following the format we design. We choose dim_retailer as the target table from the new database, and we select the primary key, retailer_id, and also the fields we want to export. We also select retailer_tk as our technical key field.

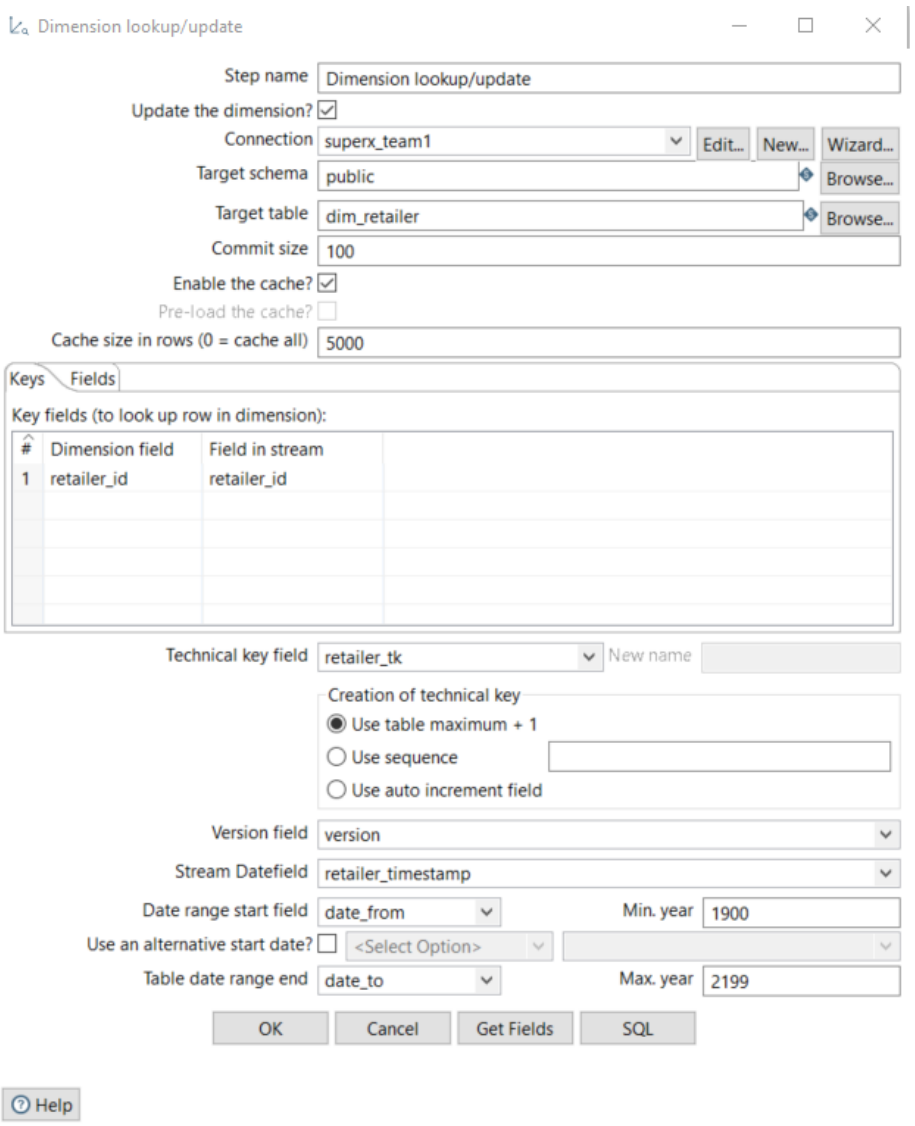


Figure 19: Dimension lookup/update - dim_retailer

After loading the Dim_Retailer, we also need to address data inconsistencies described earlier. First, we have multiple duplicated retailers, which we filter based on three criteria.

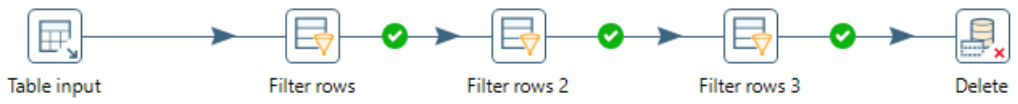


Figure 20: Transformation to Filter Retailers

First we filter all rows, where the address of the retailer contains the symbol “%”. In the second and third step, this is repeated for the symbols “*” and “$”. The results fitting the criteria will finally be deleted from the already loaded Dim_Retailer. At this point it is noteworthy, that we only can use human logic to identify duplicated retailers and the criteria to filter them, as doing it automatically with fuzzy matching would not find any results in two

cases (where the name is identical) and couldn't decide which retailer row is the correct one. As such, we used this workaround, which works nicely for this specific case.

Second, we want to map the retailer category to only contain "online" and "offline" and extract the retailer's country.

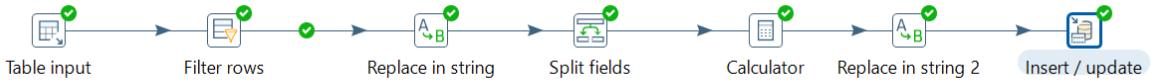


Figure 21: Transformation to Map Retailer's Channel and Country

For this, we first filter rows, where retailer_id is not equal to 0. In the next step, we replace "offline retailer" with the already existing value "offline" and "e-shop" with the already existing value "online".

Then we split the address attribute by comma, creating five new attributes, the last two of which are stating the country. In most cases, only four of these attributes are needed, leaving the fifth attribute empty.

Split fields

Step name: Split fields
Field to split: address
Delimiter: ,
Enclosure:

Fields

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group	Decimal	Currency	Nullif	Default	Trim type
1	address1		N	String									none
2	address2		N	String									none
3	canton		N	String									none
4	country		N	String									none
5	country_2		N	String									none

Figure 22: Split Fields - Mapping Retailer's Country

As such, we replace the empty values in the calculator with the values from the fourth attribute, using the calculation function "NVL(A,B)".

Step name
Calculator

Throw an error on non existing files

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	country_stream	NVL(A, B)	country_2	country		String			N

Figure 23: Getting Retailer's Country

Here again, we noticed some data quality issues, Germany sometimes referenced as "Deutschland" and the U.S.A. sometimes referenced as "USA". To solve this, we once again replace in string accordingly. Finally, we update the rows where the category has been replaced or country was added.

5.1.3 Dim_employee

To build the Dimension Employee, we will follow the same steps as the Dimension Retailer. This is how the loading Dim_Employee process looks:

Figure 24: Transformation dim_employee

First we create a Table Input which will take the raw data from the old employee table. By using an SQL query, we select the fields and their new names, if needed, that we want to input into the process:

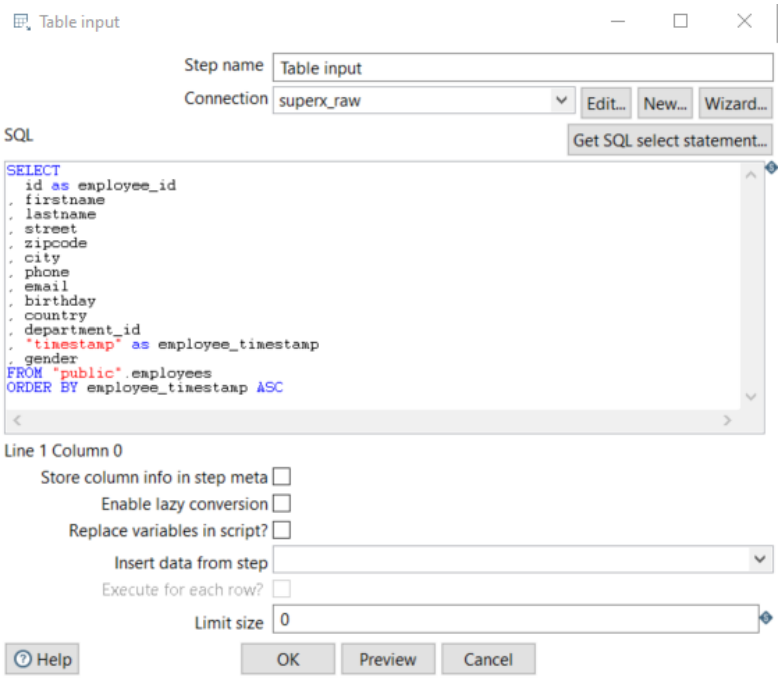


Figure 25: Table Input - dim_employee

After that, we will use a Dimension lookup/update, which will load the data we selected into the new employee table. We choose dim_employee as the target table from the new database, we select the primary key, employee_id, and also the fields we want to export. We also select employee_tk as our technical key field.

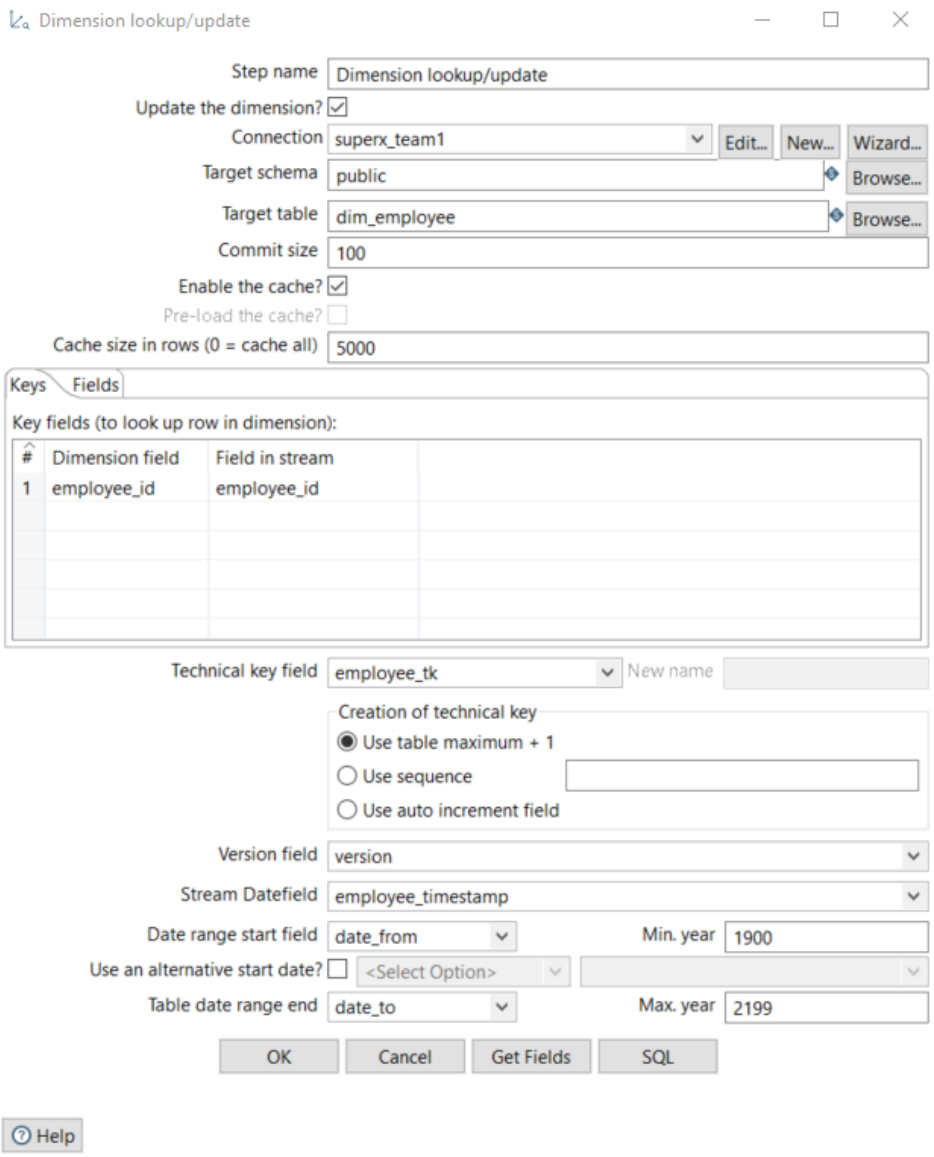


Figure 26: Dimension Lookup/Update - dim_employee

5.1.4 Dim_material

For the dimension table Material, we want to implement Slowly Changing Dimension 1, meaning that we only want the latest information for order items and any data update will overwrite the old one.
This is how the final transformation looks at the end.

Figure 27: Transformation dim_material

The transformation starts with importing the needed data from the raw Super X database with a SQL query.

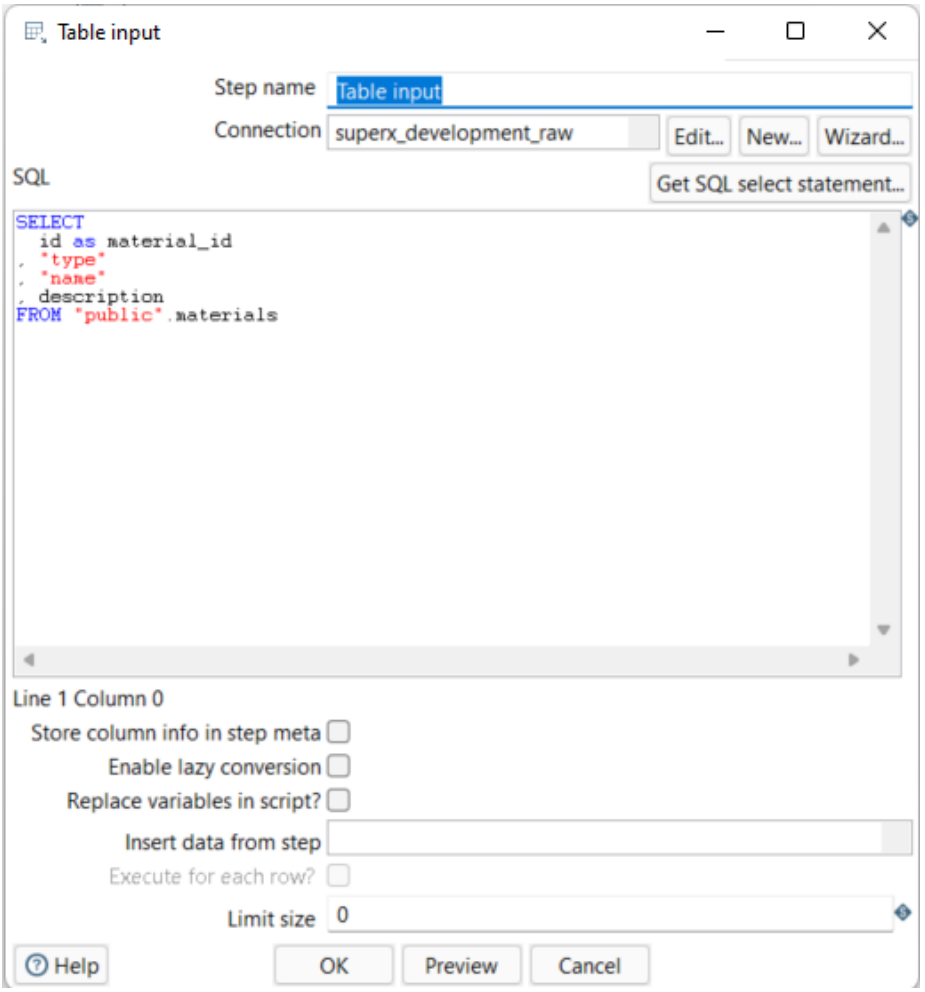


Figure 28: Table Input - dim_material

The next step in the transformation is Insert/Update, where we insert the data into our dimension table dim_material. The key to look up value used is material_id, and we changed the Update option for material_id from Y (Yes) to N (No).

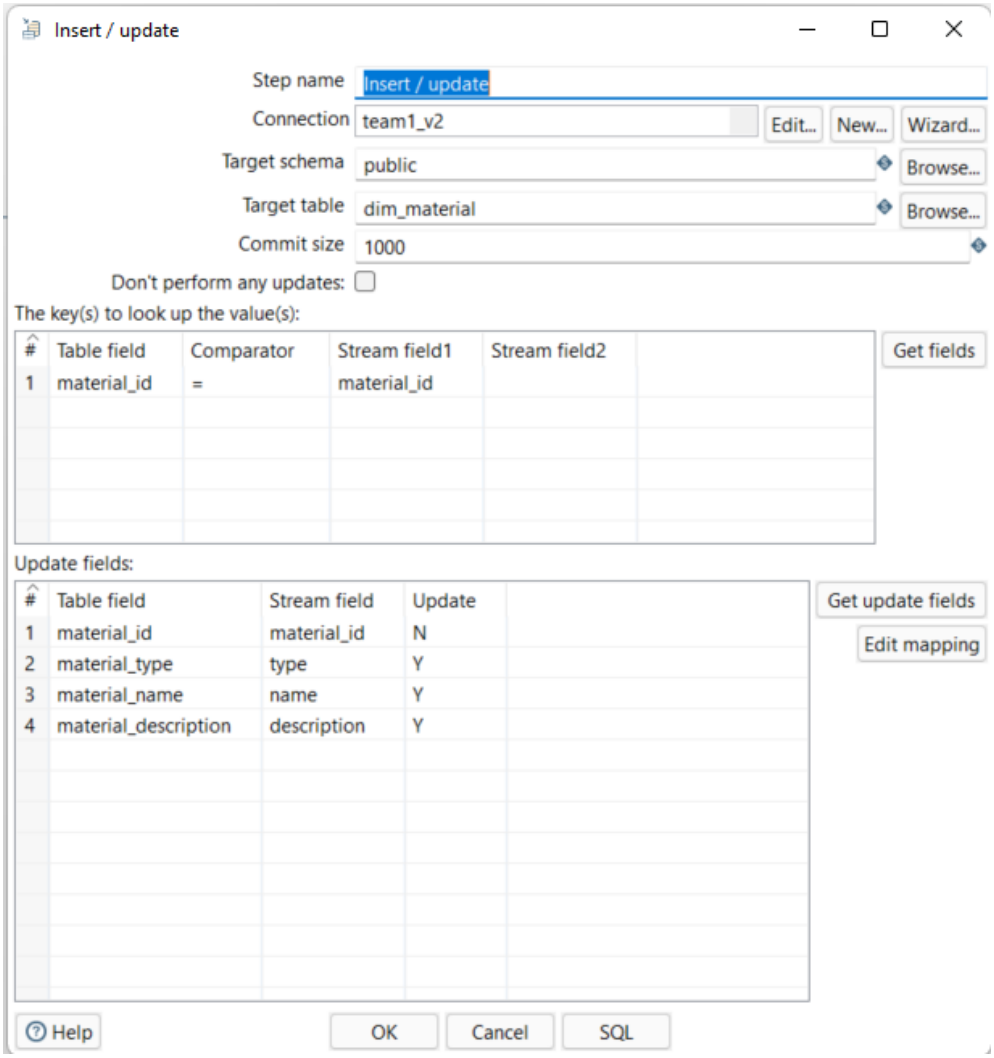


Figure 29: Insert/Update - dim_material

5.2 Loading the fact tables

5.2.1 Fact_sales

When loading Fact_Sales, it is important to keep in mind, we have 5 different currencies in our OLTP, as such we should convert them all to Euro to make the revenue comparable in the final dashboard.

