Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Designing a Data Mart for the Procurement Department of SUPER-X

Authors
Annelie Schridde (1834424)
Joshua Campos Chiny (1848601)
Lucas Silbernagel (1838093)

Supervisor
Prof. Dr. Roland Müller

1. Analysis of the business requirements of the data mart, including identifying important KPI's for the data mart

As a first step for creating a data mart for the procurement department, we need to get familiar with their processes and the current database that SUPER-X uses. With the database it is possible to identify what data already exists and can be used for the data mart. Therefore, a source-driven approach was used to derive different business requirements and to analyze the department's specific activities. Each team member noted down 3 to 5 business requirements that appeared to them as useful. In a meeting the ideas were shared and evaluated. Further, the tables from the database that seemed relevant for answering these business questions were discussed. The seemingly most important requirements were then added to the final business requirement table. The results can be found below.

No.	Business Requirement (Question)	Importance	High Level Entities	Measures
1	How accurate is the forecast compared to the actual demand by material and by month?	High	Forecast, Demand, Material, Month	Quantities difference
2	What is the average level of inventory of each material type in each month?	Medium	Inventory, Material, Month	Level of inventory
3	How high is the rate of missing materials that cannot be compensated by the inventory? What are the missing materials? Are they common to be missing?	Medium	Material, Inventory	Rate of missing materials
4	Which are the top 10 suppliers which we purchase the most from monthly?	Low	Suppliers, Month	Cost of orders
5	Which materials do we buy the most and how much do we spend on them monthly?	High	Material, Month	Cost of orders
6	What is the average duration between the purchase date and the delivery date by material and by month?	Medium	Purchase, Delivery, Material, Month	Average time difference
7	Are there any inconsistencies between the purchased items and the delivered items by material and by month?	High	Purchase, Delivery, Material, Month	Average quantity difference

Table 1 Business Requirements

The table shows the business questions that could be relevant for the new data mart along with their importance, entities, and measures.

From this table it is already possible to also note down KPI's that are needed to answer the business requirement questions. Later, this is the basis for modeling the data mart. These thoughts resulted in the following information package table that gathers information about the entities and the possible dimensions as well as their hierarchies and KPI's.

Entities (potential future dimensions)	Hierarchies in entity (potential future dimensions)
Suppliers (Category, currency, state)	
Material (Type)	
Date (Month, Quarter, Year)	Month -> Quarter -> Year
Purchases (Currency, supplier)	
Deliveries (Supplier)	
Inventory	
Forecast (Retailer)	
Demand	
Measures (Key Performance Indicators) (potential future measures in fact table)	Cost of orders, (rate of missing materials), (stock level of inventory), (quantities difference), average time difference, average quantity difference

Table 2 Information Packages

As the problem occurred that not only one but two fact tables would be needed to answer all these questions, we decided to limit the requirements in a way that a simpler data mart with a star schema can be created. Only procurement's activity within their purchases and deliveries are in focus for this data mart. Otherwise, a second fact table describing the inventory with materials, the monthly demands and supplies and the forecast table would be needed to answer the first three business requirement questions. Also, the average stock level of the inventory cannot be answered as the inventory table only shows the current state of the stock level. Therefore, we decided to exclude the business requirement questions 1 to 3 as they do not seem to be answerable within the scope of this project.

For the remaining requirements we then conducted a data quality analysis. The procedure is described in the next chapter.

2. Analysis of the relevant data sources including an analysis of the data quality

Before starting the analysis of the data quality, we first must identify the data sources that are relevant for our data mart. Since we chose to focus on the purchase and delivery section of the procurement process, we identified that our business questions could be answered with the following tables:

- Material
- Supplier
- Purchase Order
- Purchase Order Item
- Delivery
- Delivery Item

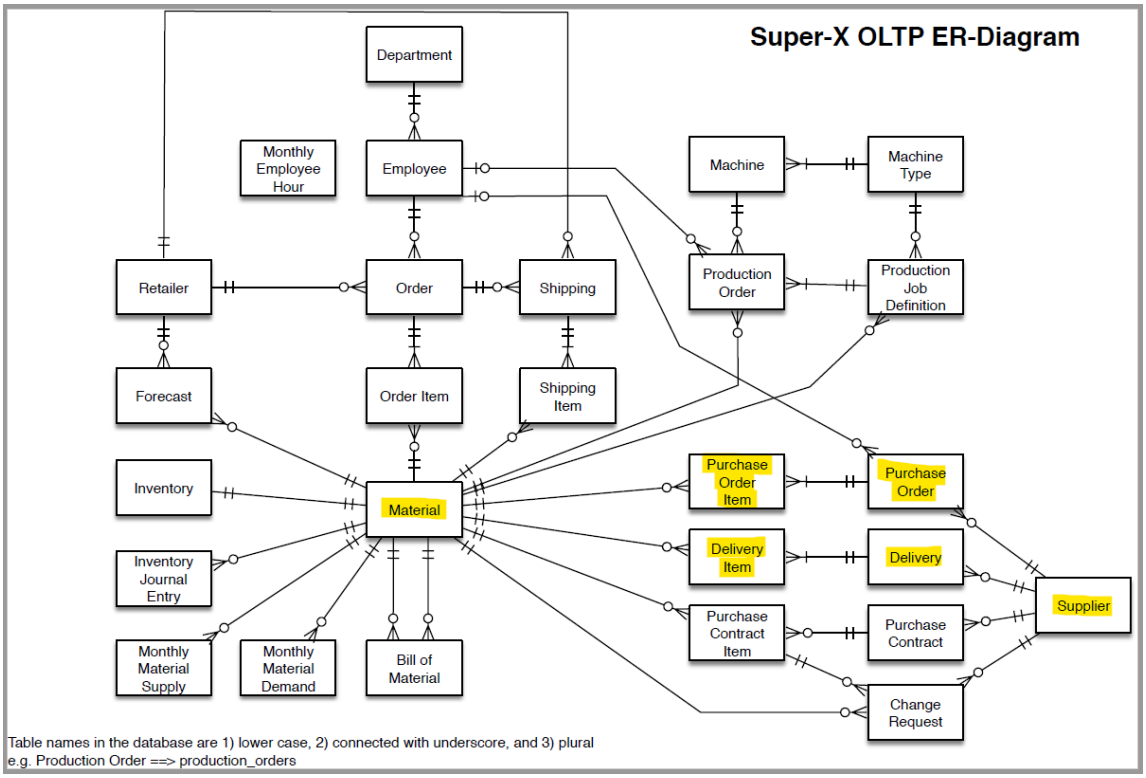

Figure 1 Selection of the relevant data sources

After identifying the required tables, we need to analyze the data's quality in order to detect any anomalies or inconsistencies in the tables mentioned above that might subsequently affect our data mart. For this task, we decided to use Talend Open Studio with which we did several types of analysis on each of the tables including schema analysis, column analyses, redundancy analyses, and matching analyses.

The schema analysis gives us a general idea about the tables we are working with, such as the number of rows, keys, and indices. The column analyses helped us focus on the most important columns for our data mart and find important factors such as the number of nulls and duplicates, the frequency of values, summary statistics, etc. The redundancy analysis was executed to verify the relationship between foreign and primary keys and to identify any possible integrity violation. The matching analyses were performed to assess the possible number of duplicates in the data caused by slight differences in their spelling.

Not all analyzed tables had significant data quality issues. For this reason, we will only talk about the tables that had actual problems or that gave us an interesting insight about the data. Before we start with the specific tables, we did a schema analysis to understand the general structure of the tables. We can see that purchase orders and deliveries have the same number of rows. The same happens between purchase order items and delivery items. That might tell us that all the purchases made were indeed delivered.

Table	#rows	#keys	#indexes
materials	31	1	1
suppliers	285	1	1
purchase_orders	4646	1	3
purchase_order_items	18044	1	3
deliveries	4646	1	3
delivery_items	18044	1	3

Figure 2 Results of schema analysis

- Suppliers

The first table that we want to talk about is the suppliers table. From the column analysis, we found out that there are two suppliers that have the exact same name. In this case, we will be using the second record as it seems to be the most recent one which tells us that maybe the supplier information changed but the record was not updated.

id	name	description	timestamp
228	Broek BV	Inverse bottom-line installation	2015-08-03 09:00:00.000
272	Broek BV	Reverse-engineered maximized hub	2016-10-03 09:00:00.000

Figure 3 Duplicate results from column analysis

From the matching analysis, we also noticed that there are a few suppliers with very similar names. For these suppliers, we will be using the first record as we can see that there are also spelling mistakes in the address column for the others which gives us a clue that the address is also wrong.

	name	address
1	Enríquez Soto e Hijos	Ronda Sergio Mondragón, 74 Esc. 719, 26010 Huelva, Aragón, Spain
2	Eoríquez Soto e Hijns	R%nda Ser*io Mondragón, 74 Esc. 719, 26010 Huelva, Aragón, Spain
3	Goldkühle, Ne und Schedler	Am Alten Schafstall 85b, 70119 Ost Fabianland, Hamburg, Deutschland
4	Gdlokühle, Ne und Schedler	Am Alten Sc$afstall %5b, 70119 Ost Fab*anland, Hamburg, Deutschland
5	Goldkühle, Ne und Schelder	Am Alten Schafst$%l 85b, 70119 O*t Fabianland, Hamburg, Deutschland
6	Henkel-Ullrich	Meckhofer Feld 18, 14490 Jarosburg, Bremen, Deutschland
7	Hinkel-Uelrich	Meckhofer*Feld 18, 1449$ Jarosburg, Breme%, Deutschland
8	Broek BV	Noapark 228 III, 9947 OE Noord Emmaenmaes, Noord-Braband, Netherlands
9	Broek BV	Svenpark 171a, 8416 FS Bergwoude, Gelderland, Netherlands

Figure 4 Results of matching analysis for supplier's table

From the column analysis, we also found that there is a wrong category for the suppliers. Instead of the correct "small" category, there is another category labeled as "smal" which must be fixed to the correct one. We can also notice that this wrong category has been used several times, so it is not an uncommon mistake.

Value	Count	%
big	74	25.96%
smal	73	25.61%
small	69	24.21%
medium	69	24.21%

Figure 5 Wrong "smal" category from column analysis

- Purchase Orders

The next table that we analyzed was the purchase order's table. From the cross-table redundancy analysis we found out that there are two suppliers from which no purchases have been made. This does not pose as a problem to our data mart but an interesting fact.

	suppliers	purchase_orders
%Match	99.30%	100.00%
%NotMatch	0.70%	0.00%
#Match	283	4646
#NotMatch	2	0
#Rows	285	4646

Figure 6 Redundancy analysis between suppliers and purchase orders

- Purchase Order Items

We will continue with the analysis of the purchase order item's table. From the column analysis, we discovered that there are over 350 null values on the currency column. In order to fix this, we could either use the default currency value or use the supplier's currency which should match with the purchase order item's currency.

