



SAP® Analytics Cloud



## The Flexible Data Warehouse - Solution Handbook

by Michael Muell

# Table of Content

- The Flexible Data Warehouse - Solution Handbook
- Table of Content
- Introduction & Remarks
  - Important
  - Chapter1:
  - Chapter2:
  - Chapter3:
- Chapter1: The Framework for Business Process Improvement
  - The dashboard that shows just numbers misses the point.
  - The main job of the middle manager is process optimization.
  - IDENTIFY / Is There a Problem?
  - ANALYZE / What Caused the Problem?
  - ACT / Which Actions Should We Take?
  - CHECK / Were the actions successful?
  - How to improve even more!
- Chapter2: Designing an IT system to support Business Process Improvement optimally
  - Current System Landscape and its Users
    - Operational Reporting:
    - Core KPI Layer
    - Agile KPI Layer
  - The Gap to Close
  - The best practice solution architecture
    - Changeability & Extendability by Business
    - Short Development Cycles
- Chapter3: Implementation of the flexible data Warehouse
  - Systems
    - Data Factory
    - SQL Server
    - SAC Fileserver:
    - SAC:
  - Overall Data Flow
    - Data Source
      - SAP R3 Tables
      - External Excel Files
      - BW Reports
      - EFLOW
  - Data Factory
    - Datasets
    - Dataflows
      - General Data Flow logic
    - Pipelines
      - Ingest Pipelines
      - Transformation Pipeline
      - Output Pipeline
      - Output Pipeline Schema
    - Transport Changes from D-System to P-System
  - SQL Server
    - Key requirements for transformation stage:
    - Coding rules :
    - Naming Conventions
      - Tables
      - Stored Procedures

- Scalar Functions and ETL
  - Schema
  - Execution from Datafactory
- CLEANED with ETL functions
  - p\_cln\_first
  - Config ETL Functions
  - CLN\_FI1000
- Transform 1 and Transform 2
  - O2C
  - P2P
  - TP1\_EKBE
- Transform 3
  - P\_TP30\_ALL\_ITEMS
    - Key Date
    - Recalculate Due Date and Arrears after Net
  - P\_TP3\_IRB
  - P\_TP30\_ALL\_CUST\_ITEMS
    - Recalculate Due Days based on the VAT issue date
    - Include only specific documents for payment behavior calculation
- Stage
  - STA\_ALL\_ITEMS
  - STA\_OPEN\_ITEMS\_MONTHLY
  - STA\_IRB\_FULL
  - STA\_IRB\_MONTHLY
  - STA\_ALL\_CUST\_ITEMS
  - STA\_OPEN\_CUST\_ITEMS
  - STA\_EFLOW\_CLR
  - STA\_EFLOW\_LIKP
  - STA\_FI1000
  - STA\_PAYMENT\_BEHAVIOR
  - STA\_\*SCHEMA
- Transport Changes from D-System to P-System
- SAP Analytics Cloud
  - File Server
  - SAC Models
    - Model Settings
    - Story Overview
    - P2P Stories
- Monitoring
  - Data Flow
  - Flow Date Tracking
    - Export date from EP1
    - Import data from EP1 and EFLOW to the data lake
    - Running Pipeline in Data Factory
    - Importing data to SAC
  - Compare Q-D System
  - Pipeline runs
- Frequent Questions / Issues / Requests
  - Requests
    - We (Business) want to add a new column to model X
    - Add new BW cube to Solution Space
- Issues
  - We (Business) want to add a new calculated dimension, but in SAC, we get the following error
  - Business thinks that the data is wrong

# Introduction & Remarks

The following whitepaper of the flexible data warehouse is a documentation of the business background (chapter 1), implementation concepts(chapter 2), and the actual implementation (chapter 3) of a project to gain insights into the business processes of a Shared Service Center. The solution enables effective process improvement, inevitably leading to cost savings that the automation KPI can track.

## Important

The knowledge that led to this document was not accumulated step by step in a waterfall fashion like it is written down, but rather through many iterations of try & error.

We wrote down the principles contained in chapter2 AFTER actually implementing them implicitly through many iterations. That makes them more valuable as they are tested and seem to work great in a real-world application.

## Chapter1:

Explains the use case from a business perspective. It is the summary of many discussions with the company's finance department and lays the foundation for the solution design.

Especially if you're an IT guy like me, this is a MUST read. Because we are fascinated by technology, we are all too often excited by flashy charts, AI analysis, and Demo videos.

**It is crucial to understand that just because something is technically possible does not mean it is valuable.**

The chapter shines a light on how business sees things.

## Chapter2:

This chapter builds on top of the first, discusses the necessity of a different approach for the agile KPI layer, and presents a working solution for its implementation. The chapter highlights the chosen architecture and outlines the benefits of the classic data warehouse and operational reporting.

## Chapter3:

This chapter is proof that this document is not just a consulting company slide-(shit)-show.

Through the last year, we implemented the flexible data warehouse described in chapter two. The chapter has two purposes. Firstly, be a maintenance guide to quickly lookup errors once they occur. Secondly, it shows the exact implementation steps and our thinking behind it to give a starting point for new use cases or extensions.

The coding samples do not include all coding but only the building blocks to explain how you can implement these principles. The code does not claim to be perfect but functional.

# Chapter1: The Framework for Business Process Improvement

## The dashboard that shows just numbers misses the point.

The first thought that comes to mind when thinking about Self Service BI is a beautiful dashboard that shows the critical business numbers while giving you an excellent overview of everything you need to know and how those numbers perform over a given time frame.

How could this not be helpful?

Well, let's think about what the average middle manager that will use the tool later needs.

You might ask why only middle managers would use these tools. The simple answer is that if you had to prepare a C-level dashboard, it would be too big to consume.

It's the Controller's job to take and prepare presentations that point out the critical numbers on a given moment and prepare presentations that highlight their specific relevance. Designing an automated Dashboard fails as the story that the data needs to tell can vary significantly in a given circumstance.

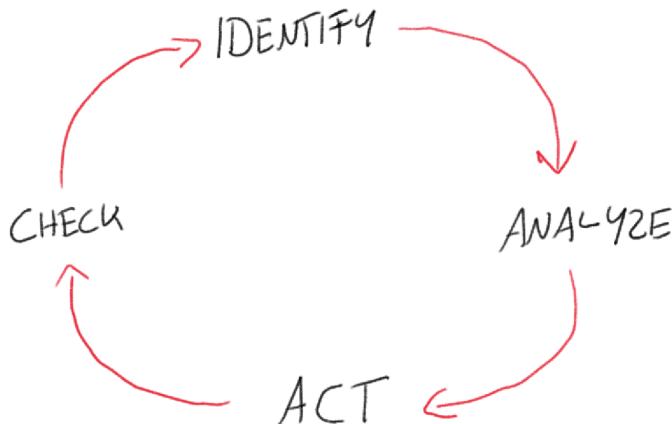
In contrast, it seems useful that a Controller uses BI tools to make sense of the data before preparing a C-level presentation. With augmented analytics, a machine might do this prioritization of abnormal data in the future. However, the future is not now. At least not in your average industrial company.

Actually, the topic is quite big and deserves a separate post. For this article's sake, let's agree that the middle manager is the main target for Self Service BI tools.

## The main job of the middle manager is process optimization.

Process optimization is a multi-step process. Depending on your source, the number of steps and their names might vary. Generally speaking, it's the following:

1. Identify that there is a problem
2. Analyze the root-cause
3. Define actions and execute them
4. Check if the actions were successful



The managers' success is measured mainly by this task. That is correct in most cases as he is usually receiving a bonus based on how well the KPIs under his responsibility perform over a given period. Our goal is to design an information system to support him with this task. The best way to do that is to help him answer the main questions related to each step based on the data we have:

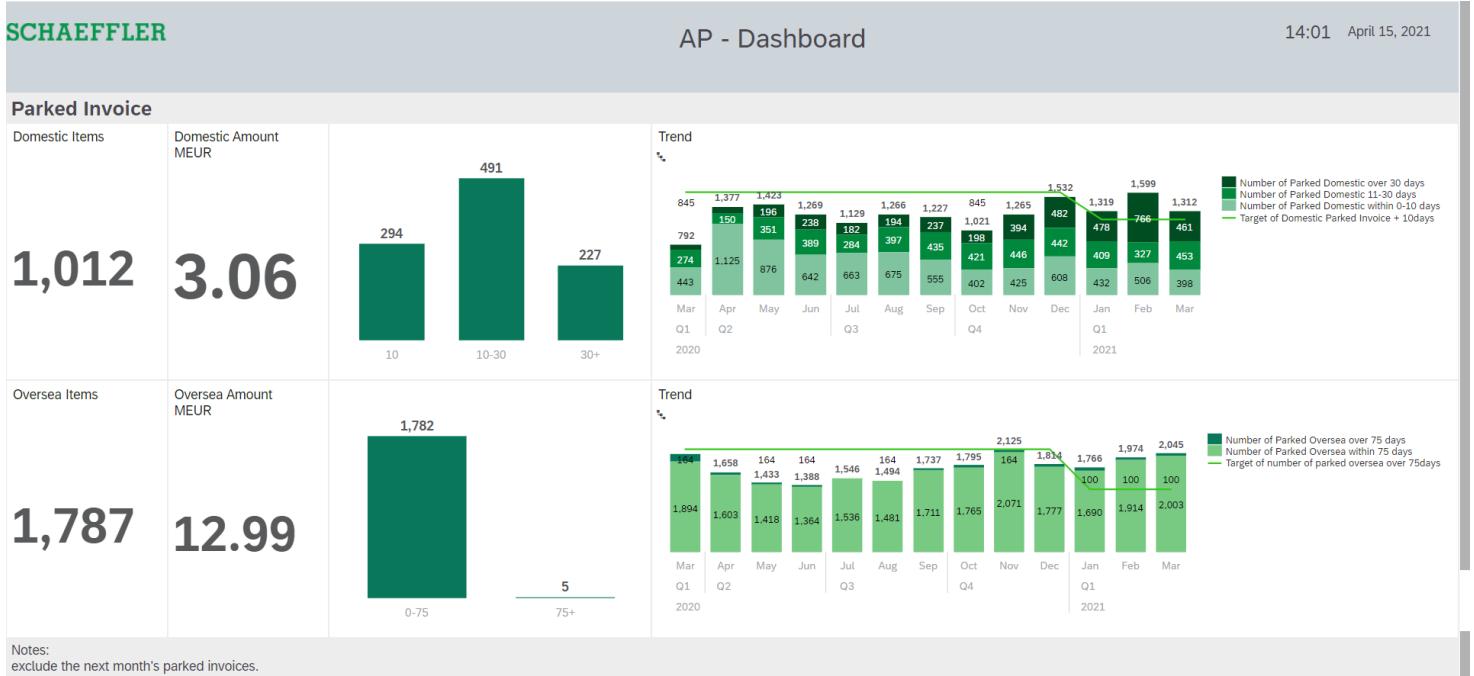
- Is there a problem?
- What caused the problem?
- Which actions should we take?

- Were the actions successful?

From looking at the questions above, you can already tell that the dashboard described at the beginning of this article only helps answer the first question. Most of the value that an automated analytics solution could have is left out.

Let's have a look at how a more sophisticated solution could answer these questions. I took the screenshots from one of the actual dashboards we implemented at my company. To make it readable at this page layout, I included only the essential parts (lego bricks) of each dashboard part to get the point across. The numbers are anonymized.

## IDENTIFY / Is There a Problem?



In this screenshot, you can see the classic dashboard that usually gets printed on marketing materials. The actual numbers of last period's KPI were placed on the left. Below you can see the trend over the previous months. If the KPI crosses a certain threshold, it turns green/red to indicate whether the result got better or worse than last month. For each KPI, you can jump to the root cause analysis by double-clicking.

## ANALYZE / What Caused the Problem?

The main idea is that a bad result is, in most cases, not caused by the average. Most of the time, outliers drag down the overall result. Consequently, showing a top-level KPI without quickly allowing for root cause analysis leads to ineffective actions as the vast majority of dimension members is not a problem. Problems often cluster around a particular attribute of a dimension that all members have in common.

For example, Suppliers based in Hong Kong perform worse than suppliers from other countries. It's easy to then drill down into suppliers from Hong Kong.

## ANALYSIS

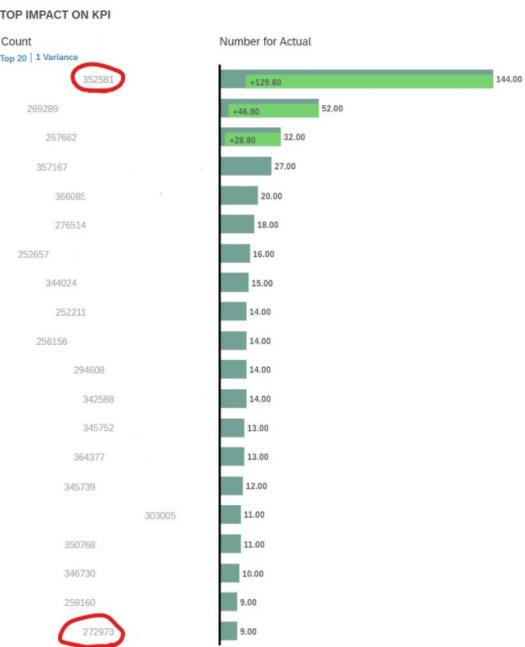


By spreading out the KPI along critical dimensions and attributes, the manager can quickly narrow down the possible causes.

## ACT / Which Actions Should We Take?

As the time available to the team for making improvements is limited, it's essential to prioritize effective actions. To find the most effective actions, we need to first think about which actions the manager can take and which dimensions will be influenced. Most of the time, it turns out to be not that many, so it's even more important to know them. Examples might be his direct reports, specific system settings, and vendors or customers.

In the analysis section, we already discussed the importance of outliers. Outliers are exponentially more beneficial to work on as the same action is usually required to improve both an outlier and an average member, while the potential increase of the overall KPI is much greater with the outlier.



The graph above shows the top customers that had the worst impact on the KPI over the last period. Most of the time, we can calculate this by the overall KPI of the customer times the transaction/activity frequency. E.g., A frequent customer performing bad will drag down the overall KPI more than a one time customer.

When we compare this potential for improvement for each customer, we can see that customer 352581 has much more potential than customer 272973. Assuming the same amount of work is required to improve each customer, it's a much better choice to work on customer 352581.

This shows that the impact we can have with our actions is more determined by which customers we chose in the first place than how well we carry out the activities.

Assuming we have time to work on the top 3 customers each month, we can now simulate the overall KPI outcome. You can see the second green bar on the leading three suppliers in image 4–1. Improving these customers with 90% confidence would result in below improvement on the overall KPI.

**95.87% (+1.26)**  
△ % KPI

image 4–2 by the author



Confidentiality and customer choice can be adjusted without additional effort, which allows the manager to simulate possible actions, formulate the best measures, and set a realistic target for the next period.

Including the managers' expert business knowledge in this step is essential as specific customers will have more or less business in the future, or some customers might be harder to handle than others.

## CHECK / Were the actions successful?

## TARGET

in % | 1 Filter

**54%**

SET

## ACTUAL

1 Filter

**52.05%**

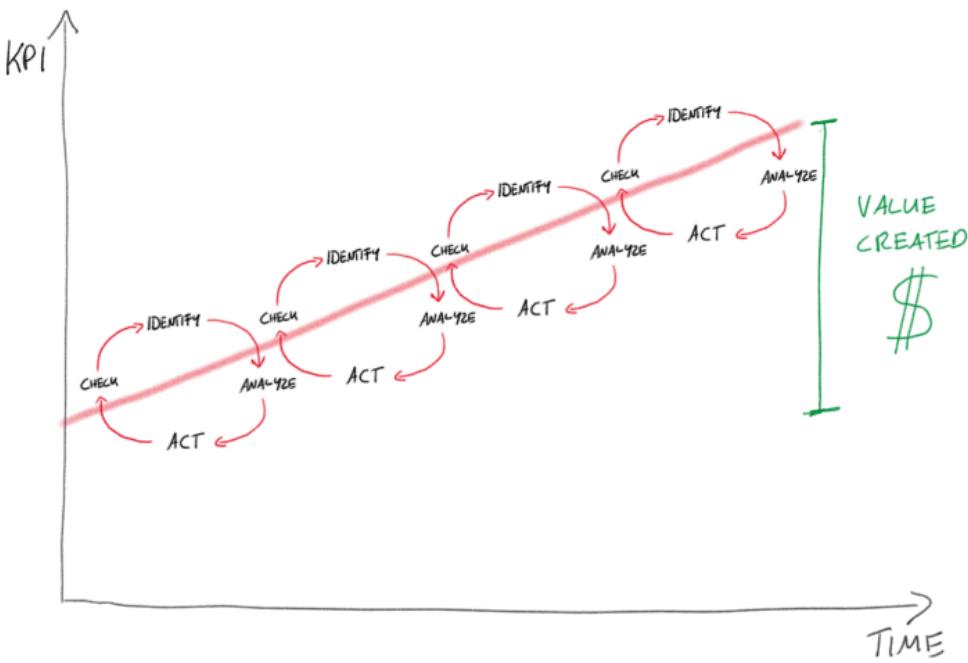
% KPI

Setting a goal to achieve for each period helps the manager understand if the defined actions were successful. It's a step often overlooked but extremely important because it shows the improvement over time and, therefore, the team's work.

In a real-world scenario, the chart never increases steadily but looks more like the stock market's ups and downs. As with the stock market, if the trend goes up, the actions are successful. If decreases can be explained (e.g., CoVid19), everything is fine.



The chart shows the target SET for each month compared to the actual figures.



You can think of the improvement process as a circle that continually iterates, moving upwards in the best case.

Similarly, the dashboard's value can be judged by the value difference between the introduction and the current period over time.

However, it's essential to understand that the actions taken and not the dashboard itself are responsible for the improvement.

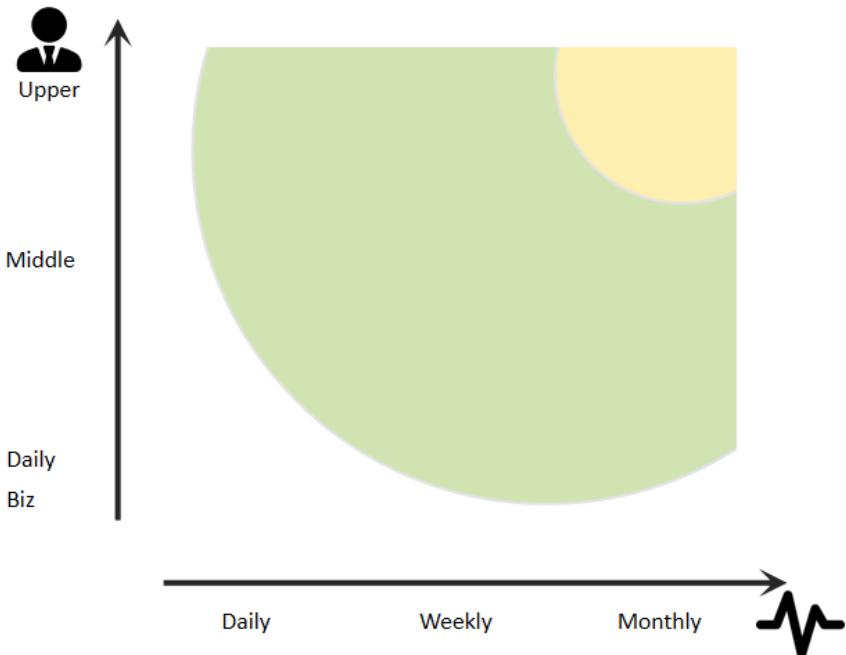
## How to improve even more!

As there is no perfect solution, there is one feature missing I would hope SAP would implement in the future:

All actions coming from the insights must be captured and followed up outside the tool. I would love to see integrated task management or direct integration to a tool like Trello that helps create and follow up on tasks derived from insights.

# Chapter2: Designing an IT system to support Business Process Improvement optimally

## Current System Landscape and its Users



The image above shows in a very simple way the user groups that consume reporting (y) and the frequency reporting is needed by them (x). Each group and frequency has different requirements on relevant information and the way to consume it.

### Operational Reporting:

Operational reporting is the reporting layer used by specialists on a day-to-day basis to verify the daily operations and smoothness of processes in the systems. Questions that a reporting system in this layer needs to answer, e.g.:

1. Were all invoices posted?
2. Have all goods been sent out to the customer?
3. Are there any processing errors?

The focus for this kind of reporting lies in the actuality of the data. This layer is also the only kind of reporting that justifies the struggle to have a real-time view of processes on the system.

Gladly, daily reporting does not have deep analysis requirements as time is usually too short to spend much of the day on analysis. Therefore a quick reporting is generally set up in the system where the specialist acts.

Summary Operational Reporting:

- Fast
- Easy
- Source System capability

### Core KPI Layer

The Core KPI Layer (yellow) contains the top KPIs that are necessary to run the company. These KPIs are crucial and must be comparable over long periods. Decisions made at this level are strategic and take time to show their impact on the actual data.

A system that produces these figures must be stable, reliable, and able to deal with vast amounts of data. These requirements cause the system to have long development cycles and data load restrictions due to the massive data.

Traditionally most companies have a sophisticated data warehouse solution that deals with the core KPI layer requirements. As described in chapter 1, it's usually the Controllers Job to find and prepare the critical data points in a meaningful way and present them to the top management.

Summary Core KPI Layer:

- Stable
- Reliable
- Long Development Cycles
- Huge Volumes of Data
- Business Warehouse system and Controllers job

## Agile KPI Layer

The Agile KPI Layer(green) contains all the error analysis, reporting, and nitty-gritty that the middle manager needs to run the business under his responsibility. The insights that a middle manager wants to derive from data are more in the short term but need to be more flexible than top management requirements, and data volumes needed are smaller but need to be more detailed.

Traditionally companies rely on the swiss army knife of software: Excel.

This flexibility is paid for by much manual work, errors, and shallow analysis.

Summary Agile KPI Layer:

- Flexible
- Short development Cycles
- Small / Midsize volumes of data
- Excel

## The Gap to Close

We see that from the three reporting layers, the agile KPI layer is the one with the least software support. The lack of software support is mainly because this space is the newest. Previously Excel and operational reports were just enough to handle those requirements. However, the increase in data collection and the decentralization of tasks in a Shared Service Center make it less and less feasible to handle these tasks in Excel.

Furthermore, computing and storage costs have decreased tremendously over the last years, turning the cost-benefit analysis in favor of an automated solution in many more cases. These developments lead to a widening gap that is not adapted by many companies today.

## The best practice solution architecture

The first thought of just increasing Operational Reporting capability or building more Data Warehouse reports falls short, looking at the very different nature of the agile KPI layer.

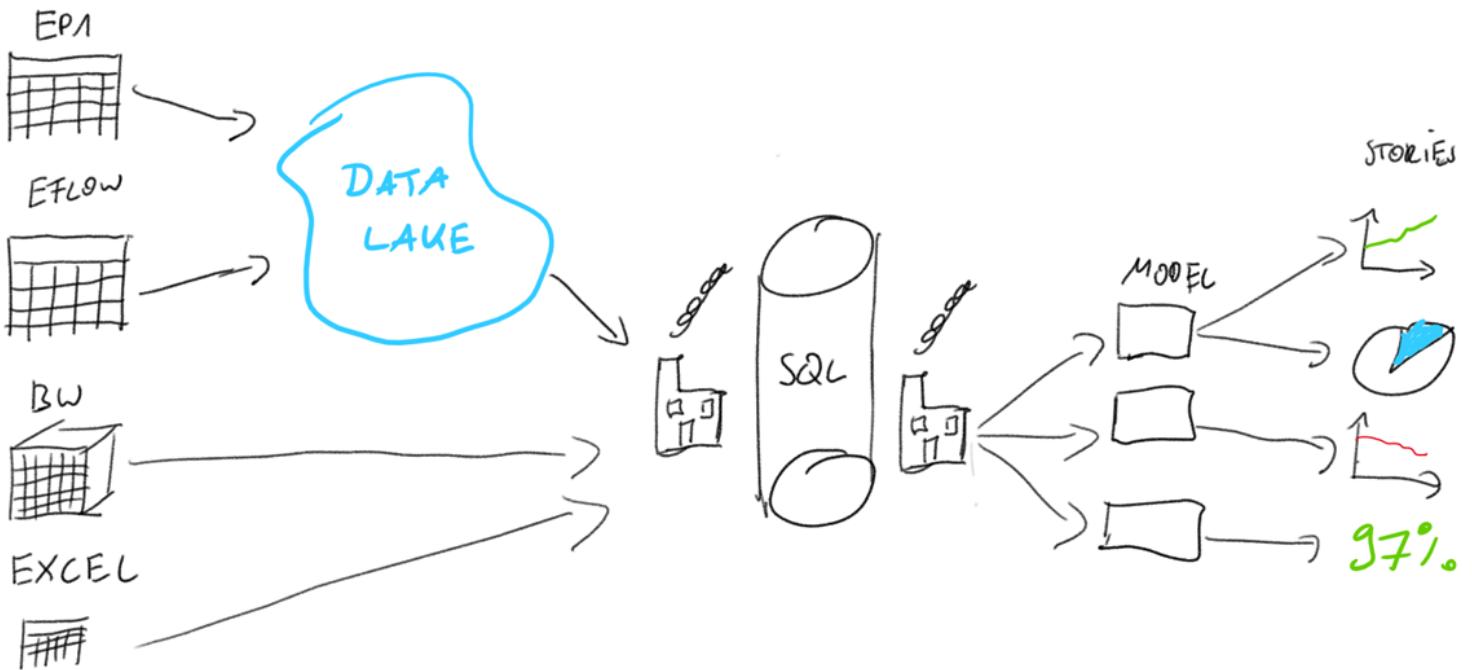
The show stopper for integrating the agile requirements into the core KPI layer is the long development cycles and IT effort involved.

The operational Reporting falls short in the source systems capabilities for advanced analysis. Additionally, a source system reporting only allows insight into the data that that specific source system holds. Often insights are gained by combining data from many data sources along the process.

We tried only using Self Service BI tools (SAP Analytics Cloud or Power BI) to fulfill the requirements in our first rotation of development. While those tools have very impressive calculation and presentation capabilities, they lack the modeling and automation ability to handle dataset joins that lead to result tables in the size of 1M rows and 100 columns.

In our experience, after around 25M cells, it gets messy. Calculations have long run times or fail. Joins with more than five tables are hard to work with at any size.

This lead to frustration and confusion for our business department and us into adapting the below approach.

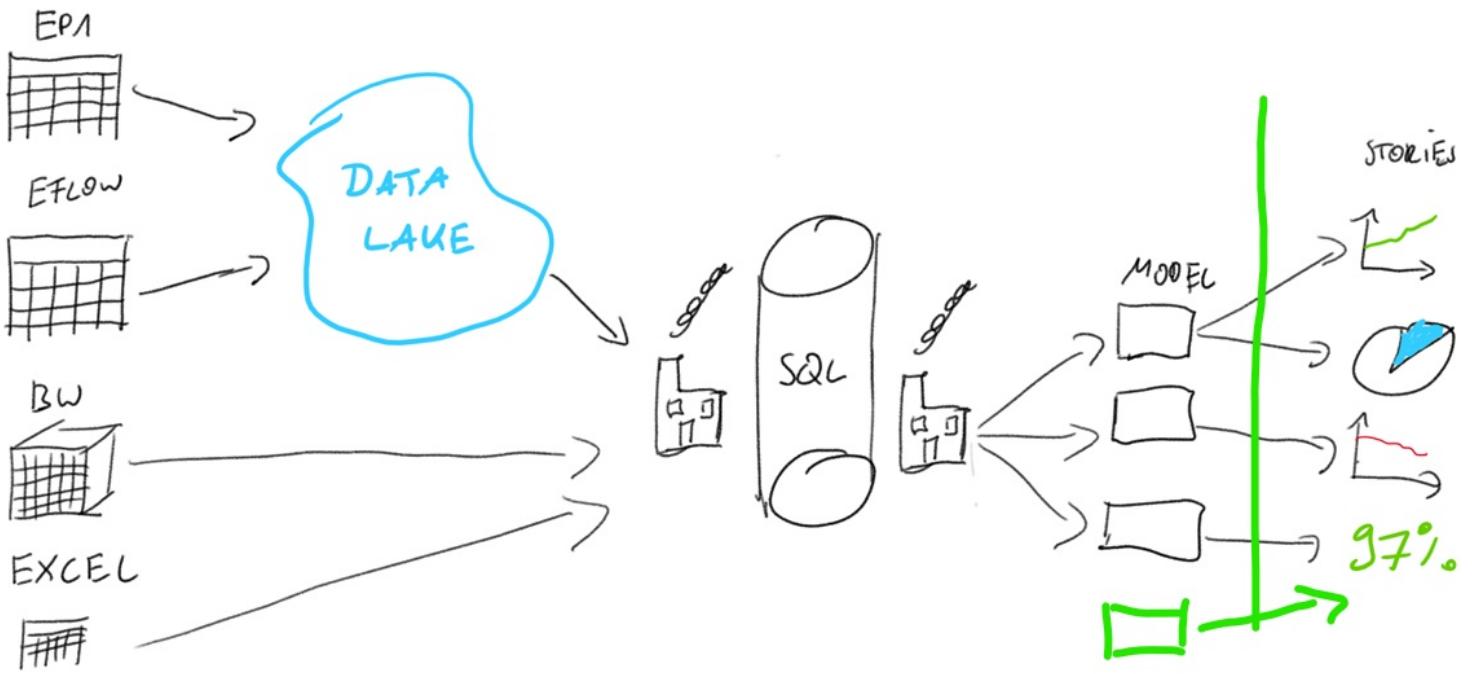


The image shows the final architecture of the flexible warehouse.

1. Data is initially extracted from many different source systems. In our case, this is:
  - an SAP R3 System (EP1)
  - an SAP BW system that sits on top of the SAP R3 System
  - an external Business Workflow system called EFLOW
  - additional Excel files that contain additional business logic or report downloads from other software that is not yet automatically connected.
2. Data is extracted from the SAP EP1 system and EFLOW system weekly into the company data lake.
3. We can access the data from the data lake, BW, and Excel with the help of the Microsoft Azure Data Factory.
4. A data warehouse is set up in the Azure SQL Server that executes ETL functions and adds business logic where necessary.
5. The result of this transformation is data model tables that BI tools like SAP Analytics Cloud or Microsoft Power BI can access.

Let's look at the advantages and trade-offs that this architecture makes to make it a much better fit for the agile KPI Layer and fulfill the requirements for agility while providing automation that allows for deep root cause analysis.

## Changeability & Extendability by Business

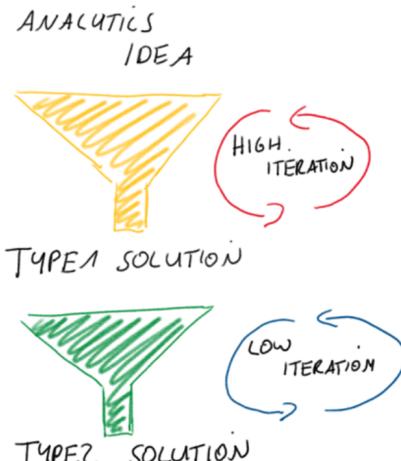


To achieve that, we have first to transfer the knowledge of how to design and build dashboards and reports with self-service BI tools to business. Handing governance of the display layer over to business gives them all the freedom they had with Excel while automating the data preparation and later the consumption of the report. Everything in the presentation area is business responsibility (right to the green line)

Training a group of key-users is only possible through hands-on training and persistence as it takes a while to get through the highs and lows of adapting to new technology.