Figure 30: Job to Convert Currency

First, we create an empty currency table with the SQL query:

SQL script to execute. (statements separated by ;) Question marks will be replaced by arguments.

```
CREATE TABLE IF NOT EXISTS currency_conversion (
    date DATE,
    USD NUMERIC(8,4),
    GBP NUMERIC(8,4),
    CAD NUMERIC(8,4),
    PLN NUMERIC(8,4),
    EUR NUMERIC(8,4)
);
```

Figure 31: Create Currency Table

Next, we need to fill the table with currency exchange data.

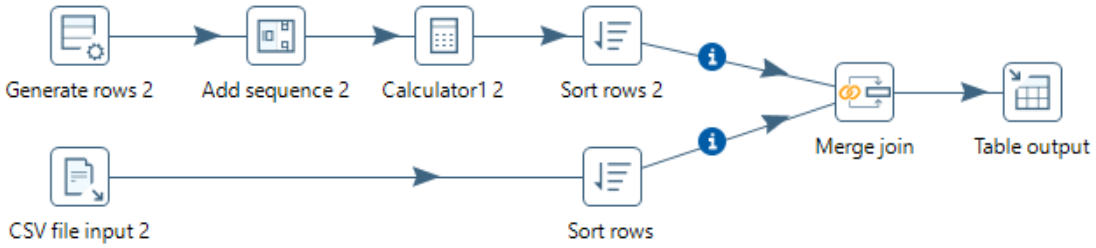


Figure 32: Transformation to Fill Currency Exchange Data

Here, we first generate date rows, similar to the Dim_Date. However, we first start with the starting date 01-12-2009. This will be important in the next transformation. At the same time, all the generated rows also have the Attribute “EUR” which will be used to convert Euro prices into Euro prices. Logically, these always have the value 1. Then we add a sequence and calculate a Date column. Finally, we have to sort by Date, in order to allow a successful merge in Pentaho.

On the other hand of the transformation, we have the currency exchange input. Normally, an API request is the correct way to go, as it allows daily accurate data. However, because of time, effort and lack of APIs we chose to use a CSV file which we got from Team 2. This CSV file has been reduced to the attributes Date, USD, GBP, PLN and CAD in the CSV file input.

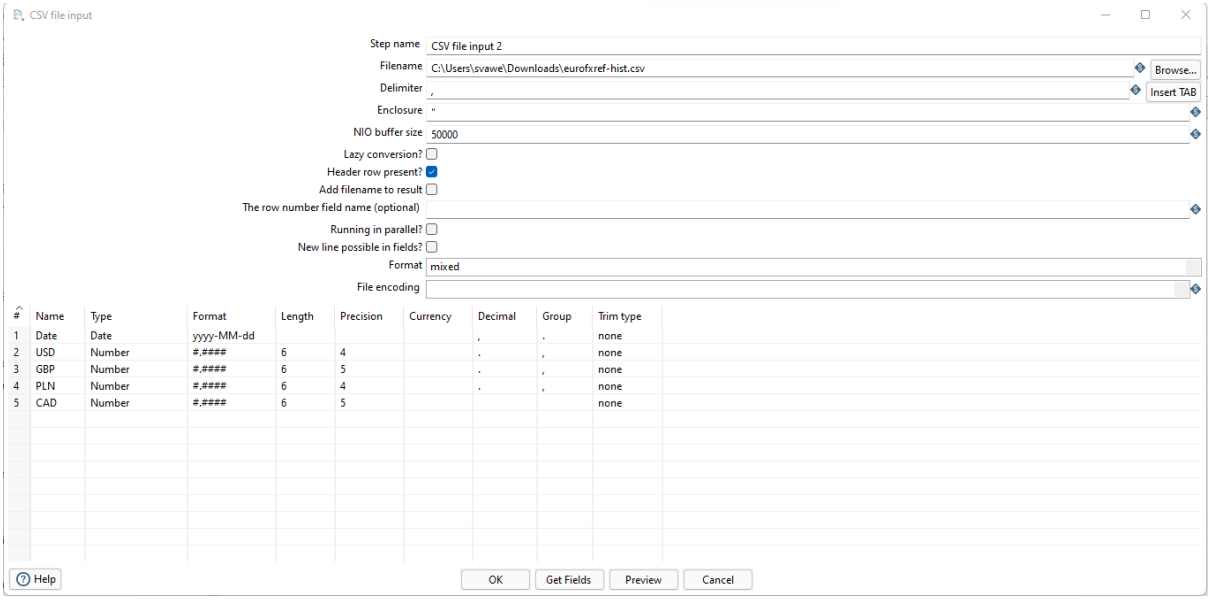


Figure 33: Currency CSV File Input

Next, we sort the CSV input by Date and then do a left outer join on both tables, such that we have a new table, starting on 01-12-2009.

Rows of step: Merge join (1000 rows)

#	first_date	EUR	day_number	Date	Date_1	USD	GBP	PLN	CAD
1	2009-12-01	1	0	2009-12-01	2009-12-01	1.5074	0.9099	4.1098	1.5761
2	2009-12-01	1	1	2009-12-02	2009-12-02	1.509	0.9043	4.1086	1.5766
3	2009-12-01	1	2	2009-12-03	2009-12-03	1.512	0.9092	4.0977	1.5873
4	2009-12-01	1	3	2009-12-04	2009-12-04	1.5068	0.9048	4.0928	1.5778
5	2009-12-01	1	4	2009-12-05	<null>	<null>	<null>	<null>	<null>
6	2009-12-01	1	5	2009-12-06	<null>	<null>	<null>	<null>	<null>
7	2009-12-01	1	6	2009-12-07	2009-12-07	1.4787	0.9051	4.08	1.5664
8	2009-12-01	1	7	2009-12-08	2009-12-08	1.4774	0.907	4.107	1.5595
9	2009-12-01	1	8	2009-12-09	2009-12-09	1.4768	0.9046	4.1336	1.5643
10	2009-12-01	1	9	2009-12-10	2009-12-10	1.473	0.9043	4.1405	1.5472
11	2009-12-01	1	10	2009-12-11	2009-12-11	1.4757	0.9052	4.1457	1.5481
12	2009-12-01	1	11	2009-12-12	<null>	<null>	<null>	<null>	<null>
13	2009-12-01	1	12	2009-12-13	<null>	<null>	<null>	<null>	<null>

Figure 34: Merging Tables Result - Currency Conversion

This table is then pushed to the currency_conversion table.

Now it's time to load in fact_sales.

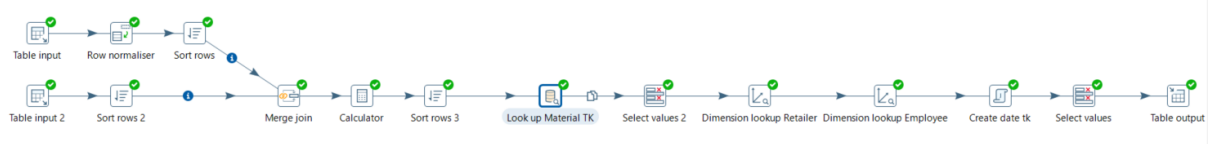


Figure 35: Transformation fact_sales

Since there are several null values, which the API cannot grab because of weekends or holidays, some of our orders wouldn't get an accurate exchange rate. In example, this applies to orders received on New Year and on Easter.

As such, we start the fact_sales transformation with filling these values using an SQL query from the currency_conversion.

SQL

```
SELECT
    to_char(date,'YYYY-MM-DD') as date,eur,
    CASE
        WHEN usd IS NULL THEN (SELECT usd FROM currency_conversion cc1 WHERE cc1.date < cc.date AND cc1.usd IS NOT NULL ORDER BY cc1.date desc LIMIT 1)
        else usd
    end as USD,
    CASE
        WHEN cad IS NULL THEN (SELECT cad FROM currency_conversion cc1 WHERE cc1.date < cc.date AND cc1.cad IS NOT NULL ORDER BY cc1.date desc LIMIT 1)
        else cad
    end as CAD,
    CASE
        WHEN pln IS NULL THEN (SELECT pln FROM currency_conversion cc1 WHERE cc1.date < cc.date AND cc1.pln IS NOT NULL ORDER BY cc1.date desc LIMIT 1)
        else pln
    end as PLN,
    CASE
        WHEN gbp IS NULL THEN (SELECT gbp FROM currency_conversion cc1 WHERE cc1.date < cc.date AND cc1.gbp IS NOT NULL ORDER BY cc1.date desc LIMIT 1)
        else gbp
    end as GBP
from currency_conversion cc
```

Figure 36: Filling Null Values for Currency

This query grabs the last known value to fill in the missing values. This is also the reason why our currency_conversion table starts on 01-12-2009, to make sure we can fill in the missing values with previous data (which we would not have starting on 01-01-2010).

Next, we use a row normalizer to have one row for every attribute and the according date. This will make the following join and the number calculating easier.

Examine preview data

Rows of step: Sort rows (1000 rows)

#	date	Currency	Exchange_rate
1	2009-12-01	CAD	1,5761
2	2009-12-01	EUR	1,0
3	2009-12-01	GBP	0,9099
4	2009-12-01	PLN	4,1098
5	2009-12-01	USD	1,5074
6	2009-12-02	CAD	1,5766
7	2009-12-02	EUR	1,0
8	2009-12-02	GBP	0,9043
9	2009-12-02	PLN	4,1086
10	2009-12-02	USD	1,509
11	2009-12-03	CAD	1,5873
12	2009-12-03	EUR	1,0
13	2009-12-03	GBP	0,9092
14	2009-12-03	PLN	4,0977
15	2009-12-03	USD	1,512
16	2009-12-04	CAD	1,5778
17	2009-12-04	EUR	1,0
18	2009-12-04	GBP	0,9048
19	2009-12-04	PLN	4,0928
20	2009-12-04	USD	1,5068
21	2009-12-05	CAD	1,5778
22	2009-12-05	EUR	1,0
23	2009-12-05	GBP	0,9048
24	2009-12-05	PLN	4,0928
25	2009-12-05	USD	1,5068
26	2009-12-06	CAD	1,5778

Figure 37: Row Normalizer Output - Currency Conversion

On the other hand, we query the data from the OLTP as a basis for our fact_sales.

SQL

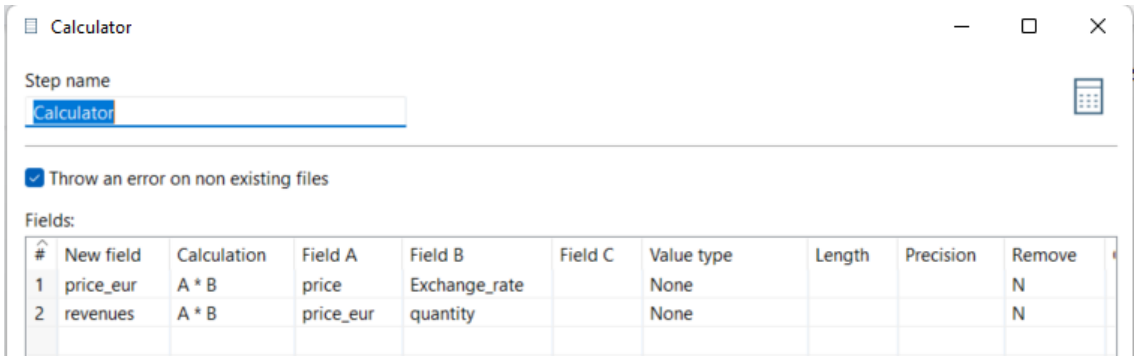
Get SQL select statement...

```
SELECT orders.id as "order_id"
, order_items.material_id
, retailer_id
, employee_id
, order_items.quantity
, order_items.price
, retailers.currency
, orders."state"
, to_char(orders.timestamp,'YYYY-MM-DD') as date
FROM "public".orders LEFT JOIN order_items
on orders.id = order_items.order_id
LEFT JOIN retailers on orders.retailer_id = retailers.id
```

Figure 38: Table Input - fact_sales

This contains all needed columns that we can get from the OLTP and also converts the timestamp into a fitting date format that will allow us to join the tables on Date and Currency, after sorting for these values in both inputs.

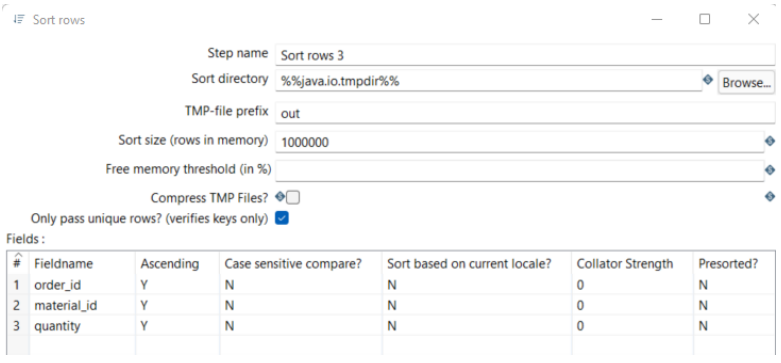
Next, we calculate price in Euro and revenues: for the final fact table, we just want to have the price in Euro as price and then use it to calculate revenues in a uniform unit of measurement.



#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	price_eur	A * B	price	Exchange_rate		None			N
2	revenues	A * B	price_eur	quantity		None			N

Figure 39: Calculator for Price Euro and Revenues - fact_sales

The final table should be sorted in ascending order by first order_id, then material_id and lastly, retailer_id. This was done by Sort Rows on Pentaho.



#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	order_id	Y	N	N	0	N
2	material_id	Y	N	N	0	N
3	quantity	Y	N	N	0	N

Figure 40: Sort Rows - fact_sales

We want to match the material_id retrieved with Table Input from the raw database with the material_id in our dimension table dim_order_item, and then only return the technical key material_tk in fact table Sales to identify material items.

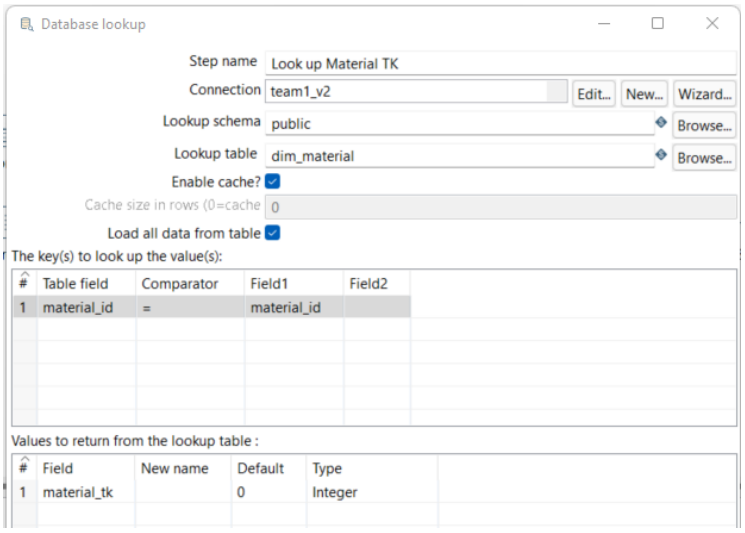


Figure 41: Lookup Material TK - fact_sales

Because we are using SCD 2 for dim_retailer and dim_employee, we need to get the right version for our fact_sale. We used Dimension Lookup in Pentaho, where we matched the retailer_id (as well as employee_id) from the table input from the raw database with the one in the dimension tables, get the right version by comparing date from fact_sales with date_from and date_to from dim_retailer (and dim_employee) and then return retailer_tk (employee_tk) to fact_sales.

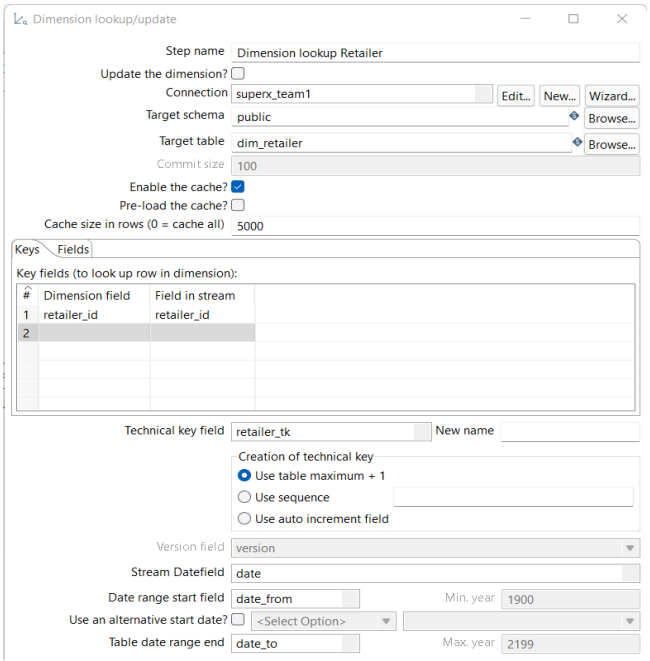


Figure 42: Dimension Lookup Retailer TK - fact_sales

The date_tk in fact_sales was created from date by using the same Java expression as before.

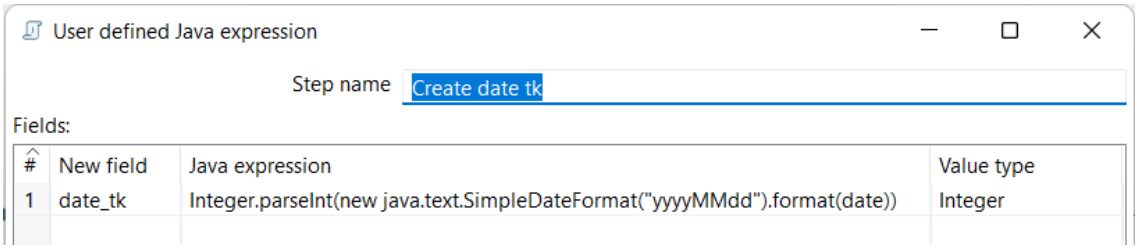


Figure 43: Create Date TK - fact_sales

The last steps were to only select the values we want in the final table and loading it into our fact_sales. Below are the list of attributes we included. In this step, we used Select Values and Table Output.