Label	Count	%
Row Count	18044	100.00%
Null Count	353	1.96%
Distinct Count	6	0.03%
Unique Count	0	0.00%
Duplicate Count	6	0.03%
Blank Count	0	0.00%
Default Value Count	11713	64.91%

Figure 7 Column analysis on currency

From the first cross-table redundancy analysis with the purchase order's table, we noticed that there are over 1318 records that are not present in the purchase order item's table. This means that there are over 1300 purchases that have no record of the items they included. For this specific problem, we do not know where to get the missing data so unfortunately, we will have to filter out the purchase orders that do not have any items linked to them.

	purchase_orders	purchase_order_items
%Match	71.63%	100.00%
%NotMatch	28.37%	0.00%
#Match	3328	18044
#NotMatch	1318	0
#Rows	4646	18044

Figure 8 Redundancy analysis between purchase orders and items

From the second cross-table redundancy analysis, we can see that there are 16 materials that are not present in the purchase order item's table. In this case it makes sense since only 15 materials are purchased from suppliers while all the others are produced by SUPER-X itself.

	materials	purchase_order_items
%Match	48.39%	100.00%
%NotMatch	51.61%	0.00%
#Match	15	18044
#NotMatch	16	0
#Rows	31	18044

Figure 9 Redundancy analysis between materials and purchase order items

- Deliveries

We then analyzed the deliveries table by doing a cross-table redundancy analysis with the suppliers table where we found out that there are also two suppliers that have never done any deliveries. This makes sense as they are both the two suppliers from whom nothing has ever been purchased.

	suppliers	deliveries
%Match	99.30%	100.00%
%NotMatch	0.70%	0.00%
#Match	283	4646
#NotMatch	2	0
#Rows	285	4646

Figure 10 Redundancy analysis between suppliers and deliveries

- Delivery Items

For the delivery item's table, we performed a cross-table redundancy analysis with the deliveries table. We found out that there are 1318 records from the deliveries table that do not match for the delivery item's table. This must be the same missing data as from the purchase orders and the purchase order items. Like we mentioned before, unfortunately, this missing data cannot be found so we will have to filter out the deliveries that do not have items linked to them.

	deliveries	delivery_items
%Match	71.63%	100.00%
%NotMatch	28.37%	0.00%
#Match	3328	18044
#NotMatch	1318	0
#Rows	4646	18044

Figure 11 Redundancy analysis between deliveries and delivery items

We also performed a cross-table redundancy analysis with the material's table, where we found out that 16 materials do not appear in the delivery item's table. Just like with the purchase order items this is logical because only 15 materials are purchased and delivered.

	materials	delivery_items
%Match	48.39%	100.00%
%NotMatch	51.61%	0.00%
#Match	15	18044
#NotMatch	16	0
#Rows	31	18044

Figure 12 Redundancy analysis between materials and delivery items

- Extra Purchase Order Items

We also have several CSV files that include extra purchase order items that are not included in the database. We first did a matching analysis on the supplier column, where we found out that there are many suppliers with slight differences in their spelling, which means that this might be typing mistakes. To fix this, it will be necessary to find the correct supplier name and replace those with slight differences.

	Supplier	SCORE
174	Gamez, Viera y Ramón Asociados	1.0
175	GamezR Viera y ,amón Asociados	0.9333333333333...
176	Gamez, Viera y Ramón Asociados	1.0
177	Gamezy Viera , Ramón Asociados	0.9333333333333...
178	Gamez, Viera y Ramón Asociados	1.0
179	Gamez, Viera y Ramón Asociados	1.0
180	Gamez, ciera y Ramón AsoViados	0.9333333333333...
181	Gamez, Vaera y Ramón Asociidos	0.9333333333333...
182	Gamez, Viera y Ramón Asociados	1.0

Figure 13 Section of matching analysis on supplier's name

We then performed the same type of analysis on the material's column where we expected the same to happen – and it did. We have many materials that have slight differences in their names which possibly result from typing mistakes. The same approach of finding the correct one and replacing the others must be performed here.

	Material	SCORE
11	Receiver 2-Ceannhl 2MHz	1.0
12	Receever 2-Channil 2MHz	0.8695652173913...
13	Receivhr 2-Ceannel 2MHz	0.9130434782608...
14	Receiveh 2-Crannel 2MHz	0.8695652173913...
15	Raceiver 2-Chennel 2MHz	1.0
16	Receiver 2-Cahnnel 2MHz	0.8695652173913...
17	Receiver H-Channel 2M2z	1.0
18	R2ceiver e-Channel 2MHz	0.8695652173913...
19	Receiver 2-CHannel 2Mhz	0.8695652173913...

Figure 14 Section of matching analysis on material's column

Unfortunately, these CSV files do not contain any information about the deliveries of the orders so we will not be able to use it in our data mart as we also need information about the deliveries for each record of our fact table.

3. Multi-dimensional design of the data mart

When it comes to the multi-dimensional design of a data mart, there are several approaches on how to do that. We decided to choose a graphical approach by modeling it as a Multidimensional Entity Relationship Diagram (ME/R). The diagram is shown in figure 15.

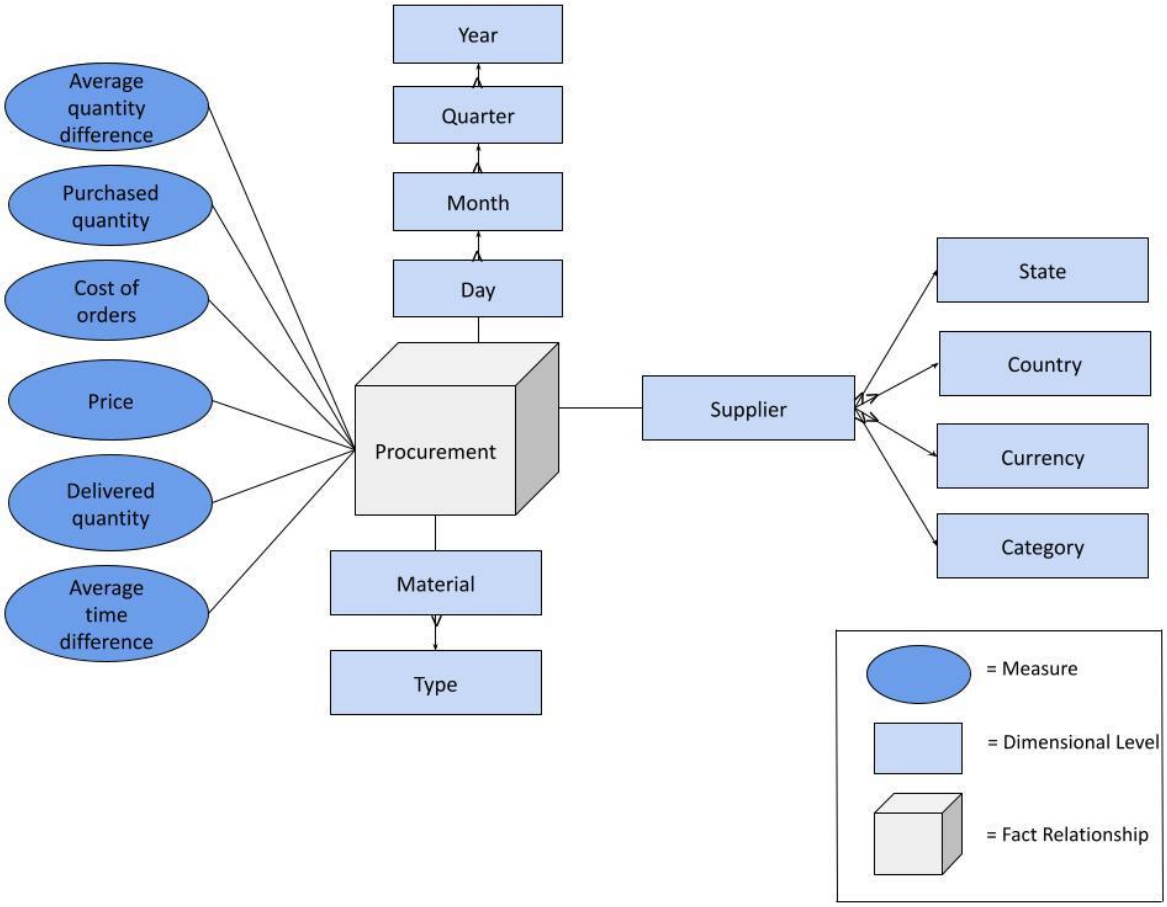

Figure 15: M/ER diagram of the data mart

In chapter one, we already described the process of getting from brainstormed business requirements to the actual information package which lists all relevant entities, facts, and dimensions. This work pays off when it comes to modeling the ME/R diagram since it already contains all necessary information.

We decided to choose “Procurement” as the name for our fact table since it includes all dimensions that we need to answer the business requirements we defined.
Our time dimension is divided into day, month, quarter, and year. For most of our Business Requirements a month granularity would have been enough but since we are asking for a time

span between the date of purchase and date of delivery, we had to opt for a day granularity in this dimension. For our material dimension, there is only one single-level hierarchy for the material type. It is the only information out of the material table that is necessary to answer our business requirements. Lastly, we have chosen a supplier dimension which includes single-level hierarchies for state, currency, and category.

For us, the most challenging part of the conceptual design was coming up with the correct measures to include. While three of them, namely the average time, the quantity difference, and the cost of orders, were obvious, we were not sure whether to include the other three figures, namely purchased and delivered quantity and price. The reason for this was the fact that these figures can already be found in the database which means they do not require any computation efforts. Thinking of Occam's Razor, we were considering not include these three measures for simplicity reasons. However, since they are not part of any of the dimensional tables, we decided to include them for the sake of completeness and increased comprehensiveness. For us, the inclusion does add value to the data warehouse.

4. Proof-of-concept implementations of the multi-dimensional design

4.1 Multidimensional implementation

For the implementation of our data mart, we first created the logical model using SQL Power Architect, based on our previously defined conceptual model. We have the same three dimensions tables, each with its most relevant descriptive columns, as well as its hierarchies. For the material dimension, we decided to use a slow changing dimension of type 1, so whenever the material changes in some way, the previous record will be overwritten. We chose this because we believe the materials will rarely change. For the supplier dimension, we decided to use a slow changing dimension of type 2, so whenever any information of the supplier changes, we will be updating the date from, date to, as well as its version. For the date dimension, we did not implement any slow changing dimension technique since it is not necessary.