In our experience, around 90% of business logic and KPI can be calculated on the fly in the BI tool and are therefore under complete control of the business department, changing them in self-service manner if needed. Contrary to a BW system where the business logic is calculated in the reporting cube and set up by IT.

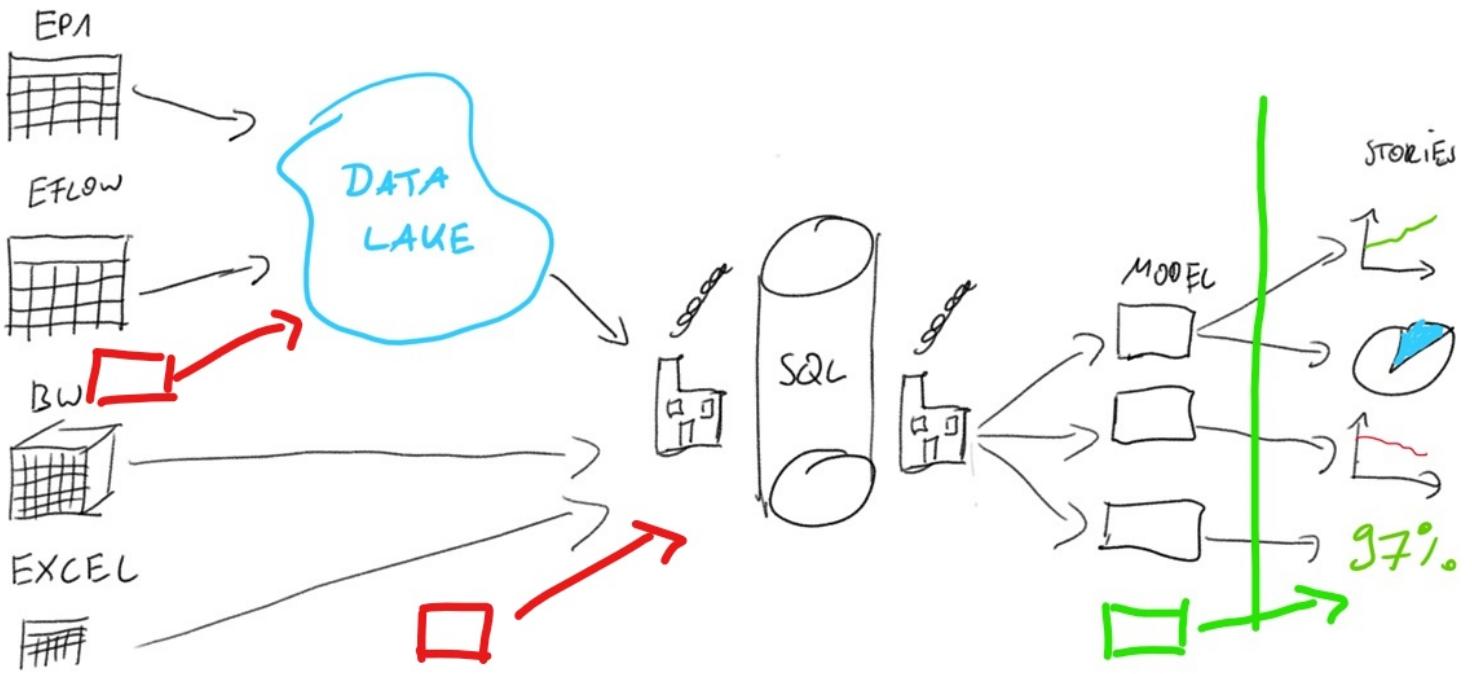
The second requirement of change is adding additional data or dimensions. Adding further data quickly is possible because we can merge each model table of the flexible data warehouse with other manually added data and then join. This way, business can quickly add additional dimensions.



These two requirements of change we call Type 1 changes. They have high flexibility, are easy to implement, and **don't involve any IT staff**. Business can try out new ideas and additional requirements his way.

There is a trade-off in that these type1 changes increase complexity for business to handle and decrease automation as the key users must maintain the additional files manually.

Because of that, there is another change we call type2 change. A type2 change is always preceded by a Type1 change that serves as the blueprint for the type 2 change. A type2 change moves the data source further back in the pipeline to automate the extraction and processing.

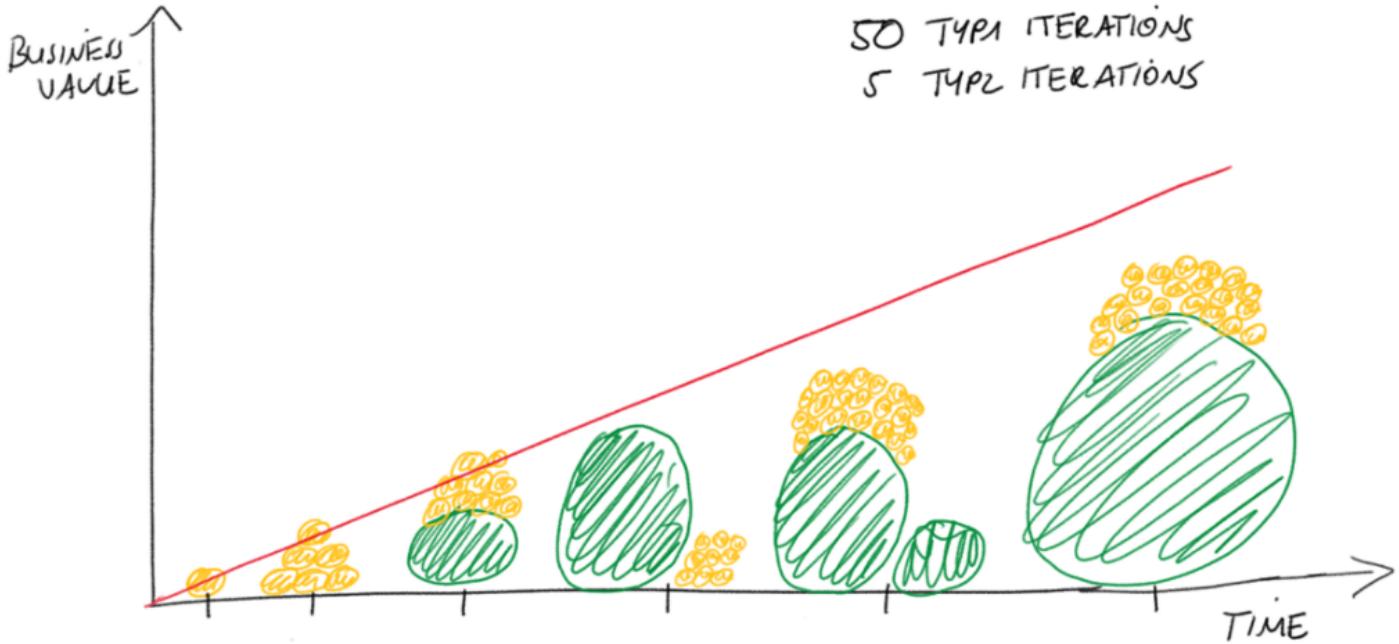


This move will increase automation and decrease complexity for business as the processing is now handled automatically in the pipeline. However, it involves IT effort and additional testing of all affected data streams.

Handling changes in this two-step approach also decreases friction between Business and IT department where business fails to exactly describe what they want (IT perspective 😊) or IT fails to understand the business idea (Business perspective 😊) as the type1 change serves as a blueprint.

Optimally, we combine many Type1 changes into one bigger Type2 change because we can significantly reduce implementation and testing effort this way.

As below shown: Business value created by changes over time.



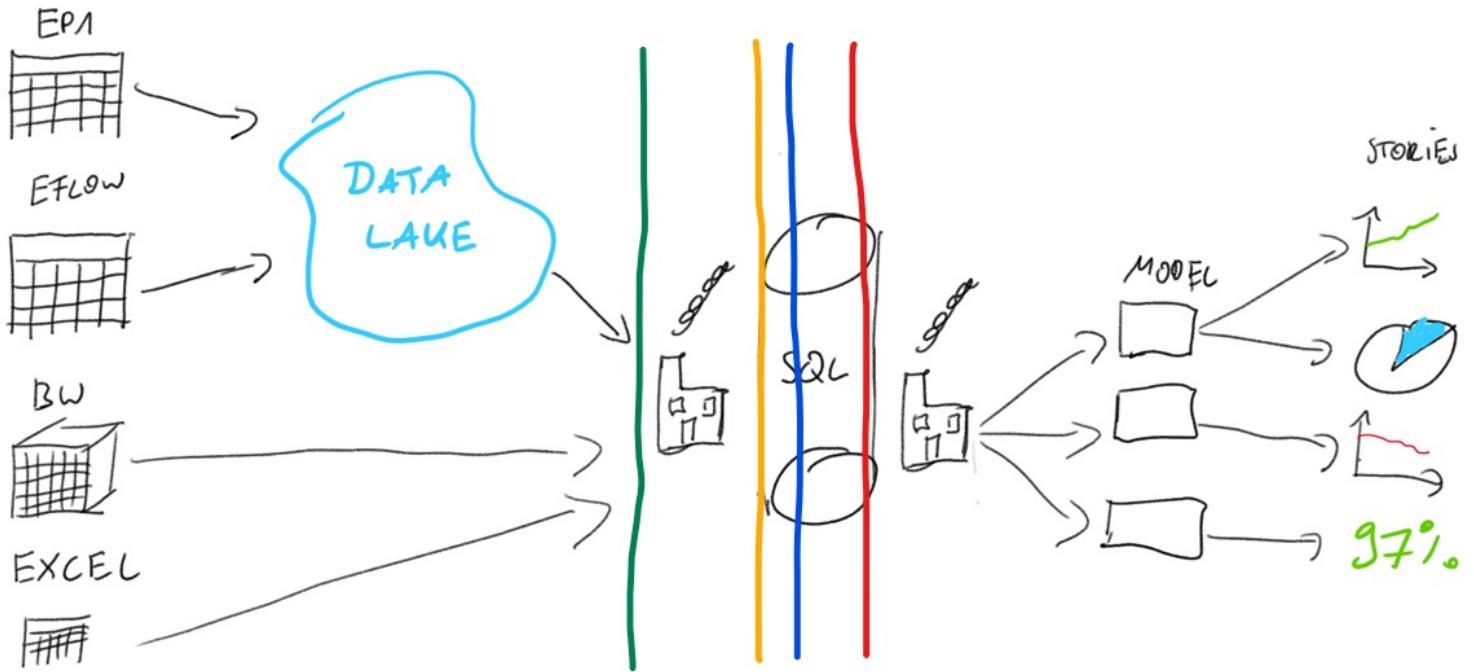
## Short Development Cycles

Short development cycles are achieved by decreasing complexity. One big part of that is the change strategy introduced in the previous chapter. It enables leaving business logic to business.

Several other technical design principles ensure short development cycles.

The more frequently a logic/procedure changes, the closer to the end we should put it in the pipeline

In our case, this means the following:



Until the green line, there is no logic whatsoever. The tables from the source systems are loaded entirely and without any delta update logic. Contrary to a BW solution, the data volume is not huge, and storage and compute power are cheap nowadays.

Until the orange line, we filter the data by very high-level case specifics, e.g., Data from Chinese Entities.

Until the blue line, only general ETL functions prepare the data. There is still no specific logic.

Until the red line, tables are prepared and joined only into their natural objects. e.g., SAP Invoice, Credit Line Workflow, etc.

Only after the red line business logic and frequently changing transformations take place.

This layered approach keeps the core stable and easy while allowing for more frequent change. We can make the changes most of the time without touching anything behind the red line. The only exception being big Type2 changes.

### Avoid storage tables completely

In the flexible data warehouse, We should completely recreate ALL tables with each pipeline run. Following this rule allows us to add additional columns and calculations without worrying about any legacy data.

I'll repeat it because this is huge! In the flexible data warehouse, we don't have to worry about any legacy data!

Getting rid of storing legacy data means that we don't have to write complex update routines for our tables or manually add legacy columns after downloading them. There are no slowly changing dimensions needed to be taken care of.

This eliminates the biggest source of errors and manual work in the development cycle and is the most significant difference compared to the classic BW system that has to do complex delta updates. Partly because in the past, it was necessary and partially because the data used is enormous.

The trade-off, in that case, is less manual work and higher stability for accuracy. If legacy data is changed in the source system, it is also changed in our flexible data warehouse. E.g., If a supplier name is changed, it is also adjusted for all legacy data under that supplier number.

In the flexible warehouse, we are very willing to make this trade-off. For the core KPI layer, it might be a different discussion.

# **Chapter3: Implementation of the flexible data Warehouse Systems**

## **Data Factory**

D-System:

[https://adf.azure.com/en-us/authoring/dataflow/O2C\\_INGEST\\_ACCOUNTS\\_RECEIVEABLES?factory=%2Fsubscriptions%2F770faa00-73c9-4505-be0e-9fd399518c7f%2FresourceGroups%2Fsdp-s-fssc%2Fproviders%2FMicrosoft.DataFactory%2Ffactories%2Fsdp-s-d-fssc-df](https://adf.azure.com/en-us/authoring/dataflow/O2C_INGEST_ACCOUNTS_RECEIVEABLES?factory=%2Fsubscriptions%2F770faa00-73c9-4505-be0e-9fd399518c7f%2FresourceGroups%2Fsdp-s-fssc%2Fproviders%2FMicrosoft.DataFactory%2Ffactories%2Fsdp-s-d-fssc-df)

P-System

<https://adf.azure.com/en-us/authoring?factory=%2Fsubscriptions%2F11737385-67b3-4577-8eb6-d5e4551e47e3%2FresourceGroups%2Fsdp-s-fssc%2Fproviders%2FMicrosoft.DataFactory%2Ffactories%2Fsdp-s-p-fssc-df>

## **SQL Server**

D-System:

[sdp-s-d-sqls.database.windows.net](http://sdp-s-d-sqls.database.windows.net)

P-System:

[sdp-s-p-sqls.database.windows.net](http://sdp-s-p-sqls.database.windows.net)

## **SAC Fileserver:**

Input External Excel files

[\\schaeffler.com\herzogenaurach\DATA\SZ-HZA-Z\Projects\BW\\_SAC\\_Upload\FI\FSSC\\_IMPROVEMENT\\_FRAMEWORK\INPUT](\\schaeffler.com\herzogenaurach\DATA\SZ-HZA-Z\Projects\BW_SAC_Upload\FI\FSSC_IMPROVEMENT_FRAMEWORK\INPUT)

Export Models

[\\schaeffler.com\herzogenaurach\DATA\SZ-HZA-Z\Projects\BW\\_SAC\\_Upload\FI\FSSC\\_IMPROVEMENT\\_FRAMEWORK\OUTPUT](\\schaeffler.com\herzogenaurach\DATA\SZ-HZA-Z\Projects\BW_SAC_Upload\FI\FSSC_IMPROVEMENT_FRAMEWORK\OUTPUT)

Additional Files by Business:

[\\schaeffler.com\herzogenaurach\DATA\SZ-HZA-Z\Projects\BW\\_SAC\\_Upload\FI\FSSC\\_IMPROVEMENT\\_FRAMEWORK\ExcelUpload](\\schaeffler.com\herzogenaurach\DATA\SZ-HZA-Z\Projects\BW_SAC_Upload\FI\FSSC_IMPROVEMENT_FRAMEWORK\ExcelUpload)

## **SAC:**

[https://schaeffler-technologies.eu10.sapanalytics.cloud/sap/fpa/ui/app.html#;view\\_id=home](https://schaeffler-technologies.eu10.sapanalytics.cloud/sap/fpa/ui/app.html#;view_id=home)

# **Overall Data Flow**

## **Data Source**

### **SAP R3 Tables**

ZSI\_IR\_IC\_OCRLOG

EKKO

EKBE

EBAN

T001

VF\_KRED

ADRC

T024  
T001S  
BSAK  
BKPF  
BSIK  
REGUP  
BSID  
BSAD  
FDM\_DCPROC  
UDMCASEATTR00  
SCMG\_T\_CASE\_ATTR  
LIK  
VBUK  
KNA1  
KNB1  
T052  
KNKK

## External Excel Files

INX\_OVERDUE\_REASON  
INX\_PAYMENT\_CALENDAR  
INX\_TRADESHIFT\_INVOICES  
INX\_SAMPLE\_ORDERS

## BW Reports

FI5000 - Vendor incoming invoices  
Fi1000 - DSO

## EFLow

v-dp-proc-048-dnno  
v-dp-tasks

# Data Factory

## Datasets

▲ Datasets		9
▲	Output	1
	DS_CSV	
▲	Source	6
	DS_BW_FI1000	
	DS_BW_FI5000	
	DS_EXCEL	
	DS_SDPHUB_EFLOW	
	DS_SDPHUB_SAPR3	
	DS_SDPHUB_SAPR3_TRUSTED	
▲	Transform	2
	DS_O2C_SQL_SERVER	
	DS_P2P_SQL_SERVER	

**Source:**

Dataset	Content
DS_BW_FI5000	Input for BW Report FI 5000
DS_BW_FI1000	Input for BW Report FI 1000
DS_EXCEL	Input for Additional Excel files
DS_SDPHUB_EFLOW	Input for EFLOW data
DS_SDPHUB_SAPR3	Input All SAP Tables
DS_SDPHUB_SAPR3	Input All SAP Tables RAW - currently not used

**Transformation**

Dataset	Content
DS_P2P_SQL_SERVER	All Tables on SQL Server WAVEI
DS_O2C_SQL_Server	All Tables on SQL Server WAVEII

**Output**

Dataset	Content
DS_CSV	Output all Files as CSV (Output as Excel not supported)

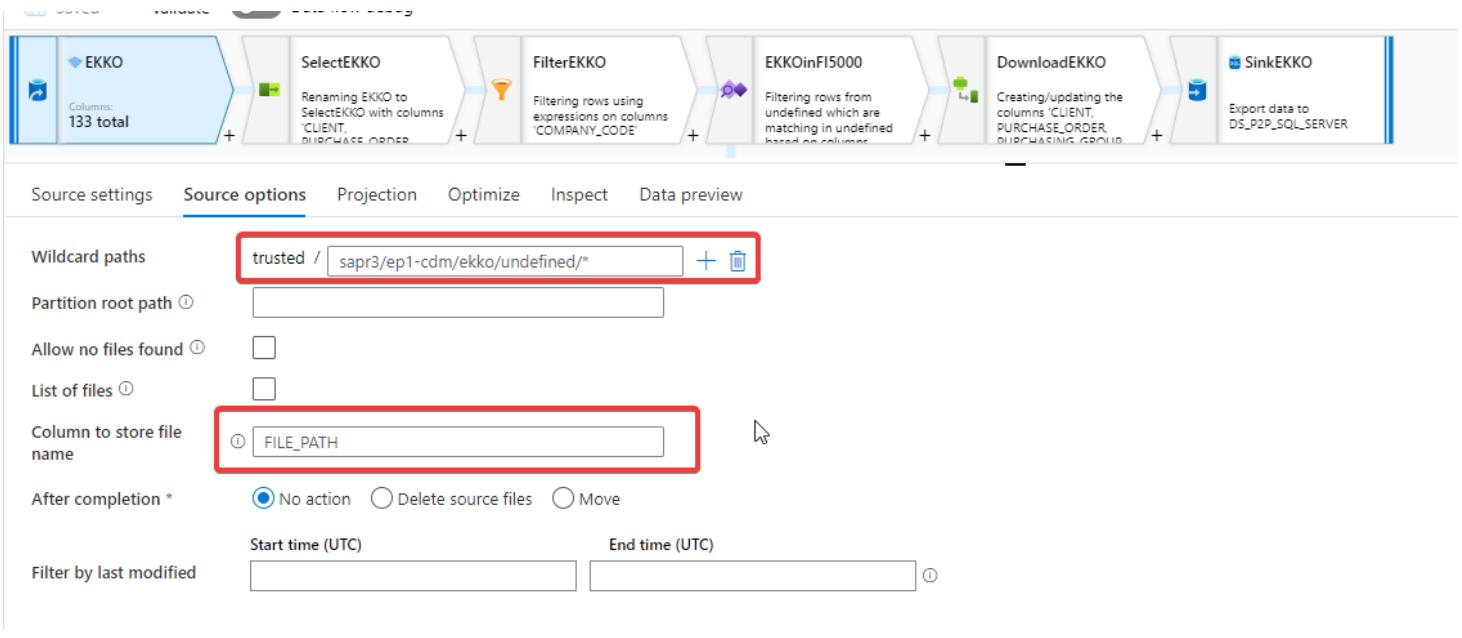
**Dataflows**

- ▲ Data flows 6
  - ❖ O2C\_INGEST\_ACCOUNTS\_RECEIVEA...
  - ❖ O2C\_INGEST\_DATA\_EFLOW
  - ❖ O2C\_INGEST\_DATA\_GENERAL
  - ❖ P2P\_INGEST\_DATA\_ACCOUNTS\_PAYA...
  - ❖ P2P\_INGEST\_DATA\_GENERAL
  - ❖ P2P\_INGEST\_DATA\_PAYMENTS

Dataflow	Content
P2P_INGEST_DATA_ACCOUNTS_PAYABLES	Ingest of Accounts Payables data to SQL Server
P2P_INGEST_DATA_PAYMENTS	Ingest of Payments data to SQL Server
P2P_INGEST_DATA_GENERAL	Ingest of all general data/master Data to SQL Server
O2C_INGEST_ACCOUNTS_RECEIVEABLES	Ingest of Accounts Receivables data to SQL Server
O2C_INGEST_DATA_EFLOW	Ingest of EFLOW and BW data to SQL server
O2C_INGEST_DATA_GENERAL	Ingest of general data to SQL Server

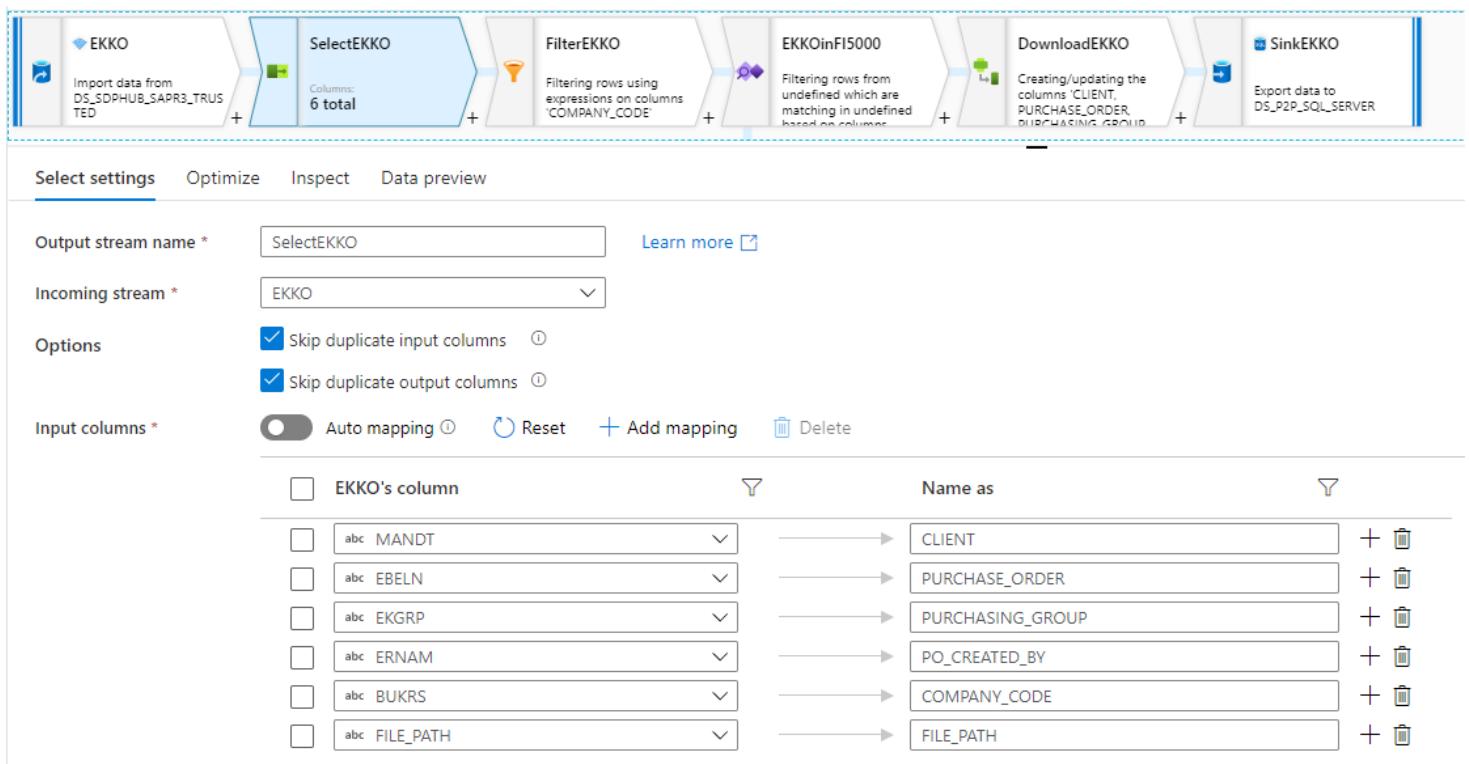
Note: In general, we could combine all of these flows into one for each o2c and p2p. However, it's easier for debugging, and in the future, the parts might have to run separately due to different business requirements by each department.

**General Data Flow logic****1. Import data from Datasource for a specific table**



- a. set correct wildcard file path to find the corresponding table on data lake
- b. store file path in a column to enable checking in SQL server later

## 2. Select relevant columns from the table and rename them into names that make sense



Do this step as early as possible to work with 'real names.' Use these consistently

## 3. Filter relevant Data

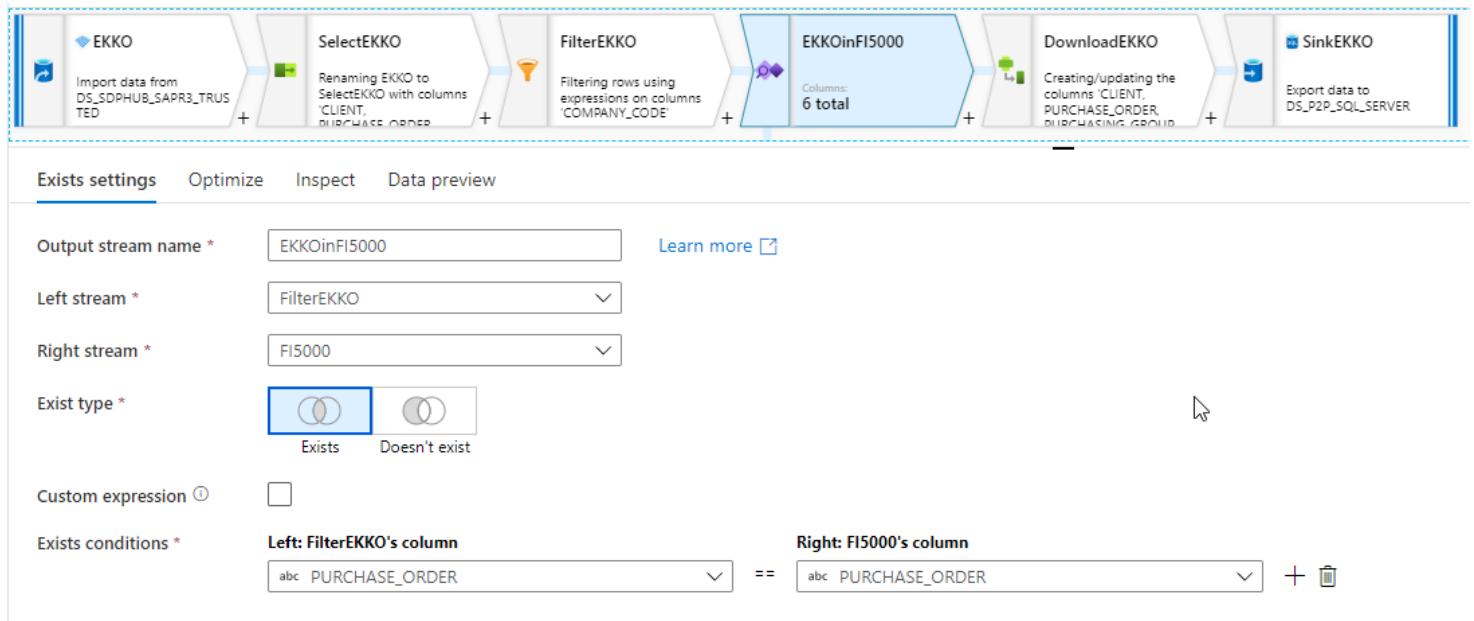
Visual expression builder   

Expression

```
COMPANY_CODE == '0083' || COMPANY_CODE == '0189' || COMPANY_CODE == '0199' || COMPANY_CODE == '0289' || COMPANY_CODE == '0369' || COMPANY_CODE == '0371' || COMPANY_CODE == '0377' || COMPANY_CODE == '0404' || COMPANY_CODE == '0426' || COMPANY_CODE == '0429'
```

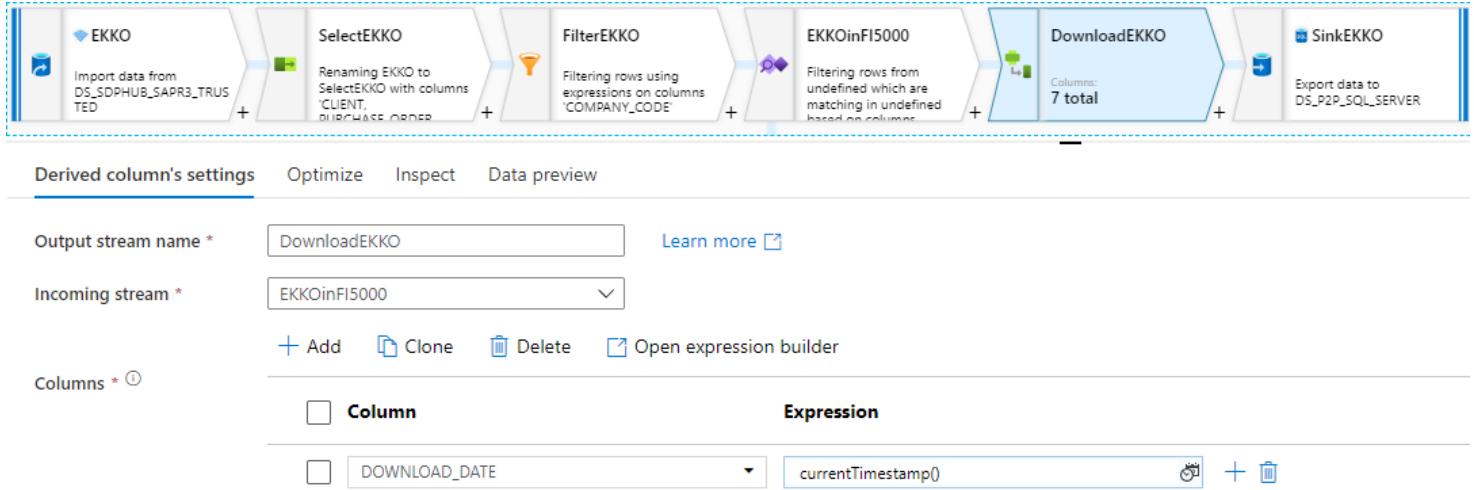
Use filters to exclude not-needed data. Exclude as much as possible in data flows as they run on Spark Cluster! Performance is much better than in the SQL Server.