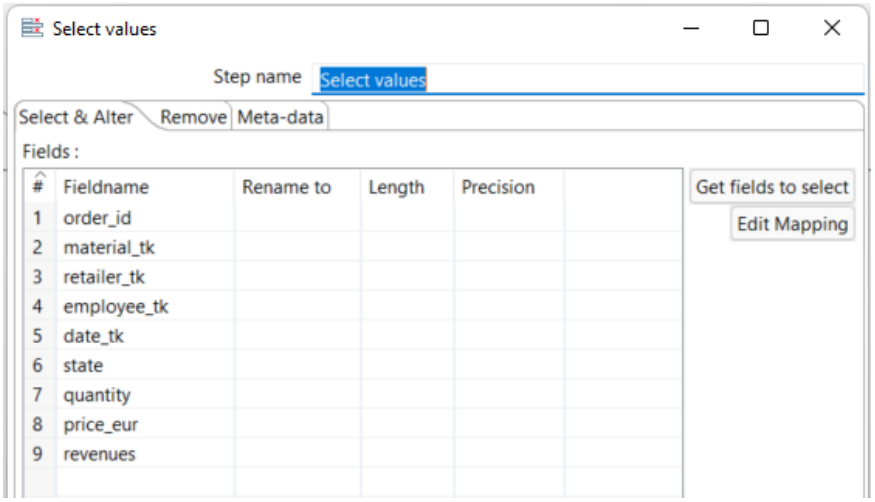


Figure 44: Select Values - fact_sales

	order_id	material_tk	retailer_tk	employee_tk	date_tk	state	quantity	price_eur	revenues
1	9,240	30	45	61	20,170,418	open	22	54	1,195
2	9,240	29	45	61	20,170,418	open	131	106	13,898
3	9,240	28	45	61	20,170,418	open	56	189	10,590
4	9,240	27	45	61	20,170,418	open	114	138	15,677
5	9,240	15	45	61	20,170,418	open	23	0	12
6	9,240	14	45	61	20,170,418	open	30	1	30
7	9,240	12	45	61	20,170,418	open	21	0	10
8	9,240	11	45	61	20,170,418	open	15	0	8
9	9,240	9	45	61	20,170,418	open	51	16	811
10	9,240	8	45	61	20,170,418	open	98	9	843

Figure 45: fact_sales Overview in DBeaver

5.2.2 Fact_forecast

Creating the fact_forecast was partly similar to creating the fact_sales table with fewer steps as all the information we needed for forecast are already available in the raw database. For forecast, we only need to have the forecast quantity as measure, so it did not require much processing with price and currency such as with fact_sales.

This is the transformation to load fact_forecast.

Figure 46: Transformation fact_forecast

For forecasts we want to have all the information in the Forecasts table in the raw database.

Figure 47: Table Input - fact_forecast

We sorted rows in the following order before selecting values we want

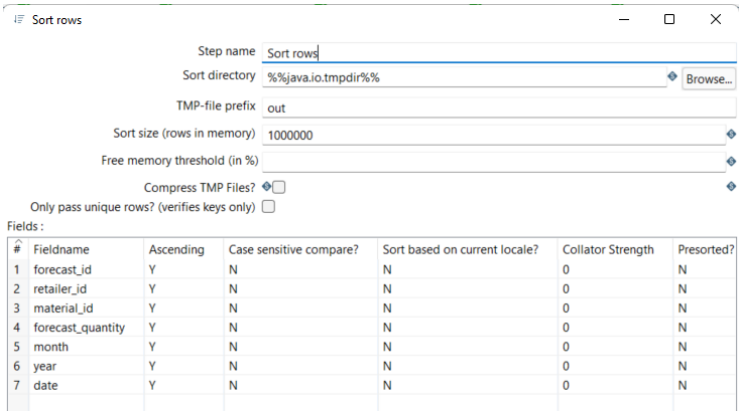


Figure 48: Sort Rows - fact_forecast

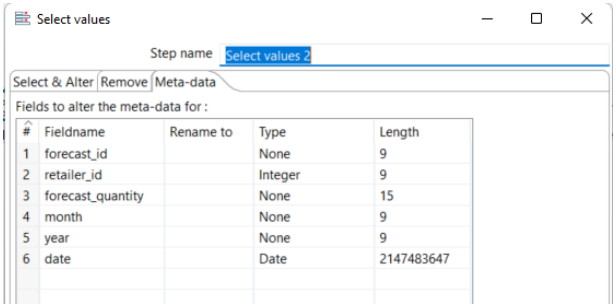


Figure 49: Select Values - fact_forecast

The later steps in the transformation are similar to what we did with fact_sales, where we looked up to get material_tk, looked up to get the right version for retailer_tk based on forecast date, created date_tk, selected the values we wanted in the final table and loaded them in our database. This is how the final table looks in DBeaver:

	forecast_id	date_tk	retailer_tk	material_tk	forecast_quantity
1	1	20,100,101	1	15	129
2	2	20,100,101	1	15	64
3	3	20,100,101	1	15	129
4	4	20,100,101	1	15	258
5	5	20,100,101	1	15	323
6	6	20,100,101	1	15	258
7	7	20,100,101	1	15	129
8	8	20,100,101	1	15	64
9	9	20,100,101	1	15	128
10	10	20,100,101	1	15	258
11	11	20,100,101	1	15	323
12	12	20,100,101	1	15	259
13	13	20,100,101	1	13	129
14	14	20,100,101	1	13	64
15	15	20,100,101	1	13	129

Figure 50: fact_forecast Overview in DBeaver

5.3 Loading CSV files

The final thing we had to do in Pentaho was loading the additional orders. The problem on the CSV files was once again data quality. Here every order did not have an order_item_id, no retailer had a retailer_id, no material had a material_id. Accordingly, we had to join the CSV files with the dimension tables to add the values we needed.
However, this also means that we are not adding new retailers that may appear in the CSV files, but only get orders from the existing 100 retailers.

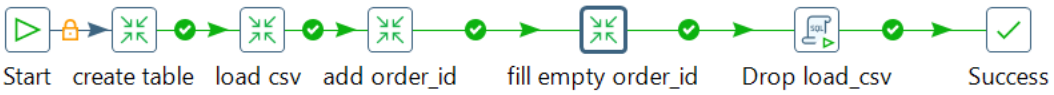


Figure 51: Job to Load CSV Files

First, we created the load_csv table with the following SQL query:

SQL script to execute. (statements separated by ;) Question marks will be replaced by arguments.

```
CREATE TABLE IF NOT EXISTS load_csv (
    order_id int
,   material_tk int
,   retailer_tk int
,   employee_tk int
,   date_tk int
,   quantity int
,   price_eur numeric(10,2)
,   revenues numeric(10,2)
,   state VARCHAR
);
```

Figure 52: SQL Query to Create load_csv Table

Having created this table, we can then load the CSV files.

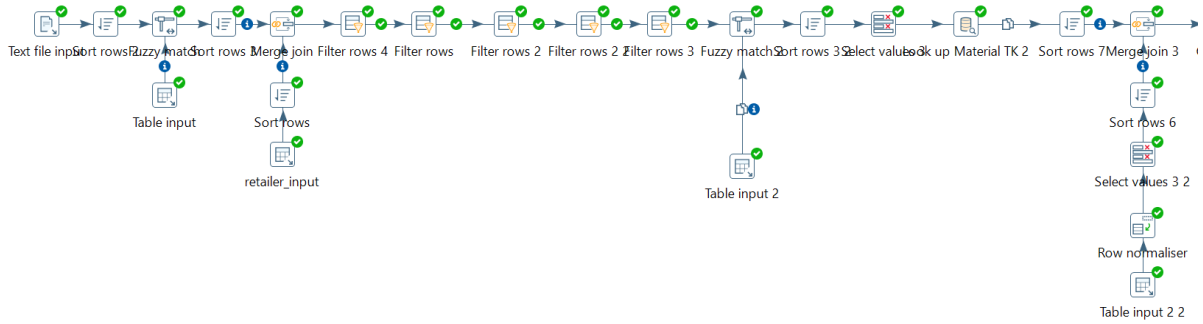


Figure 53: Transformation to Load CSV Files

First, we input all CSV files at once, using the text file input:

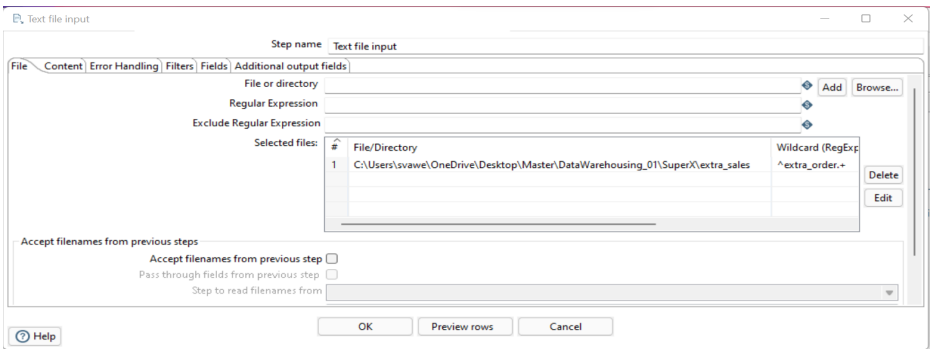


Figure 54: Text File Input - Load CSV Files

For this, we need to specify the directory, the Regular Expression and the encoding in “Content” to UTF-8, making sure that all special symbols are correctly extracted.

Next, we want to fuzzy-match wrong retailers from the CSV to our retailers from the OLTP. For this, we need to sort the retailers from our input alphabetically, then add a reference from Dim_Retailer and sort the retailers accordingly. The fuzzy matching was specified on Levenshtein distance with a maximum of 3. This value was chosen as a cut-off to have no false matches. Another option would be a maximum distance of 4, which would add one more correct match, however, it would also mean including 7 false matches.

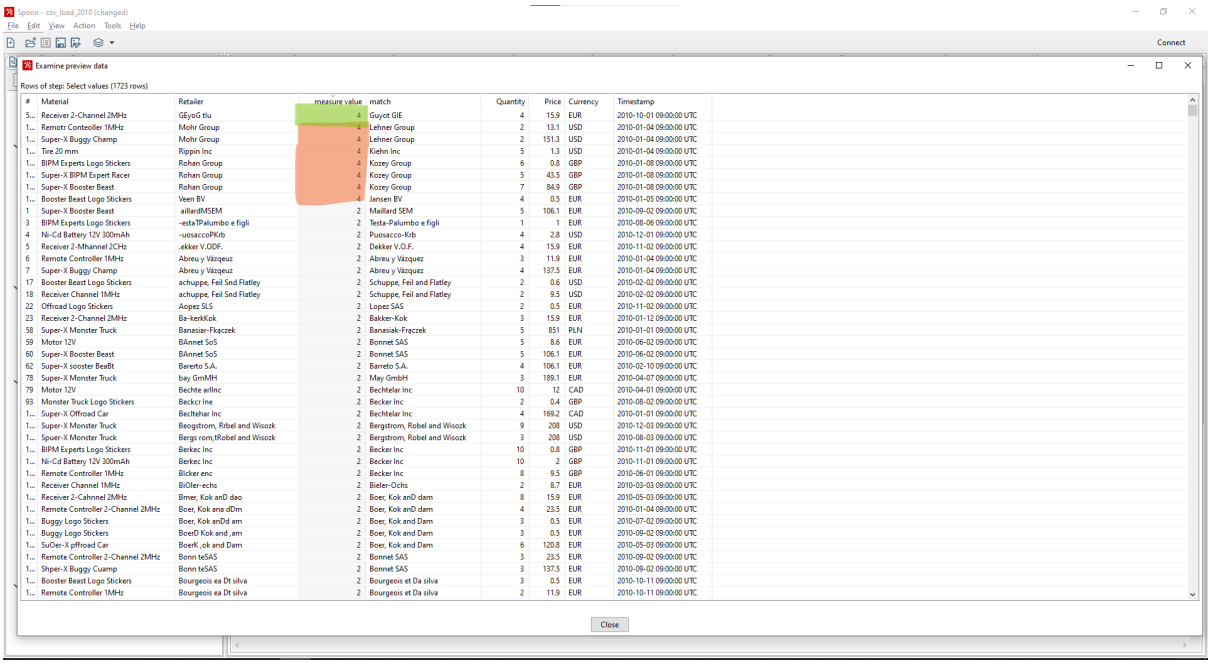
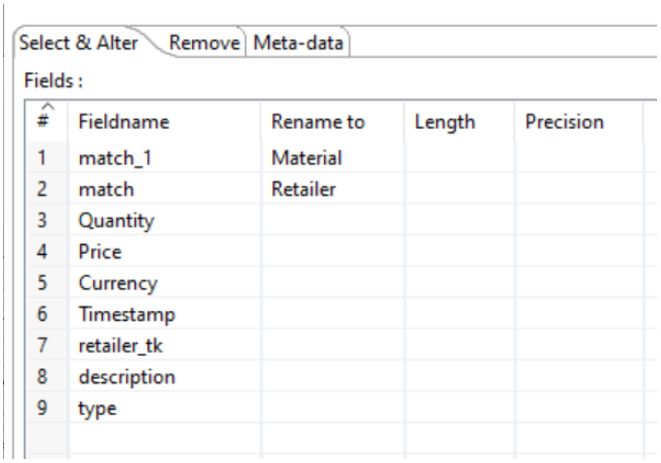


Figure 55: Matching Retailer - Load CSV Files

Next we sort rows by fuzzy matched retailer name and get the other needed retailer info from Dim_Retailer, namely “retailer_id”, “retailer_tk”, “retailer_name”, “retailer_description”, “retailer_category”, “retailer_address” and “retailer_state”. These are also sorted by retailer name and joined on the fuzzy matched names. Afterwards, we filter rows, first only keeping matched rows. The following filters make sure the retailer_tk is not null, and then we repeat the address filter from Dim_Retailer, to delete all wrong retailers. This means, the retailer part of the missing data has been solved.

Next, we need to do the same for material. As such, we fuzzy-match the material names with the materials from the OLTP. Here we again use Levenshtein distance, however, this time setting a max value of 2 will have every material matched. We then select the values we need afterwards, using the fuzzy matched names as new Material and Retailer.



#	Fieldname	Rename to	Length	Precision
1	match_1	Material		
2	match	Retailer		
3	Quantity			
4	Price			
5	Currency			
6	Timestamp			
7	retailer_tk			
8	description			
9	type			

Figure 56: Select Values - Load CSV Files

In the same step, we also change the type of Timestamp from Timestamp to a yyyy-MM-dd date format, and rename the attribute to date. This will be needed to join the table with the currency_conversion table from earlier.

Then, we use a dimension lookup on the Material to get the material_tk from Dim_Material.

For the currency conversion, we sort by Date and Currency, both on the table and the identical currency_conversion input from earlier.

Again, we calculate the price_eur and revenue, as these will be needed in the Fact_Sales table. Next, we create the date_tk from the timestamp with the Java expression.

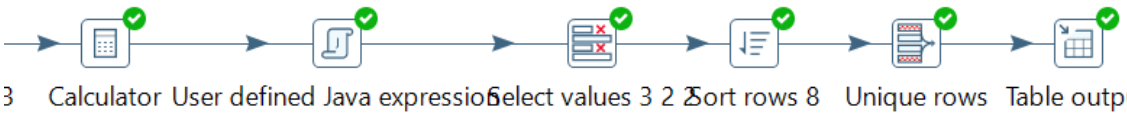
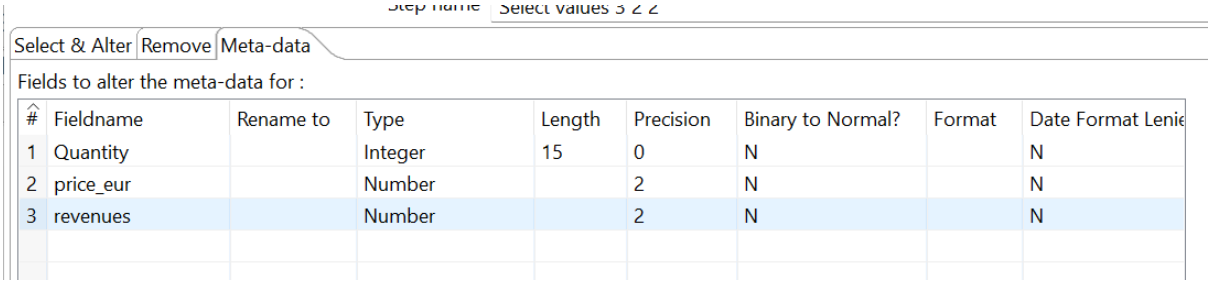


Figure 57: Transformation - Load CSV Files

Concluding, we sort rows for Date and Retailer and then change the type of quantity, price_eur, and revenues in the Meta-Data of the Select Values.



#	Fieldname	Rename to	Type	Length	Precision	Binary to Normal?	Format	Date Format Lenie
1	Quantity		Integer	15	0	N		N
2	price_eur		Number		2	N		N
3	revenues		Number		2	N		N

Figure 58: Meta-Data Editing - Load CSV Files

Next, we sort by retailer_tk, date_tk, material_tk and quantity. This is needed to drop 122 duplicated rows based on these attributes, as this was another data issue identified in the CSV files. Finally, we upload the data into the load_csv table.

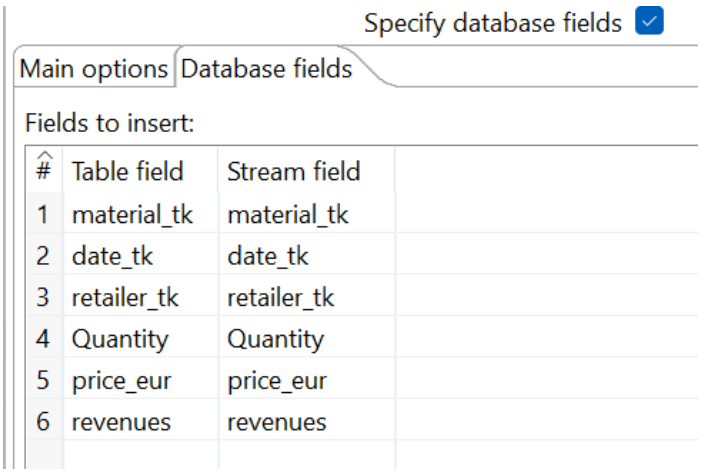


Figure 59: Loading Data into the load_csv Table

In the next transformation, we assign order_id to the CSV orders.

The big question was how to assign the order_id. The two options we considered were starting with order_id numbers that are so high, that they will never be reached, such as 900,000,000. However, this may cause confusion. Instead, we decided to continue where the fact_sales order_id stops. This was done loading the load_csv input while also generating a number of series in the query.