For the procurement fact table, we will have one primary foreign key for the material and supplier dimension, but two for the date dimension, one for the purchase date and one for the delivery date. The measures will also match the conceptual model previously explained, where two measures will be taken from the transactional data sources, specifically the purchase and delivery quantity, and the rest will be calculated during the ETL process.

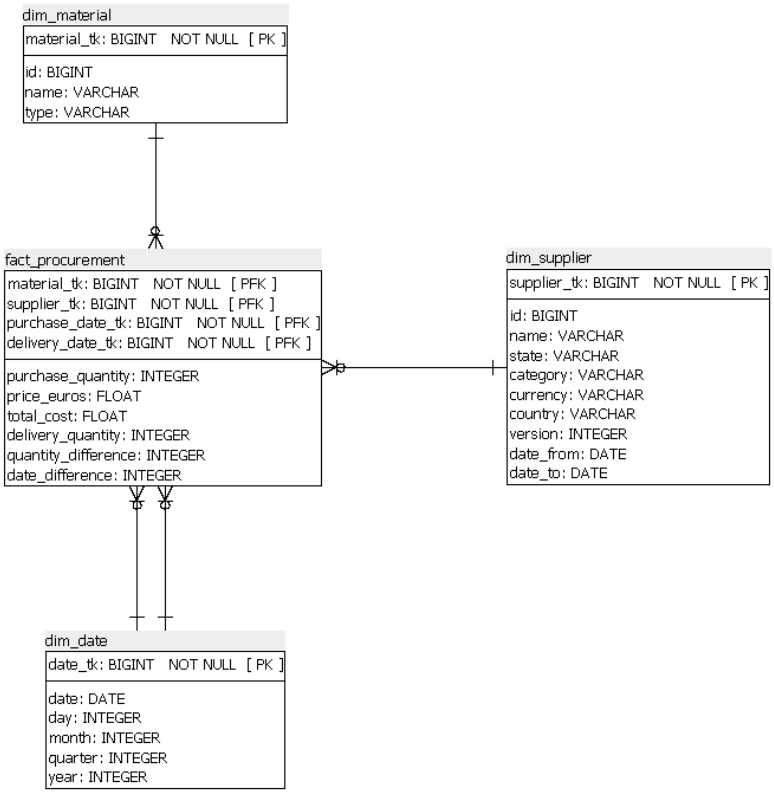

Figure 16 Logical model of data mart

We decided to create our data mart using Amazon Relational Database Service (RDS), where we chose to use the PostgreSQL database engine. We created the database in the Central European region to improve the general connectivity and latency. We also had configured the accessibility to the virtual public cloud by allowing a connection from all IP addresses.

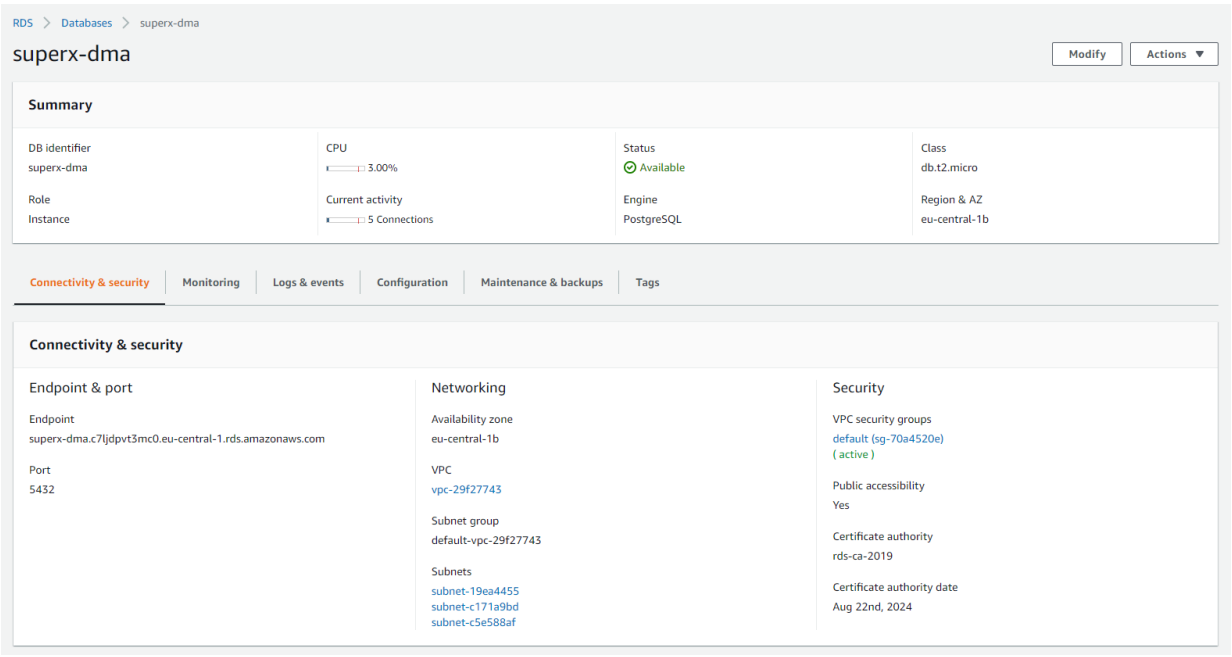

Figure 17 Database setup in AWS

Once everything was configured in Amazon RDS, we forward engineered our logical model from SQL Power Architect to our database. After connecting from DBeaver, we were able to see the ER diagram to confirm that all the tables and relations were created correctly, as we had previously designed them.

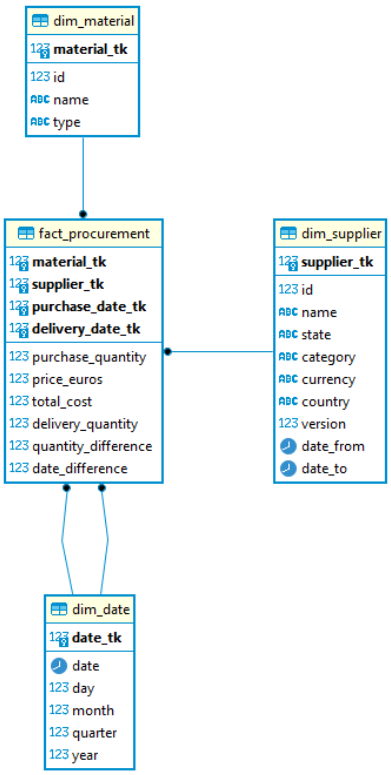

Figure 18 Datamart ER diagram

4.2 ETL Process

For the ETL process we used Pentaho Data Integration, where we created four transformation workflows, one for each dimension table and one for the fact table. We also created two jobs, one for the dimensions' transformation and one for the entire ETL process. We will discuss more in detail each of the workflows created.

Material Dimension Transformation

Since we could not locate any quality issues within the material table, the ETL process for the material dimension did not require any transformation. Thus, the only two steps in the Pentaho workflow are a "Table Input" ("Read Materials Input") to extract the columns "type" and "name" form the material table and an "Insert / Update" to load the extracted data in our data mart. Further, the "Insert / Update" not only loads the data but also overwrites entries with the same ID since we decided to apply type 1 of Slowly Changing Dimensions to the material dimension.

Figure 19 Pentaho workflow for material dimension

dim_material

Properties | Data | ER Diagram

dim_material | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	material_tk	id	name	type
1	1	1	Plastic Plane	RawMaterial
2	2	2	Screw 6mm	RawMaterial
3	3	3	Axle 60mm	RawMaterial
4	4	4	Remote Controller 2-Channel 2MHz	OemProduct
5	5	5	Remote Controller 1MHz	OemProduct
6	6	6	Tire 20 mm	OemProduct
7	7	7	Ni-Cd Battery 12V 300mAh	OemProduct
8	8	8	Motor 12V	OemProduct
9	9	9	Receiver 2-Channel 2MHz	OemProduct
10	10	10	Receiver Channel 1MHz	OemProduct

Figure 20 Final result for the material dimension

Supplier Dimension Transformation

As stated in the data quality section in the second chapter, in our raw data there were several data quality issues found in the table “suppliers”. Since this table is the underlying one for our supplier dimension, certain transformations were necessary to eliminate them.

As shown in *Figure 21*, the first step (“Read Suppliers Table”) is again a “Table Input” which extracts the columns “name”, “address”, “state”, “category”, “currency” and “timestamp” from the supplier’s table of the raw data. However, in the SQL query that defines which data to extract already excludes the rows 22 and 23 because it is duplicate data (see chapter 2 “Goldkühle, Ne und Schedler”). Since we could not figure out how to dynamically duplicate records where small differences in supplier name and address indicate them being a duplicate, we used this hard-coded solution to exclude them.

The second element of the supplier dimension workflow is the Regex evaluation “Get Country”. Using the regex expression “(.*,\s)(.*)” on the address column, we were able to isolate the country names from the rest of the address. In the next step a “Value mapper” (“Change countries”), we were now able to tackle the second data quality issue: inconsistent spelling of country names. This issue was resolved by replacing “U.S.A” and “USA” with “United States” and “Deutschland” with “Germany”. Another case of inconsistent spelling was found I the “category” column of the supplier’s table. To resolve this, another “Value mapper” (“Change category”) was used to replace “smal” with “small”.

The last remaining at this point was another duplicate value. Unlike the duplicates that were mentioned earlier in this paragraph, this pair of duplicates had identical values in the “name” column which is why we could use the in-built step “Unique rows” to get rid of one of them. However, we were not indifferent about which entry to drop but wanted to keep the more

recent one. Thus, we included a “Sort rows” step before “Unique rows” which ordered the data based on “name” and then “timestamp”. This was important because the “Unique rows” step always removes the first duplicate row and by imposing this order, we could ensure that it is the less recent one.

The last step is “Dimension lookup/update”. This step not only loads our transformed data in the data mart but also guarantees compliance to the requirements of type 2 of Slowly Changing Dimensions. It does so by adding a new row and incrementing the value of the “version” column of the dim_supplier table by one whenever values of an already existing supplier were changed. Changes made to the “id” column, however, will not be taken over.