#### 4. Merge Data



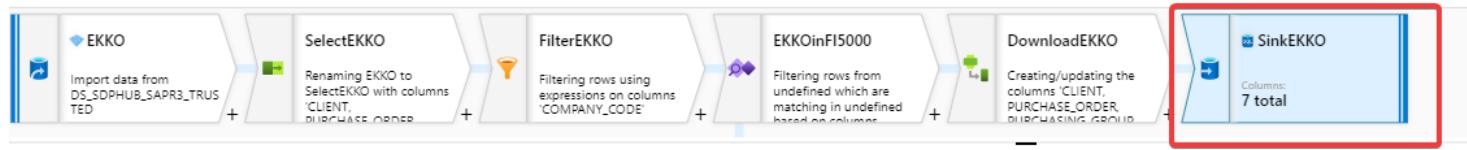
Only include data from the tables contained in the facts tables from BW >report FI5000. Again, exclude as much as possible on Spark Cluster.

#### 5. Add Download date to double-check when the system last downloaded data



Very important to be able to follow through on when the data was downloaded. Check Monitoring Chapter for details.

#### 6. Insert remaining rows into target SQL Server



Sink   Settings   Mapping   Optimize   Inspect   Data preview

Update method  Allow insert

Allow delete

Allow upsert

Allow update

Table action  None  Recreate table  Truncate table

Batch size ⓘ

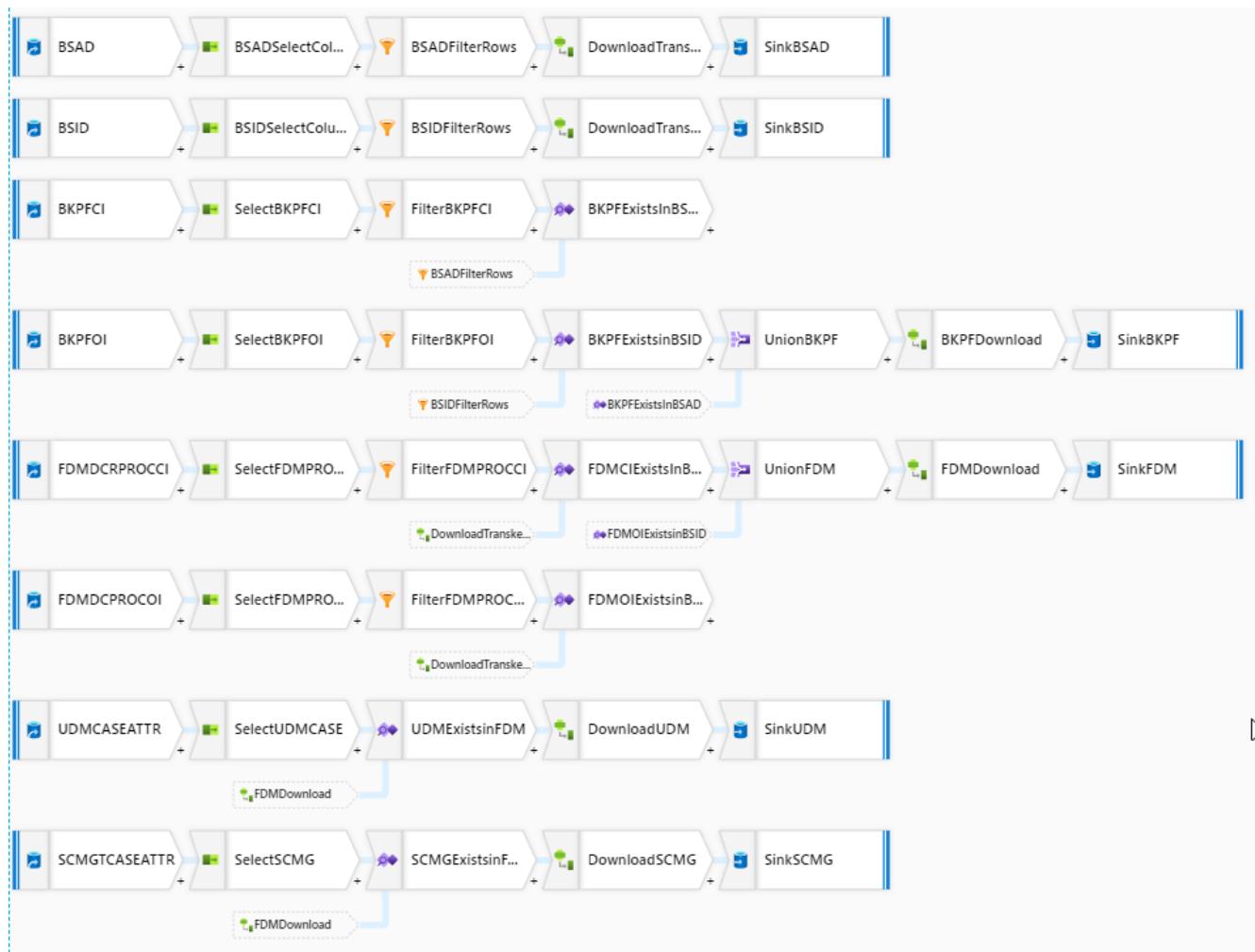
Use TempDB ⓘ



Pre SQL scripts  List of scripts  Custom expression ⓘ

Use 'recreate table' to delete and recreate the target tables with each load. Avoid duplicates.

**For each table, there is a similar dataflow according to the above schema**



## Pipelines

▲ Pipelines	9
PIP_TEST	
▲ O2C	4
PIP_O2C_INGEST	
PIP_O2C_OUTPUT	
PIP_O2C_OUTPUT_SCHEMA	
PIP_O2C_TRANSFORM	
▲ P2P	4
PIP_P2P_INGEST	
PIP_P2P_OUTPUT	
PIP_P2P_OUTPUT_SCHEMA	
PIP_P2P_TRANSFORMATION	

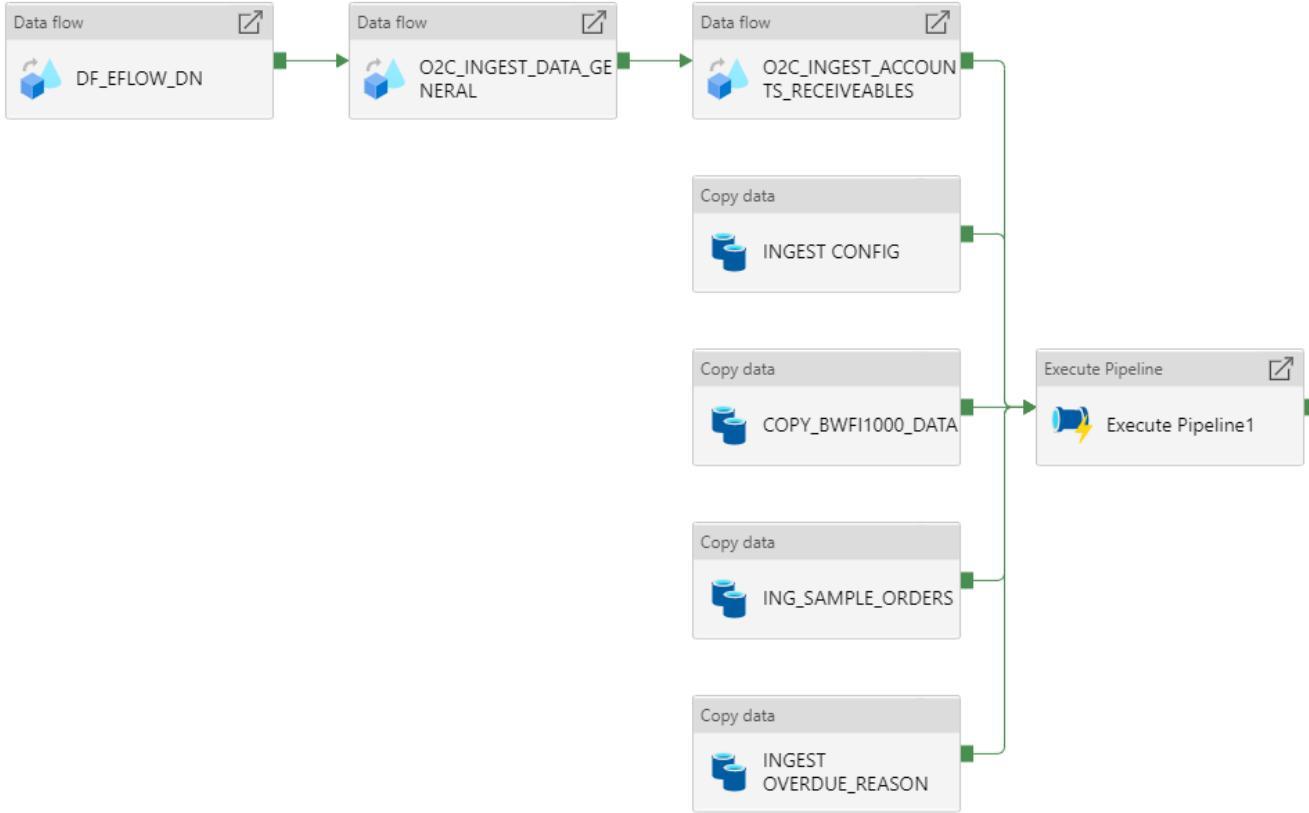
Pipeline	Function
PIP_TEST	Dummy used for tests. Not used in the weekly run.
PIP_O2C_INGEST	Ingest all O2C data to SQL Server
PIP_O2C_TRANSFORM	Execute all O2C Transformation in SQL Server
PIP_O2C_OUTPUT	Output all O2C data onto SAC Fileserver
PIP_O2C_OUTPUT_SCHEMA	Prints the first 500 rows of each O2C model to the SAC server.Used to speed up model updates in SAC. <b>Not used in weekly run</b>
PIP_P2P_INGEST	Ingest all P2P data to SQL Server
PIP_P2P_TRANSFORM	Execute all P2P Transformation in SQL Server
PIP_P2P_OUTPUT	Output all P2P data onto SAC Fileserver
PIP_P2P_OUTPUT_SCHEMA	Prints the first 500 rows of each P2P model to SAC server. Used to speed up model updates in SAC. <b>Not used in weekly run</b>

The Pipelines are structured in 3 parts:

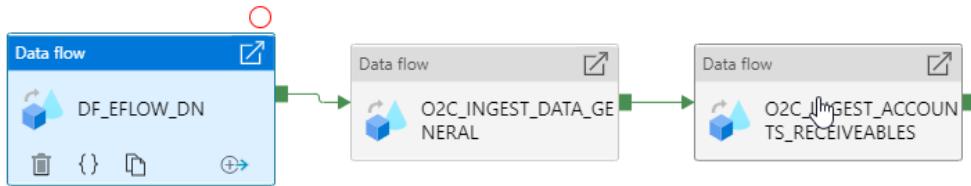
1. Ingest from the data lake
2. Transformation on the SQL Server
3. Output to SAC Fileserver

## Ingest Pipelines

Both Ingest Pipelines work in the same way:



#### 1. Executes Data Dataflows to get SAP Tables from Data lake



General    **Settings**    Parameters    User properties

Data flow \*  Open + New

▲ SINKTOEFlowTask parameters ⓘ

Name	Value	Type
tablename	ing_eflowtask	string

▲ SinkToEFlowDN parameters ⓘ

Name	Value	Type
tablename	ing_eflowdn	string

▲ SinkToLIKp parameters ⓘ

Name	Value	Type
tablename	ing_likp	string

▲ SinkToVBUK parameters ⓘ

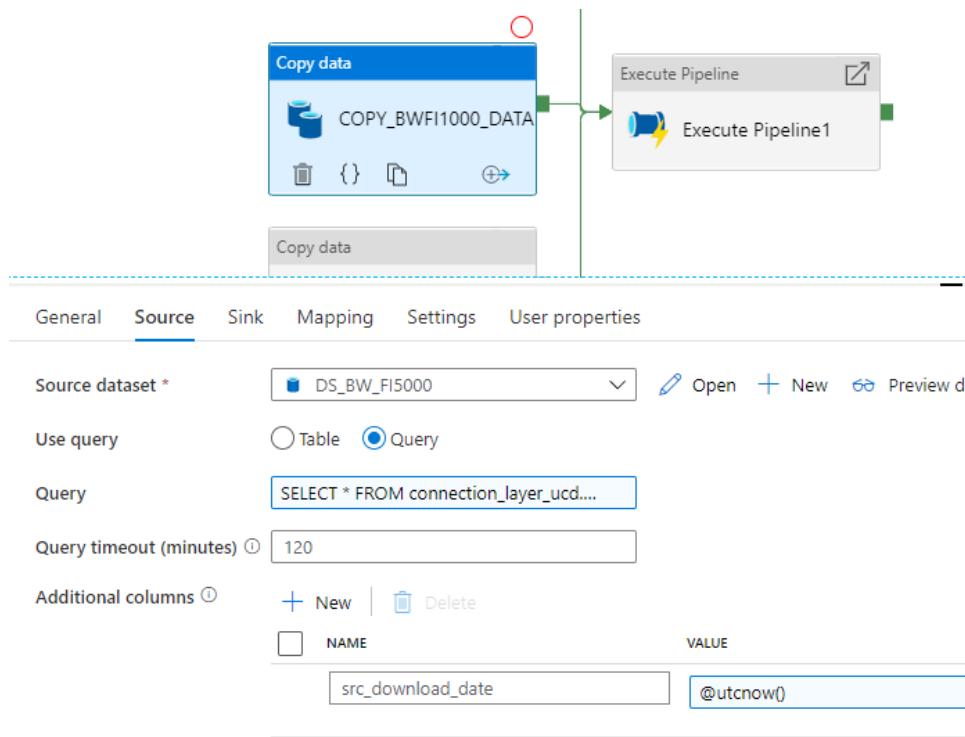
Name	Value	Type
tablename	ing_vbuk	string

Run on (Azure IR) \* ⓘ

a. Parameters from Dataflows must be entered here. In our case, it's the SQL Server table names

For each dataflow, we can variably select the "Run On". If there is a lot of data, we can use more CPUs to speed up the processing. The more CPU power we use, the more expensive the processing becomes, though.

## 2. Copy Data from BW Cubes



In the copy data step, you can directly query SQL statements against the Denodo base-view of the query. There is no delay in getting the data from BW.

You can easily use dynamic parameters in this select statement.

```

SELECT *
FROM connection_layer_udc.bv_sapbw_bp1_fssc_framework_improvement_fi1000
WHERE monthyearfrompostingdatedefaultfromexitme_0 = '@{getPastTime(1,'Month','MM.yyyy')}' AND
(
companycode_key_0 = 'EP1_100/0083' OR
companycode_key_0 = 'EP1_100/0189' OR
companycode_key_0 = 'EP1_100/0199' OR
companycode_key_0 = 'EP1_100/0289' OR
companycode_key_0 = 'EP1_100/0371' OR
companycode_key_0 = 'EP1_100/0369' OR
companycode_key_0 = 'EP1_100/0377' OR
companycode_key_0 = 'EP1_100/0404' OR
companycode_key_0 = 'EP1_100/0426' OR
companycode_key_0 = 'EP1_100/0429' OR
companycode_key_0 = 'EP1_100/0078'
)

```

### 3. Copy external Excel files

Excel files can be imported directly from the directory in the dataset. You can enter the filename in the settings as a parameter.

Source	Type	Destination
DOCUMENT_NUMBER	String	document_number
COMPANY_CODE	String	company_code
YEAR	String	year
LINE_ITEM	String	line_item
KEY_DATE	String	key_date
SAMPLE_ORDER	String	sample_order

#### 4. Special

The only exception is this SQL server procedure that the pipeline must execute in the ingest pipeline. It's needed to pre-format the data from BW so that the dataflows can merge the data with the SAP tables.

**General Settings User properties**

Linked service \* LS\_ASQ\_FSSC

Stored procedure name \* [dbo].[P\_ZZZ\_EXEC\_FI5000]

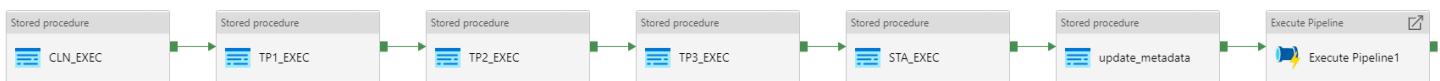
Edit

Stored procedure parameters

Import  New

## Transformation Pipeline

The Transformation Pipeline is used to execute the stored procedures on the SQL-Server in a procedural manner. We will provide more information in the Chapter about the SQL Server.



## Output Pipeline

The output pipelines copy the data from the model tables in the SQL server onto the SAC fileserver. It's possible to do another mapping here between SQL Server column name and SAC column name.

The screenshot shows the Azure Data Factory pipeline editor interface. At the top, there are two 'Copy data' activities:

- Copy IRB Full**: This activity is mapped to the 'Source' table.
- Copy ALL ITEMS**: This activity is mapped to the 'Destination' table.

The 'Mapping' tab is selected, showing the detailed mapping configuration:

Source	Type	Destination	Type
DOCUMENT_NUMBER	nvarchar	Document_Number	String
LINE_ITEM	decimal	Line_item	Decimal
Precision: 3 Scale: 0			
VENDOR_NUMBER	nvarchar	Vendor_Number_	String
DOCUMENT_TYPE	nvarchar	Document_type	String
SPECIAL_GL_INDICAT...	nvarchar	Special_GL_ind	String
PAYMENT_BLOCK	nvarchar	Payment_Block	String
PAYMENT_TERMS	nvarchar	Terms_of_Payment	String

## Output Pipeline Schema

The model mapping function in SAC is very slow if there is a lot of data uploaded. That is why this pipeline only prints the 500 first rows of each model into the CSV file. If there needs to be a new mapping business will ask IT to print only the schema to do the mapping. Afterward, the full dataset can be loaded automatically.

## Transport Changes from D-System to P-System

The screenshot shows the Microsoft Azure Data Factory interface. The top navigation bar shows the path: Microsoft Azure | Data Factory > sdp-s-d-fssc-df. The main area displays the 'Factory Resources' sidebar with 'Pipelines' selected, showing the 'PIP\_TEST' pipeline. The pipeline details pane on the right lists the following activities:

- Move & transform
- Azure Data Explorer
- Azure Function
- Batch Service

Two buttons are highlighted with red boxes:

- main branch**: A dropdown menu for branching.
- Publish**: A button to publish changes.

The data factory has GIT version control. After changes in the d-system are done, they can be published via the publish button. Afterward, HQ needs to be contacted to move the changes to P-System.

# SQL Server

## Key requirements for transformation stage:

1. The pipeline has to be procedural on each step in the data factory. If one step fails, the pipeline must be able to only rerun from the failed activity to avoid long runtimes  
--> Oil in a Pipeline also does not flow back 😊
2. Errors should be traceable as easily and fast as possible.
3. Make solution scalable for future demands

## Coding rules :

1. Each Stored procedure is only allowed to work with tables of the same name stage or lower:  
E.g., Stored procedure CLN can only work with tables CLN and ING  
TP1 only with CLN/ING/TP1 etc.
2. Each task for each table in each step gets its own stored procedure with the table's name as procedure name.  
e.g., Step to clean table CLN\_ADRC is called P\_CLN\_ADRC
3. Strictly follow transformation structure for all tables

## Naming Conventions

### Tables

1. INGESTED from Data Factory  
Table Names ING\_\* for SAP&BW / INX\_\* for Excel
- Original tables from the data lake. procedures should never change the data to avoid having to run the INGEST Pipeline again
2. CLEANED with ETL functions  
Table Names CLN\_\*
3. TRANSFORM1 Join Data together  
Table Names TP1\_\*
4. TRANSFORM2 Join Data together  
Table Names TP2\_\*
5. TRANSFORM3 Calculate additional Columns  
Table Names TP3\_\*
6. STAGE data for Export to Fileserver  
Table Names STA\_\*

### Stored Procedures

1. Stored Procedures Names \*\_EXEC  
Are executed from Azure Data Factory
2. Stored Procedure Names P\_CLN\_\*
- Apply all the ETL functions and data cleaning
3. Stored Procedure Names P\_TP1\*  
Join Cleaned Tables
4. Stored Procedure Names P\_TP2\*  
Join the Dimensions and Facts tables together
5. Stored Procedure Names P\_TP3\*  
Apply Business logic and calculate additional columns
6. Stored Procedure Names P\_STA\*  
Create different views that can be consumed via Analytics tools.

## Scalar Functions and ETL

ETL functions and Scalar functions are named after what they do. They only do one thing. Having small functions and procedures is especially important because SQL Server does not have efficient debugging capabilities.

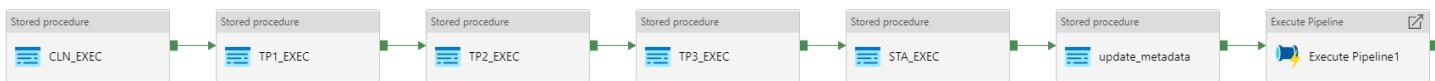
## Schema

1. dbo  
Everything related to Purchase to Pay Process
2. o2c  
Everything related to Order to Cash Process

Currently, everything in schema dbo is written in the upper case. Everything in o2c is written in lower case. This is a legacy and should be changed in the future to all lower case. Also, all p2p tables should be moved to a new schema called p2p. Everything that's used for both, like ETL functions, should be kept in dbo (default schema)

## Execution from Datafactory

Not all stored procedures are executed directly from Azure Cloud. Azure Data Factory only controls the five main procedures:



This is a trade-off between monitoring capability in Data Factory and practicality of use. If every procedure is in ADF, the stored procedure that can be identified directly must be set up and published one by one and adjusted with every change. The \*\_EXEC Procedures help structure the procedures while allowing quick changes without changing the overall flow in ADF.

## CLEANED with ETL functions

```

alter procedure [dbo].[p_cln_exec] as
-- first-----
exec p_cln_first
exec o2c.p_cln_src_download_date @schema = 'dbo'
exec p_cln_load_details
-----  

exec p_cln_adrc
exec p_cln_bkpf
exec p_cln_bsik
exec p_cln_eban
exec p_cln_ekbe
exec p_cln_ekko
exec p_cln_overdue_reason
exec p_cln_payment_calendar
exec p_cln_t001s
exec p_cln_t024
exec p_cln_vf_kred
exec p_cln_ocrlog
exec p_cln_t001
exec p_cln_bsak
exec p_cln_regup
exec p_cln_ts_invoices
--last  -----
exec o2c.p_cln_clean_columns @schema = 'dbo'

```

## **p\_cln\_first**

1. Copies all data from ING\_\* to CLN\_\*, creates indexed tables one by one, and adds additional columns.

```

declare @table table
(
tablename varchar(50),
id int identity(1,1)
)

insert into @table
select distinct table_name from information_schema.columns
where left(table_name,2) = 'in' and
      table_name <> 'ing_fi5000' and
      table_name <> 'ing_eflowtask' and
      table_schema = @schema

declare @max int
declare @sql varchar(max)
declare @tablename varchar(50)
declare @id int = 1

select @max = max(id) from @table

while (@id <= @max)
begin

select @tablename = tablename from @table where id = @id
set @sql =      'drop table if exists '+@schema+'.cln'+substring(@tablename,4,20)+';
                  select * into '+@schema+'.cln'+substring(@tablename,4,20)+'
                  from '+@schema+'.'+@tablename+''

--print(@sql)
exec(@sql)
set @id = @id +1
end

```

2. Sometimes there are two files on the datalake. Therefore it's necessary to remove duplicates for each of the ingested tables.

Example for P2P:

```

alter procedure [dbo].[p_cln_bsik] as

with bsik_duplicates as (
    select *,
           row_number() over (
               partition by
                   company_code,
                           document_number,
                           [year],
                           line_item
               order by
                   company_code,
                           document_number,
                           [year],
                           line_item
           ) row_num
    from
        cln_bsik
)

```

**delete from bsik\_duplicates  
where row\_num > 1**

Example For O2C:

```

alter procedure [o2c].[p_cln_bsad] as

delete from o2c.cln_bsad
where file_path <> (select max(file_path) from o2c.cln_bsad)

update o2c.cln_bsad
set o2c.cln_bsad.src_download_date = src.src_download_date
from o2c.cln_load_details as src
where table_name = 'BSAD'

```

The P2P Method has a longer runtime but makes sure that there can't be any duplicates, even if there would be an error in SAP. While the option that was implemented for O2C is faster, it, in principle, could still contain duplicates if they existed on the Source side.

### 3. Remove Characters that can't be used in CSV File

```

declare @table table
(
tablename varchar(50),
columnname varchar(50),
id int identity(1,1)
)

insert into @table(tablename,columnname)
select table_name, column_name from config
where function_name = 'CLEAN_COLUMNS'
and db_schema = @schema

declare @max int
declare @sql varchar(max)
declare @tablename varchar(50)
declare @columnname varchar(50)
declare @id int = 1

select @max = max(id) from @table

while (@id <= @max)
begin

select @tablename = tablename, @columnname =columnname from @table where id = @id

set @sql = 'update '+@schema+'. '+@tablename+' set '+@columnname+' = replace('+@columnname+', ''',''');'
          update '+@schema+'. '+@tablename+' set '+@columnname+' = replace('+@columnname+', '''','''');    '

exec(@sql)

set @id = @id +1
end

```

The STA\_\* tables are later printed as CSV files because SAP Analytics Cloud can currently not work with direct SQL access. Using CSV files has the disadvantage that the characters ',' and "" can't be used.

', ': It will cause the CSV file to think the next column already starts and therefore mess up the whole file.

"" ': This character will tell the system to not regard this ',' as a separator which means that if it is at the end of a column can combine two columns and mess up the file.

## Config ETL Functions

Most ETL functions can be controlled via the config file in Excel.

A	B	C	D	E	F
FUNCTION_NAME	TABLE_NAME	COLUMN_NAME	PARAMETER	ACTIVE	DB_SCHEMA
ADD_ZERO	CLN_SAMPLE_ORDERS	DOCUMENT_NUMBER		10 X	o2c
ADD_ZERO	CLN_SAMPLE_ORDERS	COMPANY_CODE		4 X	o2c
REMOVE_ZERO	sta_fi1000	customer_number		X	o2c
REMOVE_ZERO	sta_fi1000	reconciliation_account		X	o2c
REMOVE_ZERO	sta_fi1000	credit_account		X	o2c
REMOVE_ZERO	sta_eflow_likp	delivery_nr		X	o2c
REMOVE_ZERO	sta_eflow_likp	soldtoparty		X	o2c
REMOVE_ZERO	sta_all_cust_items	document_number		X	o2c
REMOVE_ZERO	sta_all_cust_items	customer_number		X	o2c
REMOVE_ZERO	sta_all_cust_items	gl_account		X	o2c
REMOVE_ZERO	sta_all_cust_items	clearing_document		X	o2c
REMOVE_ZERO	sta_all_cust_items	reconciliation_account		X	o2c
REMOVE_ZERO	sta_all_cust_items	trading_partner		X	o2c

- Function name: Name of the ETL function
- Table Name: Table for which the function should be executed
- Column Name: Column for which the function should be executed
- Parameter: Additional Parameters that the function might need

- Active: The line is active or not
- DB Schema: Database Schema for which the function should be executed

Currently, there are three procedures controlled with the config file. The ETL procedures is used at several places in the pipeline:

- ADD\_ZERO
- REMOVE\_ZERO
- CLEAN\_COLUMNS

```

ALTER procedure [o2c].[p_execute_etl_function] @imp_function nvarchar(max),
                                                @imp_tablename nvarchar(max),
                                                @schema varchar(max)
as
declare @table table
(
table_name varchar(max),
column_name varchar(max),
parameter varchar(max),
id varchar(max)
)

declare @max int
declare @sql varchar(max)
declare @tablename varchar(50)
declare @columnname varchar(50)
declare @parameter varchar(max)
declare @id int = 1

insert into @table(table_name, column_name, parameter, id)
select table_name, column_name, parameter, row_number() over (order by table_name, column_name desc) as id from config
where function_name = @imp_function and active = 'x' and table_name = @imp_tablename and db_schema = @schema

select @max = max(id) from @table

while (@id <= @max)
begin

    select @tablename = table_name, @columnname = column_name, @parameter = parameter from @table where id = @id

    if @imp_function = 'REMOVE_ZERO'
    begin
        set @sql = 'update '+@schema+'. '+@tablename+ ' set '+@columnname+' = substring('+@columnname+', patindex(''%[^0]%'', '+@columnname+
        end

    if @imp_function = 'ADD_ZERO'
    begin
        if @parameter = '10'
        begin
            set @sql = 'update '+@schema+'. '+@tablename+ ' set '+@columnname+' = right(''000000000'+isnull('+@columnname+', '''),10)'
        end
        if @parameter = '4'
        begin
            set @sql = 'update '+@schema+'. '+@tablename+ ' set '+@columnname+' = right(''0000'+isnull('+@columnname+', '''),4)'
        end
        end

        exec (@sql)

        set @id = @id +1
    end
end

```

## CLN\_FI1000

We must first unpivot the FI1000 BW Query to make good use of it. Currently, it contains one measure for each month of sales. This way, it can not be displayed correctly in any BI tool. This routine takes columns for each month and combines them into one column with the month's values and one date column.

```

alter procedure [o2c].[p_cln_fi1000] as

declare @inyearmonth nvarchar(10)
declare @fulldate nvarchar(10)
declare @initdate date
select @inyearmonth = max(monthyearfrom) from o2c.ing_fi1000
select @fulldate = concat(right(@inyearmonth,4),left(@inyearmonth,2),'01')
select @initdate = convert(date,@fulldate,102)

drop table if exists o2c.#bwfi1000ar
select company_code, customer_number,business_division,business_unit,credit_control_area,division,
case yearmonth
when 'ar_month_0' then @initdate
when 'ar_month_1' then dateadd(month,-1,@initdate)
when 'ar_month_2' then dateadd(month,-2,@initdate)
when 'ar_month_3' then dateadd(month,-3,@initdate)
when 'ar_month_4' then dateadd(month,-4,@initdate)
when 'ar_month_5' then dateadd(month,-5,@initdate)
when 'ar_month_6' then dateadd(month,-6,@initdate)
when 'ar_month_7' then dateadd(month,-7,@initdate)
when 'ar_month_8' then dateadd(month,-8,@initdate)
when 'ar_month_9' then dateadd(month,-9,@initdate)
when 'ar_month_10' then dateadd(month,-10,@initdate)
when 'ar_month_11' then dateadd(month,-11,@initdate)
when 'ar_month_12' then dateadd(month,-12,@initdate)
end as postdate, 'aramount' as category, amount
into o2c.#bwfi1000ar
from
(
    select company_code,customer_number,business_division,business_unit,credit_control_area,division,ar_month_0 , ar_month_1,ar_month_2
        ,ar_month_3,ar_month_4,ar_month_5,ar_month_6,ar_month_7,ar_month_8,ar_month_9,ar_month_10,ar_month_11,ar_month_12
    from o2c.ing_fi1000
) p

unpivot (
    amount for yearmonth in (ar_month_0,ar_month_1,ar_month_2,ar_month_3,ar_month_4,ar_month_5,ar_month_6,ar_month_7,ar_month_8,ar_month_9,a
        )
) as unpvtar;

drop table if exists o2c.#bwfi1000sales
select company_code, customer_number,business_division,business_unit,credit_control_area,division,
case yearmonth
when 'sales_month_0' then @initdate
when 'sales_month_1' then dateadd(month,-1,@initdate)
when 'sales_month_2' then dateadd(month,-2,@initdate)
when 'sales_month_3' then dateadd(month,-3,@initdate)
when 'sales_month_4' then dateadd(month,-4,@initdate)
when 'sales_month_5' then dateadd(month,-5,@initdate)
when 'sales_month_6' then dateadd(month,-6,@initdate)
when 'sales_month_7' then dateadd(month,-7,@initdate)
when 'sales_month_8' then dateadd(month,-8,@initdate)
when 'sales_month_9' then dateadd(month,-9,@initdate)
when 'sales_month_10' then dateadd(month,-10,@initdate)
when 'sales_month_11' then dateadd(month,-11,@initdate)
when 'sales_month_12' then dateadd(month,-12,@initdate)
end as postdate, 'salesamount' as category,amount
into o2c.#bwfi1000sales
from
(
    select company_code,customer_number,business_division,business_unit,credit_control_area,division,sales_month_0 , sales_month_1,sale
        ,sales_month_3,sales_month_4,sales_month_5,sales_month_6,sales_month_7,sales_month_8,sales_month_9,sales_month_10,sales_month_1
    from o2c.ing_fi1000
) p

unpivot (
    amount for yearmonth in (sales_month_0,sales_month_1,sales_month_2,sales_month_3,sales_month_4,sales_month_5,sales_month_6,sales_month_7
        )
) as unpvtsales;

-- unpivot overdue amount

```

```

drop table if exists o2c.#bwfi1000overdue
select company_code, customer_number,business_division,business_unit,credit_control_area,division,
case yearmonth
when 'overdue_month_0' then @initdate
when 'overdue_month_1' then dateadd(month,-1,@initdate)
when 'overdue_month_2' then dateadd(month,-2,@initdate)
when 'overdue_month_3' then dateadd(month,-3,@initdate)
when 'overdue_month_4' then dateadd(month,-4,@initdate)
when 'overdue_month_5' then dateadd(month,-5,@initdate)
when 'overdue_month_6' then dateadd(month,-6,@initdate)
when 'overdue_month_7' then dateadd(month,-7,@initdate)
when 'overdue_month_8' then dateadd(month,-8,@initdate)
when 'overdue_month_9' then dateadd(month,-9,@initdate)
when 'overdue_month_10' then dateadd(month,-10,@initdate)
when 'overdue_month_11' then dateadd(month,-11,@initdate)
when 'overdue_month_12' then dateadd(month,-12,@initdate)
end as postdate, 'overdueamount' as category, amount
into o2c.#bwfi1000overdue
from
(
    select company_code,customer_number,business_division,business_unit,credit_control_area,division,
    overdue_month_0 , overdue_month_1,overdue_month_2,overdue_month_3,overdue_month_4,overdue_month_5,overdue_month_6,overdue_month_7,ov
    overdue_month_9,overdue_month_10,overdue_month_11,overdue_month_12
    from o2c.ing_fi1000
) p
unpivot (
    amount for yearmonth in (overdue_month_0 , overdue_month_1,overdue_month_2,overdue_month_3,overdue_month_4,overdue_month_5,overdue_month
    overdue_month_7,overdue_month_8,overdue_month_9,overdue_month_10,overdue_month_11,overdue_month_12)
)
as unpvtoverdue;

drop table if exists o2c.#bwfi1000
select * into o2c.#bwfi1000 from o2c.#bwfi1000ar
union all
select * from o2c.#bwfi1000sales
union all
select * from o2c.#bwfi1000overdue

--pivot
drop table if exists o2c.cln_fi1000
select company_code,customer_number,business_division,business_unit,credit_control_area,division,postdate,
isnull(aramount,0) aramount ,isnull(salesamount,0) salesamount ,isnull(overdueamount,0) overdueamount
into o2c.cln_fi1000
from
(
    select company_code,customer_number,business_division,business_unit,credit_control_area,division,postdate,category,amount from o2c.#bwfi1000
)p
pivot
(
    sum(amount) for category in (aramount,salesamount,overdueamount)
) as pvt

order by postdate

drop table if exists o2c.#bwfi1000ar
drop table if exists o2c.#bwfi1000sales
drop table if exists o2c.#bwfi1000overdue
drop table if exists o2c.#bwfi1000

```

## Transform 1 and Transform 2

In these stages, all Joins are done. The Joins can only access tables that are in the CLN stage or CLN and TP1 for Transform 2.