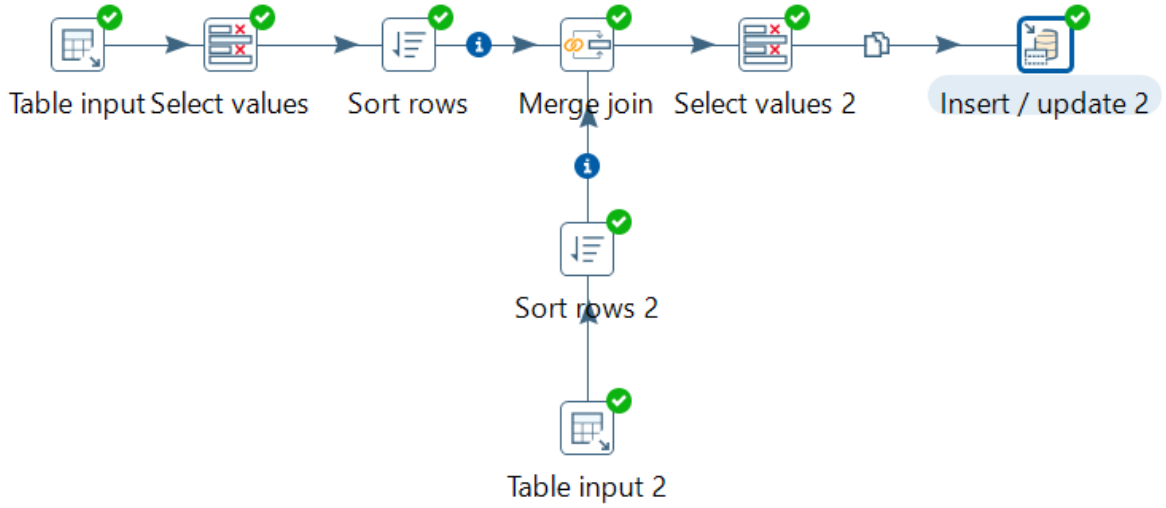


Figure 60: Transformation overview - add order_id

The table input selects one order item (one date_tk, material_tk, retailer_tk combination) from each order, and matches it with a generated series starting with the last order_id known from fact_Sales.

SQL

```
with sub1 as (
    SELECT ROW_NUMBER() OVER (), *
    FROM (select distinct on (date_tk, retailer_tk) *
from load_csv
order by date_tk, retailer_tk, material_tk
)as subsub)
select * from sub1
left join (SELECT ROW_NUMBER() OVER (),*
           FROM (SELECT *
FROM generate_series(1, (SELECT max(order_id + 10000)
from fact_sales))
where generate_series > (SELECT max(order_id)
from fact_sales)) as sub)
as sub2 on sub2.row_number = sub1.row_number
```

Figure 61: Table Input - add order_id

Next, we only keep the date_tk, retailer_tk, material_tk, and order_id, which the generated series is renamed to. To merge this data into the load_csv, we query the complete load_csv table, sort all values by date_tk, retailer_tk, and material_tk and then right join on these attributes. Accordingly, for every order, one row has now an order_id while all the other rows of the order have missing values for now. In hindsight, the missing values could probably have been avoided by joining on retailer_tk and date_tk, not also joining on material_tk.

We then drop the duplicates columns created by the join, keeping only the filled columns, and update this table into load_csv.

The last part of loading the CSV files is to fill the missing order_id rows and deal with missing values in employee_tk and state, which are not allowed in fact_sales.

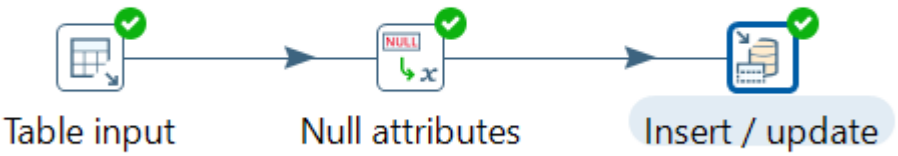


Figure 62: Transformation overview - fill empty order_id

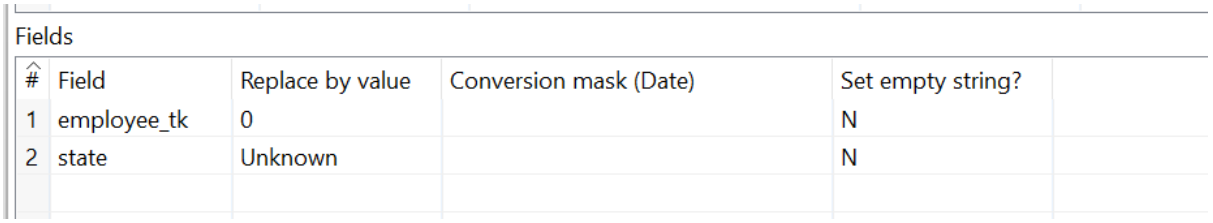
Filling the missing order_id is done in an SQL query, with the same idea as filling the missing currencies in the currency_conversion table.

SQL Get SQL

```
with lc1 as
(SELECT ROW_NUMBER() OVER (), * FROM (select * from load_csv lc order by date_tk, retailer_tk, order_id) as sub1)
select *, first_value(order_id) over (partition by grp) as order_id_filled
from (
      select *,
             sum(case when order_id is not null then 1 end) over (order by row_number) as grp
      from   lc1
     ) as sub
```

Figure 63: Fill Missing order_id - Load CSV Files

The null values in employee_tk are next filled with the employee_tk “0”, indicating the missing information, as employee_tk is not connected to any name. The null values in state are filled with “Unknown” as this is the clearest indicator to this data quality issue.

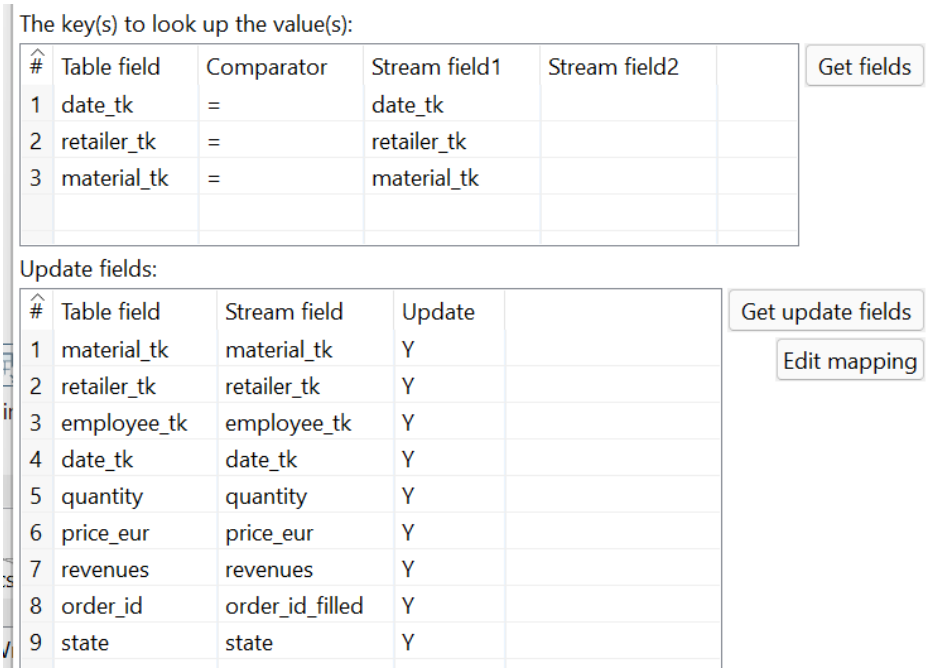


Fields

#	Field	Replace by value	Conversion mask (Date)	Set empty string?
1	employee_tk	0		N
2	state	Unknown		N

Figure 64: Assigning State “Unknown” - Load CSV Files

Then we upload the new table into fact_sales, matching orders on date_tk, retailer_tk, and material_tk. This allows us to find orders in the CSV files which were already existing in the OLTP and update them instead, which was the case for 87 orders.



The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	date_tk	=	date_tk	
2	retailer_tk	=	retailer_tk	
3	material_tk	=	material_tk	

Get fields

Update fields:

#	Table field	Stream field	Update
1	material_tk	material_tk	Y
2	retailer_tk	retailer_tk	Y
3	employee_tk	employee_tk	Y
4	date_tk	date_tk	Y
5	quantity	quantity	Y
6	price_eur	price_eur	Y
7	revenues	revenues	Y
8	order_id	order_id_filled	Y
9	state	state	Y

Get update fields

Edit mapping

Figure 65: Insert / Update - Load CSV Files

5.4 Change Data Capturing (CDC)

For our data mart, we decided on the CDC method of Queryable Change Data, where change data comes from a data source via a query. Although we do have timestamps for our sales order, it is possible that there are multiple orders on the same day and the hour is not recorded, therefore we used order_id as our so-called “timestamp”.

Since we want to detect the change based on the order_id, we need to retrieve the orders that have not been recorded in our data mart, whose order_id is larger than the order with the largest order_id number. First, we created a transformation in Pentaho to set the variable max_factsales_order_id. The transformation starts with a Table Input where we select the maximum order_id and connect it to “Set Variables” to set up the variable name.

Figure 66: Transformation to Set max_factsales_order_id Variable

The next step is to change the Table Input in our transformation to load fact_sales. Here, we only need to capture the new orders that have not been loaded yet. We added a conditional SQL statement in the Table Input query and ticked the “Replace variables in script” which in the end looks like this:

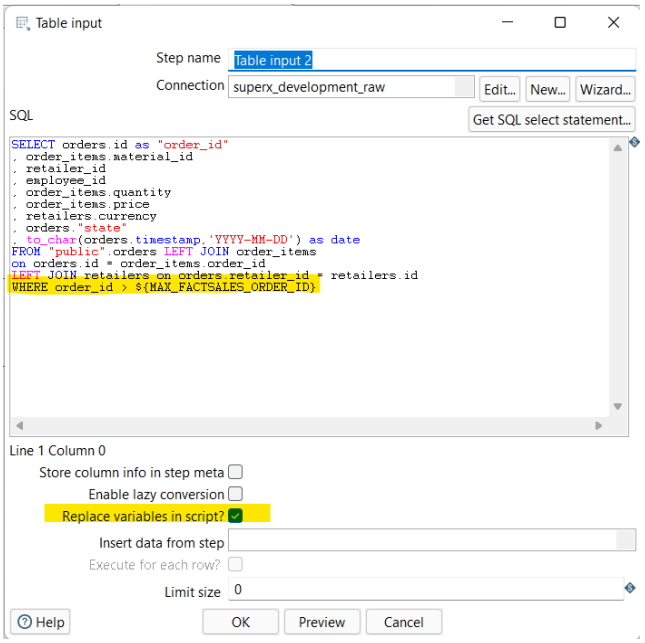


Figure 67: Table Input - fact_sales_cdc

As the last step, we created a job to carry out both transformations at the same time: first, the tr_cdc_set_var to set up max_factsales_order_id, connect it to an SQL query where we

delete any data whose order_id is larger than the maximum ID and finally connect it to the transformation to load fact_sales.

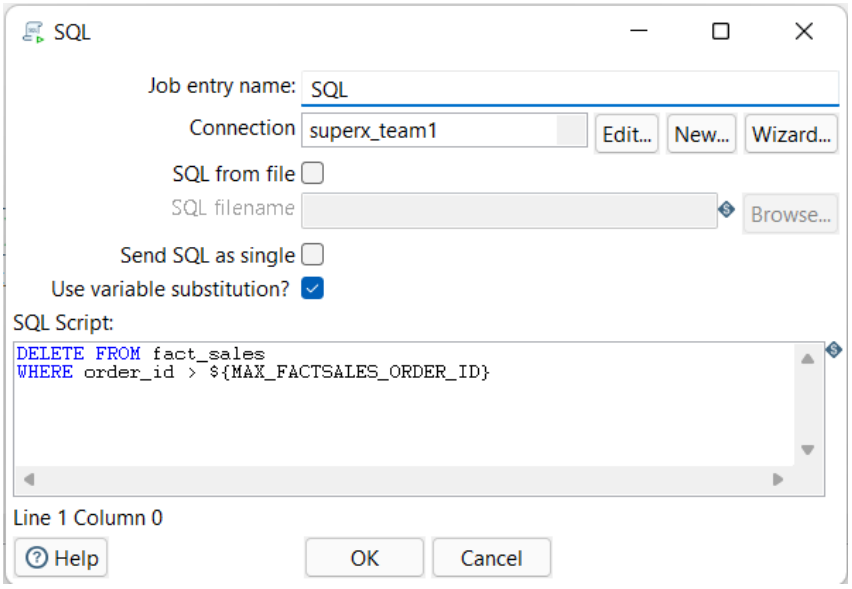


Figure 68: Filter to Get New Orders - CDC

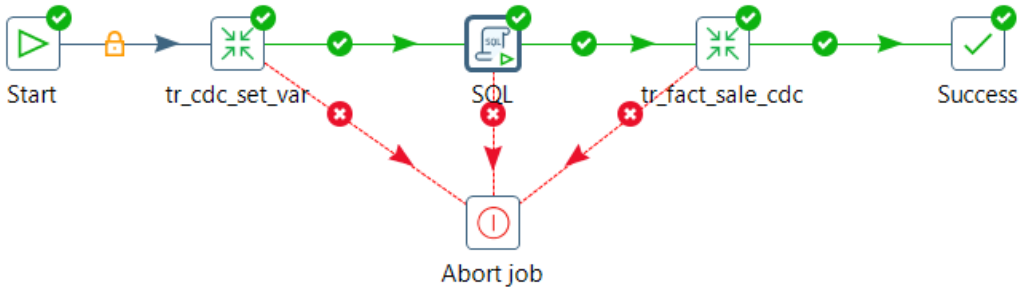


Figure 69: Job for CDC

6. TABLEAU DASHBOARD

We wanted to be able to answer all of our high-importance Business Requirements with data visualization in Tableau and, at the end, to have a dashboard that would give us the most comprehensive and holistic overview of our Sales data. The dashboard should also be flexible enough by having useful filters for us to keep track by year, by month or by retailer.

Business Requirement #1: What is the average monthly net sales?
and **Business Requirement #3:** What does our sales development look like throughout the months?
For all diagrams using the measure “Revenues” from the data mart, we decided to take only ***orders that are not cancelled*** into considerations, which means Net Sales.

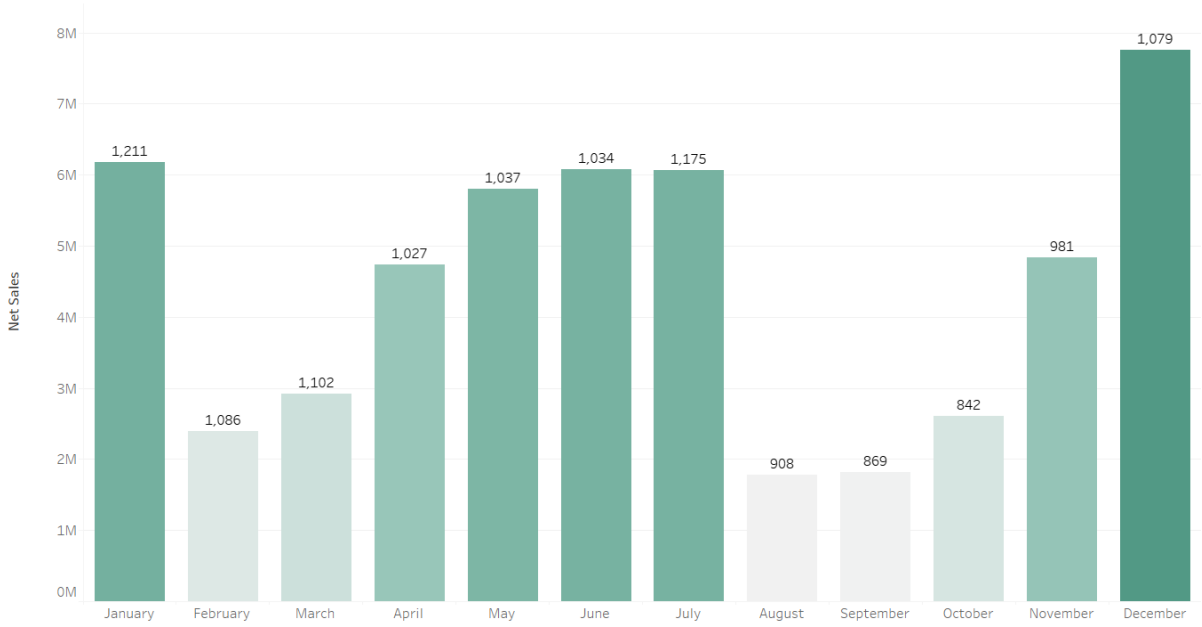


Figure 70: Sales Development by Calendar Months (with Total Number of Orders)

The chart shows the net sales by the 12 months in a calendar year, with the total number of orders on top of each bar. The net sales range from 1,7M to 7,7M €. Looking at the bar, we can see that December is the most revenue-generating month for our business. Although the number of orders might not be as many as other months of the year, we manage to sell more finished products that cost more. The reason for this might be December being the holiday season that would have a positive impact on toy purchase.

The following second graph shows the average net sales based on the months of the date throughout the years. The movement, as can be seen, is nowhere near stable as our net sales varied tremendously from month to month. The highest average monthly net sales recorded was in December 2015 (1493 €).

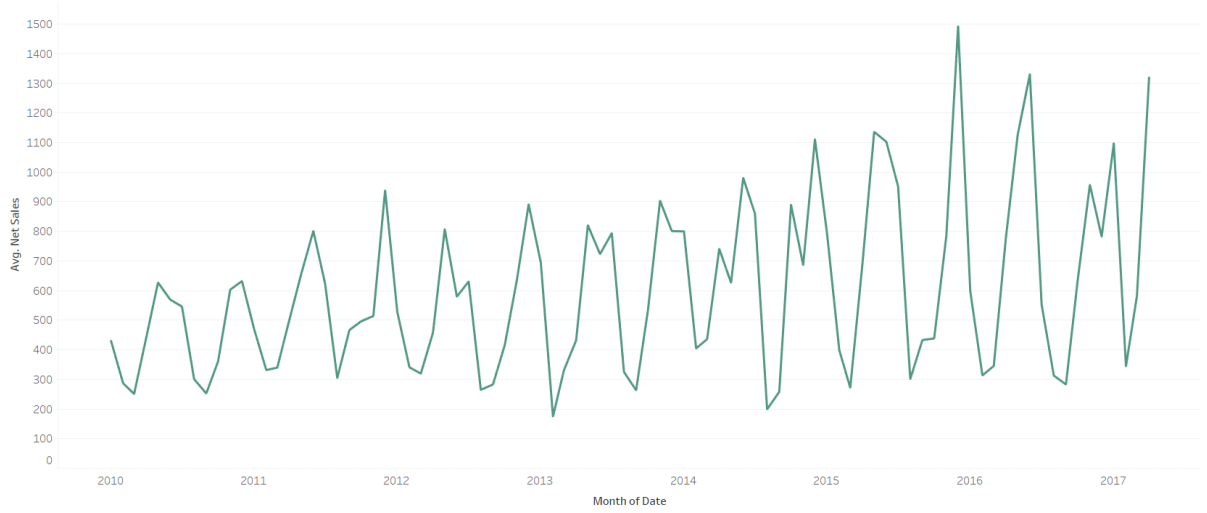


Figure 71: Average Net Sales

Here is an overview of the revenue movement by year. Since the database only has orders until April 2017 available, therefore there are much fewer revenues shown on the chart. If considering ***all types of orde***r (which means including cancelled orders), we can see that Super X has experienced steady growth of business from 2010-2016, with revenues in 2016 almost doubled the amount in 2010.