Figure 21 Pentaho workflow for supplier dimension

dim_supplier

Properties | Data | ER Diagram

dim_supplier | Enter a SQL expression to filter results (use Ctrl+Space)

	supplier_tk	id	name	state	category	currency	country	version	date_from	date_to
1	0	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	1	[NULL]	[NULL]
2	1	1	Brouwer BV	active	small	EUR	Netherlands	1	1900-01-01	2199-12-31
3	2	2	Goldkühle, Ne und Schedler	active	small	EUR	Germany	1	1900-01-01	2199-12-31
4	3	3	Santoro-Barbieri s.r.l.	active	small	EUR	Italy	1	1900-01-01	2199-12-31
5	4	4	Henkel-Ullrich	active	medium	EUR	Germany	1	1900-01-01	2199-12-31
6	5	5	Crona, Huels and Koelpin	active	medium	CAD	Canada	1	1900-01-01	2199-12-31
7	6	6	Bins, Kuvalis and Hand	active	small	CAD	Canada	1	1900-01-01	2199-12-31
8	7	7	Gamez, Viera y Ramón Asociados	active	big	EUR	Spain	1	1900-01-01	2199-12-31
9	8	8	Leeuwen V.O.F.	active	small	EUR	Netherlands	1	1900-01-01	2199-12-31
10	9	9	Wisozk-Tremblay	active	medium	USD	United States	1	1900-01-01	2199-12-31

Figure 22 Final result for the supplier dimension

Date Dimension Transformation

For the date dimension we started off with a “Generate rows” step. This step is used to generate 3500 rows with the value “2009-01-01” in the new column “start_date” in the format “yy-MM-dd”. Afterwards, the step “Add sequence” initiates another column “day_number” which starts of with the value one in row one and increments by one in each row. The next step “Calculator” continuously creates in each row the sum of the “start_date” and the “day_number” value which results in 3500 consecutive dates starting from the 1st of January 2009. A second “Calculator” step called “Calculator 2” then initializes four more columns called “day”, “month”, “quarter” and “year”. “Calculator 2” populates these columns by splitting the respective date value and assigning the correct sub-value to the correct column. Since our business requirements only entail comparisons on a monthly level, the values in the “day” column refer to days in a month. In the next step “User defined Java expression” we create a new column “date_tk” which serves as our technical key in the data mart. This column is populated by the java expression “Integer.parseInt(new java.text.SimpleDateFormat ("yyyyMMdd").format(the_date))” so that it contains each date value as an integer. Since the “start_date” and “day_number” were only needed for computational purposes, we drop in the next step “Select values”. Further, we bring our remaining columns in the correct order that the data mart requires. The last step is a “Table Output” that loads our transformed data in the data mart.

Figure 23 Pentaho workflow for date dimension

dim_date

Properties | Data | ER Diagram

dim_date | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text

	date_tk	date	day	month	quarter	year
1	20,090,101	2009-01-01	1	1	1	2,009
2	20,090,102	2009-01-02	2	1	1	2,009
3	20,090,103	2009-01-03	3	1	1	2,009
4	20,090,104	2009-01-04	4	1	1	2,009
5	20,090,105	2009-01-05	5	1	1	2,009
6	20,090,106	2009-01-06	6	1	1	2,009
7	20,090,107	2009-01-07	7	1	1	2,009
8	20,090,108	2009-01-08	8	1	1	2,009
9	20,090,109	2009-01-09	9	1	1	2,009
10	20,090,110	2009-01-10	10	1	1	2,009

Figure 24 Final result for the date dimension

Procurement Fact Transformation

For the transformation of the procurement fact table, we started querying the four tables where we will get all our necessary transactional data. These tables are Purchase Order Items, Purchase Orders, Delivery Items, and Deliveries. After we query these tables, we join Purchase Order Items and Purchase Orders on the Purchase Order Id; at the same time, we join Delivery Items and Deliveries on the Deliveries Id. After this, we remove the duplicate suppliers in both tables, which we discovered during the data quality process. We then sort both tables by Supplier Id, and then look up the Supplier dimension to find their technical keys. For the Purchase Orders path, we also get the supplier's currency, as we will need it soon.

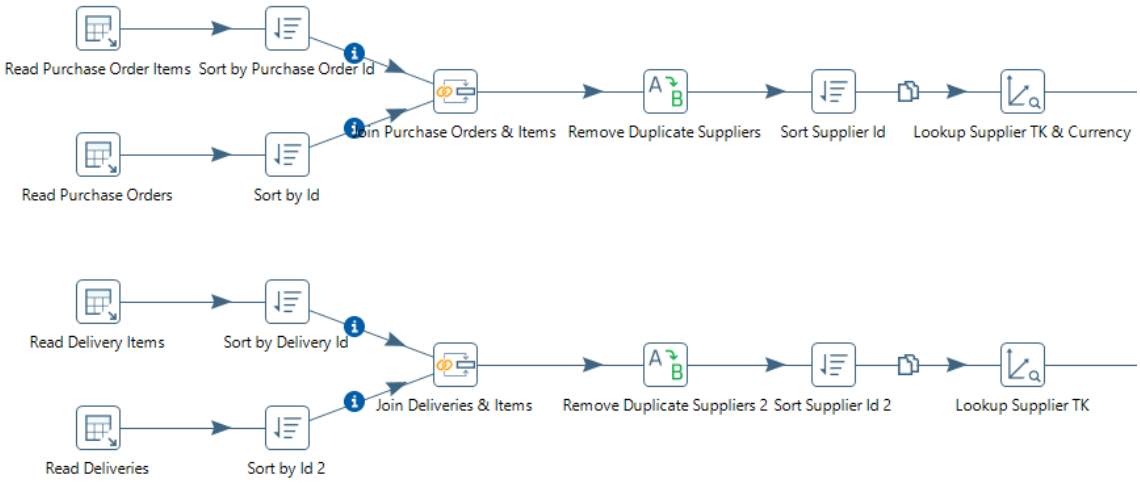

Figure 25 Fact Table ETL (Part 1)

In order to compute the total cost of the order in Euros, we have to convert the price used in each order to Euros, for which we use some pre-defined values. For a more accurate conversion rate, we could have used JavaScript to call an API and get the exchange rate for the purchase date; however, we did not do this as it is out of our project scope. After we calculate the currency conversion, we can compute the total cost for each order. We then get the month and year from the purchase and delivery date, as we will need to group by them later. Afterwards, we remove some unnecessary columns and sort the rows according to Material Id, Supplier Id, Month and Year, which we then use to group by and join our purchase orders to our deliveries.

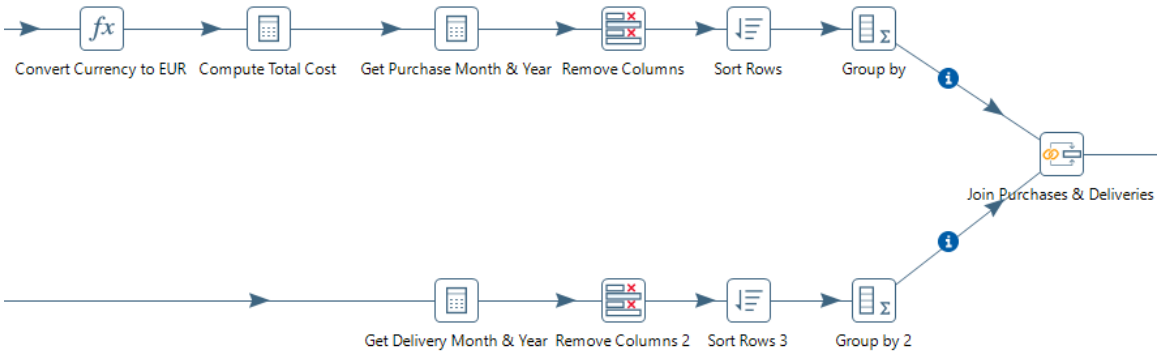

Figure 26 Fact Table ETL (Part 2)

After we have done this, we look up for the technical keys from the material dimension. Once we have this, we calculate the quantity difference and date difference between the purchase orders and deliveries. We then create the date technical keys, remove certain columns, adjust the price, date, and technical keys format, and finally output our results to the fact table in our database.

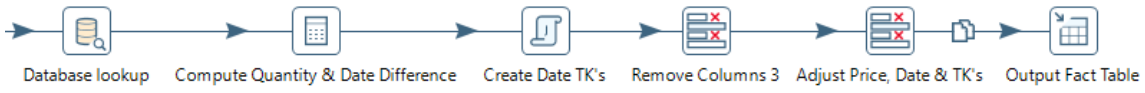

Figure 27 Fact Table ETL (Part 3)

In the following figure, we can see the output from the ETL process of the procurement fact table transformation, where we have four technical keys that relate to our dimensions, and six difference measures to answer our business questions.

fact_procurement

Properties | Data | ER Diagram | AWS Database | superx_dma | Schemas

fact_procurement | Enter a SQL expression to filter results (use Ctrl+Space)

	material_tk	supplier_tk	purchase_date_tk	delivery_date_tk	purchase_quantity	price_euros	total_cost	delivery_quantity	quantity_difference	date_difference
1	1	4	20,100,111	20,100,114	37,542	4.26999998	160,304.34375	37,542	0	2
2	1	4	20,110,110	20,110,113	37,542	4.26999998	160,304.34375	37,542	0	2
3	1	4	20,120,109	20,120,112	37,542	4.26999998	160,304.34375	37,542	0	2
4	1	4	20,130,109	20,130,111	18,771	4.26999998	80,152.171875	18,771	0	1
5	1	4	20,140,109	20,140,114	37,542	4.26999998	160,304.34375	37,542	0	4
6	1	4	20,150,109	20,150,114	18,771	4.26999998	80,152.171875	18,771	0	4
7	1	4	20,160,111	20,160,114	37,542	4.26999998	160,304.34375	37,542	0	2
8	1	4	20,170,109	20,170,112	37,542	4.26999998	160,304.34375	37,542	0	2
9	1	4	20,100,209	20,100,212	37,542	4.26999998	160,304.34375	37,542	0	2
10	1	4	20,110,209	20,110,214	37,542	4.26999998	160,304.34375	37,542	0	4

Figure 28 Final result of the procurement fact table

ETL Jobs

Once we have all our transformation workflows, we create our first job. In this job, we specify to run all the dimension transformations in parallel. We also specify to run the date transformation only once when the table is empty.