Transform 1 joins tables to Dimensions that can then be reused to join multiple other FACT tables to the base models. e.g., combines the customer master tables KNA1 and KNB1 to the dimension customer.

## O2C

Model	sta_all_cust_items	sta_open_cust_items	sta_eflow_cir	sta_eflow_likp	sta_fi1000	sta_payment_behavior
TP3	tp3_all_cust_items	tp3_all_cust_items	-	-	tp3_fi1000	combination of (sta_all_cust_items, sta_open_cust_items, tp1_customer)
TP2	tp2_all_cust_items (tp1_all_cust_items tp1_customer tp1_dispute cln_sample_orders cln_overdue_reason)	tp2_all_cust_items (tp1_all_cust_items tp1_customer tp1_dispute cln_sample_orders cln_overdue_reason)	-	-	tp2_fi1000 (tp1_fi1000 tp1_customer)	-
TP1	tp1_all_cust_items (cln_bsld cln_bsad cln_bkpf)	tp1_all_cust_items (cln_bsld cln_bsad cln_bkpf)	cln_eflow_likp (cln_eflowtask cln_eflow_likp)  cln_vbuk  cln_kna1	cln_eflow_task	tp1_customer (cln_knb1 cln_kna1 cln_t001s cln_adrc cln_knkk cln_t001)	-
	tp1_customer (cln_knb1 cln_kna1 cln_t001s cln_adrc cln_knkk cln_t001)	tp1_customer (cln_knb1 cln_kna1 cln_t001s cln_adrc cln_knkk cln_t001)				
	tp1_dispute (cln_fdm_dcproc cln_udmcaseattr00 cln_scmg_t_case_attr)	tp1_dispute (cln_fdm_dcproc cln_udmcaseattr00 cln_scmg_t_case_attr)			cln_fi1000	
SOURCE	EP1:bsid,bsad,bkpf,knb1,kna1,t001s,adrc,knkk, t001, cln_fdm_dcproc,cln_udmcaseattr00, cln_scmg_t_case_attr,t052 EXCEL: cln_sample_orders, cln_overdue_reason	EP1:bsid,bsad,bkpf,knb1,kna1,t001s,adrc,knkk, t001, cln_fdm_dcproc,cln_udmcaseattr00, cln_scmg_t_case_attr, t052 EXCEL: cln_sample_orders, cln_overdue_reason	EFLOW:v-dp- proc-048-dnno,v-dp-tasks EP1: likp,vbuk	EFLOW: v-dp-tasks	BW: fi1000 EP1: knb1, kna1,t001s, knkk, t001, adrc	EP1:bsid,bsad,bkpf,knb1,kna1,t001s,adrc,knkk, t001

## P2P

Model	sta_all_items	sta_open_items_monthly	sta_irb_full	sta_irb_monthly
TP3	tp3_all_items (tp2_open_items union tp2_cleared_items)	tp3_all_items (tp2_open_items union tp2_cleared_items)	tp3_irb	tp3_irb
TP2	tp2_open_items/tp2_cleared_items (cln_bsak/cln_bsik tp1_vendor_dim cln_t001 cln_regup cln_bkpf cln_overdue_reason cln_payment_calendar)	tp2_open_items/tp2_cleared_items (cln_bsak/cln_bsik tp1_vendor_dim cln_t001 cln_regup cln_bkpf cln_overdue_reason cln_payment_calendar)	tp2_irb (tp1_irb cln_tradeshift_invoices tp1_ekbe_ref tp1_ekbe_po tp1_vendor_dim)	tp2_irb (tp1_irb cln_tradeshift_invoices tp1_ekbe_ref tp1_ekbe_po tp1_vendor_dim)
TP1	tp1_vendor_dim (cln_vf_kred cln_t001s cln_adrc)	tp1_vendor_dim (cln_vf_kred cln_t001s cln_adrc)	tp1_irb (cln_fi5000 cln_ocrlog cln_ekko cln_eban cln_t024 cln_t001)  tp1_ekbe (cln_ekbe)	tp1_irb (cln_fi5000 cln_ocrlog cln_ekko cln_eban cln_t024 cln_t001)  tp1_ekbe (cln_ekbe)
SOURCE	EP1: bsik, bsak, t001, vf_kred, adrc, regup, bkpf EXCEL: overdue_reason, payment_calendar	EP1: bsik, bsak, t001, vf_kred, adrc, regup, bkpf EXCEL: overdue_reason, payment_calendar	BW:FI5000 EP1: zsi_ir_ic_ocrlog, ekko, ekbe, eban, vf_kred, t001, t001s, adrc, t024 EXCEL: tradeshift_invoices	BW:FI5000 EP1: zsi_ir_ic_ocrlog, ekko, ekbe, eban, vf_kred, t001, t001s, adrc, t024 EXCEL: tradeshift_invoices

Two Transformation stages will be necessary to allow scaling for more and more use cases.

## TP1\_EKBE

In the IRB model, we want to determine the invoice quantity and goods received quantity for each purchase order and the last GR posting for the invoice. SAP stores all of this information in table EKBE. However, the document for the goods receipt value is in a different line than the invoice receipt. They are linked together with a reference document. The logic below works as follows.

1. create two helper tables with indexes as the workload can be many million rows
2. find and sum up all invoice lines (Q) and good receipt lines (E)
3. join the newest good receipt for the follow on document of each invoice
4. delete duplicates based on PO for the join of IR/GR PO quantity
5. delete duplicates based on the invoice document to later join on IRB material document.

```

drop table if exists #ekbe_q
drop table if exists #ekbe_e
drop table if exists tp1_ekbe_po;
drop table if exists tp1_ekbe_ref;

create table #ekbe_q(
    [purchase_order] [nvarchar](max) null,
    [material_document] [nvarchar](10) null,
    [year_mat_doc] [decimal](4, 0) null,
    [reference_document] [nvarchar](10) null,
    [year_ref_doc] [decimal](4, 0) null,
    [posting_date] [date] null,
    [entry_date] [date] null,
    [created_by] [nvarchar](max) null,
    [po_history_category] [nvarchar](max) null,
    [plant] [nvarchar](max) null,
    [quantity] [decimal](13, 3) null,
    [file_path] [nvarchar](max) null,
    [aa_number] [decimal](2, 0) null,
    [movement_type] [nvarchar](max) null,
    [download_date] [datetime2](7) null
) on [primary] textimage_on [primary]

create nonclustered index ekbe_q_ref
    on #ekbe_q (reference_document, year_ref_doc);

create table #ekbe_e(
    [purchase_order] [nvarchar](max) null,
    [material_document] [nvarchar](10) null,
    [year_mat_doc] [decimal](4, 0) null,
    [reference_document] [nvarchar](10) null,
    [year_ref_doc] [decimal](4, 0) null,
    [posting_date] [date] null,
    [entry_date] [date] null,
    [created_by] [nvarchar](max) null,
    [po_history_category] [nvarchar](max) null,
    [plant] [nvarchar](max) null,
    [quantity] [decimal](13, 3) null,
    [file_path] [nvarchar](max) null,
    [aa_number] [decimal](2, 0) null,
    [movement_type] [nvarchar](max) null,
    [download_date] [datetime2](7) null
) on [primary] textimage_on [primary]
-- for reverse movement types reverse quantity

create nonclustered index ekbe_e_ref
    on #ekbe_e (reference_document, year_ref_doc, entry_date desc, material_document desc );

update cln_ekbe set quantity = quantity * -1 where debit_credit = 'h' ;

with ekbe_sum_q as (
select
    *,
    sum(quantity) over (
        partition by
            purchase_order
    ) quantity_sum,
    row_number() over (
        partition by
            purchase_order
        order by
            purchase_order
    ) row_num
from
    cln_ekbe
    where (po_history_category = 'q')
    and reference_document <> ''
)
insert into #ekbe_q (
    [purchase_order]
    ,[material_document]
    ,[year_mat_doc]
    ,[reference_document]

```

```

,[year_ref_doc]
,[posting_date]
,[entry_date]
,[created_by]
,[po_history_category]
,[plant]
,[quantity]
,[file_path]
,[aa_number]
,[movement_type]
,[download_date]
)
select
[purchase_order]
,[material_document]
,[year_mat_doc]
,[reference_document]
,[year_ref_doc]
,[posting_date]
,[entry_date]
,[created_by]
,[po_history_category]
,[plant]
,[quantity_sum]
,[file_path]
,[aa_number]
,[movement_type]
,[download_date]
from ekbe_sum_q;

with ekbe_sum_e as (
select
*,  

    sum(quantity) over (
        partition by
            purchase_order
    ) quantity_sum,
    row_number() over (
        partition by
            purchase_order
        order by
            purchase_order
    ) row_num
from
cln_ekbe
where po_history_category = 'e'  

    and reference_document <> ''
)

insert into #ekbe_e (
[purchase_order]
,[material_document]
,[year_mat_doc]
,[reference_document]
,[year_ref_doc]
,[posting_date]
,[entry_date]
,[created_by]
,[po_history_category]
,[plant]
,[quantity]
,[file_path]
,[aa_number]
,[movement_type]
,[download_date]
)
select
[purchase_order]
,[material_document]
,[year_mat_doc]
,[reference_document]
,[year_ref_doc]
,[posting_date]
,[entry_date]

```

```

,[created_by]
,[po_history_category]
,[plant]
,[quantity_sum]
,[file_path]
,[aa_number]
,[movement_type]
,[download_date]
from ekbe_sum_e;

with ekbe_delete_duplicates as (
select
  *,
  row_number() over (
    partition by
      reference_document,
      year_ref_doc
    order by
      reference_document,
      year_ref_doc,
      entry_date desc,
      material_document desc
  ) row_num
from
#ekbe_e
where po_history_category = 'e'
)
delete from ekbe_delete_duplicates
where row_num > 1

select q.material_document,
       q.year_mat_doc,
       q.purchase_order,
       e.material_document as reference_document,
       e.year_ref_doc as year_ref_doc,
       e.entry_date as ref_doc_entry_date,
       e.posting_date as ref_doc_posting_date,
       e.created_by as ref_doc_created_by,
       e.quantity as gr_quantity,
       q.quantity as ir_quantity
into dbo.tp1_ekbe_ref from #ekbe_q as q
left outer join #ekbe_e as e on
q.reference_document = e.reference_document and
q.year_ref_doc = e.year_ref_doc;

select * into dbo.tp1_ekbe_po from tp1_ekbe_ref;

with ekbe_ref_del_duplicates as (
select
  *,
  row_number() over (
    partition by
      year_mat_doc,
      material_document
    order by
      year_mat_doc,
      material_document,
      ref_doc_entry_date desc
  ) row_num
from
tp1_ekbe_ref
)
delete from ekbe_ref_del_duplicates
where row_num > 1;

with ekbe_po_del_duplicates as (
select
  *,
  row_number() over (
    partition by
      purchase_order
    order by

```

```

        purchase_order
    ) row_num
  from
  tp1_ekbe_po
)

delete from ekbe_po_del_duplicates
where row_num > 1

drop table if exists #ekbe_q
drop table if exists #ekbe_e

```

## Transform 3

In this stage, we finally add the business logic.

### P\_TP30\_ALL\_ITEMS

```

alter procedure [dbo].[p_tp30_all_items] as

update tp3_all_items set key_date = eomonth(src_download_date,-1)

update tp3_all_items set due_date = dbo.fc_calculate_due_date_ap(convert(date, baseline_date,104),convert(int,days1),convert(int,days2), deb

update tp3_all_items set arrears_after_net = dbo.fc_calculate_arrears(due_date,key_date)

update tp3_all_items set transaction_key = concat([year],company_code,document_number,line_item)

update tp3_all_items set amount_local = amount_local * -1 where debit_credit = 'h'

update tp3_all_items set amount_document = amount_document * -1 where debit_credit = 'h'

update tp3_all_items set wht = 'withholding tax' where left(reference,3) = 'wht' or right(reference,3) = 'eit'

update tp3_all_items set duplicate = 'x' where charindex('v',reference) <> 0

```

### Key Date

The Key date is a critical column as it reflects the day of reporting from the financial perspective. Therefore all models contain this field to mark the day of reporting. It's always set as the last day of the last month. e.g. today is 2021-04-05 --> Key date = 2021-03-31

### Recalculate Due Date and Arrears after Net

These columns indicate when an item is due and how many days passed after the due date, respectively, how many days until the item will be due. This information is dynamically generated in EP1 and must therefore be recalculated

#### Recalculate due date based on days1 / days 2

```

alter function [dbo].[fc_calculate_due_date_ap]
(@start date, @days1 int, @days2 int, @debit_credit varchar(max), @follow_on_doc varchar(max))
returns date
as
begin

declare @duedate date

if @debit_credit = 's' and @follow_on_doc = ''
begin
set @duedate = @start
end
else
begin
if @days2 = 0
begin
set @duedate = dateadd(day,@days1,@start)
end
else
begin
set @duedate = dateadd(day, @days2, @start)
end
end
end

return @duedate
end

```

#### **Recalculate time passed after due date / before due date**

```

alter function [dbo].[fc_calculate_arrears]
(@start date, @end date)
returns int
as
begin

declare @arrears int

set @arrears = datediff(day, @start, @end)

return @arrears
end

```

## **P\_TP3\_IRB**

```

alter procedure [dbo].[p_tp30_irb] as
    update tp3_irb set scan_date_to_input_date = dbo.fc_get_business_days(scan_date, convert(date, input_date,104))
    update tp3_irb set input_date_to_posting_date = dbo.fc_get_business_days(convert(date, input_date,104), convert(date, entered_on_dat

    update tp3_irb set eiv_autopost= 'yes'
    where invoice_input_channel = 'eiv' and
        ts_error_01 is null and
        ts_error_02 is null and
        ts_error_03 is null and
        ts_error_04 is null and
        ts_error_05 is null and
        ts_error_06 is null and
        ts_error_07 is null and
        ts_error_08 is null

    update tp3_irb set eiv_autopost = 'no'
    where eiv_autopost is null and
        invoice_input_channel = 'eiv'

    update tp3_irb set eiv_autopost= 'n/a'
    where invoice_input_channel <> 'eiv'

update tp3_irb set key_date = eomonth(src_download_date,-1)

update tp3_irb set amount_eur = o2c.fc_convert_currency(currency,convert(decimal(30,2),amount_document), 'eur')

update tp3_irb set gr_quantity = 0
where gr_quantity is null

update tp3_irb set ir_quantity = 0
where ir_quantity is null

```

### Get Business days between two dates

We can not calculate some KPI like scan\_date\_to\_input\_date excluding business days and weekends without knowing the days to exclude. To maintain the days that we should exclude from the calculation, we use a column in the Excel Import INX\_PAYMENT\_CALENDAR inside which business maintains these special days.

	A	B	C	D	E	F
1	DATES	DOMESTIC	OVERSEA_	OVERSEA	CHINA_PUBLIC	HOLIDAY
2	1/1/2020				X	
3	1/2/2020	X		X		
4	1/3/2020					
5	1/4/2020					
6	1/5/2020			X		
7	1/6/2020					
8	1/7/2020					
9	1/8/2020					
10	1/9/2020					
11	1/10/2020					
12	1/11/2020			X		
13	1/12/2020			X		
14	1/13/2020					
15	1/14/2020					
16	1/15/2020	X		X		
17	1/16/2020					
18	1/17/2020					
19	1/18/2020				X	
20	1/19/2020					
21	1/20/2020					
22	1/21/2020					
23	1/22/2020		X			
24	1/23/2020					
25	1/24/2020			X		
26	1/25/2020			X		
27	1/26/2020			X		
28	1/27/2020			X		
29	1/28/2020			X		

This table is then used in a function to calculate the business days between Scan date & Input date resp. Input date & Posting date

```

alter function [dbo].[fc_get_business_days]
(@from datetime, @to datetime)
returns int
as
begin

declare @days int

if @from = '' or @to = ''
begin
    set @days = 0
end
else
begin
    select @days = count(*)+1
    from cln_payment_calendar
    where datepart(dw, dates) not in (1,7)
    and china_public_holiday is null
    and dates > @from and dates <= @to
end

return ( @days )
end

```

## Convert Currency

In SAC, there is no real option to recalculate the exchange rate if more than one currency is involved. E.g. CNY --> EUR no problem. HKD or CNY --> EUR = problem

Therefore the Exchange rate table dbo.currency was created:

	currency1	currency2	exchangerate
1	CNY	EUR	8.23
2	HKD	EUR	9.16
3	EUR	EUR	1
4	USD	EUR	1.18
5	GBP	EUR	0.92
6	JPY	EUR	124.11
7	CHF	EUR	1.08

In this table, the Budget rate is maintained. Currently, this is not controlled with Excel because it only changes once a year. Business will create a ticket for IT to change it manually. For now, it's not necessary to use different rates for different periods.

This table is then used in the convert currency function.

```

alter function [o2c].[fc_convert_currency]
(@curr1 nvarchar(max), @amount1 decimal(30,2), @curr2 nvarchar(max))
returns decimal(30,2)
as
begin

declare @amount2 decimal(30,2)

set @amount2 = @amount1 / (select max(exchangerate)
                           from o2c.currency
                           where currency1 = @curr1
                             and
                             currency2 = @curr2)

return @amount2
end

```

## P\_TP30\_ALL\_CUST\_ITEMS

```

alter procedure [o2c].[p_tp30_all_cust_items] as
update o2c.tp3_all_cust_items set key_date = eomonth(src_download_date,-1)

update o2c.tp3_all_cust_items set due_date = o2c.fc_calculate_due_date_ar(convert(date, baseline_date,104),convert(int, days1),convert(int,da
update o2c.tp3_all_cust_items set arrears_after_net = dbo.fc_calculate_arrears(due_date, key_date)
where clearing_date is null

update o2c.tp3_all_cust_items set arrears_after_net = dbo.fc_calculate_arrears(due_date, clearing_date)
where clearing_date is not null

update o2c.tp3_all_cust_items set amount_local = amount_local * -1 where debit_credit = 'H'

update o2c.tp3_all_cust_items set amount_tax = amount_tax * -1 where debit_credit = 'H'

update o2c.tp3_all_cust_items set amount_document = amount_document * -1 where debit_credit = 'H'

update o2c.tp3_all_cust_items set company_code_currency = 'HKD' where company_code = '0078'

update o2c.tp3_all_cust_items set company_code_currency = 'CNY' where company_code <> '0078'

update o2c.tp3_all_cust_items set amount_eur = o2c.fc_convert_currency(company_code_currency,convert(decimal(30,2),amount_local), 'EUR')

update o2c.tp3_all_cust_items set dispute_created_on = LEFT(dispute_created_on,8)

update o2c.tp3_all_cust_items set dispute_changed_on = LEFT(dispute_changed_on,8)

update o2c.tp3_all_cust_items set dispute_closed_on = LEFT(dispute_closed_on,8)

update o2c.tp3_all_cust_items set dispute_closed_on = null
where left(dispute_closed_on,1) <> '2' and
      left(dispute_closed_on,1) <> ''

update o2c.tp3_all_cust_items set reference_key1 = ''
where left(reference_key1,1) <> '2' and
      left(reference_key1,1) <> ''

update o2c.tp3_all_cust_items set days1_vat = o2c.fc_calculate_days1(payment_terms, reference_key1)
where payment_terms <> '' and
      reference_key1 <> ''

update o2c.tp3_all_cust_items set days1_vat = days1
where payment_terms = '' or
      reference_key1 = ''

update o2c.tp3_all_cust_items set days2_vat = o2c.fc_calculate_days2(payment_terms, reference_key1)
where payment_terms <> '' and
      reference_key1 <> ''

update o2c.tp3_all_cust_items set days2_vat = days2
where payment_terms = '' or
      reference_key1 = ''

update o2c.tp3_all_cust_items set due_date_vat = o2c.fc_calculate_due_date_ar(convert(date, reference_key1,112),days1_vat,days2_vat, debit_c
where payment_terms <> '' and
      reference_key1 <> ''

update o2c.tp3_all_cust_items set due_date_vat = due_date
where payment_terms = '' or
      reference_key1 = ''

update o2c.tp3_all_cust_items set arrears_after_net_vat = dbo.fc_calculate_arrears(due_date_vat, key_date)
where payment_terms <> '' and
      reference_key1 <> '' and
      clearing_date is null

update o2c.tp3_all_cust_items set arrears_after_net_vat = dbo.fc_calculate_arrears(due_date_vat, clearing_date)
where payment_terms <> '' and
      reference_key1 <> '' and
      clearing_date is not null

update o2c.tp3_all_cust_items set arrears_after_net_vat = arrears_after_net
where payment_terms = '' or
      reference_key1 = ''

```

```

update o2c.tp3_all_cust_items set posting_to_clearing_days = dbo.FC_GET_BUSINESS_DAYS(CONVERT(date, posting_date,104), CONVERT(date, clearin
update o2c.tp3_all_cust_items set relevant_for_payment_behavior = 'X'
where
    (company_code = '0078'
    and debit_credit = 'S')
    or
    (company_code = '0083'
    and debit_credit = 'S'
    and not contains(item_text, 'quality')
    and not contains(item_text, 'price')
    and not contains(item_text, 'write')
    and not contains(item_text, 'sample'))
    or
    (company_code = '0289'
    and debit_credit = 'S')
    or
    (company_code = '0369'
    and debit_credit = 'S')
    or
    (company_code = '0199'
    and debit_credit = 'S'
    and reference <> LEFT('INV.',4)
    and not contains(item_text, 'price')
    and not contains(item_text, 'deduction')
    and not contains(item_text, '保证金')
    and not contains(item_text, '质量')
    and not contains(item_text, '质保金')
    and not contains(item_text, '三包')
    and not contains(item_text, '扣款')
    and not contains(item_text, '折扣')
    and not contains(item_text, '折让')
    and not contains(item_text, '税')))

update o2c.tp3_all_cust_items set relevant_for_payment_behavior = ''
where reverse_document = 'X' and MONTH(key_date) < MONTH(clearing_date)

update o2c.tp3_all_cust_items set overdue_rank = '1-30'
where arrears_after_net > 0 and arrears_after_net <= 30

update o2c.tp3_all_cust_items set overdue_rank = '31-90'
where arrears_after_net > 30 and arrears_after_net <= 90

update o2c.tp3_all_cust_items set overdue_rank = '90+'
where arrears_after_net > 90

update o2c.tp3_all_cust_items set overdue_rank = 'not_due'
where arrears_after_net <= 0

update o2c.tp3_all_cust_items set overdue_rank_vat = '1-30'
where arrears_after_net_vat > 0 and arrears_after_net_vat <= 30

update o2c.tp3_all_cust_items set overdue_rank_vat = '31-90'
where arrears_after_net_vat > 30 and arrears_after_net_vat <= 90

update o2c.tp3_all_cust_items set overdue_rank_vat = '90+'
where arrears_after_net_vat > 90

update o2c.tp3_all_cust_items set overdue_rank_vat = 'not_due'
where arrears_after_net_vat <= 0

update o2c.tp3_all_cust_items set overdue_value = amount_local
where relevant_for_payment_behavior = 'X'

update o2c.tp3_all_cust_items set vat_issued = 'VAT issued' where left(REFERENCE,3) = 'INV'
update o2c.tp3_all_cust_items set vat_issued = 'no VAT' where left(REFERENCE,3) <> 'INV'

```

## Recalculate Due Days based on the VAT issue date

Like the P2P side, for the O2C side, too, the due date and Arrears after net must also be recalculated. Additionally, there is the requirement to calculate the due date based on the VAT due date (Reference\_key1 field). This way, we can compare the Due date based on the Baseline date and

Real VAT date in a report. To enable this comparison, we need also to recalculate Days1 and Days2 based on the reference\_key\_1 and payment term table t052:

```

alter function [o2c].[fc_calculate_days1]
(@payment_terms nvarchar(max), @baseline_date nvarchar(max))
returns int
as
begin

--declare @payment_terms as nvarchar(max)
--declare @baseline_date as nvarchar(max)
declare @days1 as int
declare @days1_fixed as int
declare @due_date_special1 as int
declare @month_special1 as int
declare @date as date
declare @day_limit as int

--set @payment_terms = 'z304'
--set @baseline_date = '2021-01-09 00:00:00.0000000'

set @day_limit = (select min(day_limit)
                  from o2c.cln_t052
                  where payment_term = @payment_terms)

set @baseline_date = try_convert(date,@baseline_date)
set @days1 = 0

if @day_limit = 0
begin
    set @days1_fixed = (select max(days1_fixed)
                         from o2c.cln_t052
                         where payment_term = @payment_terms)

    set @due_date_special1 = (
        select max(due_date_special1)
        from o2c.cln_t052
        where payment_term = @payment_terms)

    set @month_special1 = (
        select max(month_special1)
        from o2c.cln_t052
        where payment_term = @payment_terms)
end
else
begin
    set @days1_fixed = (select top 1 days1_fixed
                         from o2c.cln_t052
                         where payment_term = @payment_terms and
                               day_limit >= right(@baseline_date,2)
                         order by day_limit)

    set @due_date_special1 = (
        select top 1 due_date_special1
        from o2c.cln_t052
        where payment_term = @payment_terms and
              day_limit >= right(@baseline_date,2)
        order by day_limit)

    set @month_special1 = (
        select top 1 month_special1
        from o2c.cln_t052
        where payment_term = @payment_terms and
              day_limit >= right(@baseline_date,2)
        order by day_limit)
end

set @days1 = @days1_fixed

if @due_date_special1 > 0 or @month_special1 > 0
begin
    set @date = @baseline_date

    if @days1_fixed > 0
    begin
        set @date = dateadd(day,@days1_fixed,convert(date,@baseline_date))
    end
end

```

```

-- date calculation
set @date = dateadd(month,@month_special1,@date)

if @due_date_special1 = 31
begin
    set @due_date_special1 = right(eomonth(@date),2)
end

set @date = datefromparts(left(@date,4),right(left(@date,7),2),@due_date_special1)

if @date < @baseline_date
begin
    set @baseline_date = @date
end

set @days1 = datediff(day,@baseline_date,@date)

end
return @days1
end

```

There's another function [o2c].[fc\_calculate\_days2] that similarly calculates Days2. Unfortunately, there was no easy way to combine the two without increasing the complexity a lot.

Based on these newly calculated Days1 and Days2, the due date and arrears are calculated again similar to the P2P function.