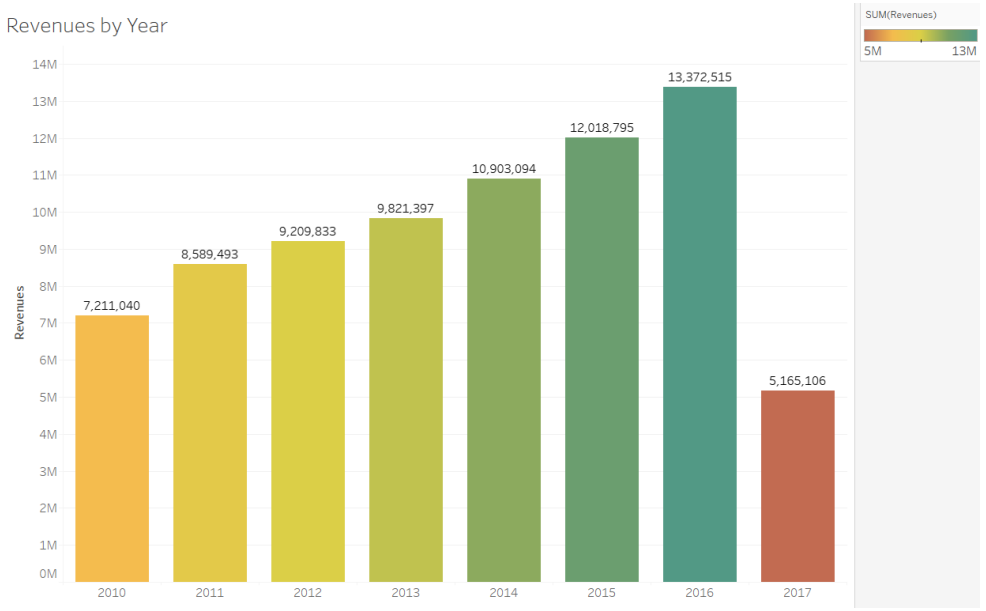


Figure 72: Sales Revenues by Year (all state)

However, when we filtered out cancelled orders and only look at ***net sales***, we observed that Super X actually did not earn much from the increase of number of orders throughout the years since the cancellation rate was still very high. There was a significant net sales income in 2015, but the amount went back to around 7M in 2016.

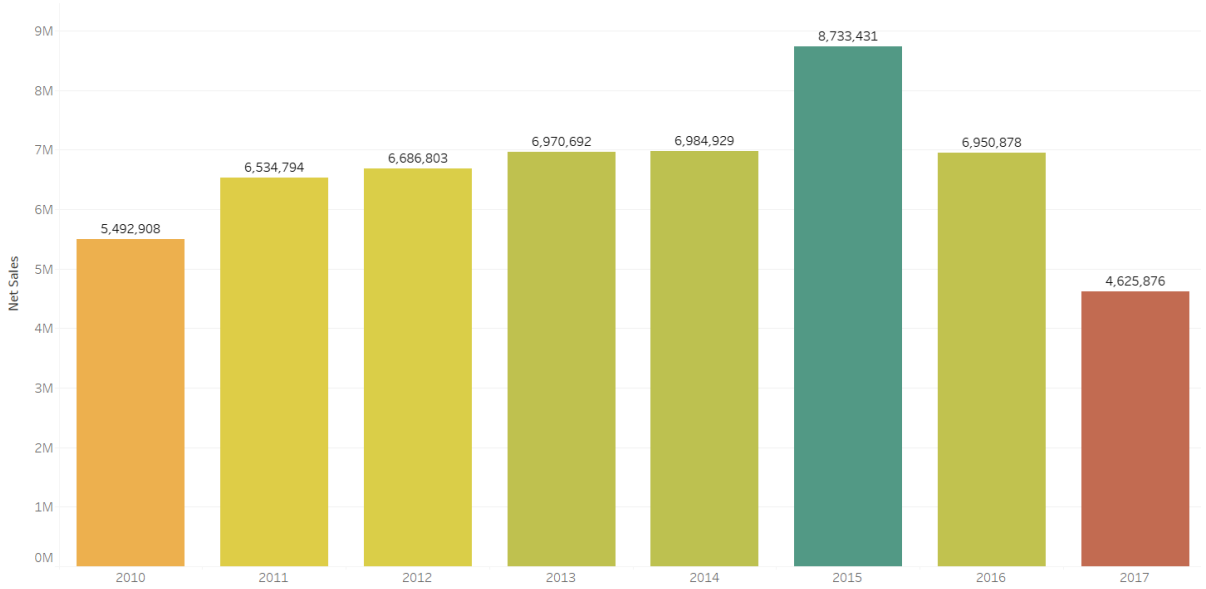


Figure 73: Net Sales by Year (not canceled)

Business Requirement #2: Which retailers have the highest/lowest monthly net sales?

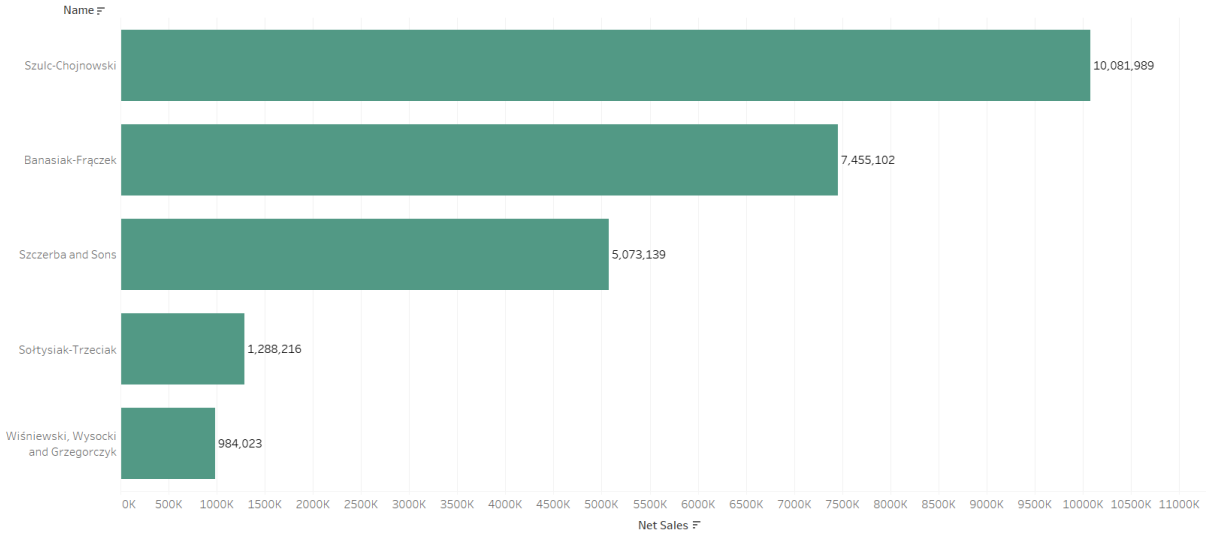


Figure 74: Top 5 Retailers Based on Net Sales

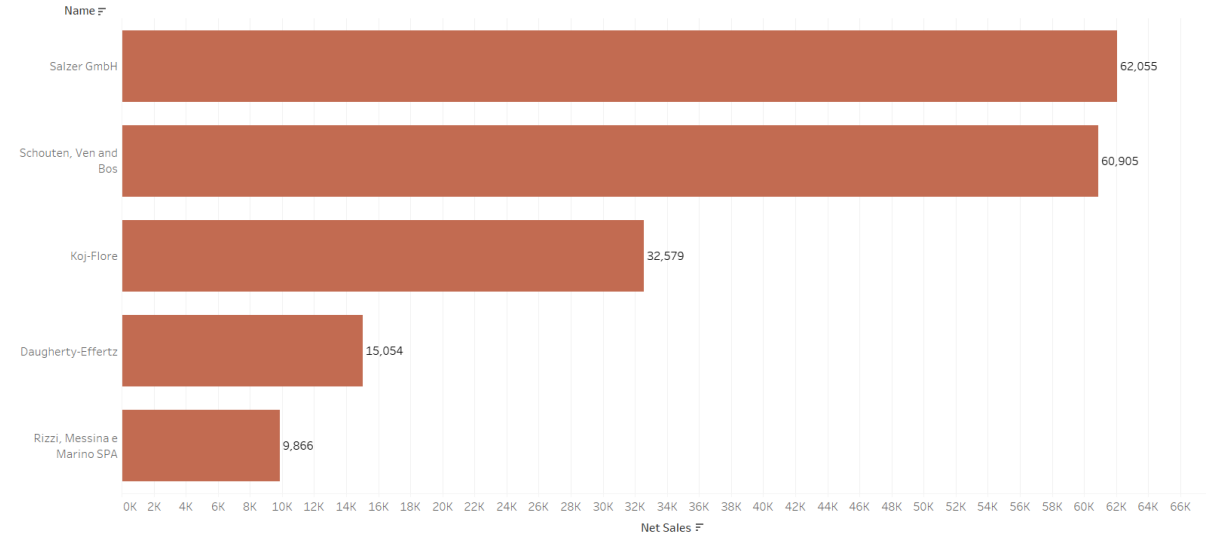


Figure 75: Bottom 4 Retailers Based on Net Sales

As the first picture shows, the top 5 retailers based on their total net sales throughout all the years (2010-2017) are Szulc-Chojnowski, Banasiak-Fraczek, Szczerba and Sons, Soltysiak-Trzekciak and Wisniewski, Wysocki and Grzegorczyk. While the retailers with the least net sales amount are Salzer GmbH; Schouten, Ven and Bos; Koj-Flore; Daugherty-Effertz, and Rizzi, Messina e Marino SPA. The highest total net sales from our retailer is 10M €, while the lowest is only approx. 10K € which is a big difference.

Furthermore, we created a **Retailer Dashboard** where we can pick out one specific retailer to look into their location and channel as well as the net sales, number of orders, most ordered materials which can be filtered by a specific year and month.

For instance, when looking at our top 1 retailer “Szulc-Chojnowski”, our dashboard tells us that they’re an offline retailer in Poland and in 2013 as an example, their total net sales were 1,3M € with 21 orders. We can also see how the net sales and the number of orders are distributed throughout the months. They mostly ordered Super-X Monster Truck, Super-X Buggy Champ and Super-X Booster Beast.

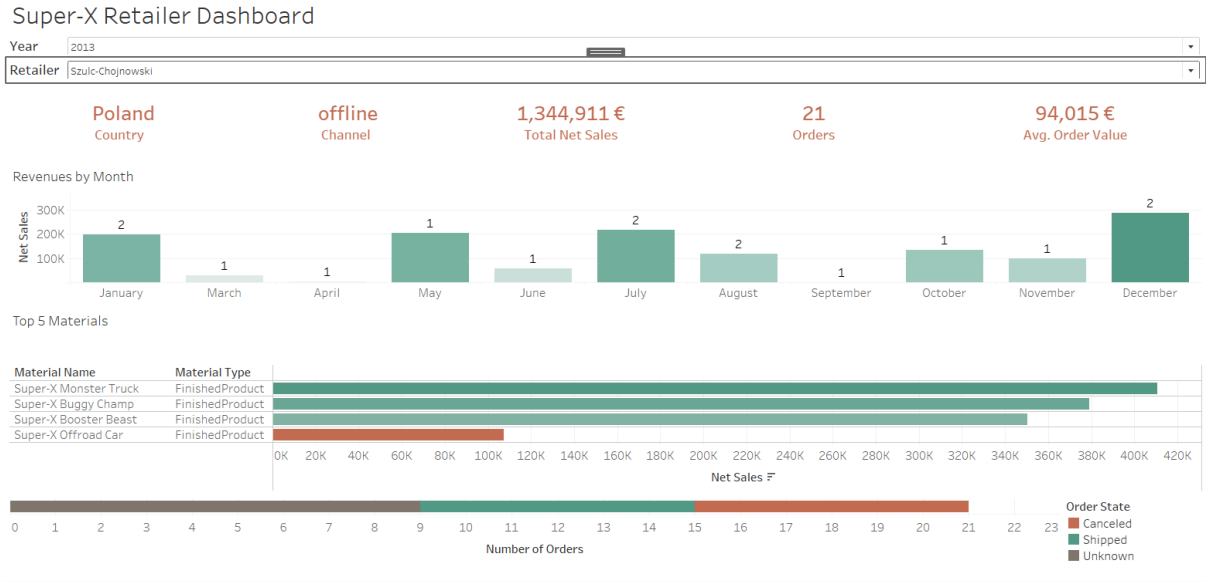


Figure 76: Retailer Dashboard

Business Requirement #4: How did the monthly number of retailers change over the last 12 months?

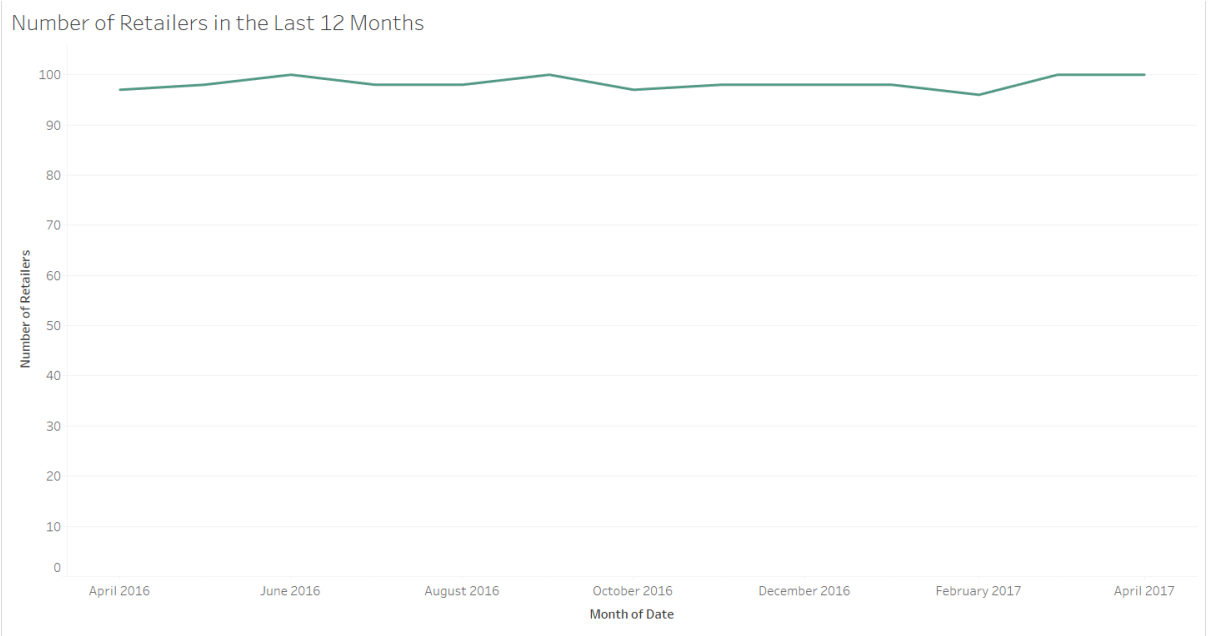


Figure 77: Number of Retailers in the Last 12 Months

Looking at the line graph, there is not much of a drastic change in the number of retailers during the last 12 months. We have a maximum of 100 retailers and the number did not vary much. The fewest number of retailers recorded was 96 in February 2017. We can conclude that the number of retailers is quite stable.

Business Requirement #5: Which 3 retailers have the most/the least cross-selling rate?

With this requirement, we want to find out which retailers are ordering the most different types of material from us, as well as the opposite. We could not calculate the exact rate of cross-selling with Tableau, but we managed to filter out the top 3 that have purchased the most number of different types of materials and the least number.
Our top 3 ordered around 15 different types of material monthly, while our bottom 3 order just less than 10. Retailer “Abreu y Vazquez” ordered as few as 5 types of material in August 2016.