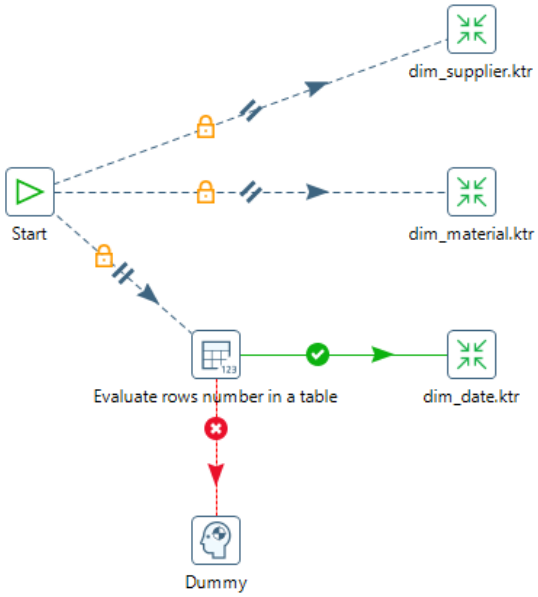

Figure 29 ETL Job for Dimension Transformations

Finally, we create a simple job to run the entire ETL process, where it starts by running the dimensions' transformation job, followed by running the transformation for the fact table. Once this job is done, we can see all our transformed data in each of the tables of our database.

Figure 30 ETL Job for Entire Process

4.3. Implementation of a dashboard for your data mart that visualizes the KPIs for the business process

For the visualization dashboard, we decided to use Tableau. For this task, we created six different worksheets, where the first four focus on answering the selected business KPIs from the first chapter, and the other two worksheets give other interesting insights about the data in our data mart.

Our fourth business question, "*Which are the top 10 suppliers which we purchase the most from monthly?*", can be answered with our first visualization on top left corner of our dashboard. Our fifth question, "*Which materials do we buy the most and how much do we spend on them monthly?*", can be answered with the middle left visualization. Our sixth

question, "*What is the average duration between the purchase date and the delivery date by material and by month*?", can be answered with our visualization on the bottom left corner of the dashboard. On the top right visualization, we included a map of all the countries the suppliers are, colored according to the total cost of their suppliers' orders. On the bottom right, we have a visualization that shows a timeline of all the months throughout the years, and we also show the total purchase cost that we have incurred in and the average number of days that all materials take to reach us. All the visualizations can be used as filters, and we also included a month filter on the top right to comply with the business requirements.

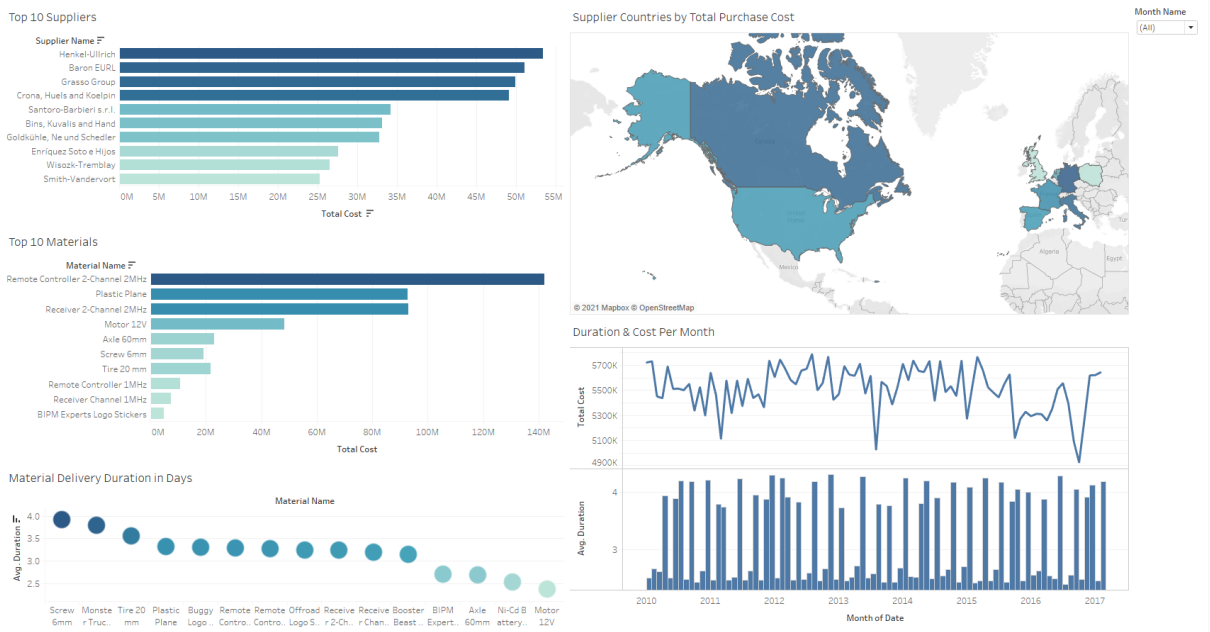

Figure 31 Final Procurement Dashboard

For our seventh question, "*Are there any inconsistencies between the purchased items and the delivered items by material and by month?*", we discovered there was no difference at any point in time; therefore, we did not include the visualization in the dashboard. Here we can see that the difference has always been zero.

Figure 32 Material Delivery Difference Result

4.4. Evaluation and comparison of the used data warehousing technologies

Local PostgreSQL Database vs. PostgreSQL Database in Amazon Web Services

Since the beginning of the project, we were thinking about using a cloud-hosted database, so we could all access a single data warehouse without each having to run the ETL process individually and create the same data warehouse locally. However, for testing purposes, we first created a local database to make sure the data model made sense, and everything worked out. We ran the first couple of ETL processes on the local database to test them too. After we checked this, we decide to setup the PostgreSQL database in AWS and run the whole ETL process, so we could all access the same data warehouse for the other tasks.

Since we had already worked with a local PostgreSQL database during the course, it was not difficult to setup. We just created a new database, forward engineered the logical model from SQL Power Architect and connected Pentaho for the ETL process. If we had not had already the PostgreSQL installed, we would have had to download and install it, which is also not that difficult. Setting up PostgreSQL in AWS was not difficult either, but it required a few more steps. We first had to create an AWS with a valid payment method, even if we are using the free tier. We then had to select that we wanted to use the RDS (Relational Database Service), where we chose PostgreSQL, and used the default settings, including the minimal free tier capacities. After we finished the setup and chose the database name, we had to wait a few minutes for the instance to be created and started. After this, we tried to connect using pgAdmin, but failed. We had to configure the public accessibility by creating inbound and outbound security group rules, so it allowed the connections from any IP address. After we did this, everything was ready; we forward engineered the SQL Power Architect model, ran the whole ETL process and now we could all connect to the same data warehouse.

Some of the advantages of using cloud-hosted database are clear. This type of implementation allows several people from different locations to connect to the database, which makes sense, since databases are made to be accessed by more than one system. A local database makes sense if you use it for development or testing. However, we could probably say that a local database is easier to setup, even though you need to download the software first. The steps to create an AWS database are a few more, although the creation of the database itself is straightforward. We also noticed that latency also played its role when running the ETL process on the AWS database, since it took some time to get all the data and then to output it into AWS. We also must consider that a database in AWS is not free after the first year, which is not the case for a local database. One last drawback that we noticed is that this approach might have is that certain software makes it more difficult when connecting to a cloud-hosted database than a local one, as we will see in the next comparison. Besides from the small differences in the setup, we could finally say that most disadvantages of a cloud-hosted database are compensated by the flexibility that it brings.

Tableau vs. Power BI

We decided to use Tableau as our main visualization tool, since it is the one that we used during the course and it is also known as a powerful visualization software. For comparison, we decided to go with Power BI, since it is also one of the most used visualization tools in companies.

For Tableau, we did not have to install anything, since we already used it; however, the installation was very simple and straightforward. After that, the connection to the AWS PostgreSQL database was also very easy, we used the default PostgreSQL connector that the software provides. After we used the correct connection settings, we could access the data stored in the data warehouse. However, for Power BI, the process was not so simple. After downloading the software, we started having trouble connecting to the AWS PostgreSQL database. Apparently, the default connector does not work for cloud-hosted databases; we tested this by connecting to our local database, which worked fine. After a few hours of troubleshooting and looking for a solution online, we found that the connection had to be done using an ODBC (Open Database Connectivity) connector. For this, we had to download and install the psqlODBC driver, then we had to setup the connection in the ODBC Data Sources application in Windows, for which we also had a bit of trouble and had to do some troubleshooting along the way. After the connection was successful in the ODBC Data Sources application, we opened Power BI, and used the ODBC connector, instead of the PostgreSQL one, which now allowed us to connect to the database.

After using both visualization tools, we learned several of their advantages and disadvantages. Without considering the difficult setup, we concluded that Power BI is more intuitive and easier to learn than Tableau. This might be a good characteristic for people who are new to visualization tools. After you select the visualization type and the data you want to use for the visualization, there is a simple pane for the customization of the visualization. It is possible to go through all the visualization possibilities in a single place. Because of this and some other factors, we consider that it is easier to use. However, the customizations for Power BI are somewhat limited, as we can see in the *Figure 33*, compared to the Tableau visualization in *Figure 31*; therefore, we think that Tableau is a more powerful tool, which allows for better looking visualizations. An example of this is the map visualizations, which are far better in Tableau; we believe this is due to the map provider, which for Tableau is Mapbox, and for Power BI is Bing. Another interesting difference that we noticed is that in Tableau you can create worksheets, dashboards, and stories as separate elements, but in Power BI you only have pages, which can be used for single visualizations or dashboards; maybe it is also due to making it easier for the user. In conclusion, we cannot say which one is better or worse, but for user friendliness, Power BI might be the way to go, but for a more powerful and better-looking visualizations, Tableau might be the solution.