## Include only specific documents for payment behavior calculation

Many documents like warranty, deduction, and write-offs must be excluded from Business perspective to make the analysis valuable. Therefore we must use the text field to identify the documents:

We will later use the result to aggregate data in STA\_PAYMENT\_BEHAVIOR

```

update o2c.tp3_all_cust_items set relevant_for_payment_behavior = 'X'
where
    (company_code = '0078'
    and debit_credit = 'S')
    or
    (company_code = '0083'
    and debit_credit = 'S'
    and not contains(item_text, 'quality')
    and not contains(item_text, 'price')
    and not contains(item_text, 'write')
    and not contains(item_text, 'sample'))
    or
    (company_code = '0289'
    and debit_credit = 'S')
    or
    (company_code = '0369'
    and debit_credit = 'S')
    or
    (company_code = '0199'
    and debit_credit = 'S'
    and reference <> LEFT('INV.',4)
    and not contains(item_text, 'price')
    and not contains(item_text, 'deduction')
    and not contains(item_text, '保证金')
    and not contains(item_text, '质量')
    and not contains(item_text, '质保金')
    and not contains(item_text, '三包')
    and not contains(item_text, '扣款')
    and not contains(item_text, '折扣')
    and not contains(item_text, '折让')
    and not contains(item_text, '税'))

```

To do that a full-text catalog be created:

- Storage
- Full Text Catalogs
  - fulltext\_all\_cust\_items

As this is a one time step, the code is not included in the stored procedures

Afterwards the table can be indexed:

```
ALTER procedure [o2c].[p_tp30_first] as  
  
create unique index i1 on o2c.tp3_all_cust_items(transaction_key);  
  
create fulltext index on o2c.tp3_all_cust_items (  
    item_text language 0  
) key index i1  
with  
    change_tracking = auto,  
    stoplist=off  
;
```

## Stage

Based on the full data models, several models are created based on business needs. These tables will later be loaded used to be loaded into SAC (could be consumed by any other BI tool)

### **STA\_ALL\_ITEMS**

The ALL\_ITEMS Model contains all items that are either open or have been cleared during the last year. The model therefore, updates monthly and contains data for one year.

### **STA\_OPEN\_ITEMS\_MONTHLY**

The Open items model contains all open items at the end of each month for the last year. e.g., all open items on 31.03, 30.04, 31.05, etc.

By calculating back in time on the cleared items column in the TP3\_ALL\_ITEMS table, we can identify which records were open at the key date. If an item is currently open or has been cleared after a given key date and was posted before that keydate, it was open at that time and therefore, will be printed into the output model.

In this model open items can occur multiple times as an item can be open in Jan and Feb and March etc. Therefore the primary key is the transaction\_key and the key\_date column.

After the open items are selected, the Arrears after net must be recalculated based on the key date.

```

declare @keydate_tp3 date
declare @month int
set @month = 0

set @keydate_tp3 = (select max(key_date) from dbo.tp3_all_items)

drop table if exists sta_open_items_monthly

select
client,
    company_code,
    document_number,
    line_item,
    vendor_number,
    document_type,
    special_gl_indicator,
    payment_block,
    payment_terms,
    scb_indicator,
    gl_account,
    clearing_document,
    currency,
    posting_date,
    clearing_date,
    amount_local,
    amount_document,
    reference,
    item_text,
    [year],
    days1,
    days2,
    baseline_date,
    file_path,
    debit_credit,
    download_date,
    clearing_document_year,
    src_download_date,
    vendor_name,
    accounting_clerk_number,
    accounting_clerk_name,
    accounting_clerk_user,
    reconciliation_account,
    vendor_name_chinese,
    vendor_country,
    trading_partner,
    company_name,
    city,
    company_name_short,
    run_id,
    run_date,
    document_posted_by,
    reason,
    reason_details,
    china_public_holiday,
    domestic_3rd_payment,
    oversea_3rd_payment,
    oversea_ic_payment,
    key_date,
    due_date,
    arrears_after_net,
    transaction_key,
    wht,
    wht,
    duplicate
into dbo.sta_open_items_monthly
from dbo.tp3_all_items
where posting_date <= @keydate_tp3 and
      ( clearing_date is null or clearing_date > @keydate_tp3 );

while (@month > -11)
begin

set @keydate_tp3 = eomonth(dateadd(month,-1,@keydate_tp3))

```

```
insert into dbo.sta_open_items_monthly (
client,
    company_code,
    document_number,
    line_item,
    vendor_number,
    document_type,
    special_gl_indicator,
    payment_block,
    payment_terms,
    scb_indicator,
    gl_account,
    clearing_document,
    currency,
    posting_date,
    clearing_date,
    amount_local,
    amount_document,
    reference,
    item_text,
    [year],
    days1,
    days2,
    baseline_date,
    file_path,
    debit_credit,
    download_date,
    clearing_document_year,
    src_download_date,
    vendor_name,
    accounting_clerk_number,
    accounting_clerk_name,
    accounting_clerk_user,
    reconciliation_account,
    vendor_name_chinese,
    vendor_country,
    trading_partner,
    company_name,
    city,
    company_name_short,
    run_id,
    run_date,
    document_posted_by,
    reason,
    reason_details,
    china_public_holiday,
    domestic_3rd_payment,
    oversea_3rd_payment,
    oversea_ic_payment,
    key_date,
    due_date,
    arrears_after_net,
    transaction_key,
    wht,
    wht,
    duplicate)
select
client,
    company_code,
    document_number,
    line_item,
    vendor_number,
    document_type,
    special_gl_indicator,
    payment_block,
    payment_terms,
    scb_indicator,
    gl_account,
    clearing_document,
    currency,
    posting_date,
    clearing_date,
    amount_local,
    amount_document,
    reference,
```

```

item_text,
[year],
days1,
days2,
baseline_date,
file_path,
debit_credit,
download_date,
clearing_document_year,
src_download_date,
vendor_name,
accounting_clerk_number,
accounting_clerk_name,
accounting_clerk_user,
reconciliation_account,
vendor_name_chinese,
vendor_country,
trading_partner,
company_name,
city,
company_name_short,
run_id,
run_date,
document_posted_by,
reason,
reason_details,
china_public_holiday,
domestic_3rd_payment,
oversea_3rd_payment,
oversea_ic_payment,
null,
due_date,
arrears_after_net,
transaction_key,
wht,
duplicate
from dbo.tp3_all_items
where posting_date <= @keydate_tp3 and
( clearing_date is null or clearing_date > @keydate_tp3 );

update dbo.sta_open_items_monthly set key_date = @keydate_tp3
where key_date is null

set @month = @month -1
end

update dbo.sta_open_items_monthly set arrears_after_net = dbo.fc_calculate_arrears(due_date,key_date)

exec o2c.p_execute_etl_function @imp_function = 'remove_zero', @imp tablename = 'sta_open_items_monthly', @schema = 'dbo'

```

## **STA\_IRB\_FULL**

This model contains the key\_date month + 12 months of prior data.

## **STA\_IRB\_MONTHLY**

STA\_IRB\_MONTHLY is the only model that can not be completely rebuilt with each run. The model should reflect the last 12 months of parked invoices on the second business day of each month. Unfortunately, the Invoice receipt book has no column to check when the change from parked to posted occurred. e.g., in the model of the open items, we can use the clearing date.

This means that we have to freeze the first result of each month in this table with every run. Because the state changes afterward, it causes the following restrictions that are acceptable by the local business.

1. Extension of this model with another column can not be done for past data. (unless there is a link)
2. Reporting can never be at the exact 2nd working day of the new month as the pipeline run is weekly and can not be adjusted with this business logic.
3. If there is an error in the weekly run, the result might be different for that month

There is a status table that we already imported to data lake during the project that contains the status changed for each invoice. ZSI\_IR\_IC\_STATLG. With this table, we could calculate back and find the parked status for each invoice. Unfortunately, there was not enough

| time to implement this during the project. To implement this change would be up to future developers 😊

```

alter procedure [dbo].[p_sta_irb_monthly] as
-- monthly: insert all lines into the table that don't have the same month

declare @keydate_tp3 date
declare @keydate_sto date

set @keydate_tp3 = (select max(key_date) from tp3_irb)
set @keydate_sto = (select max(key_date) from sta_irb_monthly)

if @keydate_tp3 > @keydate_sto
begin

insert into [dbo].[sta_irb_monthly]
([activity_status]
,[year]
,[cash_discount2]
,[clearing_date]
,[company_code]
,[currency]
,[entered_on_date]
,[fi_document_no]
,[input_date]
,[invoice_date]
,[invoice_input_channel]
,[invoice_state]
,[legal_entity]
,[ocr_invoice_correction]
,[ocr_stack_name]
,[ocr_supplier_correction]
,[paying_date]
,[posting_date]
,[purchase_order]
,[reference]
,[scantime]
,[source_system]
,[state_auto_posting]
,[vendor_number]
,[team_hist]
,[transaction_key]
,[amount_document]
,[amount_local]
,[tax_amount]
,[auth_legal_region]
,[activity_status_description]
,[invoice_state_description]
,[currency_description]
,[supplier_correction_description]
,[state_auto_posting_description]
,[invoice_correction_description]
,[scan_date]
,[delivery_note]
,[mat_document_no]
,[src_download_date]
,[ts_filename]
,[ts_fapiao_code]
,[purchasing_group]
,[po_created_by]
,[po_company_code]
,[requisitioner]
,[pr_creator]
,[company_name]
,[company_name_short]
,[purchaser_name]
,[gr_quantity]
,[ir_quantity]
,[ts_error_01]
,[ts_error_02]
,[ts_error_03]
,[ts_error_04]
,[ts_error_05]
,[ts_error_06]
,[ts_error_07]
,[ts_error_08]
)
```

```

,[ts_po_remark]
,[accounting_clerk_name]
,[accounting_clerk_number]
,[accounting_clerk_user]
,[reconciliation_account]
,[trading_partner]
,[vendor_country]
,[vendor_name]
,[vendor_name_chinese]
,[reference_document]
,[year_ref_doc]
,[ref_doc_entry_date]
,[ref_doc_posting_date]
,[ref_doc_created_by]
,[scan_date_to_input_date]
,[input_date_to_posting_date]
,[eiv_autopost]
,[key_date]
,[amount_eur])

select [activity_status]
,[year]
,[cash_discount2]
,[clearing_date]
,[company_code]
,[currency]
,[entered_on_date]
,[fi_document_no]
,[input_date]
,[invoice_date]
,[invoice_input_channel]
,[invoice_state]
,[legal_entity]
,[ocr_invoice_correction]
,[ocr_stack_name]
,[ocr_supplier_correction]
,[paying_date]
,[posting_date]
,[purchase_order]
,[reference]
,[scantime]
,[source_system]
,[state_auto_posting]
,[vendor_number]
,[team_hist]
,[transaction_key]
,[amount_document]
,[amount_local]
,[tax_amount]
,[auth_legal_region]
,[activity_status_description]
,[invoice_state_description]
,[currency_description]
,[supplier_correction_description]
,[state_auto_posting_description]
,[invoice_correction_description]
,[scan_date]
,[delivery_note]
,[mat_document_no]
,[src_download_date]
,[ts_filename]
,[ts_fapiao_code]
,[purchasing_group]
,[po_created_by]
,[po_company_code]
,[requisitioner]
,[pr_creator]
,[company_name]
,[company_name_short]
,[purchaser_name]
,[gr_quantity]
,[ir_quantity]
,[ts_error_01]
,[ts_error_02]
,[ts_error_03]

```

```

,[ts_error_04]
,[ts_error_05]
,[ts_error_06]
,[ts_error_07]
,[ts_error_08]
,[ts_po_remark]
,[accounting_clerk_name]
,[accounting_clerk_number]
,[accounting_clerk_user]
,[reconciliation_account]
,[trading_partner]
,[vendor_country]
,[vendor_name]
,[vendor_name_chinese]
,[reference_document]
,[year_ref_doc]
,[ref_doc_entry_date]
,[ref_doc_posting_date]
,[ref_doc_created_by]
,[scan_date_to_input_date]
,[input_date_to_posting_date]
,[eiv_autopost]
,[key_date]
,[amount_eur]
from tp3_irb
where invoice_state = '11' or invoice_state = '12'

-- update slowly changing dimensions vendor and company and ekko

update
    irb
set
    irb.accounting_clerk_name = vendor.accounting_clerk_name,
    irb.accounting_clerk_number = vendor.accounting_clerk_number,
    irb.accounting_clerk_user = vendor.accounting_clerk_user,
    irb.vendor_name = vendor.vendor_name,
    irb.vendor_name_chinese = vendor.vendor_name_chinese,
    irb.vendor_country = vendor.vendor_country,
    irb.company_name = company.company_name,
    irb.company_name_short = company.company_name_short,
    irb.purchaser_name = purchaser.purchaser_name
from
    sta_irb_monthly as irb
        left join
            tp1_vendor_dimension as vendor on
                cast(irb.vendor_number as int) = cast(vendor.vendor_number as int) and
                    irb.company_code = vendor.company_code
        left join
            cln_t001 as company on
                irb.company_code = company.company_code
        left join
            cln_t024 as purchaser on
                irb.purchasing_group = purchaser.purchasing_group

```

## **STA\_ALL\_CUST\_ITEMS**

This model is similar to STA\_ALL\_ITEMS, just for the o2c side.

## **STA\_OPEN\_CUST\_ITEMS**

This model is similar to STA\_OPEN\_ITEMS\_MONTHLY just for the o2c side.

## **STA\_EFLOW\_CLR**

This model contains EFLOW CLR workflow information. To track the KPI on release time.

```

ALTER PROCEDURE [o2c].[p_sta_eflow_clr]
AS
BEGIN

drop table if exists o2c.sta_eflow_clr
select processname,incident,steplabel,status,
case
    when status = 1 then 'open'
    when status =3 then 'Complete'
    when status = 4 then 'Return'
    when status = 7 then 'Rejected'
end as StatusText,
substatus ,taskuser, assignedtouser,starttime,endtime,task_id,
o2c.fc_cal_eflow_duration(starttime,endtime) as 'durationmin'
into o2c.sta_eflow_clr
from o2c.cln_eflowtask
where processname = 'P047_CLR_01'
and endtime >='2020-01-01'

```

It involves a function to dynamically calculate the time in minutes between start and end time for each step similar to the calculation logic in eflow system.

```

alter function [o2c].[fc_cal_eflow_duration]
(@from datetime, @to datetime)
returns int
as
begin

declare @days int
declare @durationmin int
declare @starttime time
declare @endtime time

if @from = '' or @to = ''
begin
    set @days = 0
    set @durationmin = 0
end
else
begin
    set @days = dbo.fc_get_business_days(@from,@to) - 1
    set @starttime = cast(@from as time(0))
    set @endtime = cast(@to as time(0))
-- only consider working time from 08:30 to 17:30
    if @starttime >= cast('00:00:00' as time(0)) and @starttime <= cast('08:30:00' as time(0))
    begin
        set @starttime = cast('08:30:00' as time(0))
    end
    if @endtime >= cast('00:00:00' as time(0)) and @endtime <= cast('08:30:00' as time(0))
    begin
        set @endtime = cast('08:30:00' as time(0))
    end

-- set max time to 17:30:00
    if @starttime >= cast('17:30:00' as time(0)) and @starttime <= cast('23:59:59' as time(0))
    begin
        set @starttime = cast('17:30:00' as time(0))
    end
    if @endtime >= cast('17:30:00' as time(0)) and @endtime <= cast('23:59:59' as time(0))
    begin
        set @endtime = cast('17:30:00' as time(0))
    end

set @durationmin = datediff(minute,@starttime,@endtime) + @days * 540

end

return ( @durationmin )
end

```

## **STA\_EFLOW\_LIKP**

This model combines the eflow for delivery note release with the delivery note data from SAP and customer data. This way business can identify which customers cause high volumes of handling.

```

alter procedure [o2c].[p_sta_eflow_likp] as
-- create temporary table for transforming...

drop table if exists o2c.#cln_dnsum
select a.processname,a.incident,right(concat('00000',dntbr),10) as delivery_nr ,
b.task_id, b.steplabel, b.status, b.substatus,
case
    when b.status = 1 then 'open'
    when b.status = 3 then 'complete'
    when b.status = 4 then 'return'
    when b.status = 7 then 'rejected'
end as statustext,
b.starttime, b.endtime,
case
    when b.status = 7 then 0
    else o2c.fc_cal_eflow_duration(b.starttime,b.endtime)
end as 'durationmin'
into o2c.#cln_dnsum
from o2c.cln_eflowdn as a
left outer join o2c.cln_eflowtask as b
on a.processname = b.processname and a.incident = b.incident
where b.processname = 'p048_gr_01' and b.endtime >='2020-01-01'

-- delete temporary table

delete from o2c.#cln_dnsum where task_id is null ;

with dn_duplicates as (
select *,
row_number() over (
partition by
delivery_nr
order by
delivery_nr,
convert(datetime,starttime) desc
) row_num
from
o2c.#cln_dnsum
)
delete from dn_duplicates
where row_num > 1

drop table if exists o2c.sta_eflow_likp
select a.*,
d.customer_country,
d.trading_partner,
d.customer_name1,
d.customer_name2,
c.delivery_status,
c.gi_status,
c.billing_status,
b.status,
b.statustext,
b.substatus,
b.task_id,
isnull(durationmin,-1) durationmin
into o2c.sta_eflow_likp
from o2c.cln_likp a
left outer join o2c.#cln_dnsum b
on a.delivery_nr = b.delivery_nr and
convert(date,a.rel_cre_date) = convert(date,b.starttime)
left outer join o2c.cln_vbuk c
on a.delivery_nr = c.delivery_nr
left outer join o2c.cln_kna1 d
on a.soldtoparty = d.customer_number
order by durationmin desc

drop table if exists o2c.#cln_dnsum

exec o2c.p_execute_etl_function @imp_function = 'remove_zero', @imp tablename = 'sta_eflow_likp', @schema = 'o2c';

with measure_based_on_task_id as (

```

```

select *,
       row_number() over (
           partition by
               task_id
           order by
               task_id
       ) row_num
  from
    o2c.sta_eflow_likp
)
update o2c.sta_eflow_likp set o2c.sta_eflow_likp.durationmin_task_id = measure_based_on_task_id.durationmin
from measure_based_on_task_id
where measure_based_on_task_id.delivery_nr = o2c.sta_eflow_likp.delivery_nr and
row_num = 1

```

## **STA\_FI1000**

This model provides a usefully pivoted version of the BW FI1000 cube to report DSO and overdue percentages.

## **STA\_PAYMENT\_BEHAVIOR**

This unique routine takes data from both the STA\_OPEN\_CUST\_ITEMS and STA\_ALL\_CUST\_ITEMS model to create a model that can compare overdues and total sales for customers. To do this, the single open line items are clustered in overdue 30 / 60 / 90+ days and aggregated for each customer. The Dashboard solution contains a scoring model that then clusters the customers in good and bad payers.

```

alter procedure [o2c].[p_sta_payment_behavior] as
begin

drop table if exists o2c.sta_payment_behavior
drop table if exists #temp_od
drop table if exists #temp_sum
drop table if exists #temp_cust;

with sum_od as (
    select distinct credit_account,
        key_date,
        overdue_rank_vat,
        sum(convert(float,overdue_value) ) over (
            partition by
                convert(varchar(10),credit_account),
                key_date,
                convert(varchar(10),overdue_rank_vat)
            order by
                convert(varchar(10),credit_account),
                key_date ,
                convert(varchar(10),overdue_rank)
        ) overdue_value_by_ca
    from
        o2c.sta_open_cust_items
    where credit_account is not null
    and relevant_for_payment_behavior = 'x')

select * into #temp_od from (
    select credit_account, key_date, overdue_rank_vat,overdue_value_by_ca
    from sum_od
) t
pivot (
    sum(overdue_value_by_ca)
    for overdue_rank_vat in (
        [1-30], [90+], [31-90], [not_due]
    )
)as p;

with sum_sales as (
    select distinct credit_account,
        sum(convert(float,amount_local) ) over (
            partition by
                convert(varchar(10),credit_account)
            order by
                convert(varchar(10),credit_account)
            ) sales_by_ca
    from
        o2c.sta_all_cust_items
    where (document_type = 'dg' or
        document_type = 'dr') and
        posting_date between eomonth(dateadd(month,-12,key_date)) and key_date)

select * into #temp_sum
from sum_sales

select * into #temp_cust
from o2c.tp1_customer
where customer_number = credit_account;

with del_cust_duplicates as (
    select
        *,
        row_number() over (
            partition by
                credit_account
            order by
                credit_account,
                company_code
        ) row_num
    from
        #temp_cust
)

```

```

delete from del_cust_duplicates
where row_num > 1

select od.credit_account,
       od.key_date,
       od.[1-30],
       od.[31-90],
       od.[90+],
       od.[not_due],
       su.sales_by_ca,
       cust.credit_limit,
       cust.credit_reporting_group,
       cust.customer_country,
       cust.customer_name_chinese,
       cust.last_internal_review,
       cust.reconciliation_account,
       cust.risk_category,
       cust.trading_partner,
       cust.credit_block
  into o2c.sta_payment_behavior
  from #temp_od as od
    left join
      #temp_sum as su on
        od.credit_account = su.credit_account
    left join
      #temp_cust as cust on
        right('0000000000' + convert(varchar(10),od.credit_account), 10) = cust.credit_account

drop table if exists #temp_od
drop table if exists #temp_sum
drop table if exists #temp_cust

end

```

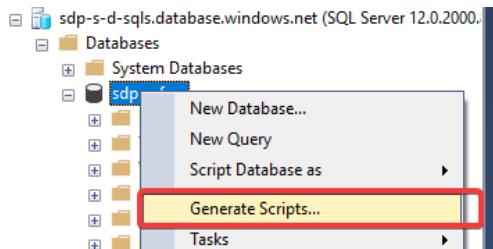
## STA\_\*SCHEMA

Every STA\_\* table also has a corresponding SCHEMA table for business to speed up the mapping process in SAC.

## Transport Changes from D-System to P-System

There is no automated transport link between D-SQL and P-SQL. Therefore the changes need to be transported by hand.

To avoid mistakes I would recommend in case of a bigger change to delete all stored procedures from P-System and Create all D-System Procedures again



We can do this with the generate script function in the SQL Server. Tables, procedures, and functions can be exported and then be run in the P-System.

## SAP Analytics Cloud

## Scandate to Input Date

**1.17 (+0.03)**

△ 2021 Feb



## Notes

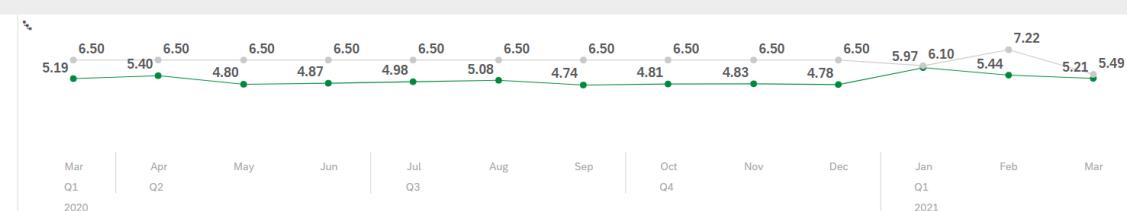
1. Job was stuck at weekend(6-7th Feb) , and recovered on 8th Feb, it caused the invoice scanned on 5th Feb , but input in 8th Feb.

2. Server is not available on 25th, and IT who is responsible for this interface also made a mistake, it caused the invoice scanned on 25th Feb , but input to SAP later, some of them enter SAP on 26th, 27th in Feb, especially 662 invoices enter SAP on 2nd of Mar.

## Input Date to Post Date

**5.21 (-0.23)**

△ 2021 Feb

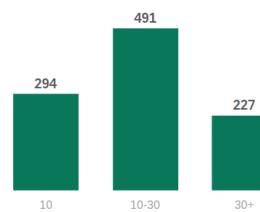


## Parked Invoice

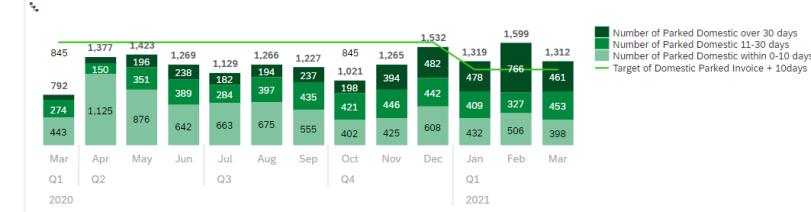
## Domestic Items

**1,012 3.06**

Domestic Amount MEUR



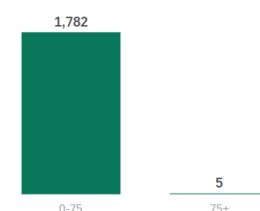
## Trend



## Oversea Items

**1,787 12.99**

Oversea Amount MEUR



## Trend



## Notes:

exclude the next month's parked invoices.

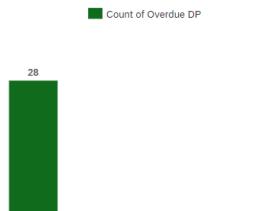
## Downpayment

## Total Open Items

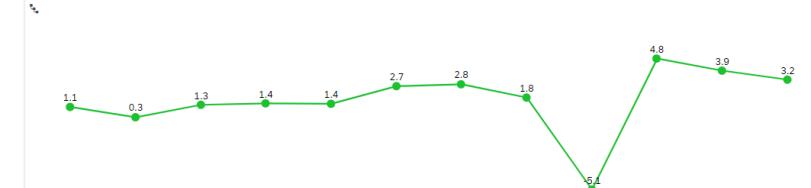
**5,264 (+4,640)**

△ 2021 Feb

Total Open Amount MEUR



## Trend

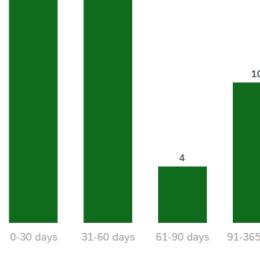


## Overdue Items

**58 (-79)**

△ 2021 Feb

Overdue Amount in MCNY



## Trend



## Notes:

1-30 days: 0199 vendor: 32000224 212 items cleared in Feb

2-31-60 days: 0371 vendor: 32000224 22 items cleared in Mar

## Exchange Rate in 2021 (Budget Rate)

<a href="#">fx Exchange Rate C...</a>	<a href="#">fx Exchange Rate U...</a>	<a href="#">fx Exchange Rate ...</a>	<a href="#">fx Exchange Rate H...</a>	<a href="#">fx Exchange Rate C...</a>	<a href="#">fx Exchange Rate J...</a>
8.23	1.18	0.92	9.16	1.08	124.11

## File Server

<a href="#">📁 Network &gt; schaeffler.com &gt; herzogenaurach &gt; DATA &gt; SZ-HZA-Z &gt; Projects &gt; BW_SAC_Upload &gt; FI &gt; FSSC_IMPROVEMENT_FRAMEWORK &gt;</a>				
Name	Date modified	Type	Size	
Excel Upload	4/14/2021 9:58 AM	File folder		
INPUT	4/14/2021 8:59 AM	File folder		
OUTPUT	4/1/2021 11:26 AM	File folder		

The STA\_\* Models are printed into the OUTPUT Folder on the SAC File Server.

Name	Date modified	Type	Size
sta_all_cust_items.csv	4/14/2021 10:50 A...	Microsoft Excel Co...	467,407 KB
STA_ALL_ITEMS.csv	4/13/2021 7:46 PM	Microsoft Excel Co...	429,334 KB
sta_eflow_clr.csv	4/14/2021 10:49 A...	Microsoft Excel Co...	4,570 KB
sta_eflow_likp.csv	4/14/2021 10:49 A...	Microsoft Excel Co...	47,038 KB
sta_fi1000.csv	4/14/2021 10:49 A...	Microsoft Excel Co...	8,471 KB
STA_IRB_FULL.csv	4/13/2021 7:46 PM	Microsoft Excel Co...	180,337 KB
STA_IRB_MONTHLY.csv	4/13/2021 7:46 PM	Microsoft Excel Co...	12,836 KB
sta_open_cust_items.csv	4/14/2021 10:50 A...	Microsoft Excel Co...	710,670 KB
STA_OPEN_ITEMS_MONTHLY.csv	4/13/2021 7:46 PM	Microsoft Excel Co...	480,259 KB
sta_payment_behavior.csv	4/14/2021 10:49 A...	Microsoft Excel Co...	1,538 KB

INPUT contains all the Excel information that is imported during the INGEST step in Data Factory

Name	Date modified	Type	Size
CONFIG.xlsx	4/1/2021 10:34 PM	Microsoft Excel W...	19 KB
OVERDUE_REASON.xlsx	4/13/2021 6:34 PM	Microsoft Excel W...	103 KB
PAYMENT_CALENDAR.xlsx	4/8/2021 5:00 PM	Microsoft Excel W...	27 KB
SAMPLE_ORDERS.xlsx	4/12/2021 10:34 A...	Microsoft Excel W...	64 KB
TRADESHIFT_INVOICES.xlsx	4/1/2021 5:46 PM	Microsoft Excel W...	1,175 KB

Excel Upload contains all additional models currently based on Excel or extensions to the current models joined during the import to SAC.

<a href="#">FF</a>	4/9/2021 3:49 PM	File folder
<a href="#">AP Data Input.xlsx</a>	4/14/2021 9:58 AM	Microsoft Excel Worksheet
<a href="#">AP KPI Per vendor.xlsx</a>	3/28/2021 3:55 PM	Microsoft Excel Worksheet
<a href="#">Business Division.XLSX</a>	4/9/2021 4:51 PM	Microsoft Excel Worksheet
<a href="#">T024.XLSX</a>	3/24/2021 11:03 AM	Microsoft Excel Worksheet

## SAC Models

All models and stories are saved in this folder owned by Summer. Additional members can be added by contacting her.

Currently, the STA\_\* tables map to the following models:

SQL_TABLE	SAC_MODEL
STA_IRB_FULL	POC_INVOICES_IRB

<b>SQL_TABLE</b>	<b>SAC_MODEL</b>
STA_OPEN_ITEMS	PAYMENTS_OPEN_ITEMS
STA_ALL_ITEMS	PAYMENTS_ALL_ITEMS
STA_IRB_MONTHLY	INVOICES_IRB_SNAP
STA_OPEN_CUST_ITEMS	OPEN ITEMS NEW AR MODEL
STA_EFLOW_CLR	E-FLOW NEW CLR MODEL
STA_EFLOW_LIKP	E-FLOW NEW DN MODEL
STA_FI1000	DSO & OVERDUE RATE BW REPOT
STA_ALL_CUST_ITEMS	ALL ITEMS NEW AR MODEL
STA_PAYMENT_BEHAVIOR	3RD AR PAYMENT BEHAVIOR NEW

## **Model Settings**

### **Story Overview**

#### **P2P Stories**

AP Processing and DP

## Scandate to Input Date

**1.17 (+0.03)**

△ 2021 Feb



## Notes

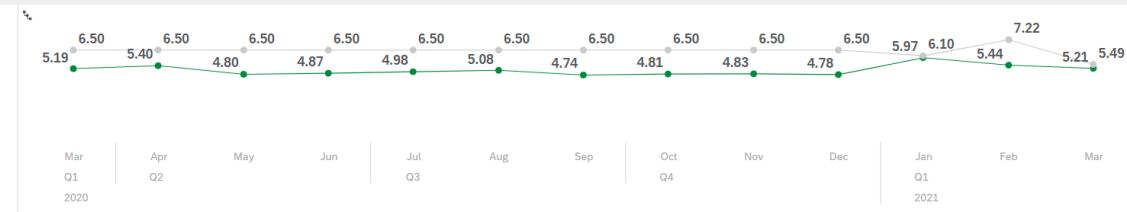
1. Job was stuck at weekend(6-7th Feb) , and recovered on 8th Feb, it caused the invoice scanned on 5th Feb , but input in 8th Feb.