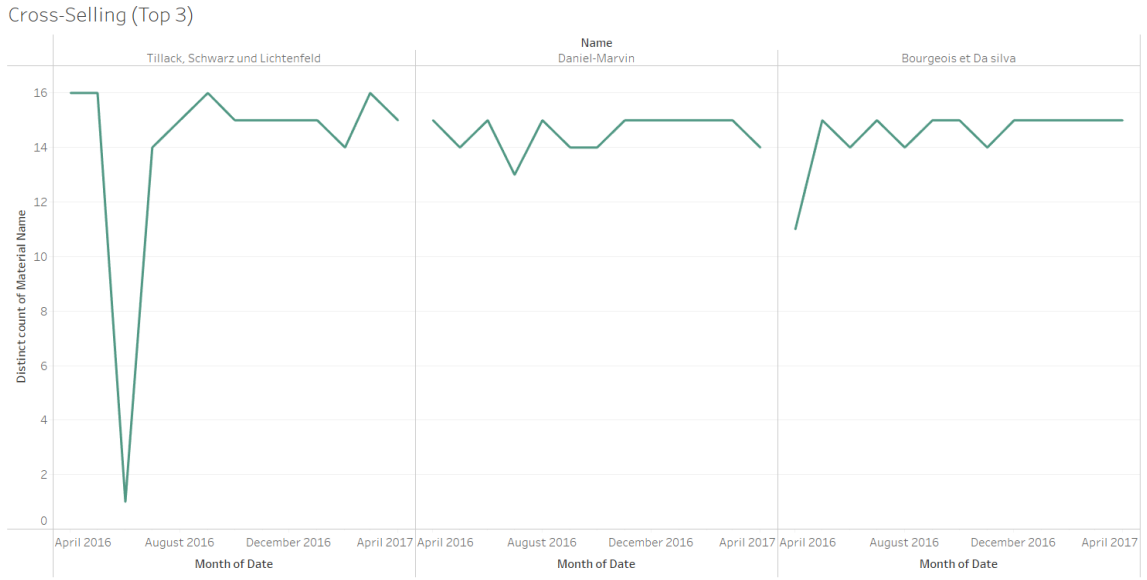


Figure 78: Cross-Selling Rate - Top 3 Retailers

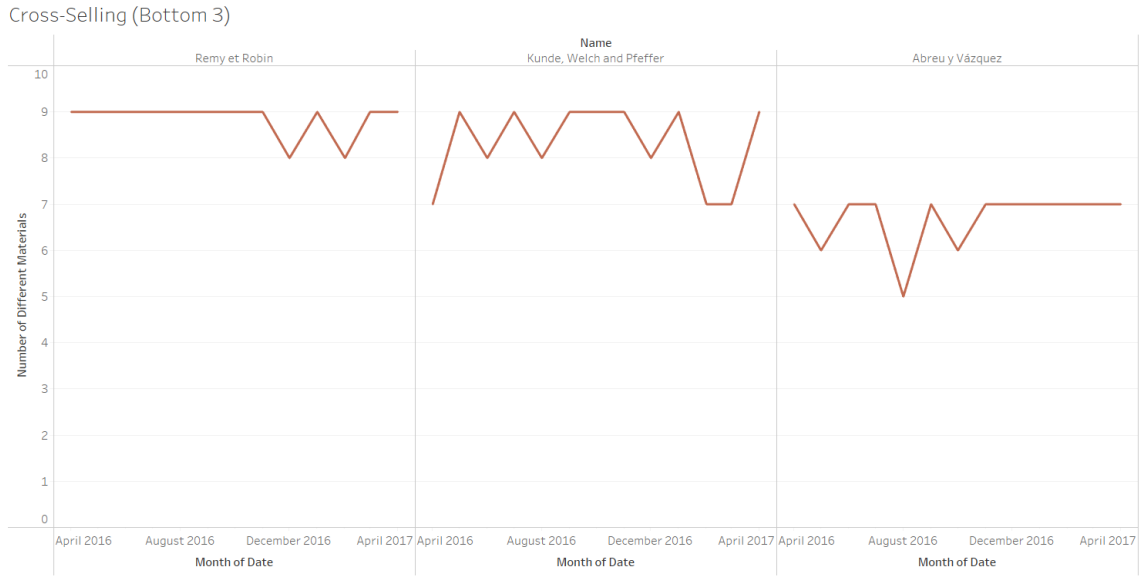


Figure 79: Cross-Selling Rate - Bottom 3 Retailers

Business Requirement #6: How did the monthly number of orders change over the last 12 months?

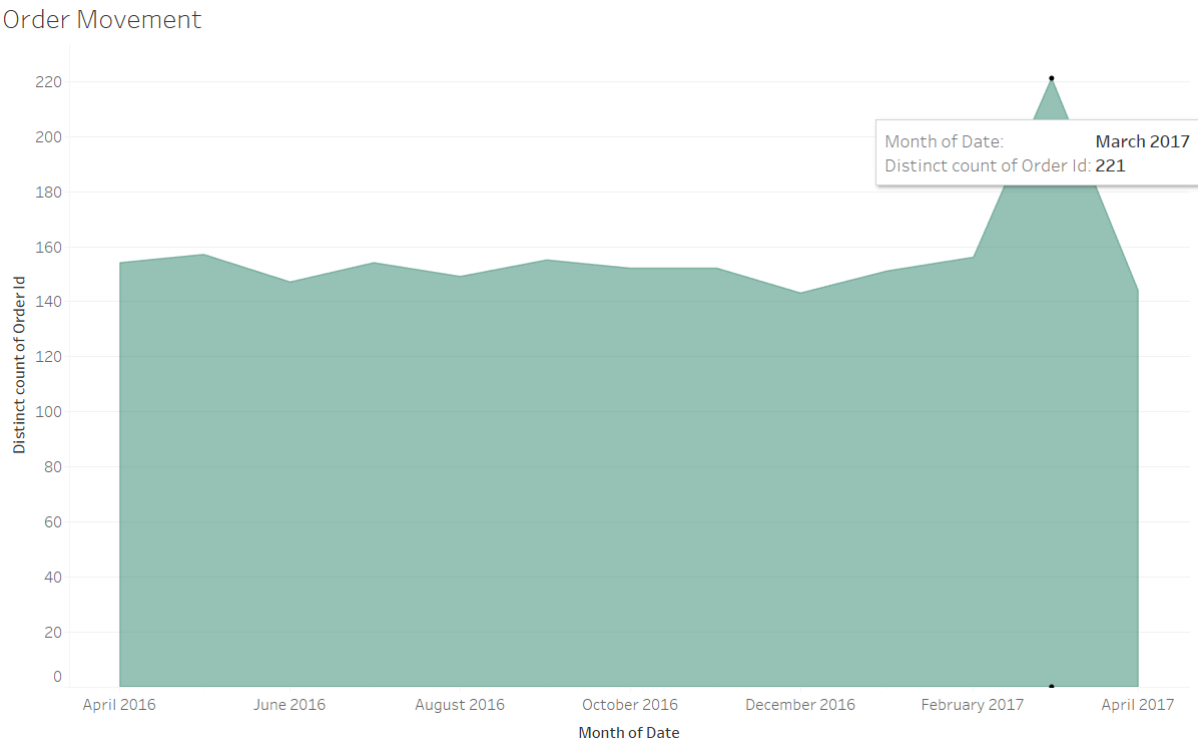


Figure 80: Number of Orders in the Last 12 Months

In the last 12 months, our number of orders stayed relatively constant around 150 orders/month, although we did observe a surge in orders in March 2017 (a total of 221 orders).

Business Requirement #7: What is our fulfillment rate?

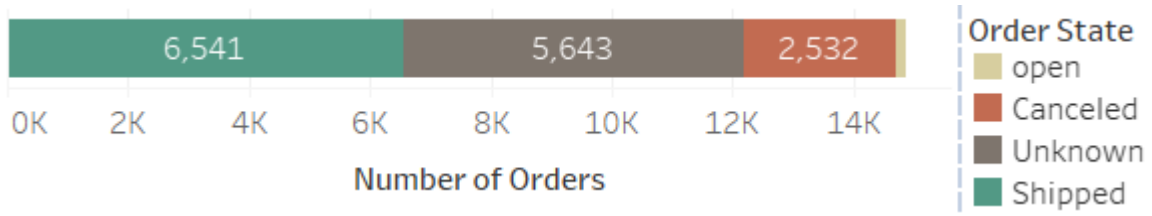


Figure 81: Number of Orders by Order State

Overall, 6541 orders were successfully shipped by us, 2532 got cancelled, 167 orders were open, and 5643 orders are unknown, which are orders from the additional CSV files and were not assigned with any particular status. So our fulfillment rate = 6541 / (6541+2532) = 72,09%.

Business Requirement #8: How much in percentage does our forecast match with the actual orders from the retailers?

Our overall forecast fulfillment is 6.28%, which is significantly low.

When calculating the forecast fulfillment rate by retailer, the highest rate we achieved is 14,44% by retailer “Kunde, Welch und Pfeffer”, meaning we managed to ship only 14% of the quantity we forecasted. The lowest forecast fulfillment rate is as low as nearly 2,5%. Looking at the numbers, we can confidently say that the forecasts are not correctly reflecting the demand in real life and would require a more precise calculation from Super-X.

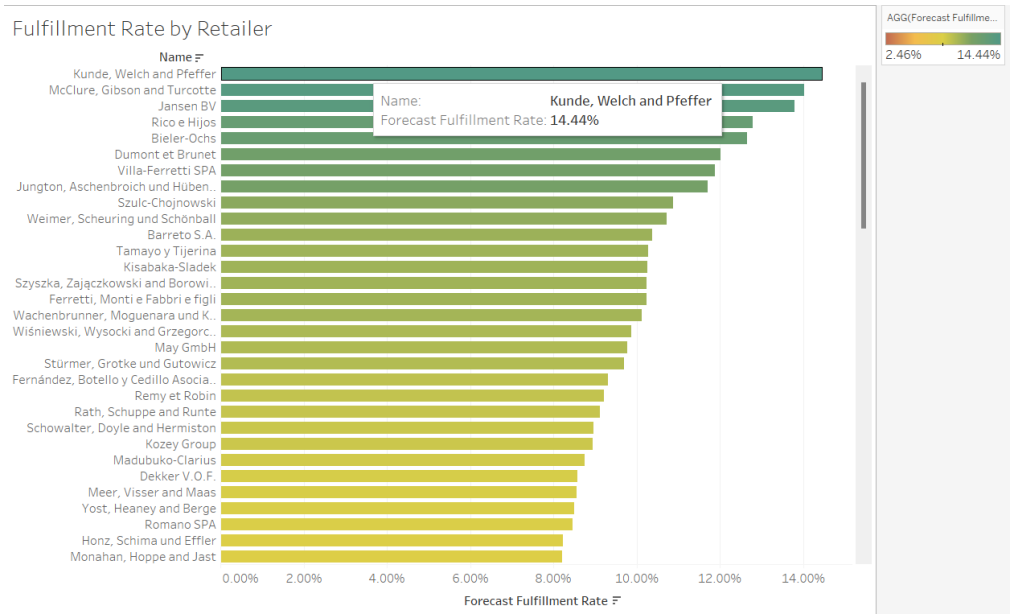


Figure 82: Forecast Fulfillment Rate 1

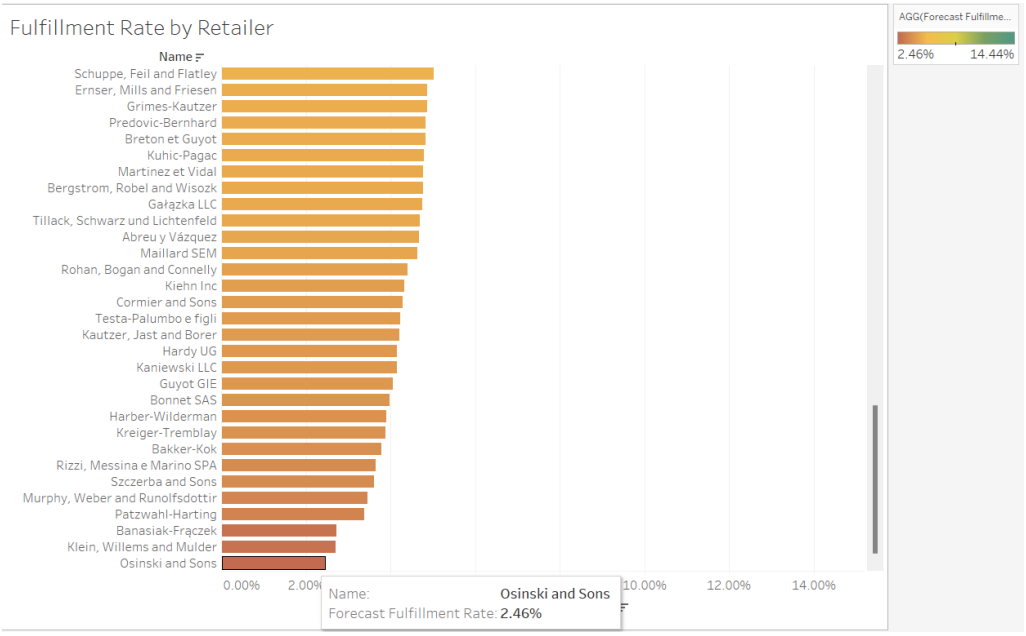


Figure 83: Forecast Fulfillment Rate 2

Business Requirement #9: Which Sales employee generates the most net sales?

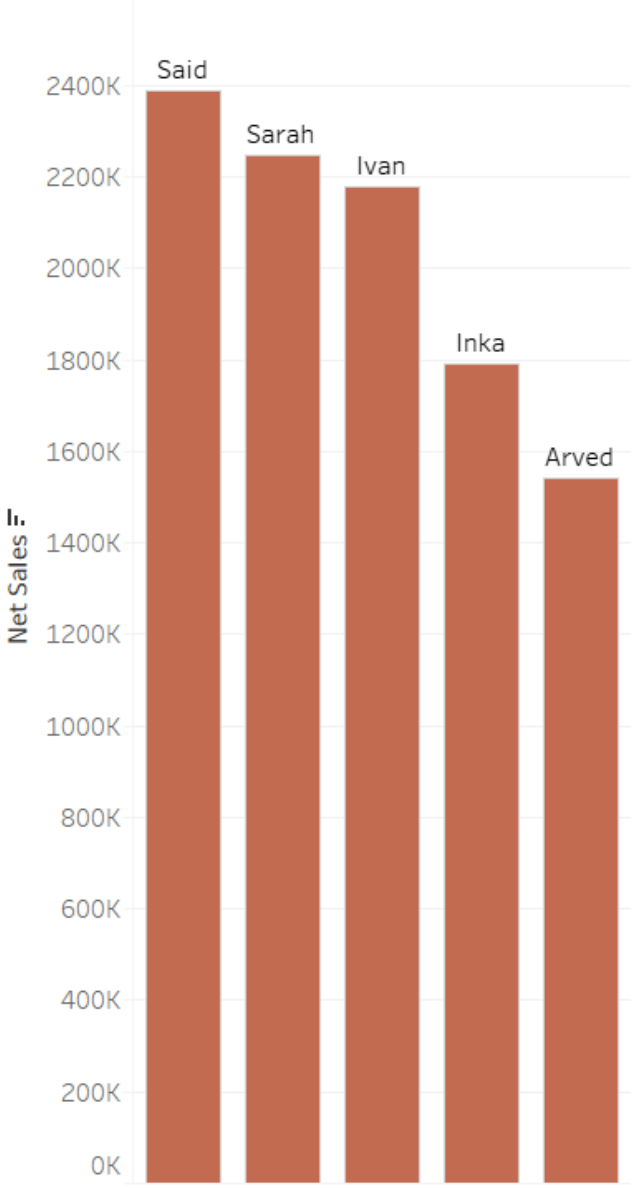


Figure 84: Top 5 Sales Employees Based on Net Sales

Our top 5 most revenue-generating Sales employees are:

1. Said
2. Sarah
3. Ivan
4. Inka
5. Arved

with Said, Sarah and Ivan generated more than 2M € in total.

This KPI is important for Super-X to see the individual contribution from its employees and to have rewards as well as incentives to keep the productivity flowing.

Business Requirement #10: What is the most demanded material (based on net sales) by our retailers?

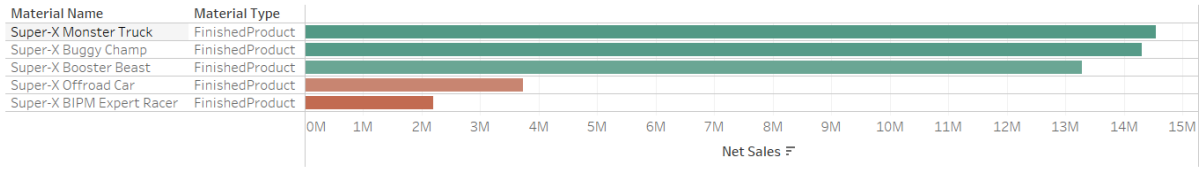


Figure 85: Top 4 Materials Based on Net Sales

We decided to choose net sales as a measure to decide the top 5 because our material range is a mix of finished product and production material part, so judging based on ordered quantity might not be the most optimal solution as their profitable value is vastly different.
Based on total net sales, the most demanded material are:

1. Super-X Monster Truck (14,5M €)
2. Super-X Buggy Champ (14,3M €)
3. Super-X Booster Beast (13,3M €)
4. Super-X Offroad Car (3,7M €)
5. Super-X BIPM Expert Racer (2,2M €)

All the materials are finished products, and they generated net sales ranging from 2,1M to 14,5M €, although we can see a large gap between the top 3 and top 4 (almost 4 times less).

Net Sales by Country

Our retailers are from 9 countries in total: 2 from North America and 7 from Europe.

We mapped out the country with their total net sales and looking at the visualization, it can be said that retailers from Poland have considerably more net sales than any other country. The total net sales from Poland are 32,3M € while the top 2, which is Germany, has significantly less, namely only roughly one-tenth.

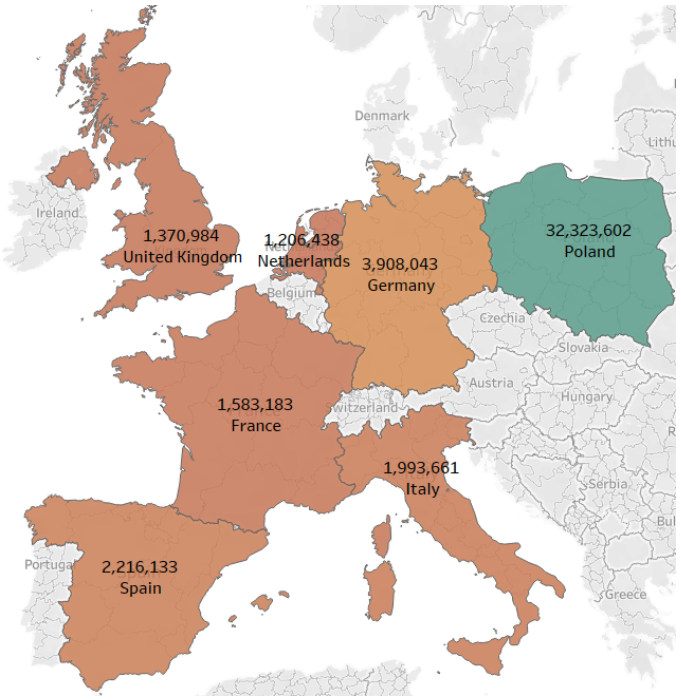


Figure 86: Net Sales by Country (Europe)

Figure 87: Net Sales by Country (North America)

The Sales Dashboard

The aim of our Sales dashboard is to have the most important KPIs packed in one view to help us make observations and, therefore, improvements or changes in the future.

The information that are included on the dashboard are:

1. Number of Orders
2. Average Order Value
3. Total Net Sales
4. Forecast Fulfillment Rate
5. Sales Movement
6. Shipped vs. Cancelled Orders
7. Top 5 Retailers
8. Top 5 Sales Employees
9. Top 5 Materials
10. Net Sales by Channel (online/offline)

All these KPIs can be filtered by year (with a drop-down menu) and further by month (by interacting with one of the charts).

The attached picture shows an example of the KPIs for June 2014.