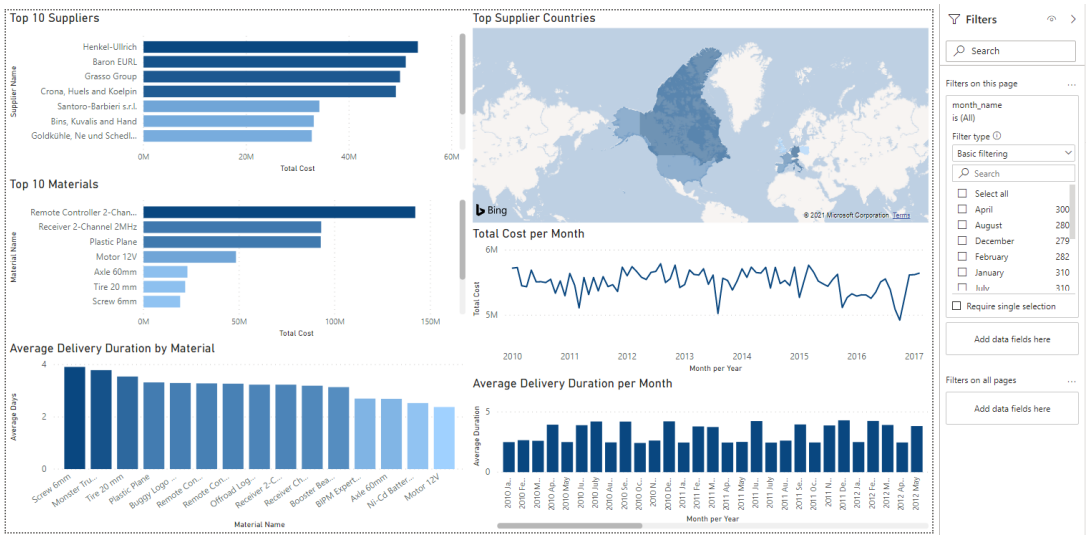

Figure 33 Visualization Dashboard in Power BI

5. Process Mining based on the event logs of the operational databases

For the Process Mining we decided to work with Disco. Just like any other Process Mining tool, Disco requires a case ID in order to properly analyze the event log that one provides as input. Since such a case ID was not directly specified in the event log, we had to extract it out of the given data. To do so, we took a closer look at the various activities in the "event_logs" table and found out that these activities were almost identical to the ones we encountered in the business process management project. From the latter we knew that the As-Is process of SUPER-X's procurement department takes place on a monthly basis and that a new process instance can only start when the prior one is finished. A new instance is always indicated by the same initial start activity. In the event log, this activity is called "Checking reserved quotas (calling suppliers at the beginning of a month)". That means that if we would order the "event_logs" table by either one of the timestamp columns, all the rows that come after such a start activity as well as the row with the start activity itself belong to the same process instance until another start activity is reached. The latter then indicates the start of a new process instance.

While this was the logical solution to the missing case ID, we now had to think about how to tackle the problem practically. Since our conceptual approach required complicated calculations, we decided to export the "event_logs" column from our database and import it in a Jupyter notebook to apply Python code.

```
import pandas as pd
import numpy as np
```

```
data = pd.read_csv("/Users/lucassilbernagel/Downloads/!Screenshots!/even_log.csv/event_logs_202101151306.csv")
data = data[data['department'] == 'Procurement']
data['case_id'] = 0
data = data.reset_index()
data.head(5)
```

	index	id	activity	start_timestamp	end_timestamp	retailer_id	supplier_id	employee_id	material_id	department	month	year	year_month	case_id
0	0	1	Checking reserved quotas (calling suppliers at...	2010-01-01 09:00:00	2010-01-01 13:26:14	NaN	NaN	37.0	NaN	Procurement	1	2010	2010/1	0
1	1	2	Engaging alternative supplier	2010-01-01 13:26:14	2010-01-01 15:12:12	NaN	NaN	80.0	NaN	Procurement	1	2010	2010/1	0
2	2	3	Drafting contract notes re penalties etc.	2010-01-01 13:26:14	2010-01-01 13:41:51	NaN	NaN	69.0	NaN	Procurement	1	2010	2010/1	0
3	3	4	Engaging alternative supplier	2010-01-01 13:26:14	2010-01-01 15:00:58	NaN	NaN	55.0	NaN	Procurement	1	2010	2010/1	0
4	4	5	Drafting contract notes re penalties etc.	2010-01-01 13:26:14	2010-01-01 13:43:08	NaN	NaN	67.0	NaN	Procurement	1	2010	2010/1	0

Figure 34 Import of "event_logs" table

As shown in figure x, we added the new column “case_id”, we filtered the data set in a way that only the activities of the procurement department remained and we reset the index.

```
v = 0
for row in range(len(data['id'])):
    if data['activity'][row] == 'Checking reserved quotas (calling suppliers at the beginning of a month)':
        data['case_id'][row] = v+1
        v += 1
    else:
        data['case_id'][row] = data['case_id'][row -1]

data.head(100)
```

Figure 35 Python code to populate “case_id” column

We now applied the Python code shown in figure x to populate the “case_id” column with the correct case ID for each row. The code uses a for loop that iterates over each row of the data set and an auxiliary variable “v” that increments by one in each iteration where the row’s activity is “Checking reserved quotas (calling suppliers at the beginning of a month)”. Further, whenever the latter is the case, the respective value for variable “v” is inserted as the row’s values for the column “case_id”. All rows which have a different activity value “copy” the case ID value form the previous row. As a result, we identified and assigned 87 different process instances with a total of 1131 activities.

	index	id	activity	start_timestamp	end_timestamp	department	month	year	year_month	case_id
0	0	1	Checking reserved quotas (calling suppliers at...	2010-01-01 09:00:00	2010-01-01 13:26:14	Procurement	1	2010	2010/1	1
1	1	2	Engaging alternative supplier	2010-01-01 13:26:14	2010-01-01 15:12:12	Procurement	1	2010	2010/1	1
2	2	3	Drafting contract notes re penalties etc.	2010-01-01 13:26:14	2010-01-01 13:41:51	Procurement	1	2010	2010/1	1
3	3	4	Engaging alternative supplier	2010-01-01 13:26:14	2010-01-01 15:00:58	Procurement	1	2010	2010/1	1
4	4	5	Drafting contract notes re penalties etc.	2010-01-01 13:26:14	2010-01-01 13:43:08	Procurement	1	2010	2010/1	1
...	...	...	...	...	...	...	...	...	...	...
1126	55372	55373	Checking reserved quotas (calling suppliers at...	2017-03-01 09:00:00	2017-03-01 10:12:46	Procurement	3	2017	2017/3	87
1127	55373	55374	Engaging alternative supplier	2017-03-01 10:12:46	2017-03-01 11:21:25	Procurement	3	2017	2017/3	87
1128	55374	55375	Drafting contract notes re penalties etc.	2017-03-01 10:12:46	2017-03-01 10:36:27	Procurement	3	2017	2017/3	87
1129	55375	55376	Engaging alternative supplier	2017-03-01 10:12:46	2017-03-01 11:43:30	Procurement	3	2017	2017/3	87
1130	55376	55377	Drafting contract notes re penalties etc.	2017-03-01 10:12:46	2017-03-01 10:26:05	Procurement	3	2017	2017/3	87

Figure 36 Final DataFame with populated “case_id” column

After exporting our modified data set, we were now ready to start mining process by using Disco. First, we compared the process which was created by Disco with the process description we obtained in our BPM course in order to find out whether the process really takes place as originally intended.

It needs to be mentioned that one of the 87 process instances only proceeded to the second level of activities and finished afterwards. Since this is logically not possible, we will not consider this process case in our further analysis. The process description stated that in 15 % of all cases suppliers cannot provide the quotas that SUPER-X agreed on with them. By filtering for the activities “Drafting contract notes re penalties etc.” and “Engaging alternative supplier”, we found out that in reality this is the case in 81 out of 86 cases which equals around 94 instead of 15 %. Another discrepancy could be found for the activity “Undertaking a reminder call to Sales”. While this is, according to the process description, necessary in 20 % of all cases, our mined process showed that in reality it was only done 6 out 86 times which is roughly 6 %. Apart from incorrect probability distributions, our analysis also yielded the occurrence of a rework-activity which was not specified at all in the process description. In

almost all (85 out of 86) cases the activity “Transferring data from Production-PDF to Procurement-PSD” was executed twice – before and after the planning procedure was conducted. However, the process description only mentioned this activity before the latter. On top of that, we found that in every process instance we examined, the activity (“Calculating negative impacts, if no alternatives can be found”) was executed. This means that SUPER-X encountered material shortfalls in every single production run.

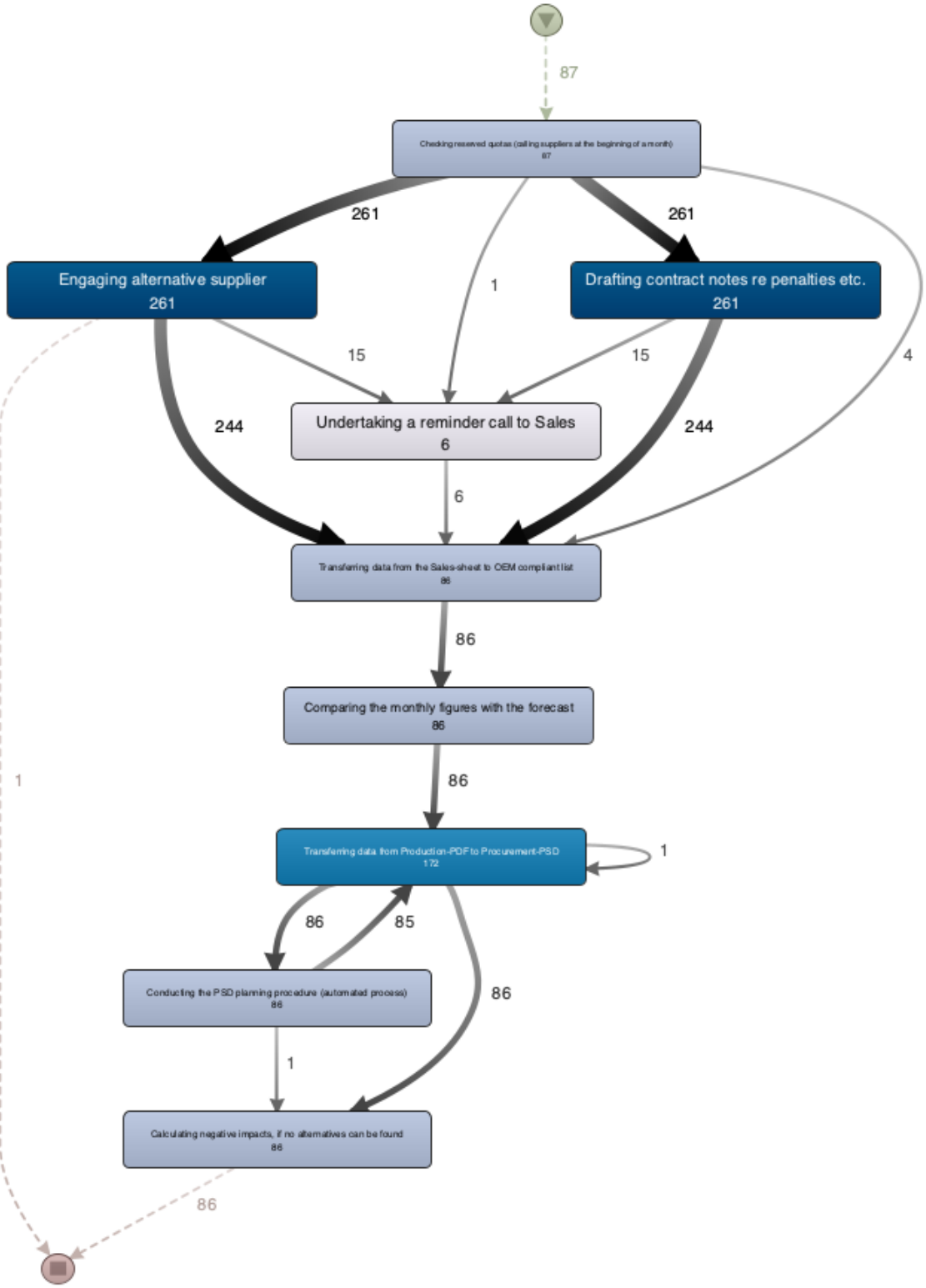

Figure 37 Mined process through application of Disco (shows all possible paths)

After analyzing differences between process description and the process based on event logs, we looked at the general statistics of the process. The processes start on the 01.01.2010 and end with the 01.03.2017. The mean duration for a case is 23.1 days, which fits the process description as well, as it is described that procurements process is handled monthly. As already mentioned, case ID 87 only takes up to 2 hours and 43 minutes and therefore not fully describes the process. For this reason, we did not consider this outlier further in our analysis. The three cases with the highest duration of 25 days and 3 hours are case 70, 84 and 4, which will be discussed later in this paragraph. Meanwhile, the fastest case (besides 87) is case 82 with a duration of 22 days and 39 minutes. As we can see from this analysis, the time in general does not differ that much for all the cases.