2. Server is not available on 25th, and IT who is responsible for this interface also made a mistake, it caused the invoice scanned on 25th Feb , but input to SAP later, some of them enter SAP on 26th, 27th in Feb, especially 662 invoices enter SAP on 2nd of Mar.

## Input Date to Post Date

**5.21 (-0.23)**

△ 2021 Feb

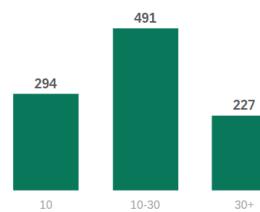


## Parked Invoice

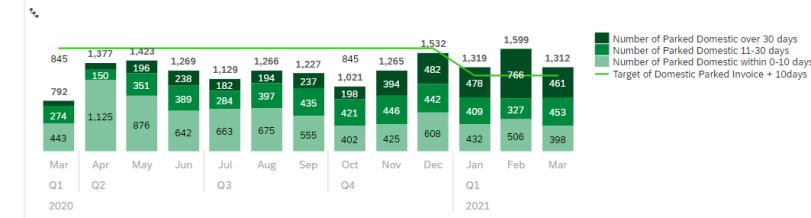
Domestic Items

**1,012 3.06**

Domestic Amount MEUR



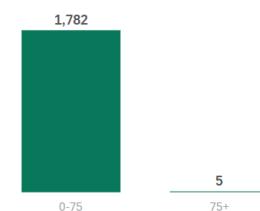
Trend



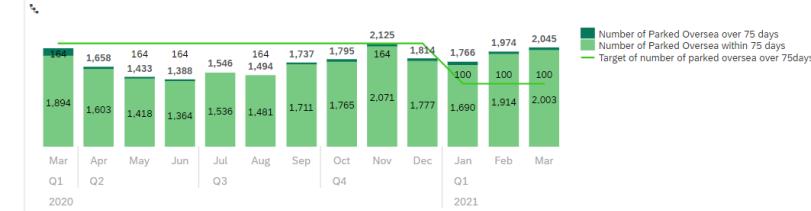
Oversea Items

**1,787 12.99**

Oversea Amount MEUR



Trend



## Notes:

exclude the next month's parked invoices.

## Downpayment

Total Open Items

**5,264 (+4,640)**

△ 2021 Feb

Total Open Amount MEUR

**34.4 m (+8.9)**

△ 2021 Feb

Count of Overdue DP

28

Trend



Overdue Items

**58 (-79)**

△ 2021 Feb

Overdue Amount in MCNY

**3.2 m (-0.7)**

△ 2021 Feb

Count of Overdue DP

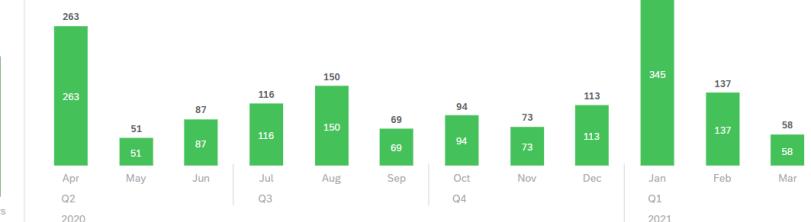
16

4

10

10

Trend



## Notes:

1-30 days: 0199 vendor: 32000224 212 items cleared in Feb

2-31-60 days: 0371 vendor: 32000224 22 items cleared in Mar

## Exchange Rate in 2021 (Budget Rate)

## SCHAFFLER

### Posting Speed

13:55 April 15, 2021

#### SELECT VENDOR

- Supplier Number
- All
- 1000 (Diverse Kreditoren Deutschland)
- 100629 (Karl Berrang GmbH Schraubengrosshandel)
- 102801 (Parker-Hannifin GmbH Engineered Materials Group Europe)
- 10626 (Roessler Oberflaechentechnik GmbH Werk Memmelsdorf)
- 10656 (Hema GmbH Maschinen-U. Apparateschutz)
- 10690 (Baumann Federn AG)
- 10806 (C. Hilzinger-Thum GmbH & Co. KG)
- 11047 (Legrom GmbH & Co.)
- 1107 (Dr. Kaiser Diamant-Werkzeuge GmbH & Co. KG)
- 11431 (Hesmer Antriebstechnik)

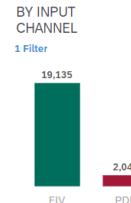
#### SCANDATE TO INPUT DATE

1 Filter

**1.17**

BY INPUT CHANNEL

1 Filter



#### TREND SCANDATE TO INPUT DATE

%



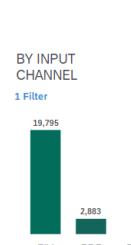
#### INPUT DATE TO POST DATE

1 Filter

**5.22**

BY INPUT CHANNEL

1 Filter



#### TREND INPUTDATE TO POST DATE

%



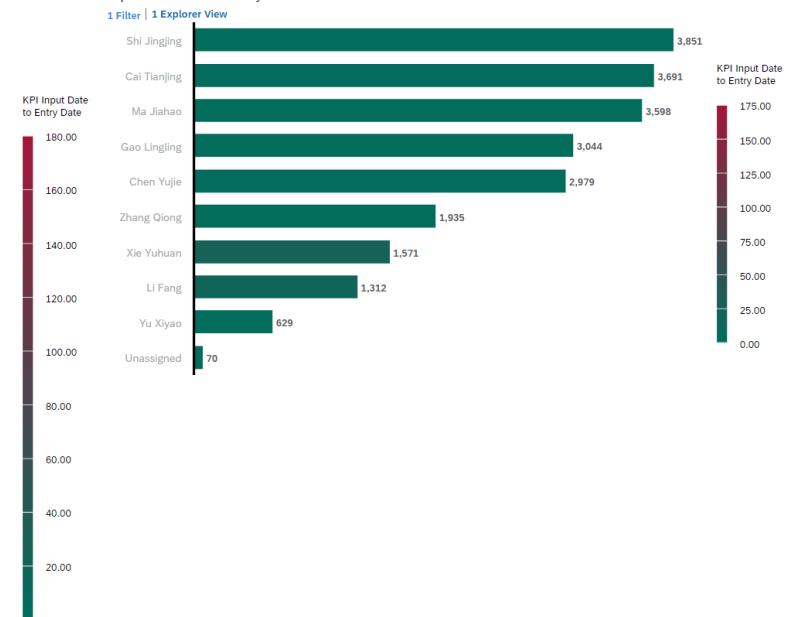
#### Input Date to Postdate by Supplier

2 Filters | Top N (AUTO) | 1 Explorer View

267682 欧盛(斯耐特材料)(太合)有限公司	449
253050 上海宝钢钢材加工配送有限公司	346
248670 施耐德电气(中国)有限公司	332
288288 苏州三星塑料制品有限公司	310
280225 光显米(中国)精密机械贸易有限公司	288
255270 宝家洁不锈钢金属制品制造有限公司	223
293237 常州凯恩斯汽车零部件有限公司	219
31177 安徽中鼎密封件股份有限公司	201
274675 上海科伦机械有限公司	192
249211 黄浦区工业品超市(上海)有限公司	189
600566 圣戈班磨料磨具(上海)有限公司	179
252656 巴杰夫传动部件(上海)有限公司	172
354455 上海君博新材料工业有限公司	169
60004 太合利雷臣机电设备有限公司	152
344537 苏州汽美橡胶有限公司	150
296042 苏州傲闻机电设备有限公司	133
354614 新莱应材制造有限公司	133
248589 上百模具有限公司(上海)有限公司	125
257063 无锡创亿自动化技术有限公司	124
354101 无锡品研精密制造有限公司	123
365278 太仓英士利电子有限公司	123
280136 苏州工具贸易(上海)有限公司	121
287810 上海新箭机电设备有限公司	120
358570 上海爱光电力设备有限公司	119
249643 西子(浙江)自动化工程有限公司	118
277903 上海市陆威试验有限公司	114
342482 盘锦工业(大连)有限公司	112
259160 克恩-里伯斯(太合)有限公司	111
60188 上海溢鸿国际贸易有限公司	110
346234 欧沃盖尔专用工具(昆山)有限公司	108
350059 太合雷鸣纳米新材料科技有限公司	108
364771 太合拓自动化设备有限公司	106
364683 昆山唯肯汽车零部件有限公司	105
354459 苏州兴华印刷包装技术有限公司	102
60231 绿叶电子元器件(上海)有限公司	102
60316 易衡电子(上海)有限公司	102
267807 必达乐(苏州)有限公司	101
267115 太合利(昆山)有限公司	101
60041 太合利(昆山)有限公司	95
60224 山特微克可逆滑切磨刀具(上海)有限公司	94
250640 长盈精密铸造(苏州)有限公司	93
279202 浙江孚辛电子有限公司	93
600525 摩斯润滑油(中国)有限公司	93
259326 布兰诺工业包装材料(上海)有限公司	93

#### Input Date to Postdate by AP

1 Filter | 1 Explorer View



## Controls

fx Focus Supplier

- All
- 1000 (1000)
- 100629 (Karl Berrang GmbH Schraubengrosshandel)
- 10149 (Scherer-Feinbau GmbH )
- 102801 (Parker-Hannifin GmbH Engineered Materials Group Europe)
- 10626 (Roessler Oberflaechechnik GmbH Werk Memmelsdorf)
- 10633 (Kuhmichel Abrasiv GmbH )
- 1065 (Krafft GmbH )
- 10656 (Hema GmbH Maschinen- U. Apparateschutz)
- 10671 (Kasto Maschinenbau GmbH & Co. KG)
- 10690 (Baumann Federn AG )
- 10806 (C. Hilzinger-Thum GmbH & Co. KG )
- 11047 (Legrom GmbH & Co. )
- 1107 (Dr. Kaiser Diamantwerkzeuge GmbH & Co. KG)
- 11443 (Carl Schloesser GmbH & Co. KG )
- 1200 (HSP Schwahlen GmbH )

## TARGET SCANDATE TO INPUT DATE

1 Filter | 1 Variance

**1.17****1.00 (+0.17)**

## DETAILS

1 Filter

**21,906.00**

P\_STI\_C\_STI Weekdays with Supplier improved

**25,593.00**

Scan Date to Input Date

## By Accountant

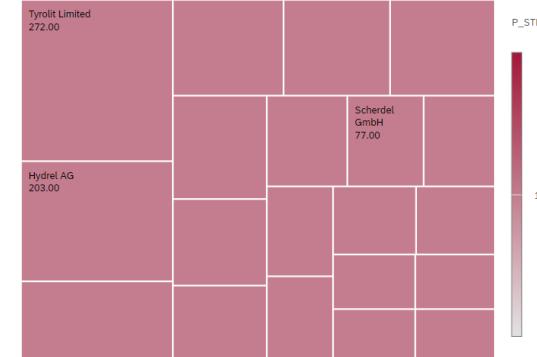
1 Filter | 1 Variance



## ANALYSIS

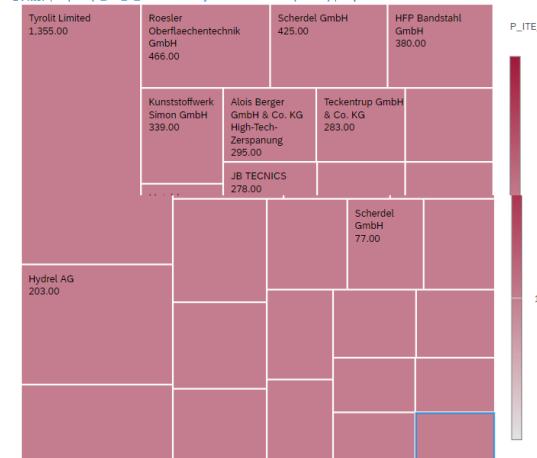
## TARGET KPI Scandate to Input Date

1 Filter | Top 20 (P\_STI\_C\_Total amount of days that can be improved) | Explorer Available



## TARGET KPI Inputdate to Poststdate

1 Filter | Top 20 (P\_ITE\_C\_amount of days that could be improved) | Explorer Available



## TARGET KPI Inputdate to Poststdate

1 Filter | Top 20 (P\_ITE\_C\_amount of days that could be improved) | Explorer Available



## Potential Posting Speed

## TARGET INPUT DATE TO POST DATE

1 Filter | 1 Variance

**5.23****-1.00 (+6.23)****-23,516.00**

## DETAILS

1 Filter

**-23,516.00**

P\_ITE\_C\_Inputdate to Entrydate if Suppliers are improved

**117,177.00**

Input Date to Post Date

## By Accountant

1 Filter | 1 Variance



## Suppliers with worst impact on Scan to Inputdate

1 Filter | Top 20 | 1 Variance



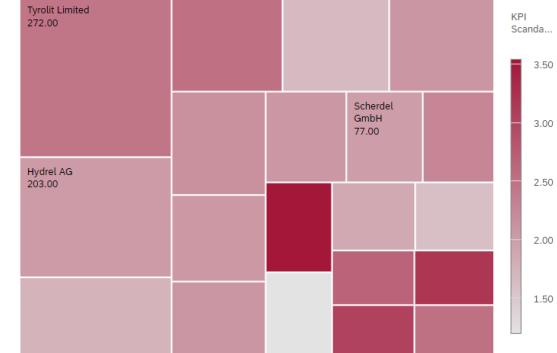
## Suppliers with worst impact on Input to Postdate

1 Filter | Top 20 | 1 Variance



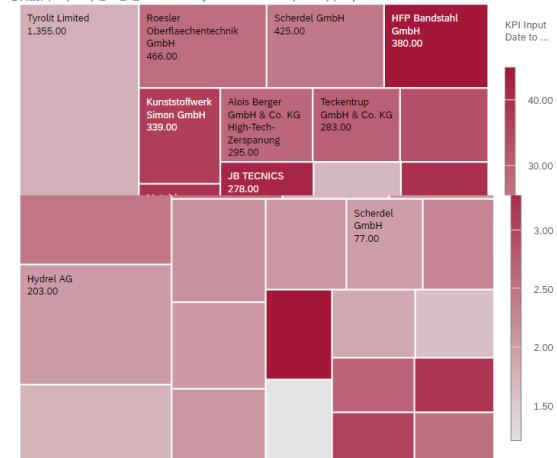
## ACTUAL KPI Scandate to Inputdate

1 Filter | Top 20 (P\_STI\_C\_Total amount of days that can be improved) | Explorer Available



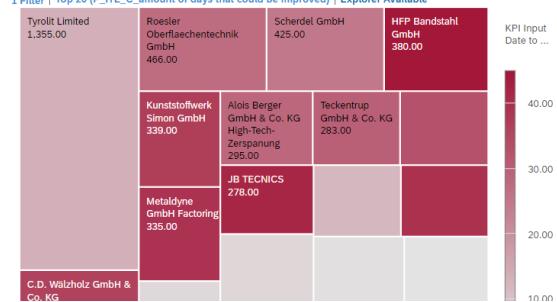
## ACTUAL KPI Inputdate to Postdate

1 Filter | Top 20 (P\_ITE\_C\_amount of days that could be improved) | Explorer Available



## ACTUAL KPI Inputdate to Postdate

1 Filter | Top 20 (P\_ITE\_C\_amount of days that could be improved) | Explorer Available





**SCHAEFFLER**

## Analysis Overdue Downpayments

13:58 April 15, 2021

OVERDUE DOWNPAYMENTS ALL DOWNPAYMENTS

**1 Filter**

**58**

Count of Overdue DP

**26,254,270.17**

Amount of Overdue DP

**1 Filter**

**5,264**

Count of Open DP

**283,209,070.77**

Amount of Open DP

Overdue DP by Items

Overdue DP by Amount

in % | **1 Filter**

**1.10%**

in % | **1 Filter**

**9.27%**

Overdue DP Analysis by Aging

Overdue DP Analysis by Aging

**Top 20**

in --, %

32000224 32000224

667

366331 南方泵业股份有限公司

30

346417 上海国际汽车城发展有限公司

28

253050 上海宝井钢材加工配送有限公司

22

285133 毕马威华振会计师事务所 (特殊普通合

19

1081979 税务局

17

6013335 MSG Gummiformtechnik GmbH

17

283669 西安天地源物业服务管理有限责任公

15

17384 ITP GmbH

12

255965 宁夏纳纳斯燃气集团有限公司

12

347394 中国人寿保险股份有限公司上海市...

11

1082662 舍弗勒 (湘潭) 有限公司工会委...

9

1083007 国家税务总局上海市嘉定区税务局

9

267688 上海虹桥德国外籍人员子女学校

9

279294 中国石化销售股份有限公司上海石...

9

32000440 32000440

9

356806 中信泰富钢铁贸易有限公司

9

60032 中国石化销售股份有限公司江苏苏...

9

362852 上海燃气有限公司

8

367088 上海固都自动化工程有限公司

8

Overdue DP Analysis by CoCd

in --, %

STS 383

IZT 204

SNJ 332

SXT 86

SAM 40

LFC 33

FXR 55

SAB 28

Overdue DP Analysis by Special Gi/L Indicator

in --, %

A 1.024

V 100

I 41

Overdue DP Analysis by User

in --, %

Unassigned 687

CAITAN 103

GAOLGL 83

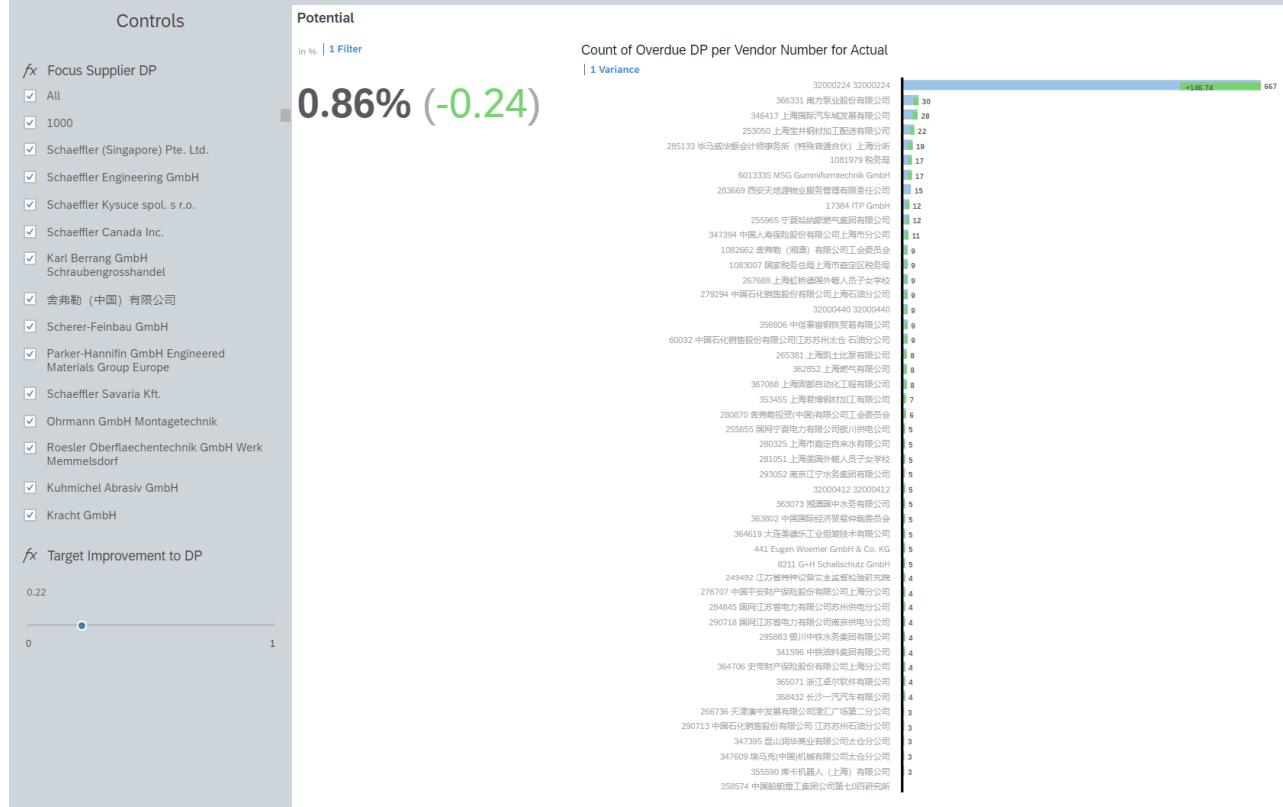
YUXYA 40

LIFN4 35

XIEYHU 41

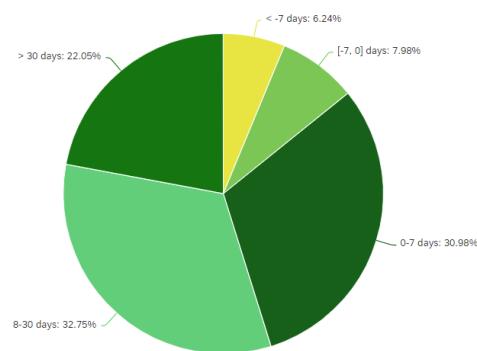
SHIJU 108

MAJAH 22



## GR Posting Date VS Invoice Issue Date [Domestic Supplier]

Overview



&lt; -7 days: Issue invoice in advance or delay receiving goods

287816 上海凯前机电设备有限公司	34
362959 湘潭市科兰达机电有限公司	29
293237 常州凯恩斯汽车零件 分选服务有限公司	25
294608 伍尔特 (中国) 有限公司	23
2956490 太仓市源展机械设备有限公司	22
249645 太仓市自来水有限公司	20
364377 太仓市佑博机械有限公司	19
356467 上海培昌信息科技有限公司	13
60056 圣戈班磨料磨具(上海)有限公司	13
365228 太仓英太思机械有限公司	11

## Calculation:

Invoice issue date - GR Posting Date

&lt; -7 days: Issue invoice in advance or delay receiving goods

[-7, 0] days: Issue Invoice and Delivery goods within 7 days, consider goods delivery time

0-7 days: Delivery goods and then Issue Invoice within 7 days, consider paper invoice delivery time

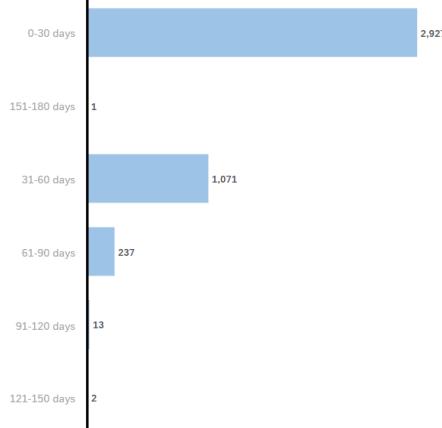
8-30 days: Delivery goods and then Issue Invoice 8-30 days

&gt;30 days: Late Issue invoice after delivery goods 30 days

## Parked Aging Overview

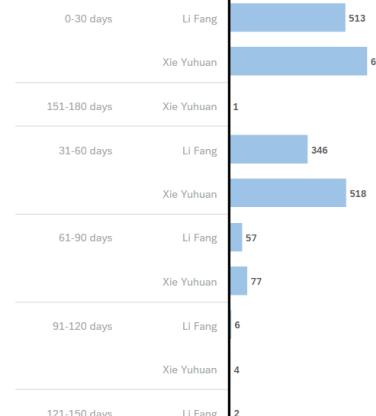
## Aging

1 Filter



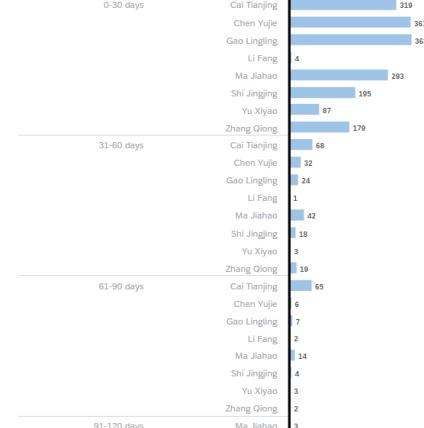
## Oversea Invoice by Aging

1 Filter



## Domestic Invoice by Aging

1 Filter



## Parked Invoice Overview

## Parked Items

1 Filter | Explorer Available

4,251 Count of Parked Invoice

2,114 Count of Parked Invoice

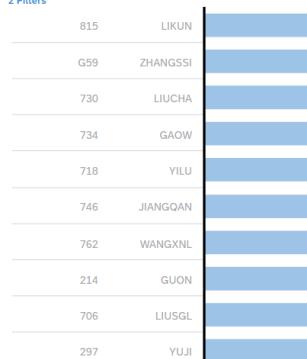
Domestic

2,135 Count of Parked Invoice

Overseas

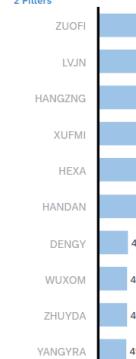
## By Purchasing Group

2 Filters



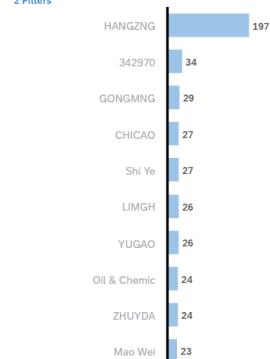
## By PR creator/MPR

2 Filters



## By Requisitioner

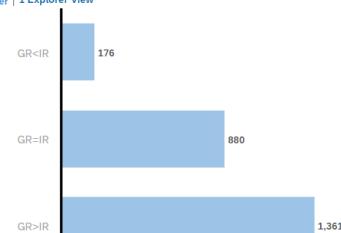
2 Filters



## Parked GR/IR Overview

## GR&gt;IR

1 Filter | 1 Explorer View



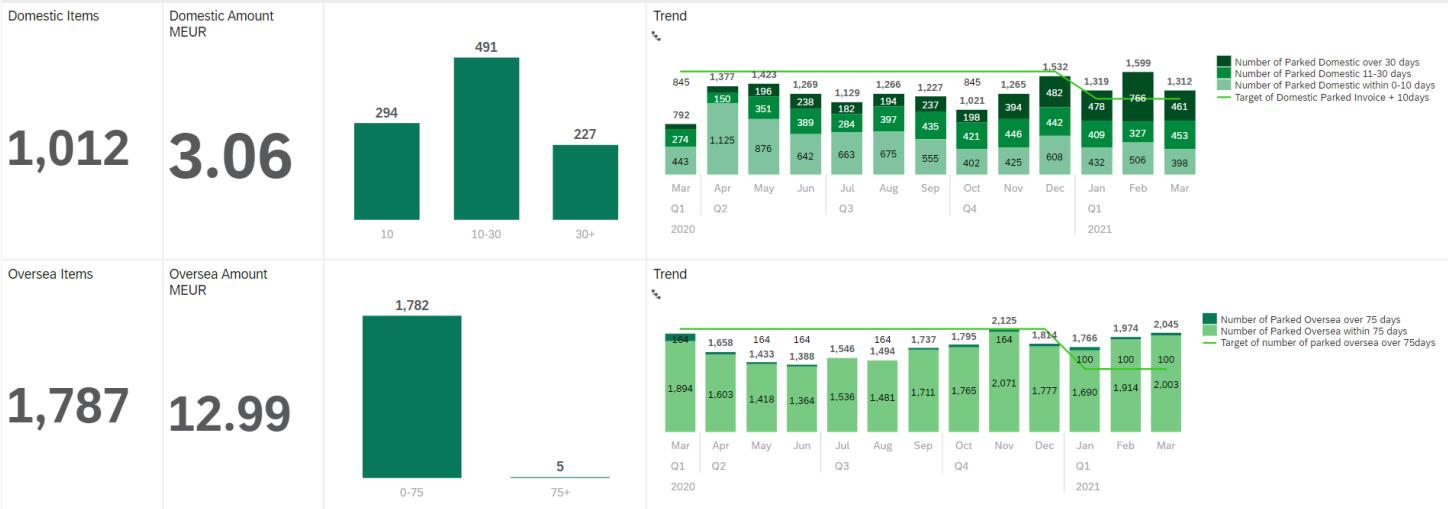
SAP no GR 1,834

## SCHAEFFLER

### AP - Dashboard

14:01 April 15, 2021

#### Parked Invoice



#### Notes:

exclude the next month's parked invoices.