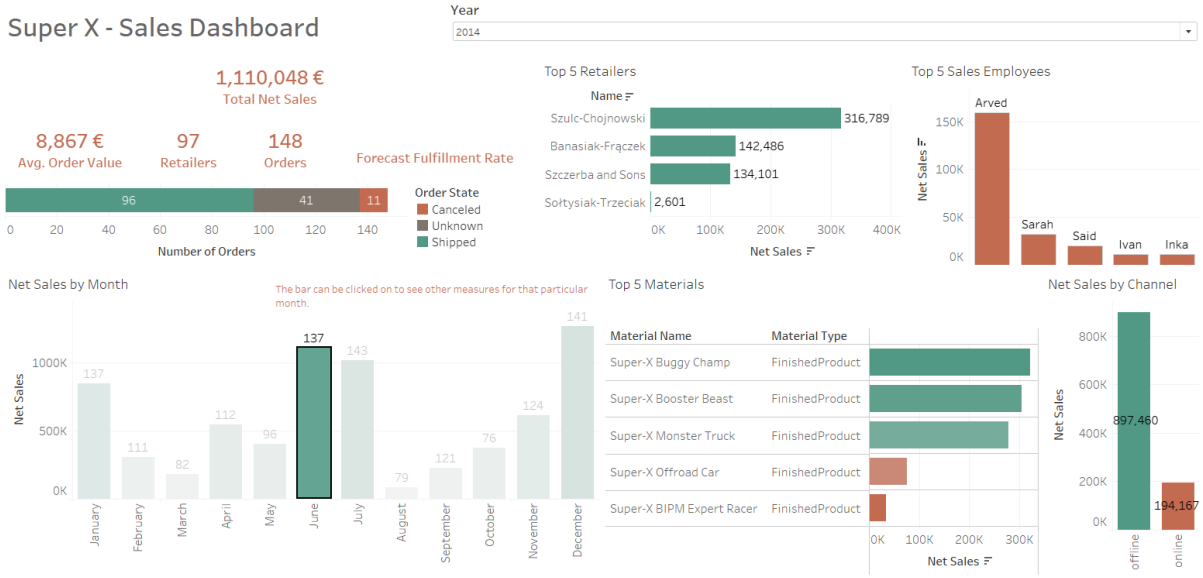


Figure 88: Sales Dashboard

7. PROCESS MINING

To perform Process Mining, the column *Month* was selected as ID, as the company processes are executed monthly and there is no specific ID available for each process.

7.1 General Process Analysis

General Process Overview 2010-2017, with 25% detail on activities and paths. We observe painted in strong red that there are some activities that require a higher time investment and that they act as a bottleneck on the process. For example, *Checking reserved quotas*, *Undertaking financial calculations* or *Calculating employee hours needed for Production*.

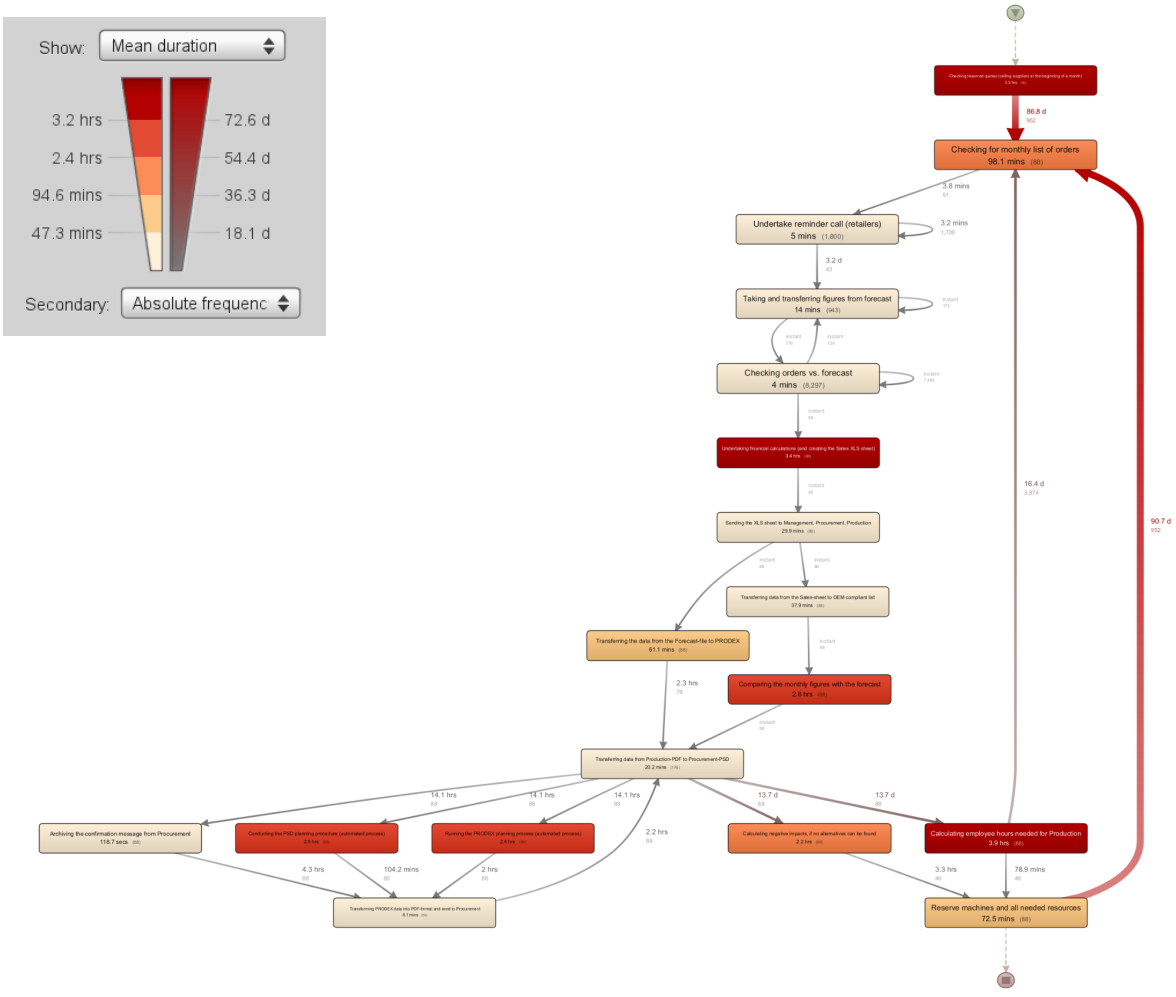


Figure 89. General Process Analysis[2]

[2] https://drive.google.com/file/d/17ozwbhvZskhg7QtzzNlaa3Dd6CVfWC-I/view

Mean Activity Duration

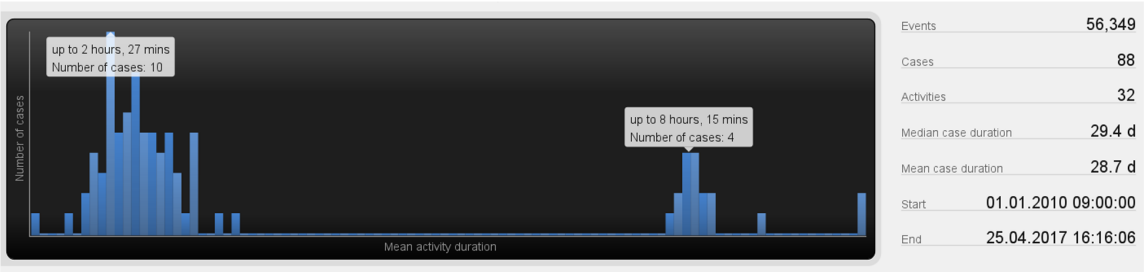


Figure 90: Mean Activity Duration

There are clearly two different kinds of activity groups depending on their duration. The first group is located in the left of the graph, and they have a short duration, and the second group is smaller, but its activities have a duration much larger than the first group.

General Process Overview for 2016-17, with 50% detail on activities and paths. The bottlenecks persist. With the increase in activities and paths, new processes have appeared, and we can see some extra steps that are sometimes taken, like *Undertake reminder call to retailers*, which is a non-value adding activity that has to be executed due to inefficiency of the process and that could be improved. Another extra step that appears is *Taking and transferring figures from forecast*, with a duration of 14 minutes, which could probably be improved by uniformizing the way of calculating and presenting the figures of the forecast, so no transformation is necessary anymore.

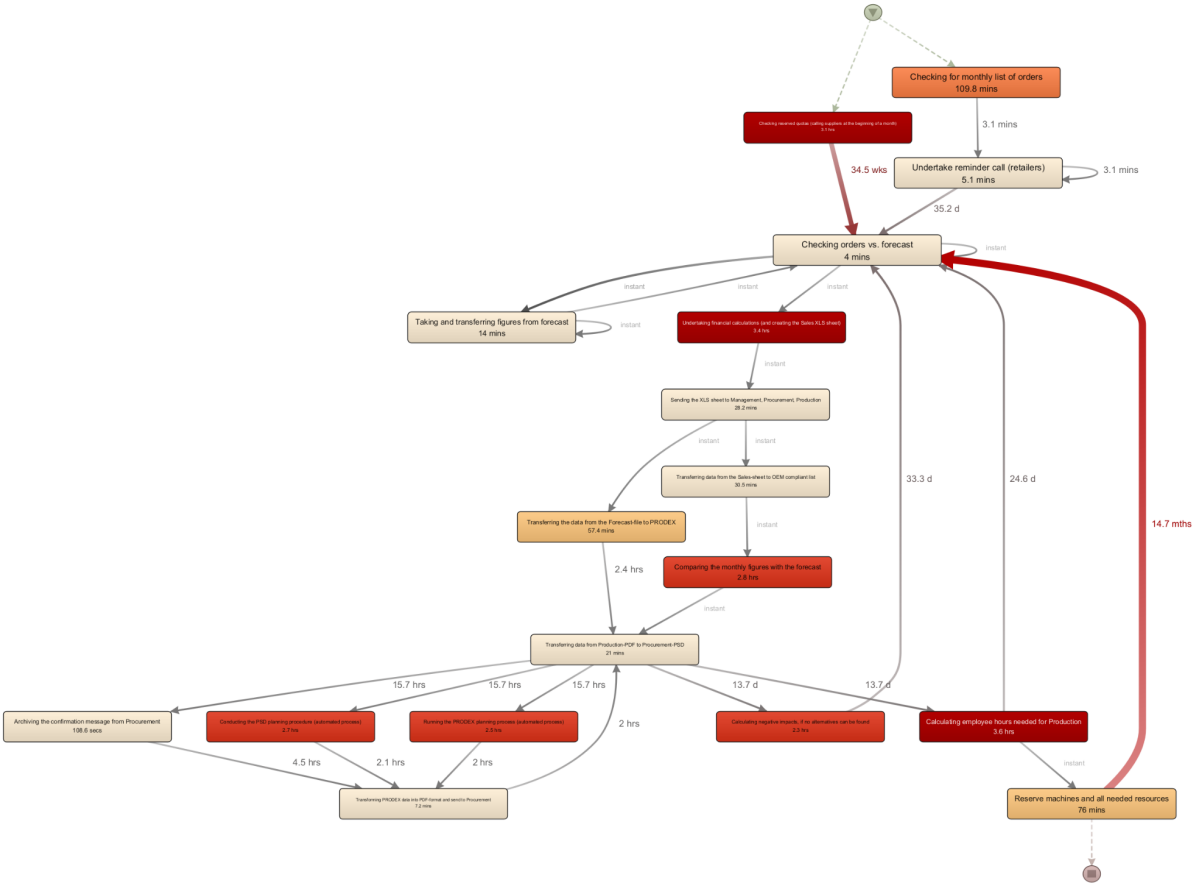


Figure 91: General Process Overview 2016-17 at 50% activities and paths.[3]

[3] https://drive.google.com/file/d/1LoAcgkMAzRD4M5up7hj4daS4bkicy3ZG/view?usp=sharing

7.2 Sales Department Analysis

Sales Process Overview for 2016-17, with 75% detail on activities and paths. The reminder call to retailers mentioned before persists on sales, as it's the responsible department to do it. The absolute frequency is high, which means that several calls are done for each process.

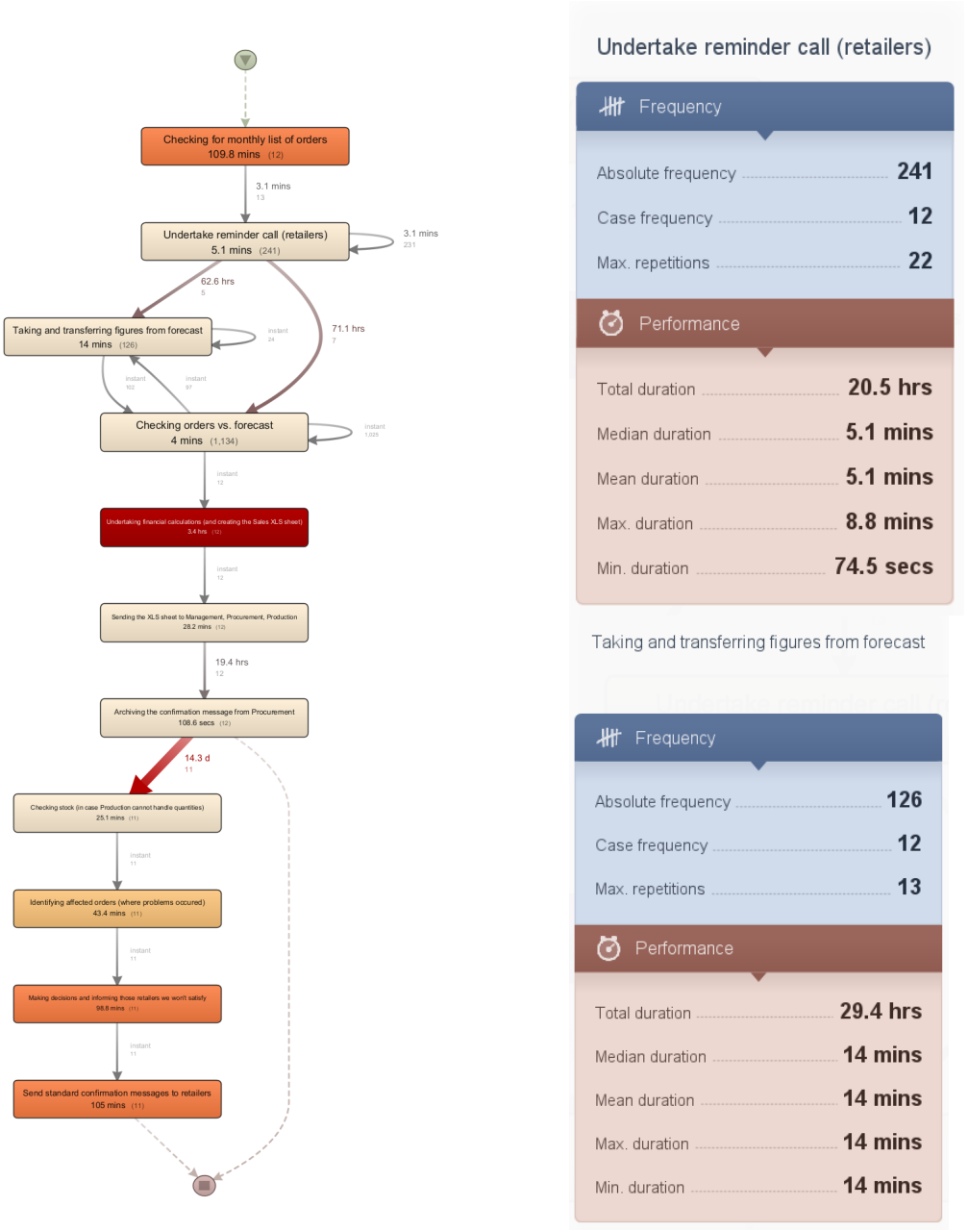


Figure 92: Sales Process Overview 2016-17, 75% activities and paths. [4]

[4] https://drive.google.com/file/d/15uA1wR0ISDbWWBJ5U_rqb6RhMBUEXwbo/view?usp=sharing

The *Taking and transferring figures from forecast* activity is still present in Sales. It's a necessary step, but it could probably be improved if the *Taking and Transfer* were not necessary and the figures were already available and ready in the appropriate format for the Sales department. This could be done by uniformizing the data format for all the processes.

The biggest bottleneck in the Sales department is the activity *Undertaking financial calculations (and creating the Sales XLS sheet)*.

At the end of the process, we have some extra steps if the Production department can't handle the required quantities. For that to happen, the Sales department has to wait 14.3 days, which is the longest path duration of the department.

Figure 93: Detailed overview of problematic processes

Performance analysis. The minimum duration of the sales department processes is 25 days and 8 hours, and the maximum is 30 days and 11 hours. 55% of the cases take longer than 28 days and 10 hours. These cases have always the same pattern: Production cannot handle quantities and Sales has to check for stock, identify affected orders, make decisions, inform retailers who they won't satisfy and send confirmation to these retailers.

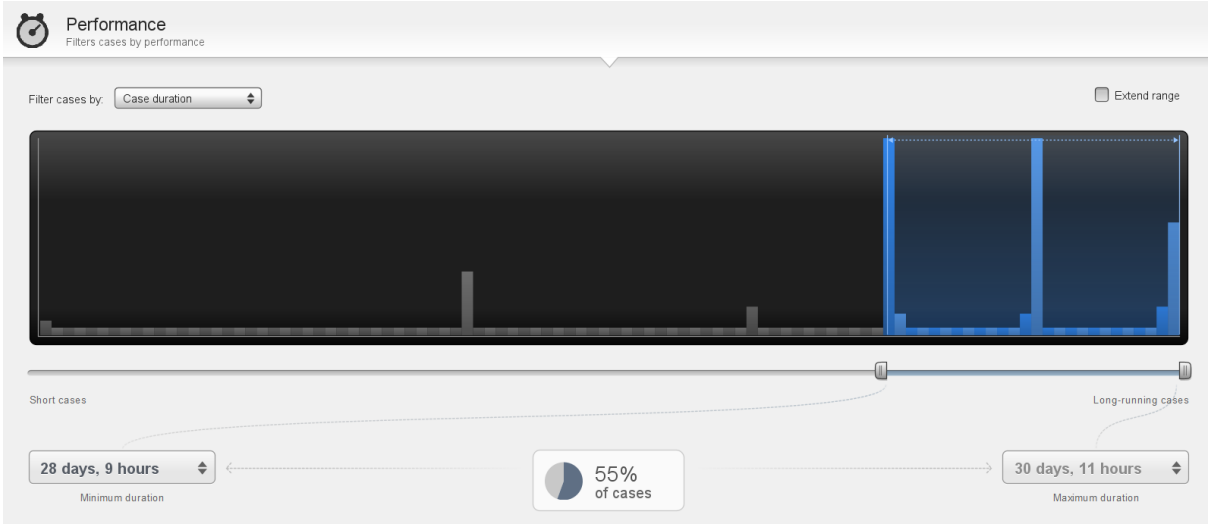


Figure 94: Performance Analysis for Sales during 2016-17

The low-valued register on the beginning of the chart is from January 2017, when the process took part also in February. As we separate processes by month, this is not taken into account in the performance of January, and it makes it have a better time, but it's a misinterpretation.