Besides that, disco shows also the likelihood of a certain variants of the process. For our 14 variants a different number of cases was identified. For example, the most likely variant with almost 20% consists of 13 activities while variant 12, which is in comparison unlikely with a chance of 1.15% to occur, has 21 events.

Activity	▲ Frequency	Relative frequency
Drafting contract notes re penalties etc.	261	23.08 %
Engaging alternative supplier	261	23.08 %
Transferring data from Production-PDF to Pro..	172	15.21 %
Checking reserved quotas (calling suppliers at .	87	7.69 %
Transferring data from the Sales-sheet to OE...	86	7.6 %
Comparing the monthly figures with the foreca..	86	7.6 %
Conducting the PSD planning procedure (auto..	86	7.6 %
Calculating negative impacts, if no alternatives .	86	7.6 %
Undertaking a reminder call to Sales	6	0.53 %

Figure 38 Process Activities

As shown in the above figure, with disco it is possible to see that certain activities take place more often than others. The most frequent activities are “Drafting contract notes re penalties etc.” and “Engaging alternative supplier”. In contrast, “Undertaking a reminder call to Sales” is the least frequent activity.

“Drafting contract notes re penalties etc.” is observable in 82 of the cases.
This activity has a mean duration of 17.7 minutes. Looking at the statistics, we can see that not just one person is actively working on that task, but all the 28 employees.

“Engaging alternative supplier” is also done by all 28 employees. This activity has a mean duration of 80.8 minutes and that this activity is also done in 82 of the cases.

Besides that, we also investigated the least frequent activity, which is “Undertaking a reminder call to Sales”. This activity occurred in 6 cases. As we can see from the previous figure in general this task was done 6 times, which indicates that this task is always done just once per case. The mean duration for this task is 13 minutes and is done for each case by a different employee.

The activity with the highest duration is “Checking reserved quotas (calling suppliers at the beginning of a month)”. It takes the 28 employees 3.4 hours on average to fulfill this task. But looking at the mean for different people we can see that the task per person varies around 1 hour on average. From the statistics column it is obvious that this task is almost equally distributed between the employees as well as for the previous described tasks.

To investigate further why certain cases took longer than others, we checked the durations and resources for the cases with the highest total duration. To see if a certain employee in general takes more time to complete certain tasks, we also investigated the statistics of this employee and the activity itself. The following table shows the result of this analysis.

Activity	Average duration of task for all employees	Cases	Employee ID	Average duration of task for specific employee	Actual duration for task
Checking reserved quotas (calling suppliers at the beginning of the month)	3 hours 21 minutes	4	99	3 hours 22 minutes	1 hours 19 minutes
		70	48	2 hours 18 minutes	3 hours 11 minutes
		84	34	3 hours 48 minutes	3 hours 33 minutes
Drafting contract notes re penalties etc .	17 minutes	4	61	16 minutes	9 minutes
			41	19 minutes	18 minutes
		70	51	16 minutes	13 minutes
			33	18 minutes	17 minutes
			61	16 minutes	30 minutes
		84	34	16 minutes	18 minutes
Engaging alternative supplier	1 hour and 20 minutes	4	47	1 hour 24 minutes	1 hour 11 minutes
			39	1 hour 40 minutes	1 hour 39 minutes
		70	22	1 hour 24 minutes	1 hour 6 minutes
			33	1 hour 25 minutes	1 hour 10 minutes
			37	1 hour 22 minutes	1 hour 31 minutes
		84	80	1 hour 20 minutes	1 hour 26 minutes
Transferring data from the Sales-sheet to OEM compliant list	38 minutes	4	102	47 minutes	47 minutes
		70	67	40 minutes	48 minutes
		84	22	31 minutes	14 minutes
Comparing monthly figures with the forecast	2 hours 47 minutes	4	110	2 hours 37 minutes	2 hours 59 minutes
		70	98	2 hours 50 minutes	2 hours 27 minutes
		84	119	3 hours 24 minutes	3 hours 17 minutes
Transferring data from Production-PDF to Procurement-PSD	20 minutes	4	102	21 minutes	18 minutes
			48	16 minutes	24 minutes
		70	24	19 minutes	33 minutes
			114	21 minutes	6 minutes
		84	99	17 minutes	21 minutes
			77	20 minutes	23 minutes

Conducting the PSD-planning procedure (automated process)	2 hours 49 minutes	4	99	2 hours 31 minutes	2 hours 28 minuites
		70	110	3 hours 33 minutes	2 hours 3 minutes
		84	48	3 hours 53 minutes	3 hours 54 minutes
Calculating negative impacts, if no alternatives can be found	2 hours 10 minutes	4	51	2 hours 15 minutes	3 hours 2 minutes
		70	5	2 hours 18 minutes	2 hours 50 minutes
		84	114	2 hours 47 minutes	3 hours 13 minutes

Table 3 Comparison of cases with highest duration

In general, we can see that "Comparing figures with the forecast", "Conducting the PSD-planning procedure (automated process)" and "Calculating negative impacts, if no alternatives can be found" are the activities that take up the most time; especially the last task. Additionally, many activities took longer than expected for these cases which are marked in red in the table. Some of these activities took longer than usual but just as a matter of minutes and therefore did not seem noteworthy enough to investigate these further.

While case 4 and 70 seem to have mainly issues within the middle of the process and at the last activity, case 84 had longer execution terms for every activity within the process. Also, we realized that the last task that includes the calculation of the negative impacts usually starts after 2 weeks for all cases, when the previous task was finished. This activity was finished with a delay of approximately 40 to 60 minutes for all the examined cases. All other cases had only taken up about 30 minutes longer than expected.

Furthermore, we can derive which employees are maybe not suitable for certain activities based on their average duration time for a certain activity and performance in these cases. For example, an employee like 114 should not do the activity "Calculating negative impacts, if no alternatives can be found" as it seems that this employee needs around 30 minutes more on average to finish the task than the other employees seen in the table.

6. Business recommendations for the management and project reflection

Throughout this project, we have concerned ourselves with various data-related topics of the SUPER-X environment. To this point, the performed tasks were by nature mostly practical. In this chapter, it is time to revisit each chapter in order to derive recommendations and advices that can be given to the SUPER-X management. To do so, we decided to divide the recommendations we came up with in three different categories depending on the addressed topic. In addition to that, we will also reflect on the general process of the project, our learnings, and the tools we used.

6.1 Business Requirements and KPI's

As stated in the first chapter it was not possible to integrate all the information that was given to create a data mart. For this project the focus was within the purchase and delivery of items and the corresponding suppliers for these. This information was seen as the most important for procurement's process. But since procurement also works with forecasts and an inventory to manage their demand and supply it is recommendable to also add them to the data mart. Again, it was already stated that this would need a further fact table in the data mart and would change the schema. This should be taken into consideration when working with the data mart in the future. Three possible business questions were already given in the first chapter and could be the base for further integration.

Also, it was possible for us to depict the specific KPI's from the first chapter. As the process is done on a monthly basis the dashboard should also be checked at this time frame. In general, the problem occurs that we do not know the priority of the objectives the SUPER-X company has. Based on this knowledge certain KPI's could be aligned with this goal to track whether the company works towards their aim. Based on the lack of information in this case, the seen KPI's were tailored to the initial business requirements in this project. Besides, the KPI's that were introduced to answer the requirement questions also further KPI's to measure procurement's performance could be used.

For example, in this project the top 10 suppliers were measured by the total cost spent on each supplier. To gain a clear view and add some dimensionality, it would also be of interest to measure how much of a discount the company gets for purchasing high quantities or if the company can rely on their deliveries. Of course, this was out of scope for this project and needed more data for the purchases. Therefore, it was not feasible for us but should be considered if the company's goal is to identify the top suppliers from whom they want to order from in the future. The same problem occurs when looking at the top ten materials. Also, the quality of the materials should then be considered.

Another point is, that we can see from the dashboard there are certain minima within the total costs per month. This might be something the company should pay attention to in order to analyze why these minima exist. Especially because they also occur within high seasons and cannot be traced back to the fact that just less materials were needed. As mentioned, it needs to be acknowledged that there seem to be seasonal differences for the duration. It takes the suppliers longer to ship the materials in summer months (June to September) and in the winter months (December, February and March). This should be considered when ordering in certain seasons. Especially, because the demand of possible customers might also change within these seasons and could lead to delays of deliveries and a bottleneck of materials and products that can be offered.

This also plays into the next KPI. We can identify three groups of materials, which vary in their duration till delivery. Group one takes 1,5 to 2 days to be delivered, group two varies around two and half days and the last group takes 3 days. It would be recommendable to adjust the time when to order certain materials based on this information. For example, a crucial part of the product like screws has the highest duration for the needed materials. These screws should be delivered within a timeframe so that production has the right materials if they need it and do not have waiting times that increase the duration of the whole process.

6.2 Operational database and data quality

From an IT perspective, there are a few changes that could be done in order to improve the effectiveness and quality of the stored data. We noticed that the same name can be inserted for several suppliers in the suppliers' table; if this is not intentional, then it would be a good idea to alter the table and assign a unique constraint to the name column to avoid duplicates. In general, a few constraints could be defined to enhance the quality of the data, like creating an enumerated type for the supplier's category, in order to prevent categories with spelling mistakes. Another constraint that could be considered is not allowing symbols in addresses, like asterisks, percentage signs, dollar signs, etc. An additional solution could be, instead of defining constraints in the database table structure, to improve the interfaces that insert new records to the database. As an improvement, we would consider implementing spelling checks on the interface and using drop-down selectors to minimize the mistakes and even the typing performed by the workers.