March's parked data in SAC is based on the data which parked on 5th April, so the parked items were less than actual parked items on 1st April.(Actual parked items in Mar is 3357, domestic 1313, oversea 2044, around 775 items was post from SAC April 5th)

#### AP Invoice Receiving and Posting

## Received Invoices

**21,691**

Compare to Last Month

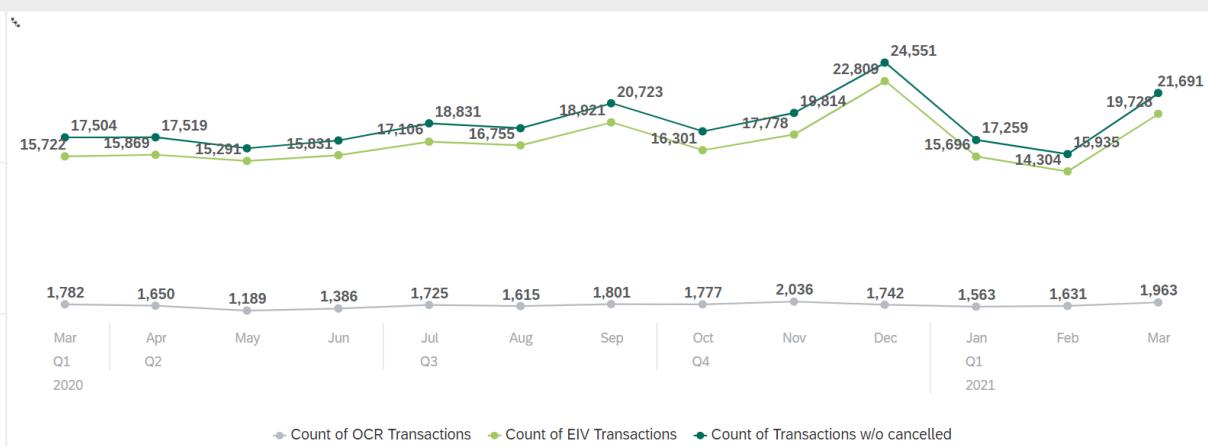
**21,691 (+5,756)**

△ 2021 Feb

Compare to Last Year

**21,691 (+4,187)**

△ 2020 Mar



## Correction Rate

E-Invoicing Correction Rate

**91.41% (-1.11)**

△ 2021 Feb

OCR Supplier Correction Rate

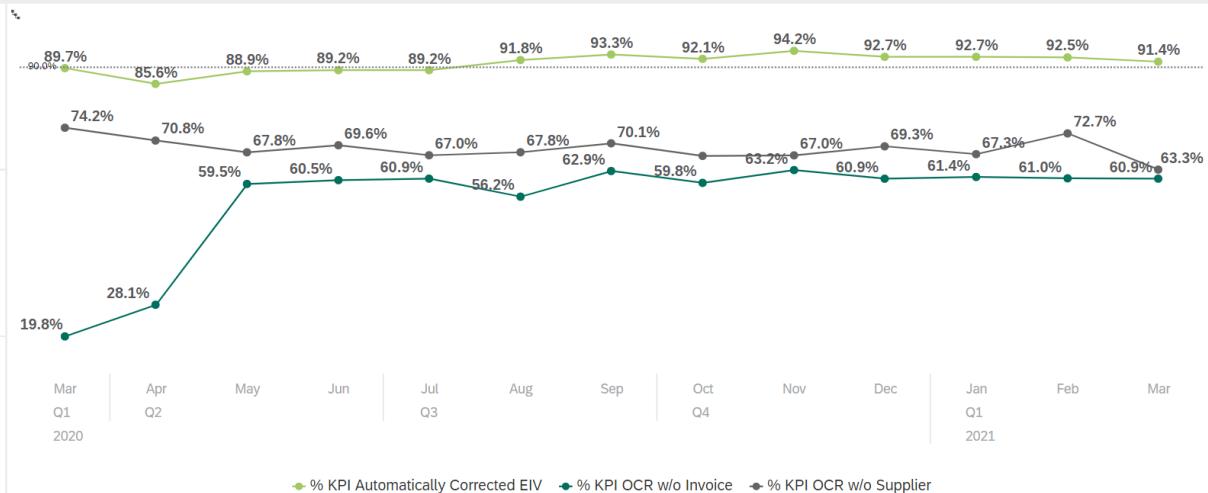
**63.27% (-9.45)**

△ 2021 Feb

OCR Invoice Correction Rate

**60.93% (-0.08)**

△ 2021 Feb

Notes:  
E

## Posting Rate

Autopost Rate

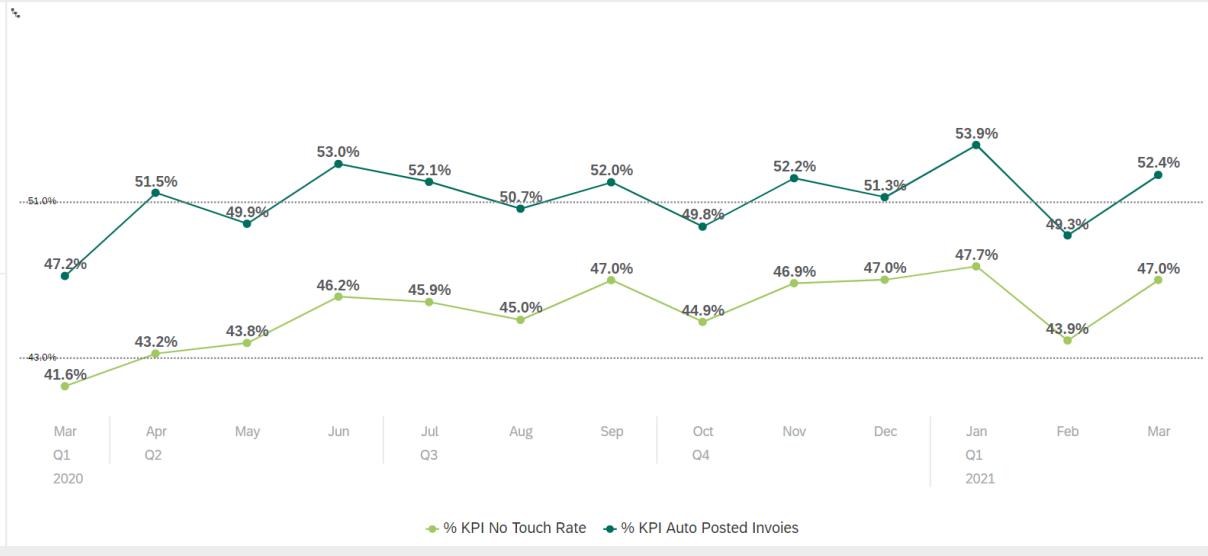
**52.41% (+3.10)**

△ 2021 Feb

No Touch Rate

**47.0% (+3.1)**

△ 2021 Feb



Notes:

SCHAEFFLER

## Auto Posted Invoices Analysis

14:03 April 15, 2021

## TOTALS

- Supplier Number
  - All
  - 1000 (1000)
  - 100629 (Karl Berrang GmbH Schraubengroßhandel)
  - 10149 (Scherer-Feinbau GmbH )
  - 102801 (Parker-Hannifin GmbH Engineered Materials Group Europe)
  - 10626 (Roesler Oberflaechentechnik GmbH Werk Memmelsdorf)
  - 10633 (Kuhmichel Abrasiv GmbH )
  - 1065 (Kracht GmbH )
  - 10656 (Hema GmbH Maschinen-U. Apparateschutz)
  - 10671 (Kasto Maschinenbau GmbH & Co. KG)
  - 10690 (Baumann Federn AG )
  - 10806 (C. Hilzinger-Thum GmbH & Co. KG )
  - 11047 (Legrom GmbH & Co.)
  - 1107 (Dr. Kaiser

## AUTO POST RATE

## 1 Filter

**52.41%**

21.557

11,299

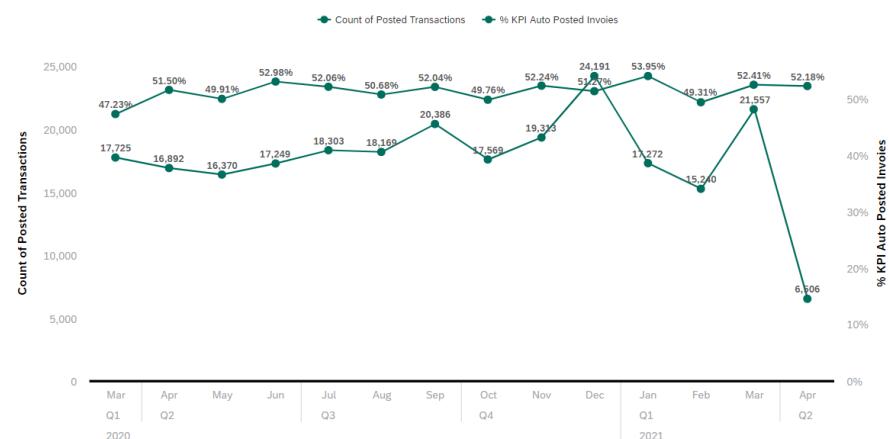
10.258

Count of Autopost not successful



## % KPI Auto Posted Invoices, Count of Posted Transactions per Posting Date for Actual

in -, % |



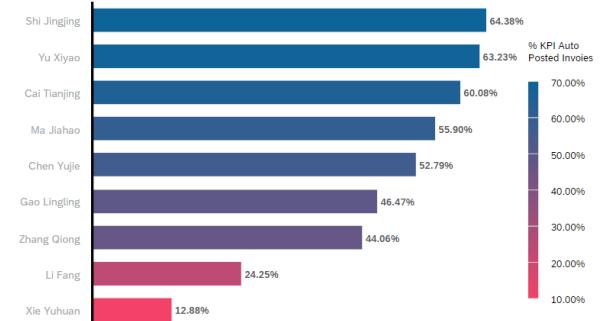
### Total post invoice

in --, % | 1 Filter | 2 Explorer Views



### % KPI Auto Posted Invoices per Accountant Name

in % | 1 Filter | 2 Explorer Views



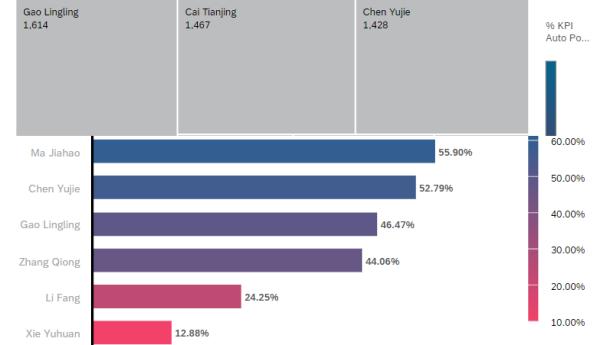
### Auto Post invoice

in -, % | 2 Filters | 2 Explorer Views



## Manual Post Invoice

2 Filters | 2 Explorer Views



## Auto Post invoice

in --, % | 2 Filters | 2 Explorer Views

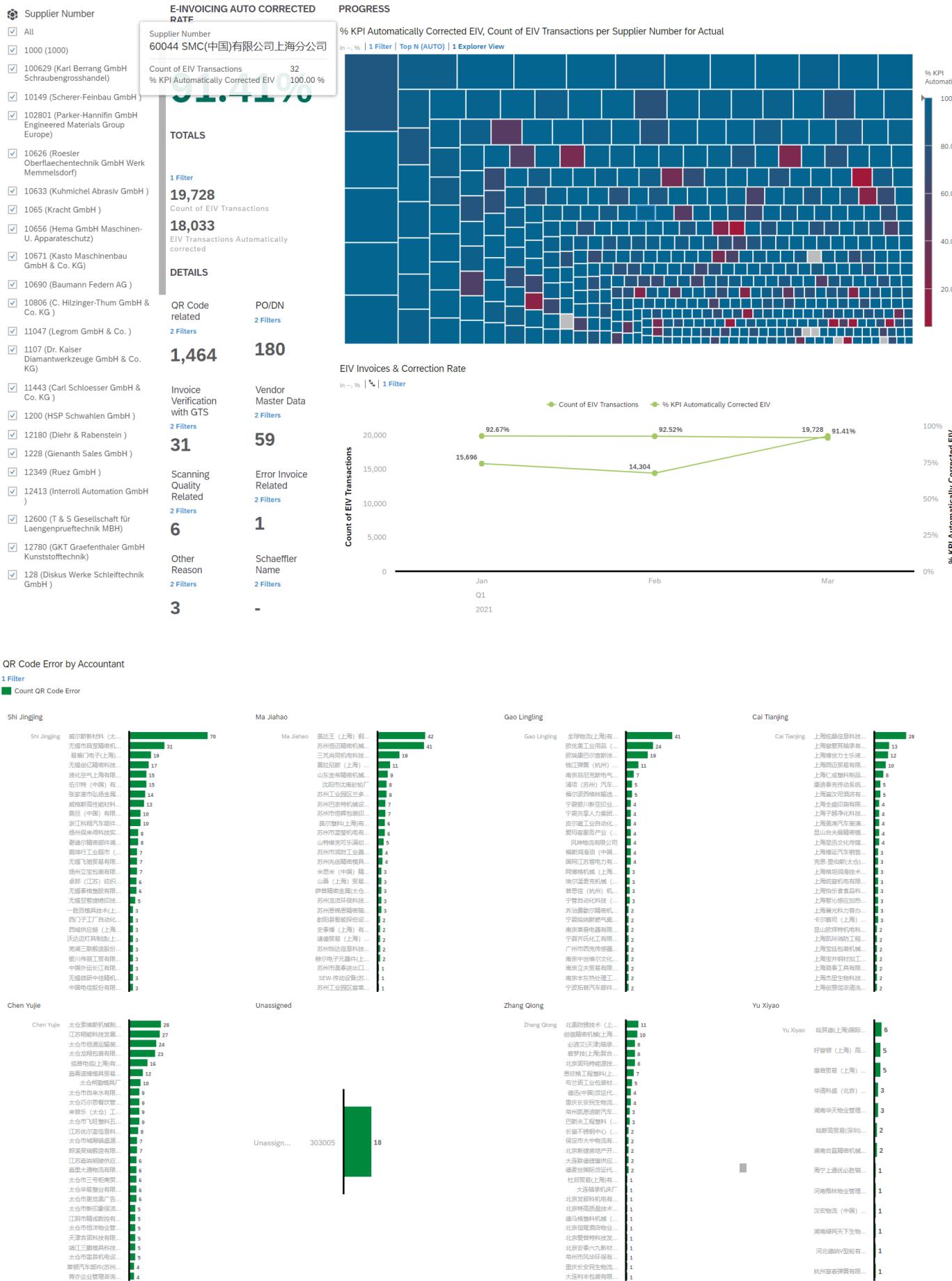


## Manual Post Invoice

2 Filters | 2 Explorer Views



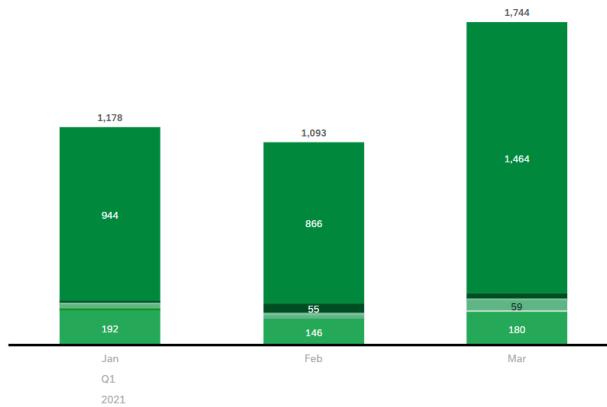


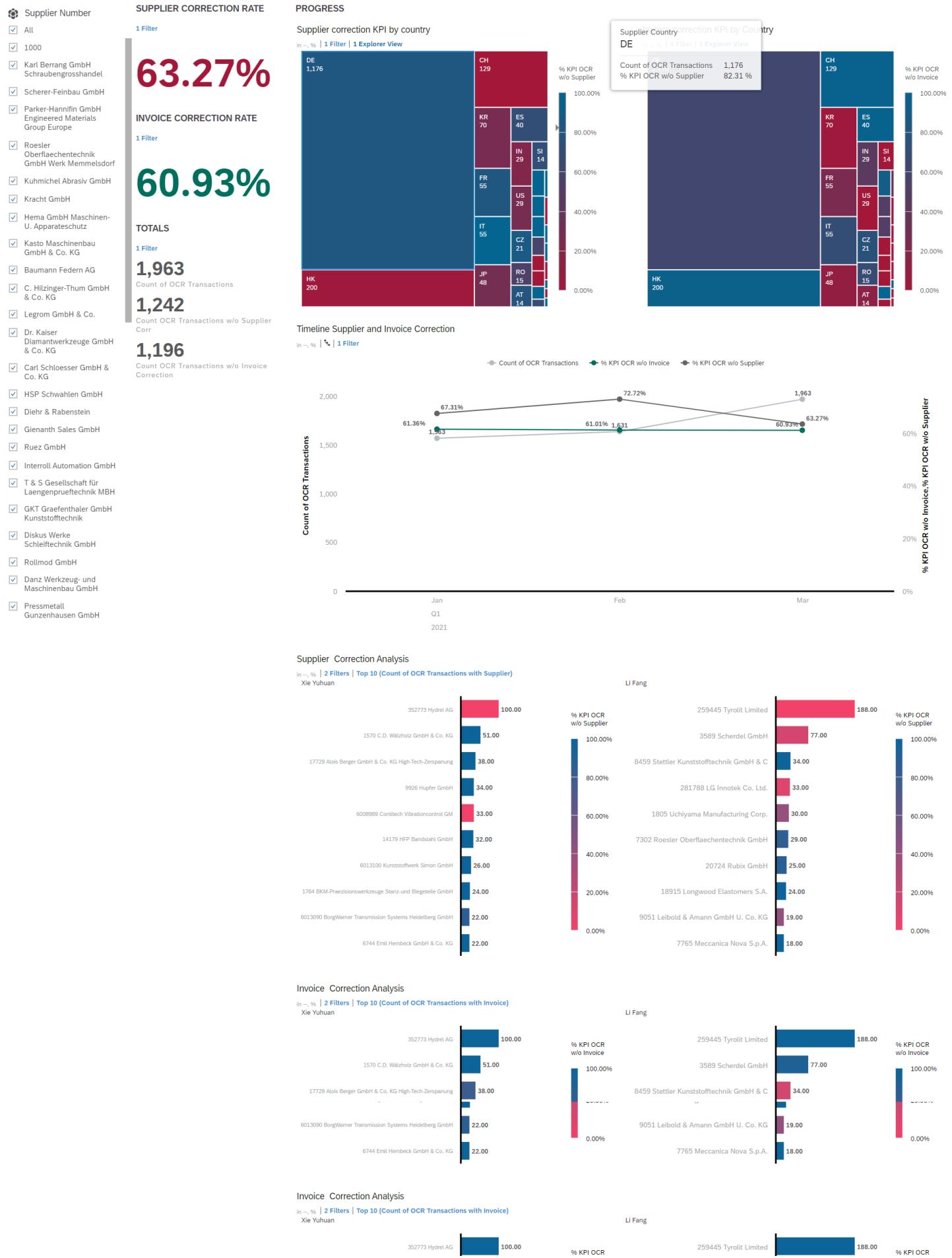


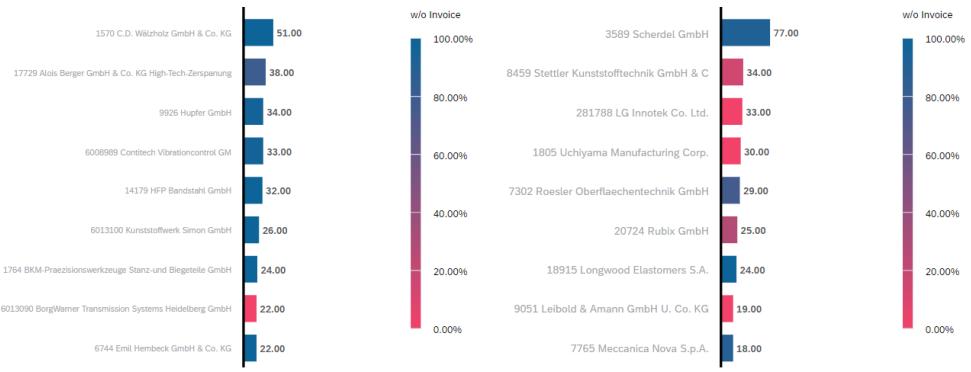
## Manual Correction Reason Overview

| 1 Filter

- Count QR Code Error
- Count GTS Error
- Count Scanning Quality Error
- Count Vendor Master Error
- Count Error Invoice Error
- Count Other Reason Error
- Count PO/DN Error







## SCHAFFLER

14:07 April 15, 2021

**Controls**

Posting Date Range Filter

2021 2022

fx Focus Supplier Autopost

All

100629 (Karl Berrang GmbH Schraubengrosshandel)

10149 (Scherer-Feinbau GmbH)

102801 (Parker-Hannifin GmbH Engineered Materials Group Europe)

10626 (Roesler Oberflaechentechnik GmbH Werk Memmelsdorf)

10690 (Baumann Federn AG)

10806 (C. Hilzinger-Thum GmbH & Co. KG)

1107 (Dr. Kaiser Diamantwerkzeuge GmbH & Co. KG)

11443 (Carl Schloesser GmbH & Co. KG)

1228 (Gienanth Sales GmbH)

12600 (T & S Gesellschaft für Laengenprueftechnik MBH)

12780 (GKT Graefenthaler GmbH Kunststofftechnik)

Show Inactive Values (1277)

fx Target Improvement to Autopost

0.43

0 1

**AUTO POST RATE**

**TOTALS**

| 1 Variance

**72.88% (+20.46)**

Target Autopost KPI ( $\Delta$  % KPI Auto Posted Invoices)

**TOP IMPACT ON KPI**

Count of Autopost not successful per Supplier Number for Actual

Top N (AUTO)

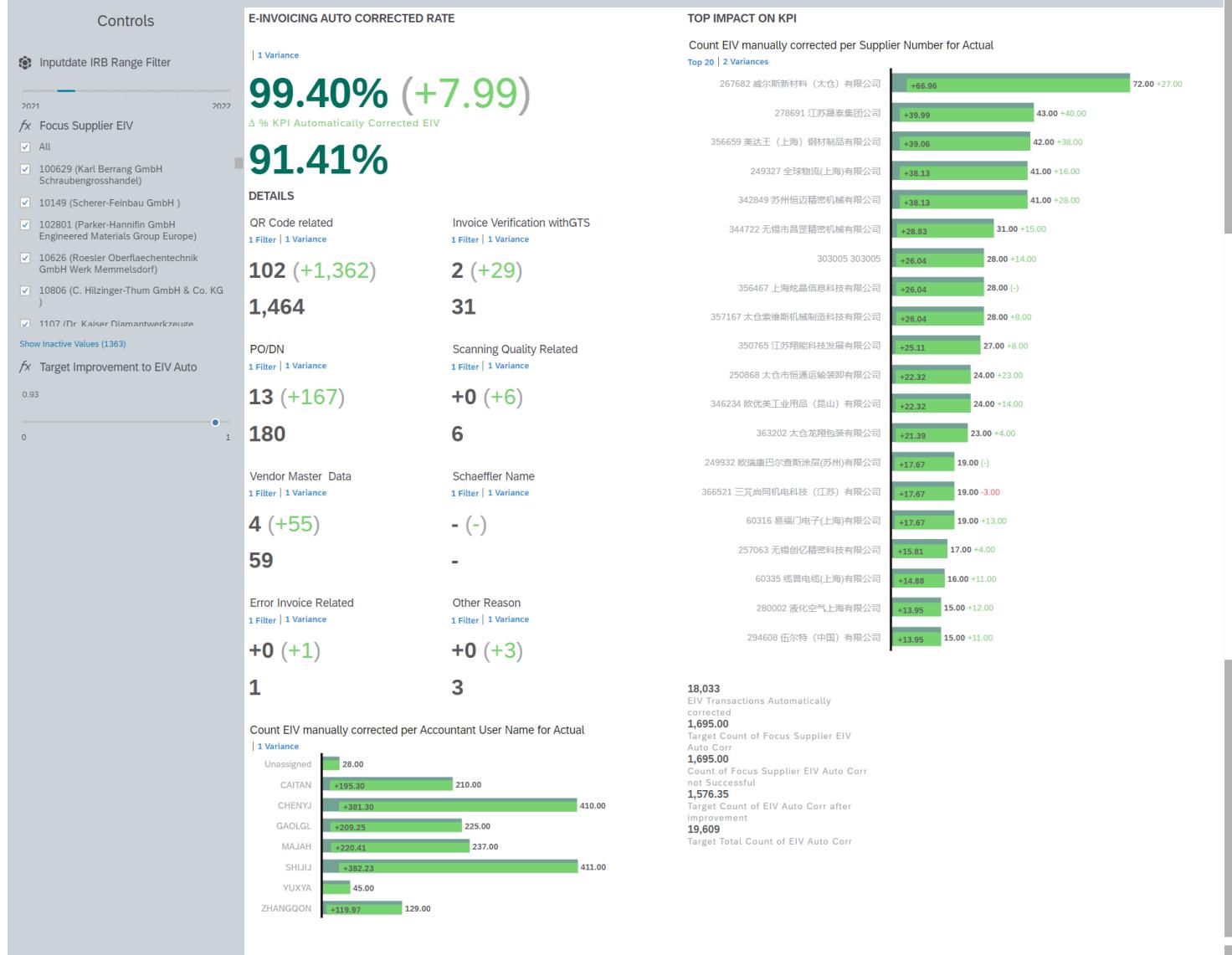
Supplier	Actual	Variance	Target
248670 菲耐托(中国)有限公司	+108.36	+52	+52
288288 上海菲耐托材料有限公司	+89.01	-207	+101
344537 苏美菱塑料有限公司	+89.34	-138	-125
353455 上海菲耐托机械有限公司	+98.47	-129	+81
354414 郑州爱耐特有限公司	+91.94	-120	-96
253050 上海开利材料加工有限公司	+97.79	-111	-1
293237 常州凯通斯通汽车零件 分销服务有限公司	+44.72	-104	-17
255270 张家港市弘进金属制品有限公司	+43.00	-100	-68
600566 重庆班纳科模具有限公司	+42.57	-99	+25
350059 太仓富勒斯的新材料科技有限公司	+41.71	-97	+23
255445 Tyrill Limited	+40.55	-95	+43
31137 安徽中鼎密封件股份有限公司	+40.55	-95	+24
250640 长盈精密技术(苏州)有限公司	+37.88	-88	+68
60189 上海爱耐特塑料有限公司	+36.84	-86	+52
354459 杭州兴华印制技术有限公司	+35.24	-82	+11
277903 江阴市精工数控设备有限公司	+34.43	-81	-77
355586 上海维斯新材料科技股份有限公司	+34.33	-81	+50
357823 苏东苏达大制品有限公司	+31.39	-73	-70
60038 浙江白金工贸有限公司	+30.10	-68	-76
274452 易博纳国际(宁波)有限公司	+29.24	-68	+39
259160 苏州盈能印太有限公司	+28.81	-67	+40
360305 303005	+28.81	-67	+21
346234 欧优美工业用品(昆山)有限公司	+25.44	-65	-44
351043 江苏伊普艾双螺杆有限公司	+25	-65	+39
339181 苏州巴特尔机械设备有限公司	+24.83	-61	-39
354284 大连海威重工有限公司	+24.83	-60	+39
356659 美达王(上海)钢构制品有限公司	+24.83	-59	+3
1570 C.D. Wilzholz GmbH & Co. KG	+24.83	-57	+11
277967 上海文尚浩洁设备有限公司	+24.83	-57	+40
267682 威尔密斯新材料(大分)有限公司	+23.25	-53	-25
284755 上海达美机械设备有限公司	+22.66	-52	+16
3591 Scherdel GmbH	+22.66	-52	+16
278691 江苏易泰集团有限公司	+21.20	-51	+20
281896 三爱通(深圳)有限公司	+20.52	-50	+2
344938 湖南飞箭新材料有限公司	+19.13	-50	+1
259326 常兰琪工业新材料有限公司	+19.13	-49	+13
344142 深圳海科检测有限公司	+19.45	-49	-45
340963 日本电产(大连)有限公司	+17.6	-47	-6
356267 南京丰安热处理工程有限公司	+16.11	-46	+11
365226 太原美太变机有限公司	+16.11	-46	+32
366085 苏州市凯盛静电科技有限公司	+16.11	-46	+32

**Potential Auto Posted Invoices**

Count of Autopost not successful per Accountant Name for Actual

| 2 Variances

Accountant	Actual	Variance	Target
Gao Lingling	+694.02	+1,614	+404
Ma Jiahao	+694.02	+1,614	+181
Cai Tianjing	+630.81	+1,467	+503
Chen Yujie	+614.04	+1,428	+518
Shi Jingjing	+588.24	+1,368	+403
Zhang Qiong	+471.71	+1,097	+165
Xie Yuhan	+316.91	+737	+123
Li Fang	634	+234	
Yu Xiyao	232	-19	
Unassigned	67	+21	



**Controls**

Inputdata IRB Range Filter

Focus Supplier OCR

- All
- 100629 (Karl Berrang GmbH Schraubengroßhandel)
- 10149 (Scherer-Feinbau GmbH)
- 102801 (Parker-Hannifin GmbH Engineered Materials Group Europe)
- 10626 (Roesler Oberflaechentechnik GmbH Werk Memmelsdorf)
- 10806 (C. Hiltzinger-Thum GmbH & Co. KG)
- 1107 (Dr. Kaiser Diamantwerkzeuge GmbH & Co. KG)
- 11443 (Carl Schloesser GmbH & Co. KG)
- 1228 (Gienanth Sales GmbH)
- 12600 (T & S Gesellschaft für Laengenprueftechnik MBH)
- 13506 (Industrieofen- und Anlagenbau GmbH)

[Show Inactive Values \(1363\)](#)

Target OCR Supplier Corr

0 1

**TARGET SUPPLIER CORRECTION RATE**

| 1 Variance

**100.00% (+36.73)**

**63.27%**

**DETAILS**

in --, %

**1,242**  
Count OCR Transactions w/o Supplier Corr

**721**  
Target Count of Focus Supplier OCR with Supplier-Corr

**721.00**  
Count of Focus Supplier OCR with Supplier Corr

**721.00**  
Target Count of OCR w/o Supplier Corr after improvement

**1,963.00**  
Target Total Count of OCR w/o Supplier Corr

**100.00%**  
Target OCR w/o Supplier Corr KPI

**Top Impact on Supplier KPI**

Top 20 | Explorer Available | 1 Variance

Supplier ID	Supplier Name	Impact
259445	Tyrolit Limited	188
352773	Hyrel AG	100
3589	Scherdel GmbH	67
6008989	Contitech Vibrationcontrol GM	33
281788	LG Innotek Co. Ltd.	30
1805	Uchiyama Manufacturing Corp.	18
4331	Famecha GmbH CNC-Drehen u. Engraten	18
342905	RPK S. Coop. Industrial	12
6008768	Daetwyler Schweiz AG	12
32542	L.G. Balakrishnan & Bros. Ltd. LGB HQ Ganapathy	11
9051	Leibold & Amann GmbH U. Co. KG	11
253368	LE JOINT FRANCAIS S.N.C. DEPARTEMENT JOINTS TORIQUES	10
267485	Shishan Diamond Industrial Co. Ltd	8
30882	Coprotec Systeme GmbH	8
4571	SFS Intec AG Automotive Products	7
1837	Nakanishi Metal Works Co. Ltd.	6
233931	AAM Powder Metal Components Inc.	6
286018	Quick-Set	6
340832	KS Gleitlager de Mexico	6
6018565	Saint-Gobain Performance Plastics L+S GmbH	6

**Target OCR Supplier Corr KPI**

0 1

**TARGET INVOICE CORRECTION RATE**

| 1 Variance

**100.00% (+39.07)**

**60.93%**

**DETAILS**

**1,196**  
Count OCR Transactions w/o Invoice Correction

**767.00**  
Target Count of Focus Supplier OCR with Invoice-Corr

**767.00**  
Count of Focus Supplier OCR with Invoice Corr

**767.00**  
Target Count of OCR w/o Invoice Corr after improvement

**1,963.00**  
Target Total Count of OCR w/o Invoice Corr

**100.00%**  
Target OCR w/o Invoice Corr KPI

**Top Impact on Invoice KPI**

Top 20 | Explorer Available | 1 Variance

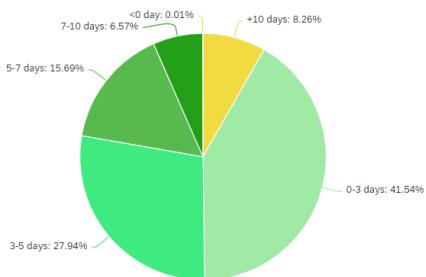
Supplier ID	Supplier Name	Impact
259445	Tyrolit Limited	188.00
352773	Hyrel AG	100.00
3589	Scherdel GmbH	+71.00
1570	C.D. Walzholz GmbH & Co. KG	51.00
17729	Alois Berger GmbH & Co. KG High-Tech-Zerspanung	+30.00
8459	Stettler Kunststofftechnik GmbH & C	+5.00
9926	Hupfer GmbH	+33.00
281788	LG Innotek Co. Ltd.	33.00
6008989	Contitech Vibrationcontrol GM	+32.00
14179	HFP Bandstahl GmbH	32.00
1805	Uchiyama Manufacturing Corp.	30.00
7302	Roesler Oberflaechentechnik GmbH	+22.00
6013100	Kunststoffwerk Simon GmbH	26.00
20724	Rubix GmbH	+7.00
1764	BKM-Praezisionswerkzeuge Stanz- und Biegeteile GmbH	24.00
18915	Longwood Elastomers S.A.	24.00
6013090	BorgWarner Transmission Systems Heidelberg GmbH	22.00
6744	Emil Hembeck GmbH & Co. KG	22.00
7002	AKS Hartmetall- Technik GmbH	21.00
17863	Kunststofftechnik ROS GmbH & Co. KG	+19.00
		20.00

**Target OCR w/o Invoice Corr KPI**

0 1

## Invoice Received Date VS Invoice Issue Date/收票日VS开票日

### Overview



**+10days: Invoice received Over 10 days after invoice issued/收票日在开票日10天后**

### 1 Explorer View

249643 西门子工厂自动化工...	SHIJUJ	117
60231 摩尔电子元器件(上海)...	MAJAH	102
252656 巴鲁夫自动化（上海...	ZHANGQON	88
280136 霍夫曼工具贸易(上海)...	YUXYA	66
294608 丘尔特（中国）有限...	SHIJUJ	58
293237 常州凯恩迪斯汽车零...	ZHANGQON	44
294916 上海裕欣五金有限公司	CAITAN	44
356467 上海炫晶信息科技有...	CAITAN	43
248589 一胜百模具技术(上海)...	SHIJUJ	41
60044 SMC(中国)有限公司上...	MAJAH	32