Mean Activity Duration analysis for Sales during 2016-17. The mean activity duration goes from 46min 48sec to 61min.

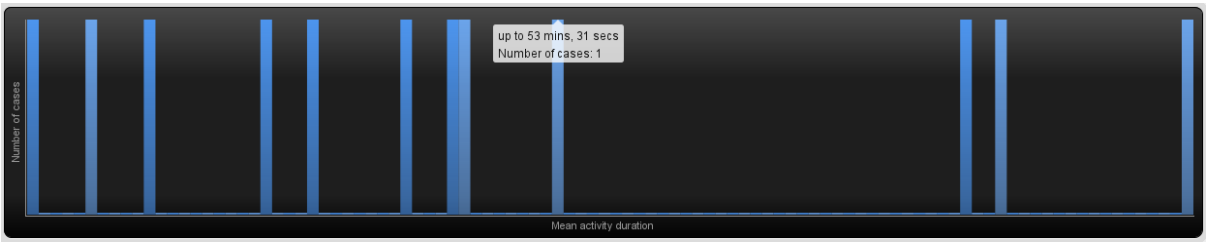


Figure 95: Mean Activity Duration for Sales during 2016-17

Mean Waiting Time analysis for Sales during 2016-17. The Mean Waiting Time has a range between 16 and 21 hours. The low-valued register on the beginning of the chart is just because the last month in the database is not finished, so the total waiting time is not yet complete.

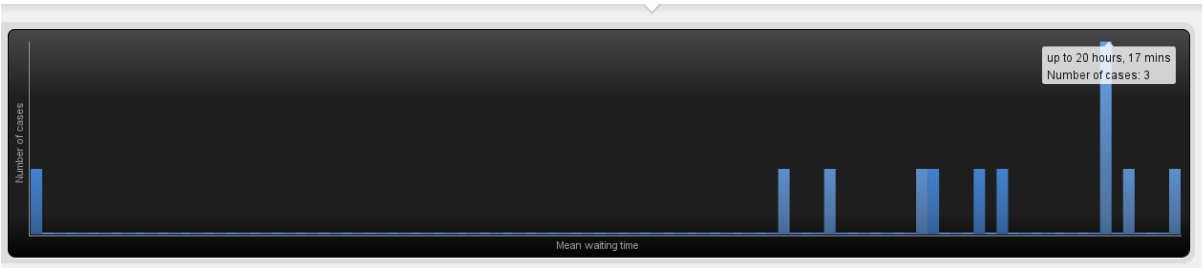


Figure 96: Mean Waiting Time for Sales during 2016-17

Activity analysis for Sales during 2016-17. The most frequent activity is *Checking orders vs forecast*. It's the most repeated one by far, having a relative frequency of 71,14. This is equilibrated by the fact that its mean duration is short, 4 minutes.

It's interesting to observe that the second most frequent activity is the *Undertake reminder call to retailers*, an activity which was already commented on before.

On the other hand, the activity with the highest Mean Duration is *Retail Demand Forecast*. Yet, this activity has a frequency of 1, which means that only happens one for each process. *Undertaking financial calculations* activity has a similar mean duration and its frequency is 12.

To finish the analysis of Sales' activities, the activity with the widest Duration range is *Undertaking financial calculations*, with a 3 hours and 12 mins range. This is because the minimum duration registered was 82 minutes, but in fact the mean duration of this activity is close to 3 hours, so that case can be named as an exception. Even though, it would be interesting to study why that process was that fast and evaluate if it can be repeated in the future.

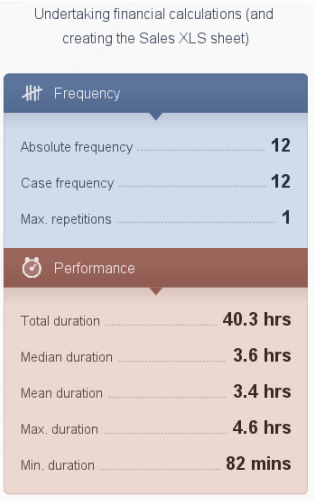


Figure 97. Closer look to the process *Undertaking financial calculations* for the year 2016-17

Activity	Frequency	Relative frequency	Mean duration	Duration range
Checking orders vs. forecast	1,134	71,14%	4 mins	
Undertake reminder call (retailers)	241	15,12%	5 mins, 6 secs	7 mins, 30 secs
Taking and transferring figures from forecast	126	0,79%	14 mins	
Checking for monthly list of orders	12	0,75%	1 hour, 49 mins	1 hour, 8 mins
Undertaking financial calculations (and creating the Sales XLS sheet)	12	0,75%	3 hours, 21 mins	3 hours, 12 mins
Sending the XLS sheet to Management, Procurement, Production	12	0,75%	28 mins, 13 secs	25 mins, 52 secs
Archiving the confirmation message from Procurement	12	0,75%	1 min, 48 secs	1 min, 58 secs
Checking stock (in case Production cannot handle quantities)	11	0,69%	25 mins, 3 secs	26 mins, 33 secs
Identifying affected orders (where problems occurred)	11	0,69%	43 mins, 22 secs	31 mins, 25 secs
Making decisions and informing those retailers we won't satisfy	11	0,69%	1 hour, 38 mins	1 hour, 59 mins
Send standard confirmation messages to retailers	11	0,69%	1 hour, 45 mins	1 hour, 53 mins
Retail Demand Forecast	1	0,06%	3 hours, 39 mins	

Table 4. Sales activities 2016-17

Employee Analysis for Sales Department during 2016-17

This table shows the top 7 and bottom 7 employees ordered by frequency of appearance in the different processes. The total number of employees in Sales is 28. The employee with the highest frequency has the ID 114 and his name is Edgar Ritschel. The employee with the

lowest frequency has the ID 80 and his name is Lewin Cordes. Take into account that frequency is just representing the number of times an employee participates in a process, but doesn't represent time spent on each process nor effectivity nor anything related. A low frequency employee can be more productive than a high frequency employee. We could just compare both employees by frequency if they both had the same responsibilities and jobs.

Value	Frequency	Relative frequency
114	70	4,39
119	69	4,33
22	66	4,14
52	66	4,14
5	63	3,95
55	63	3,95
99	63	3,95
…	…	…
61	51	3,2
47	50	3,14
34	49	3,07
102	49	3,07
24	48	3,01
41	48	3,01
80	45	2,82

Table 5. Employee Analysis for Sales Department during 2016-17

Process Map of Employee 114:

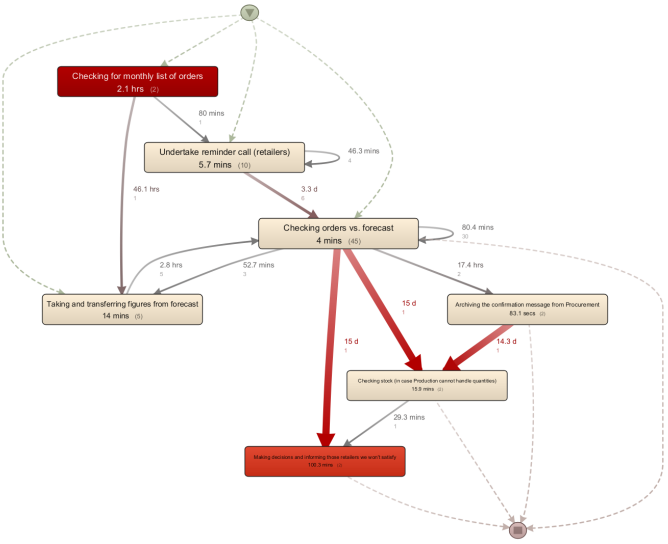


Figure 98: Process Map Employee 114. [5]

Process Map of Employee 80:

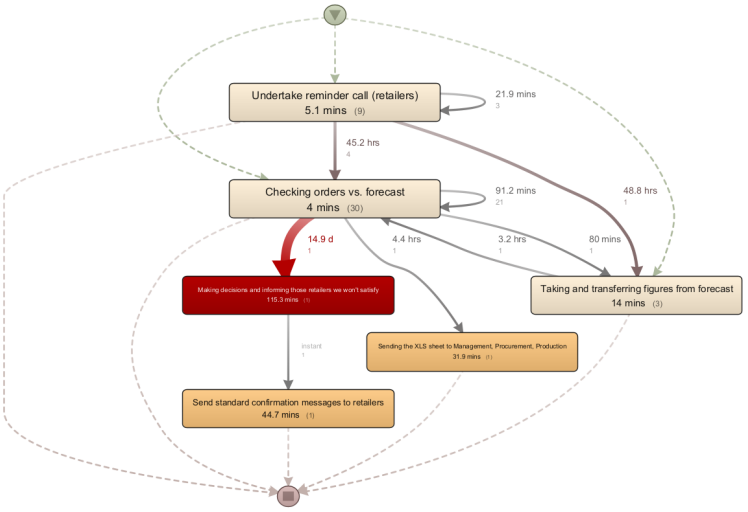


Figure 99: Process Map Employee 80.[6]

As it can be seen in the process maps, each one of these employees is performing different tasks, so frequency is not a good indicator of productivity in this case. Yet, it is a good indicator to calculate how many processes the employee is involved in.

[5] https://drive.google.com/file/d/1gEJ1gBXTguQym0it1wQe_gIYGtrkSQLX/view?usp=sharing
[6] https://drive.google.com/file/d/1f24xME-mNw0CM6SxWjcHubx6pwCqk5S8/view?usp=sharing

8. BUSINESS RECOMMENDATIONS

The first thing we would like to recommend is addressing data quality issues. It is important to have correct and complete data to create a full and realistic picture with the information. It is also important to realize that this information will be the basis for informed decision-making, allowing statistically sound decisions. As such, we recommend increasing data governance, assigning responsibility of data quality to an IT manager.

The data quality issues we would like to address are:

1) Duplicated retailers in the retailer table of OLTP, which have their own orders, extra to the original orders.
2) There are two different types of inconsistent selling prices in the database. For nine retailers, all of them are Polish, we found selling prices reaching 1000% of average price. At the same time, most retailers play multiple orders with the same items in the same month, partially even on the same day. While we did not expect this to be an inconsistency, we find that there are different selling prices in orders from the same day.
3) Generally, the CSV-files seem not to be a good approach to have extra orders. Several of the orders in these CSV files were duplicated from the orders in the OLTP, where we have no chance to know whether these are additional or duplicated orders. However, if you need to use CSV files for this purpose, make sure they contain employee data to maintain clarity, order IDs to avoid redundancies and maintain clarity and state of the order to maintain clarity.
4) Other inconsistencies that were not relevant for this data mart, however, may be relevant for other data marts, such as retailer's telephone numbers, employee addresses and zip codes not matching and so on (see Data Profiling for full information)

Next, we want to evaluate our business requirements set at the beginning:

1) The number of retailers the company is working with was consistent over the last 8 years at around 100 retailers. While the company is still growing, keeping the number of retailers consistent, acquiring new retailers can give another accelerator of growth. Here it is also worth noting that the revenue share of the 42 online retailers have been kept consistent at ~25% as well, even though global eCommerce sales have been increasing strongly every year since 2010 growing from $572 billion in 2010 to $4.2 trillion in 2020[7]. At the same time, acquiring new retailers can also mean global

[7] e.g., Jake Rheude, *ECommerce Growth from 2010 to 2020*. Red Stag Fulfillment, Octobre 9, 2021, https://redstagfulfillment.com/2010s-ecommerce-growth-decade/

diversification. This could mean reaching new markets, such as Asia, targeting Japan, South Korea, Russia, China or India; Northern Europe, targeting Denmark, Sweden, Norway and Finland; Other European countries, targeting Switzerland, Austria, Czechia and Portugal; and Australia.

2) Every single forecast for each retailer and material is missed. The forecast numbers overall don't seem reasonable and need to be reconsidered. From the process mining (and also from the BPM Project) we know that the forecast is also being used in case retailers don't place their orders in time. While this is not replicated in the database (we don't see orders with those numbers), it is a decent approach to have consistent Sales with the retailers. Forecasting can roughly be done in Tableau, the software we used for the dashboards, but also in other Software programs or in manual work.
3) The number of Sales is heavily seasonal, peaking in winter (November to January) and summer (April to July). As such, we have 5 months that could be considered off-season. Having seasonality in a company, it makes sense to adjust human resources accordingly. As we can see right now, the production staff is consistent over time, which likely means unnecessary costs for production in these months. Using placement agencies as supplement to a smaller base-staff in the on-season months can secure keeping knowledge about production in the company while being more flexible and cost-efficient.
4) The number of canceled orders was high every year, in the range of 20% and 36%. However, we noticed that the cancellation rate was higher in the last few years. What we don't know is whether the retailer or SuperX is canceling. If SuperX is canceling, we recommend adapting production and procurement processes, making sure retailer orders can be met. Lean processing or keeping higher stocks may help solve this issue. If the problem is on the retailer side, it might make sense to sign agreements on maximum buyer-cancelation rates or similar KPIs. Overall, due to cancellation, SuperX lost 6.4M in revenues in 2016 and 23.3M in revenues since 2010.
5) Through process mining, some activities were observed to be improvable. *Taking and transferring figures from forecast* and *Undertake reminder call to retailers* have margins to be enhanced by uniformizing the data format and automatizing the process, respectively. These tasks take up over 29 hours and 20 hours yearly, both slowing down the overall process time, and increasing waiting time for other departments, which may be critical for meeting retailer demands.

9. TOOL REVIEWS

9.1 PostgreSQL and pgAdmin 4

Overall, PostgreSQL seems suitable for what it does. The use of pgAdmin 4, the user interface we used, seems overall easy to navigate and is learned very quickly. PostgreSQL also offers some special SQL functions that can help find solutions, such as the generate_rows function we used for the load_csv ETL. In some cases, simple commands, such as dropping a table, took over 60 minutes. We are not sure whether that was a local problem or a server problem. However, other than that, we can't report any issue with PostgreSQL or pgAdmin.

9.2 SQL Power Architect

SQL Power Architect is a fairly easy tool to create schema for your database. It can be connected to a PostgreSQL server where you can create a connection to the database, which is convenient for us as we use PostgreSQL for both DBeaver and Tableau. Having understood all the theories about relational models, we did not have any issue creating tables, columns and setting up the attributes in the columns as well as the interconnection (primary key/foreign key) between tables. However, I do find the software interface not optimal, as some symbols (upper and side toolbar) are very small and one cannot zoom in when using the app. The layout tables are not aesthetically good-looking either and definitely could be improved. The way to forward the model to Dbeaver was not intuitive for a first time user, only after I was instructed in lecture did I know the process. Overall, the software gets the job done, but clearly has room for improvements.

9.3 Talend

Talend is a useful software for analyzing the data quality of a dataset. The main critic that could be done is that it requires a previous knowledge of the data in order to perform the best possible analysis. You have to know what you are searching for, and maybe that's not the best option because there can be errors that you might have not noticed, and if you don't execute a specific analysis on that field, Talend is not going to point it out. But if you know which are the weak points in your dataset, Talend is a powerful engine that will allow you to filter and detect the data that has to be corrected.

9.4 DBeaver

DBeaver gives a nice interface for SQL requests and works great with postgreSQL. While it can be annoying that every table in a server needs to be connected separately, this only needs to be done once and doesn't need maintenance afterwards. It is easy to start SQL queries, look at data, edit tables on the fly and export queries into other file formats like CSV. The latter can be very helpful to analyze some problems that may have come up while doing the ETL.

9.5 Pentaho

We have spent multiple days equivalents in Pentaho, figuring out how some transformations work and what the program's limits are. While we solved everything that we needed to solve, it overall feels like Pentaho has a very long learning curve. While the amount of choices for transformations are overwhelming, it is not easy for the user to understand what exactly they are doing. At the same time, we found mixed results when looking for help online. Many posts are old, and it generally feels like it's not worth it to create a forum account to ask for help in the community, as the chance for fast support feels slim. Some forums referred to transformations that are not existing in the version we were using, such as a python script transformation. As python was not available to us, which would often have made the transformation a lot easier, we instead used SQL scripts. The problem with SQL scripts in Pentaho is, however, that tables can only be loaded with SQL as a table input. In result, many transformations, that technically is one transformation (e.g., load_csv), had to be split up into 3-4 separate transformations, decreasing both overview of the whole ETL and readability for a third person. Overall, we were not able to find Pentaho's limits - as such, we can imagine that the program is well suited for people who are working on a regular basis with it and get comfortable and knowledgeable with the options Pentaho offers.

9.6 Tableau

Tableau is already a widely-used software for data visualization, and we greatly enjoy using the software for the project. I think the skills and tips we learned when making the dashboard would help us tremendously in the future. One great thing is that the software is very straightforward and has a variety of visualizing options (types of diagram, color, labels, calculations, filtering etc.) for us to play around and get creative. Personally, what I like about the software is how it recognizes the attribute named "Country" as a geographical feature and auto-match the text from our database with the correct country name, so that we could plot out the map. I have never used other data visualization software other than Tableau, so I

cannot make comparisons, but the user experience has been positive for me and I would recommend it.

9.7 Disco

Disco is able to offer a clear overview on the activities and processes of a company. In this specific case, the event log didn't have a particular ID for each entire process, so *Month* had to be used, resulting in a good but not perfect fit into the processes. Disco is not culpable of this factor, but maybe an optimal software would be able to detect the processes by analyzing their repetitions and timeframes. Apart from that, Disco has been useful, and it creates a precise visualization of how the company performs its actions.

10. TEAM MEMBER RESPONSIBILITIES

Even though there is a particular main responsible for each step, we communicated on every part of the project, gave feedback on it and solved the problems together.

Area	**Main responsible person**
KPIs and Business Requirements	Everybody
Data Quality	Roger
Data Mart Design	Minh Anh
Extract, Transform and Load Data (ETL) - **Create SCD** - **Fix data quality** - **Load CSV files** - **Load fact tables** - **Create CDC**	 Minh Anh & Roger Valentin Valentin Valentin & Minh Anh Minh Anh
Visualization Dashboard	Minh Anh
Process Mining	Roger
Business Recommendations	Valentin

Table 6. Main Responsibilities

11. REFERENCES

Rheude, Jake. “Ecommerce Growth from 2010 to 2020.” *Red Stag Fulfillment*, 9 Oct. 2021, https://redstagfulfillment.com/2010s-ecommerce-growth-decade/