Another issue that you might want to pay attention to is the inconsistency between purchase orders and purchase order items. Like we mentioned before in the data quality chapter, there are too many purchase orders that do not have any items, therefore it looks like there is data missing. The same phenomenon occurs between the deliveries and the delivery items, there are too many deliveries without any items. This should be something to be aware of and try to fix. Another situation that we do not recommend is having purchase order items in CSV files, as this information should be in a centralized non-volatile infrastructure like the database to prevent any data loss or corruption. If there is not a way to insert the data immediately into the database, this should be done as soon as possible, maybe even with some sort of ETL tool.

Now, just as general recommendations that might improve certain aspects of the database is by having a not null constraint for the currency column in the purchase order items, which is something quite important for the transaction. Although the currency can be obtained by the supplier, it would be smarter to not have nulls there. Another area of improvement could be to add a purchase id to the delivery for better tracking, so you can know which purchase relates to a specific delivery. Finally, to enrich and facilitate the process mining procedure that can be fulfilled with the event logs from the database, it would be good to include the case id for each process execution, so it is easier and more efficient for the people in charge of this activity.

These are some of the possible areas of improvement that we noticed during our project execution. The solution for these cases could be implemented to increase the efficiency and effectiveness of the IT related processes as well as for any other data warehousing that might be implemented in the future.

6.3 Business process

Before one can start to write about recommendations based on the mined process, it has to be mentioned that the actual mining process could be massively facilitated by introducing a case ID column to the table "event logs". As described in chapter five, it required an additional step of Python code to get from this table to a file which can be used to perform process

mining. Since the latter is not an activity which a company does ones and then never again, it is recommendable to start tracking the process instances beforehand.
In chapter five, it was mentioned that one of the 87 process instances only proceeded to the second level of activities and finished right after. Since from a process logic point of view, this is not possible, it can be assumed that this case was not documented correctly and therefore negatively affects the event log's data quality. To avoid such cases and their undesirable consequences in the future, we recommend SUPER-X to put a mechanism in place that recognizes such irrational process instances so that they can be investigated and resolved.
Secondly, we discovered that at the moment in 94 % of the cases SUPER-X has to deal with suppliers that cannot meet the reserved quotas. That results on average in 98.5 minutes of additional work per process instance for engaging with alternative suppliers and drafting contract notes. Thus, it is advisable to SUPER-X to work on their supplier relationships in a way that they become more reliable. If this does not yield a reduction of these 94 %, it might be reasonable to consider switching suppliers.

Thirdly, we found that in the mined process there was a surprisingly high amount of rework identified for the activity "Transferring data from Production-PDF to Procurement-PSD". For us, it is not evident why this activity is performed twice in almost all process instances and thus we recommend SUPER-X to examine this problem.
Lastly, in terms of mined versus designed process, we want to address the very last activity - "Calculating negative impacts, if no alternatives can be found". The fact that this activity is performed in every single process instance means that SUPER-X loses money due to unsatisfied material demand. The reason for this happening so frequently can either be found in poor stock management or, again, poor supplier relationship management. For the latter we have already mentioned the necessity to improve and for the former, we want to encourage SUPER-X to examine whether it would make sense to opt for higher stock levels in order to avoid material shortages.

In chapter five, we identified that the net processing time which is the productive time where actual value is created is rather short with around 0.6 days relative to the total processing time with 23.1 days per process instance.

That implies that 22.5 days in the process are waiting and set up times. Since this metric is very process specific, we cannot evaluate, based on the information we received, whether this is reasonable or not. However, it would make sense for SUPER-X to investigate whether the total processing time can be reduced by reducing waiting and set up times.

6.4 Project Reflection

In the very last chapter of this report, we will now look back on the past couple of weeks and reflect on how this project went. Our reflection will be covering both our learnings in regards of hard skills as well as the general process of the project and how we interacted as a group.

We started things off in the first meeting with presenting the business requirements that we all independently brainstormed before. While it was straight-forward to filter for the most important ones that are satisfiable with the data we have, we, at the time, did not pay any attention to which data source might be the underlying one for the respective business requirements. This became an issue for us at a later stage because we realized that our

business requirements cannot all be satisfied with one fact table. Since the latter was important us, we had to partly redefine our business requirements. For this we used the templates known from the classes in order to structure our requirements. Putting our ideas together in these two templates made it easier for everyone to access our final requirements and KPI's for further tasks if needed.

In our data quality analysis, we found various issues in our data. The problem here was not finding them but making a decision on how to eliminate them. Two supplier records where everything but the name is the same is undoubtedly a duplicate. However, what if the name of two supplier records is identical but the address or the country isn't? Is it a second subsidiary or a duplicate? Questions that are hard to answer without any internal knowledge about SUPER-X and their suppliers and thus, they represented a challenge for us. To evaluate data quality every group member used Talend. For the task we applied our knowledge from the data quality homework resulting in an analysis of the schema, matching patterns, column analysis and cross table redundancy analysis. We did not use any further analysis that Talend provides as we had no experience with them and therefore, we only covered the familiar analyses. But overall, we managed to get an understanding of the data and what tables or columns needed further processing before adding them to the data mart. Thus, Talend was a helpful tool to start the process and to find quality issues that needed to be eliminated in the upcoming ETL process.

Another issue we encountered during the design of the logical model was the defining of the Slowly Changing Dimension types for each dimension. We figured out that, logically, we need to have two different date columns to form the primary key of our fact table. Luckily, our initial concerns that this violates data mart conventions, turned out to be wrong.

For our multi-dimensional design and implementation, we used Google Drawings and Power Architect. The results were depicted in chapter 3 and 4. Again, these tools were already introduced in class which let us save time as we were already familiar with the procedure of designing a multi-dimensional model and creating the related star schema in Power Architect in order to connect it to a database. Because of this we were able to work through these process steps fluently besides the fact that we had some concerns regarding the violation of data mart conventions.

The ETL process was conducted with Pentaho. The most interesting part for us was the transformation of the data within the process. For example, we were able to identify even more quality issues that were not covered by Talend in the first place. As described already in chapter 4.2 we found that in the supplier table there was "Germany" and "Deutschland" while all other countries were only visible in English. We were not able to find this kind of quality issues with Talend, probably because these two strings are not relatable enough and were therefore not found by the matching pattern analysis.

In general, the ETL process seemed overwhelming at first as we were not able to simply apply what we learned in class but also add our own steps to the process. As an example, we were struggling with the fact that there were some suppliers that had similar names. We wanted to apply a regular expression that generally identifies duplicates (similar addresses and names) and keeps only the latest version of that supplier. Unfortunately, we were not able to create such an expression and had to manually look up the supplier id's and delete the oldest entries

directly within the SQL statement of the input table. In the end, we were still happy to manage eliminating the quality issues well although we wished we would have found a more sophisticated approach.

Also, for the creation of the dashboard with tableau there were no problems at all although it required the creativity to visualize the important KPI's. As already mentioned in chapter 4, there were some problems with the installation of Power BI and connection to the AWS database in the beginning that took many hours to resolve. But besides these issues both programs fulfilled their job of presenting the answers to our business questions entirely. Tableau in this case might have been a slightly better choice as it contains more powerful visualizations.

A, apparently, common challenge in this project is deriving the case ID of the event log column for the process mining task of this project. Fortunately, it did not take us too long to figure a way out how to do that. Our increasing knowledge through several courses within the BIPM program allowed us to simply form the case id within python. While struggling at first with the execution of our idea, we could manage eventually to write a working code snippet that presented us the case id correctly. From there on we simply applied what we have learned about process mining in class and added the saved CSV file of the data with the case id to disco.

We were hoping to find some interesting insights, but unfortunately, we could not identify any. For example, we did not find any bottlenecks with disco as the time span for the animation was too long to capture where certain activities hold up further tasks, although we already used the highest speed for the animation. Because of that, we worked around the problem by applying filters for the cases that took the longest and investigated the statistics of each task and each employee that executes that task. From this, we could derive some implications why the process takes so long but these were vague. Generally, working with disco is easy to learn. The only thing that we did not like as much is the filtering. Some students seemed to have some troubles at first to understand how these filters are applied and what results they get after applying. Especially, when different filters are needed to be added together in order to get the findings a student is looking for or when filters of the same category as for example for different 'resources' need to be applied.

However, it is likely that this task would have taken us longer without having a teammate with prior knowledge about the business process from the Business Process Management module. Ensuring the latter when forming the groups for this project, could maybe help the next BIPM cohorts with this challenge too.

Generally, we really enjoyed this group project since it was perfectly balanced between practical and theoretical work. After having an introduction into all necessary tools and technologies in class, it was a great opportunity to deepen and strengthen the skills gained in the prior exercises. Since the instructions in our homework are usually really detailed, it helped a lot to apply the things learned on an independent use case. The only thing that confused us and what could maybe be improved, is the phrasing of the task 4.4. - "Evaluation and comparison of the used data warehousing technologies". For us, it was not clear that we are supposed to use another Data Warehousing technology which was not covered in class in order to compare it with one that we already know.

7. Appendix: Main Responsibilities in the Project

Task	Sub-task	Person Responsible
1. Analyze the business requirements and identify important KPIs		Annelie Schridde
		Lucas Silbernagel
		Joshua Campos
2. Identify relevant data sources and analyze data quality		Annelie Schridde
		Lucas Silbernagel
		Joshua Campos
3. Create the conceptual model/design		Annelie Schridde
		Lucas Silbernagel
4. Implement proof of concept of the multi-dimensional design	Implement the model/design of the data mart	Joshua Campos
	Design and execute the ETL process	Annelie Schridde
		Lucas Silbernagel
		Joshua Campos
	Create dashboards to visualize KPIs	Joshua Campos
	Evaluate and compare the technologies	Joshua Campos
5. Perform the process mining on the event logs		Annelie Schridde
		Lucas Silbernagel
6. Create business recommendations and project reflection	Write business recommendations	Annelie Schridde
		Lucas Silbernagel
		Joshua Campos
	Write project reflection	Annelie Schridde
		Lucas Silbernagel
7. Write report		Annelie Schridde
		Lucas Silbernagel
		Joshua Campos
8. Create presentation		Annelie Schridde
		Lucas Silbernagel
		Joshua Campos