### Calculation:

Invoice Received Date - Invoice Issue Date:

<0 days: issue invoice in advance

0-10 days: normal cycle, consider the paper invoice delivery time

>10 days: delay sending invoice, impact DPO and payment forecast

### PoC\_Invoices\_IRB

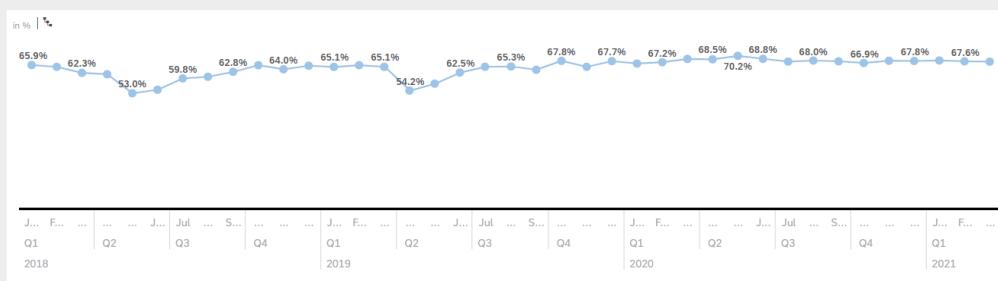
1 Filter 2

Supplier Number	Scan to Invoice issue Aging	Invoice date	Scandate	Account	Amount in LC
1000 1000	+10 days	2020-03-27 Mar 27, 2020 (2020)	Jun 30, 2020 (2020)		19,582.94
10149 Scherer-Feinbau GmbH	+10 days	2019-08-28 Aug 28, 2019 (2019)	Mar 25, 2020 (2020)		63,600.45
		2020-02-04 Feb 4, 2020 (2020)	Mar 4, 2020 (2020)		46,730.38
		2020-04-28 Apr 28, 2020 (2020)	Oct 13, 2020 (2020)		8,980.45
		2020-08-31 Aug 31, 2020 (2020)	Oct 13, 2020 (2020)		9,053.41
102801 Parker-Hannifin GmbH Engineered Materials Group Europe	+10 days	2020-04-15 Apr 15, 2020 (2020)	May 7, 2020 (2020)		73,682.89
		2021-01-07 Jan 7, 2021 (2021)	Feb 4, 2021 (2021)		0.00
		2021-01-08 Jan 8, 2021 (2021)	Feb 4, 2021 (2021)		11,166.05
		2021-01-15 Jan 15, 2021 (2021)	Feb 4, 2021 (2021)		0.00
10626 Roesler Oberflaechentechnik GmbH Werk Memmelsdorf	+10 days	2019-09-25 Sep 25, 2019 (2019)	Feb 22, 2021 (2021)		12,512.30
		2021-01-12 Jan 12, 2021 (2021)	Feb 22, 2021 (2021)		2,055.31
		2021-01-18 Jan 18, 2021 (2021)	Feb 22, 2021 (2021)		21,055.17
		2021-01-20 Jan 20, 2021 (2021)	Feb 22, 2021 (2021)		9,288.12
		2021-01-21 Jan 21, 2021 (2021)	Feb 22, 2021 (2021)		3,334.00
		2021-01-25 Jan 25, 2021 (2021)	Feb 22, 2021 (2021)		2,271.22
		2021-02-04 Feb 4, 2021 (2021)	Feb 22, 2021 (2021)		7,675.16
		2021-02-08 Feb 8, 2021 (2021)	Feb 22, 2021 (2021)		79,727.72
10633 Kuhmichel Abrasiv GmbH	+10 days	2020-09-29 Sep 29, 2020 (2020)	Oct 13, 2020 (2020)		11,694.74
1065 Kracht GmbH	+10 days	2020-08-28 Aug 28, 2020 (2020)	Oct 20, 2020 (2020)		3,460.55
		2020-11-30 Nov 30, 2020 (2020)	Feb 7, 2021 (2021)		3,337.76
10656 Hema GmbH Maschinen- U. Apparateschutz	+10 days	2019-05-16 May 16, 2019 (2019)	Jun 8, 2020 (2020)		14,328.17
10671 Kasto Maschinenbau GmbH & Co. KG	+10 days	2020-12-18 Dec 18, 2020 (2020)	Jan 11, 2021 (2021)		260,209.53
11047 Legrom GmbH & Co.	+10 days	2019-11-18 Nov 18, 2019 (2019)	Apr 2, 2020 (2020)		12,213.29
		2020-04-02 Apr 2, 2020 (2020)	Apr 20, 2020 (2020)		12,123.66
1107 Dr. Kaiser Diamantwerkzeuge GmbH & Co. KG	+10 days	2019-10-29 Oct 29, 2019 (2019)	Mar 2, 2020 (2020)		16,666.32
		2019-11-14 Nov 14, 2019 (2019)	Mar 2, 2020 (2020)		0.00
		2019-11-26 Nov 26, 2019 (2019)	Mar 2, 2020 (2020)		1,609.90

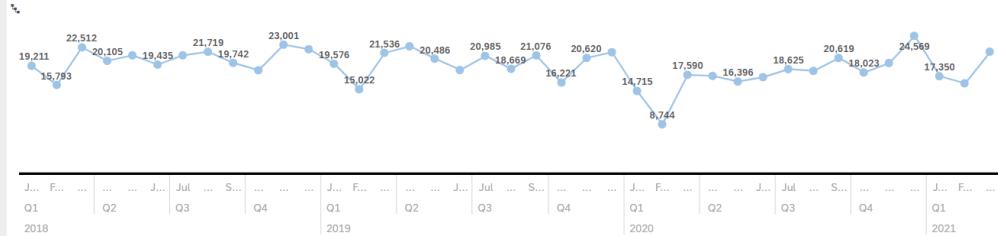
## Trend

<input checked="" type="checkbox"/> Vendor	<input checked="" type="checkbox"/> Account...
<input checked="" type="checkbox"/> All	<input checked="" type="checkbox"/> All
<input checked="" type="checkbox"/> 100629 (Karl Berrang GmbH)	<input checked="" type="checkbox"/> Cal, Tianjing
<input checked="" type="checkbox"/> 101049 (Krohne Messtechnik)	<input checked="" type="checkbox"/> Chen Yujie
<input checked="" type="checkbox"/> 10149 (Scherer-Feinbau GmbH)	<input checked="" type="checkbox"/> Gao, Lingling
<input checked="" type="checkbox"/> 102801 (Parker-Hannifin GmbH)	<input checked="" type="checkbox"/> Li, Fang
<input checked="" type="checkbox"/> 10389 (Magnetische Pruefanlagen)	<input checked="" type="checkbox"/> Ma Jiahao
<input checked="" type="checkbox"/> 10482 (Kurt Huettinger GmbH & Co. KG)	<input checked="" type="checkbox"/> Shi Jingling
<input checked="" type="checkbox"/> 10500 (Mafu GmbH Automation)	<input checked="" type="checkbox"/> Xie Yuhuan
<input checked="" type="checkbox"/> 10569 (Ohrmann GmbH)	<input checked="" type="checkbox"/> Yu, Xiyao
<input checked="" type="checkbox"/> 10626 (Roessler Oberflaechentechnik GmbH)	<input checked="" type="checkbox"/> Zhang Qiong
<input checked="" type="checkbox"/> 10633 (Kuhmichel Abrasivs GmbH)	<input checked="" type="checkbox"/> Zhao, Weijuan
<input checked="" type="checkbox"/> 1065 (Kracht GmbH)	
<input checked="" type="checkbox"/> 10656 (Hema GmbH)	
<input checked="" type="checkbox"/> 10671 (Kasto Maschinenbau)	
<input checked="" type="checkbox"/> 10690 (Baumann Federn AG)	
<input checked="" type="checkbox"/> 10806 (C. Hiltzinger-Thum GmbH & Co. KG)	
<input checked="" type="checkbox"/> 11047 (Legrom GmbH & Co.)	
<input checked="" type="checkbox"/> 1107 (Dr. Kaiser)	
<input checked="" type="checkbox"/> 1123 (Murtfeldt Kunststoffe GmbH & Co. KG)	
<input checked="" type="checkbox"/> 11260 (Josef Blässinger GmbH & Co.)	
<input type="checkbox"/> 11280 (FELV-Festoren GmbH)	
Show Inactive Values	
<input checked="" type="checkbox"/> PM/NPM	<input checked="" type="checkbox"/> Indus.
<input checked="" type="checkbox"/> All	<input checked="" type="checkbox"/> All
<input checked="" type="checkbox"/> NPM	
<input checked="" type="checkbox"/> PM	
Show Inactive Values	
<input checked="" type="checkbox"/> Indus.	<input checked="" type="checkbox"/> Show Inactive Values
<input checked="" type="checkbox"/> M011	<input checked="" type="checkbox"/> M011
<input checked="" type="checkbox"/> M012	<input checked="" type="checkbox"/> M012
<input checked="" type="checkbox"/> M013	<input checked="" type="checkbox"/> M013
<input checked="" type="checkbox"/> M014	<input checked="" type="checkbox"/> M014
<input checked="" type="checkbox"/> M016	<input checked="" type="checkbox"/> M016
<input checked="" type="checkbox"/> M017	<input checked="" type="checkbox"/> M017
<input checked="" type="checkbox"/> M018	<input checked="" type="checkbox"/> M018
<input checked="" type="checkbox"/> M019	<input checked="" type="checkbox"/> M019
<input checked="" type="checkbox"/> M031	<input checked="" type="checkbox"/> M031
<input checked="" type="checkbox"/> M029	<input checked="" type="checkbox"/> M029

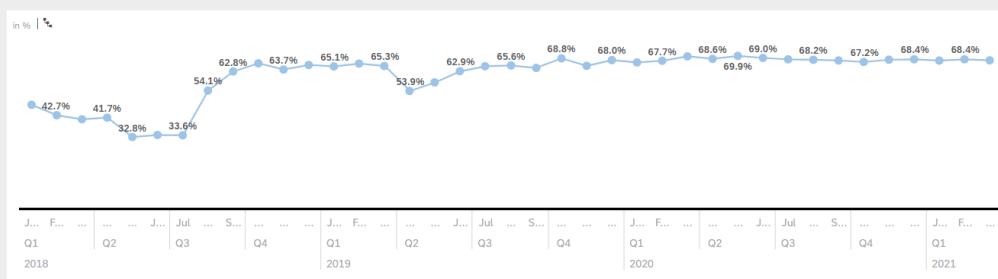
## Auto Post Rate



## Invoice Volume per Month



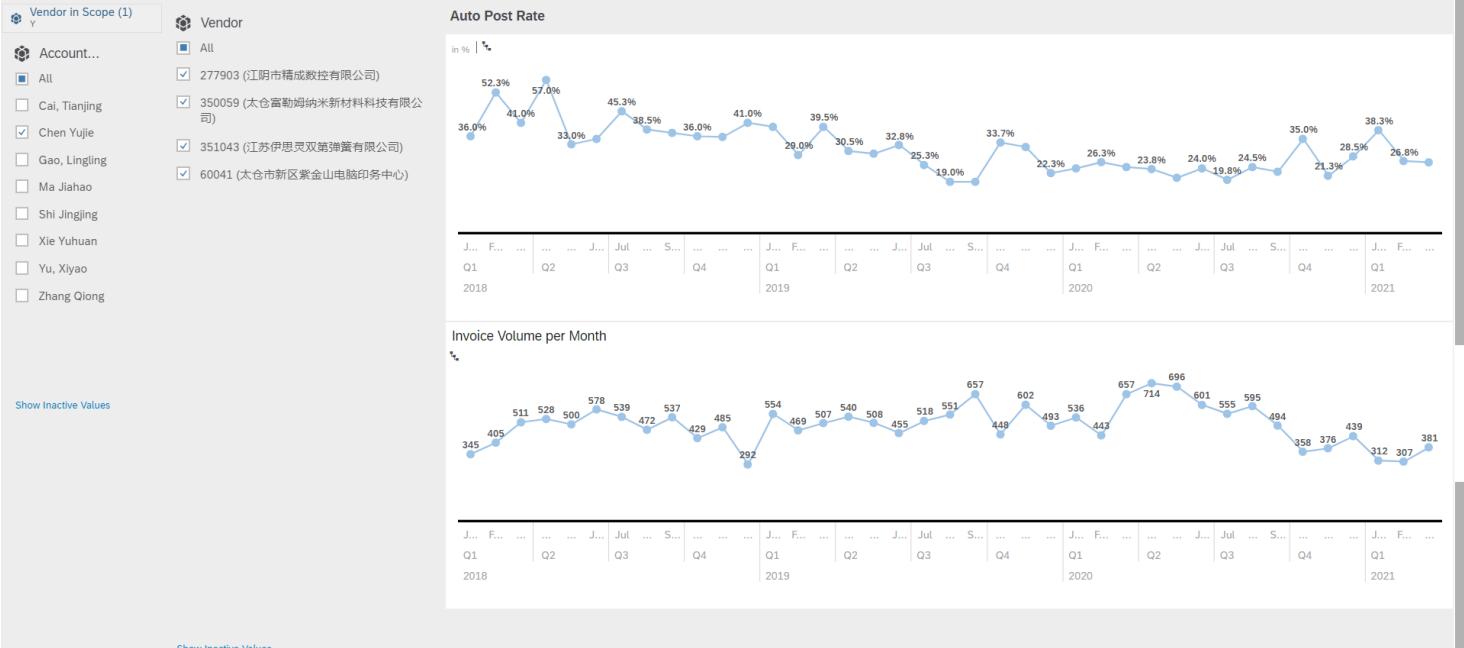
## NO Touch Rate



## SSC Sub7 NPM GR Optimization



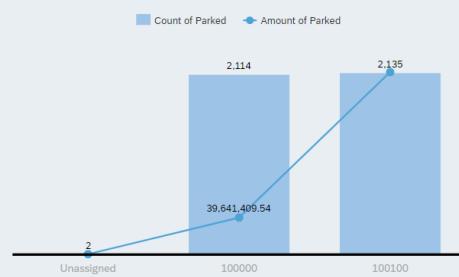
## PMGM Target Vendor KPI Monitor



## Park Overview

% Amount of Parked

Parked Items&amp;Amount

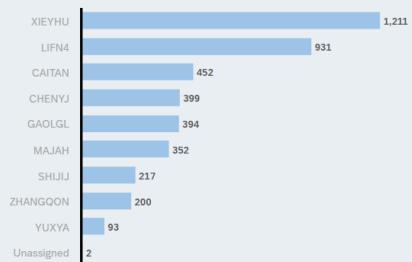
**1.78%**

% Count of Parked

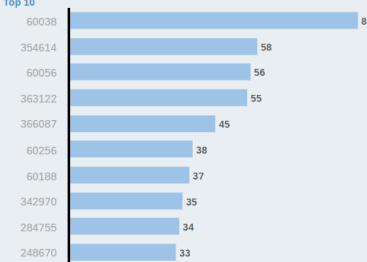
**1.63%**

## Park Analysis

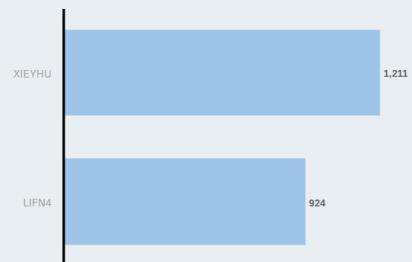
## Domestic Parked Items by AP accountant



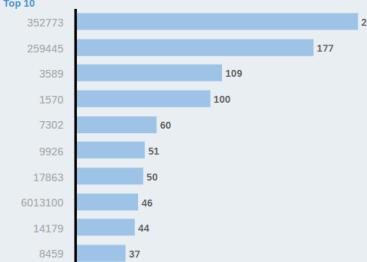
## Parked Items by Supplier-Domestic



## Oversea Parked Items by AP accountant



## Parked Items by Supplier-Domestic



## Parked Items Ratio by AP account

In %

Unassigned

CAITAN

CHENYJ

GAOLGL

**0.19%**

% Count of Parked

**1.05%**

% Count of Parked

**1.16%**

% Count of Parked

**1.18%**

% Count of Parked

LIFN4

MAJAH

SHIJU

XIEYHU

**7.20%**

% Count of Parked

**0.81%**

% Count of Parked

**0.50%**

% Count of Parked

**8.63%**

% Count of Parked

YUXYA

ZHANGQON

ZHAOWJU

## AP Overdue Items

## % of Overdue Items

**2.34% (-0.78)**

Δ 2021 Feb

**969 (-577)**

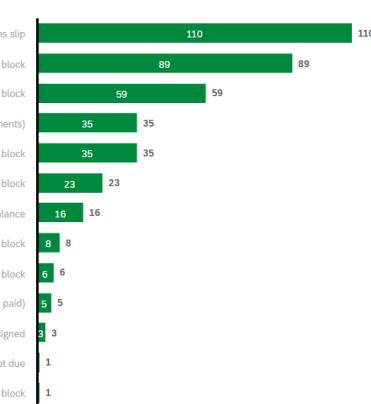
Δ 2021 Feb

## Trend



## Overdue Reason

\*exclude the Reason Paid &amp; will be Paid



## Aging



## 41,399

Count of Open Items 3rd AP

## 969

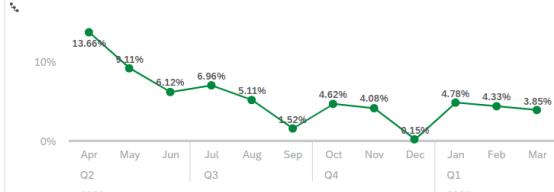
Count of Overdue OI 3rd AP

## % of Overdue Amount

**3.85% (-0.47)**

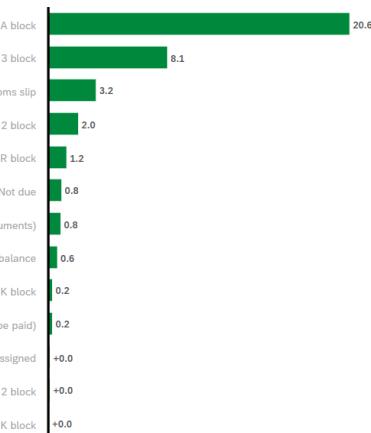
Δ 2021 Feb

## Trend



## Overdue Reason

\*exclude the Reason Paid &amp; will be Paid



## Aging



## 67.38

ABS Overdue OI Amount MCNY

## Controls

 End of Month Range Filter

2021 — 2022

 Vendor Number

- All
- 1000 (1000)
- 100629 (Karl Berrang GmbH Schraubengroßhandel)
- 10149 (Scherer-Feinbau GmbH)
- 102801 (Parker-Hannifin GmbH Engineered Materials Group Europe)
- 10569 (Ohrmann GmbH Montagetechnik)
- 10626 (Roesler OberflächenTechnik GmbH Werk Memmelsdorf)
- 10633 (Kuhmicel Abrasiv GmbH)
- 1065 (Kracht GmbH)
- 10671 (Kasto Maschinenbau GmbH & Co. KG)
- 10690 (Baumann Federn AG)

 Show Inactive Values

## ANALYSIS

## Items &amp; Amount of 3rd AP Overdue

in --, Million and others | **1 Filter****41,399**

Count of Open Items 3rd AP

**38,714**

Count of OI 3rd AP Domestic

**2,685**

Count of OI 3rd AP Oversea

**1,748.5 Million**

ABS Amount of open items 3rd AP

**67,384,612.52**

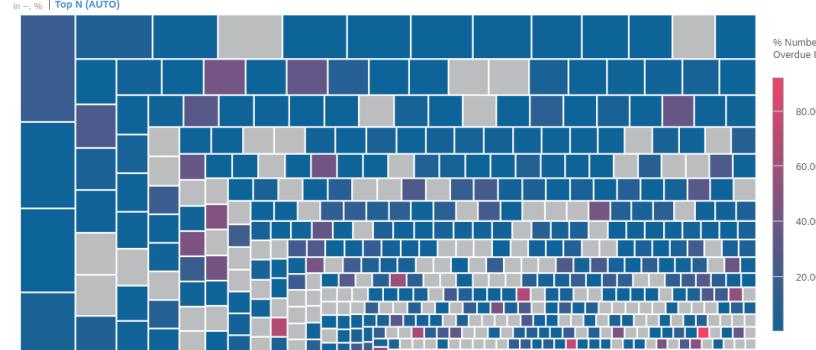
ABS Amount of Overdue OI 3rd-AP

**3.85%**

% Amount of Overdue OI 3rd AP

## AP Aging by all legal entities

% Number of Overdue Items 3rd AP, Count of Open Items 3rd AP per Vendor Number for Actual

in --, % | **Top N (AUTO)**% Number of Overdue It...  
80.00%  
60.00%  
40.00%  
20.00%

## ABS Amount of open items 3rd AP per Key Date for Actual

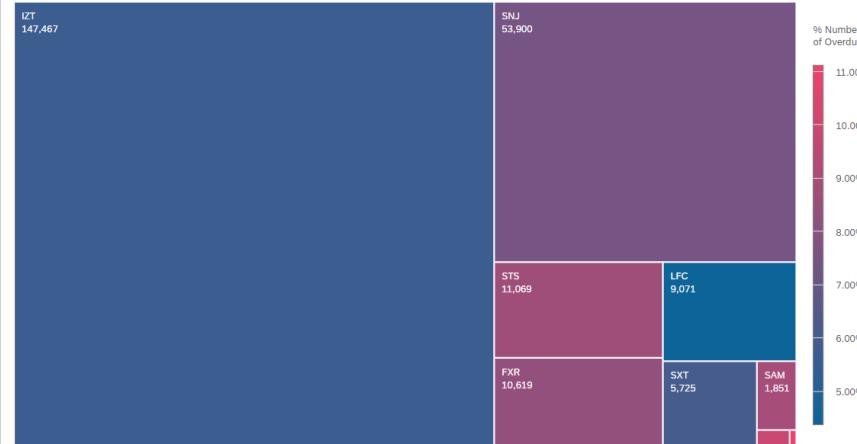
in Million |



## AP Aging analysis by company code

% Number of Overdue Items 3rd AP, Count of Open Items 3rd AP per Company Code for Actual

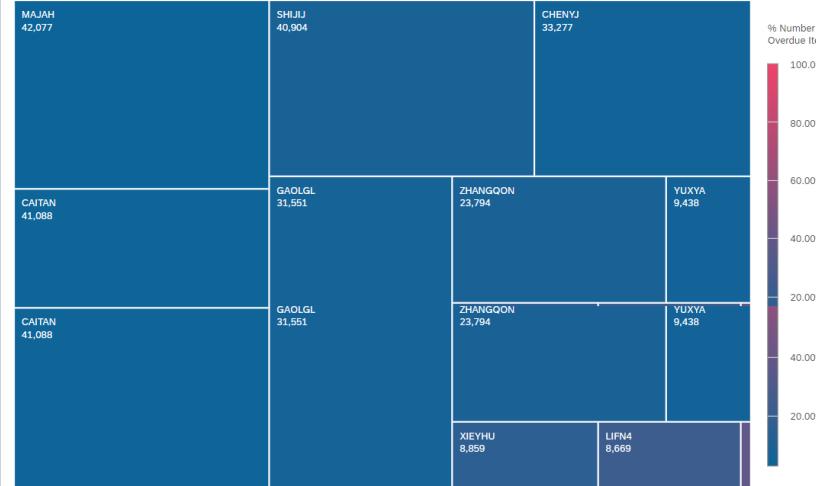
in --, %

% Number of Overdue...  
11.00%  
10.00%  
9.00%  
8.00%  
7.00%  
6.00%  
5.00%

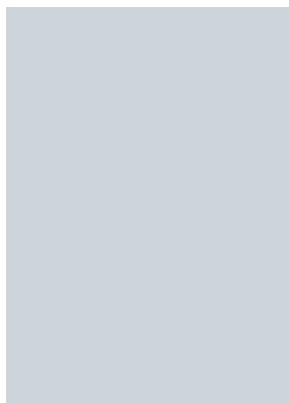
## AP Aging analysis by accounting clerk

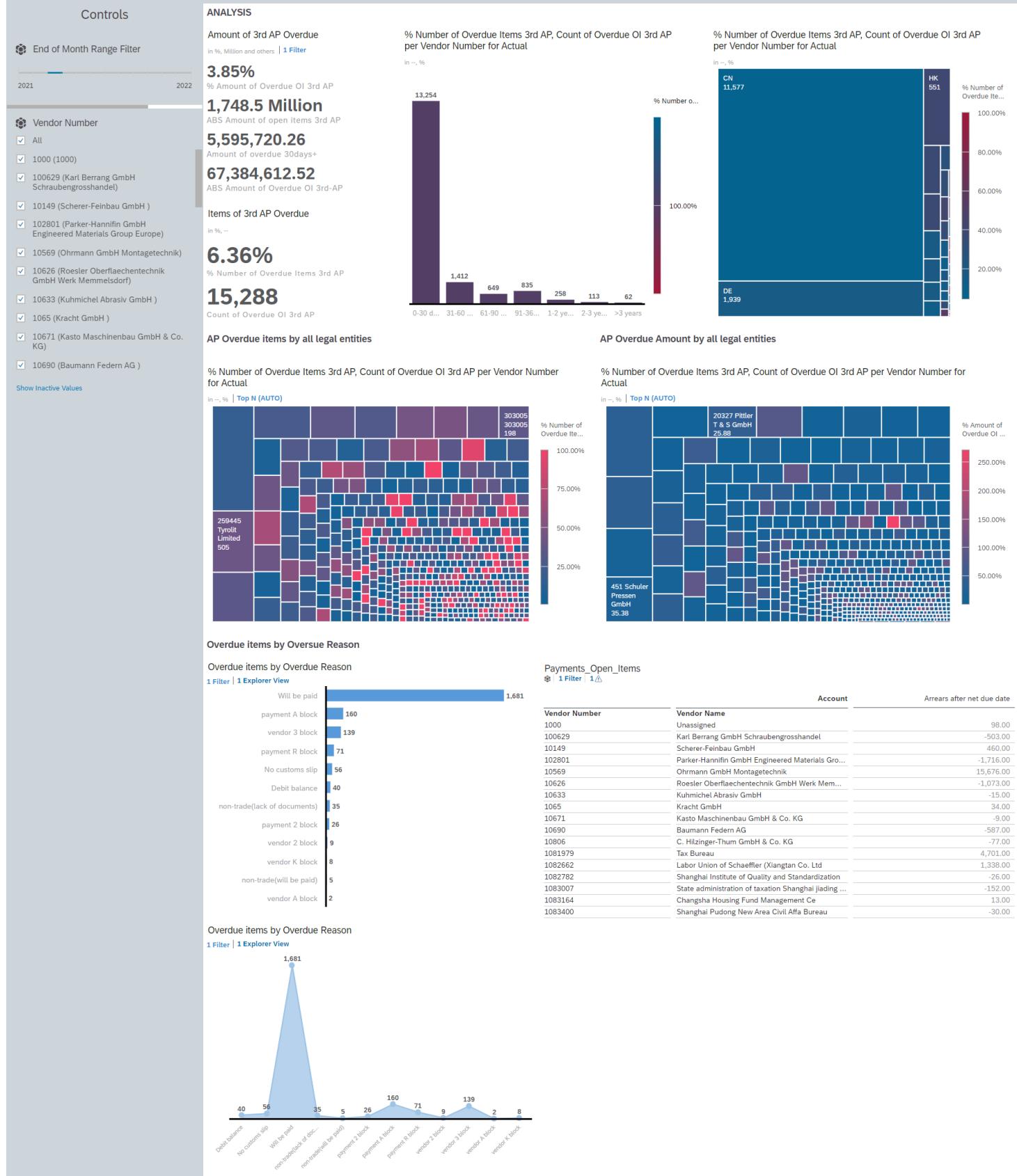
% Number of Overdue Items 3rd AP, Count of Open Items 3rd AP per User Name for Actual

in --, %

% Number of Overdue It...  
100.00%  
80.00%  
60.00%  
40.00%  
20.00%

% Amount of Overdue OI 3rd AP, % Number of Overdue Items 3rd AP and others per Vendor Number for Actual





**Controls**

Focus Supplier AP

All

1000 (1000)

10000 (Schaeffler (Singapore) Pte. Ltd.)

10015 (Schaeffler Engineering GmbH)

10045 (Schaeffler Kysuce spol. s r.o.)

10050 (Schaeffler Canada Inc.)

100629 (Karl Berrang GmbH Schraubengroßhandel)

10101 (舍弗勒 (中国) 有限公司)

10149 (Scherer-Feinbau GmbH)

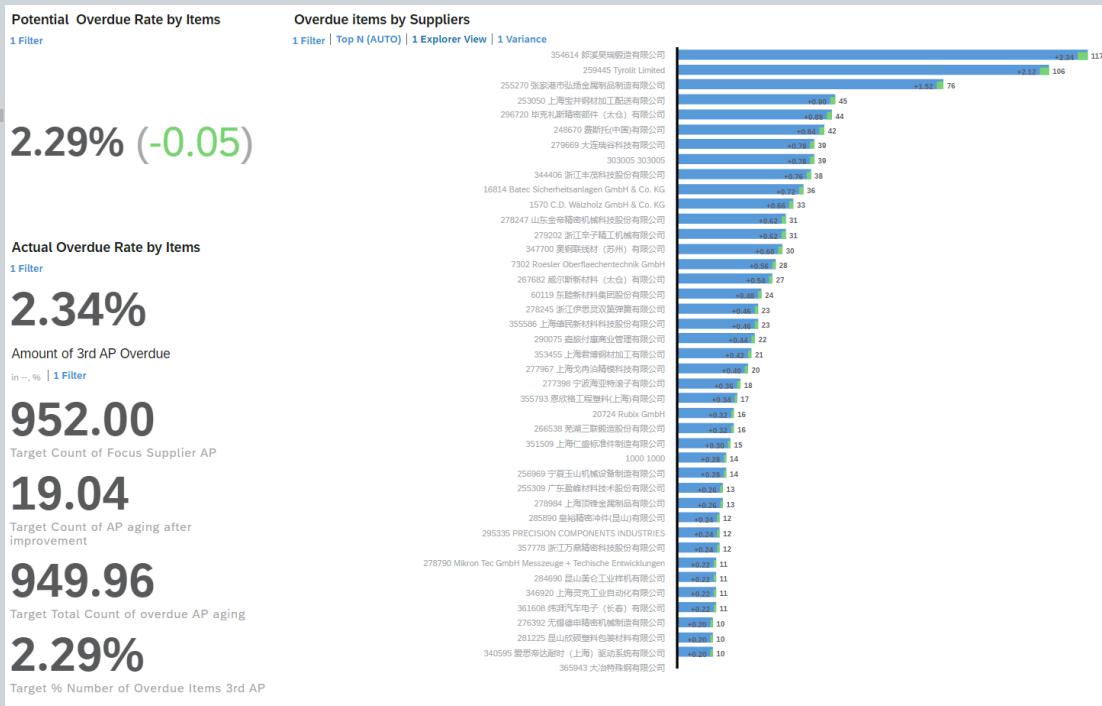
102801 (Parker-Hannifin GmbH Engineered Materials Group Europe)

1030 (Schaeffler Savaria Kt.)

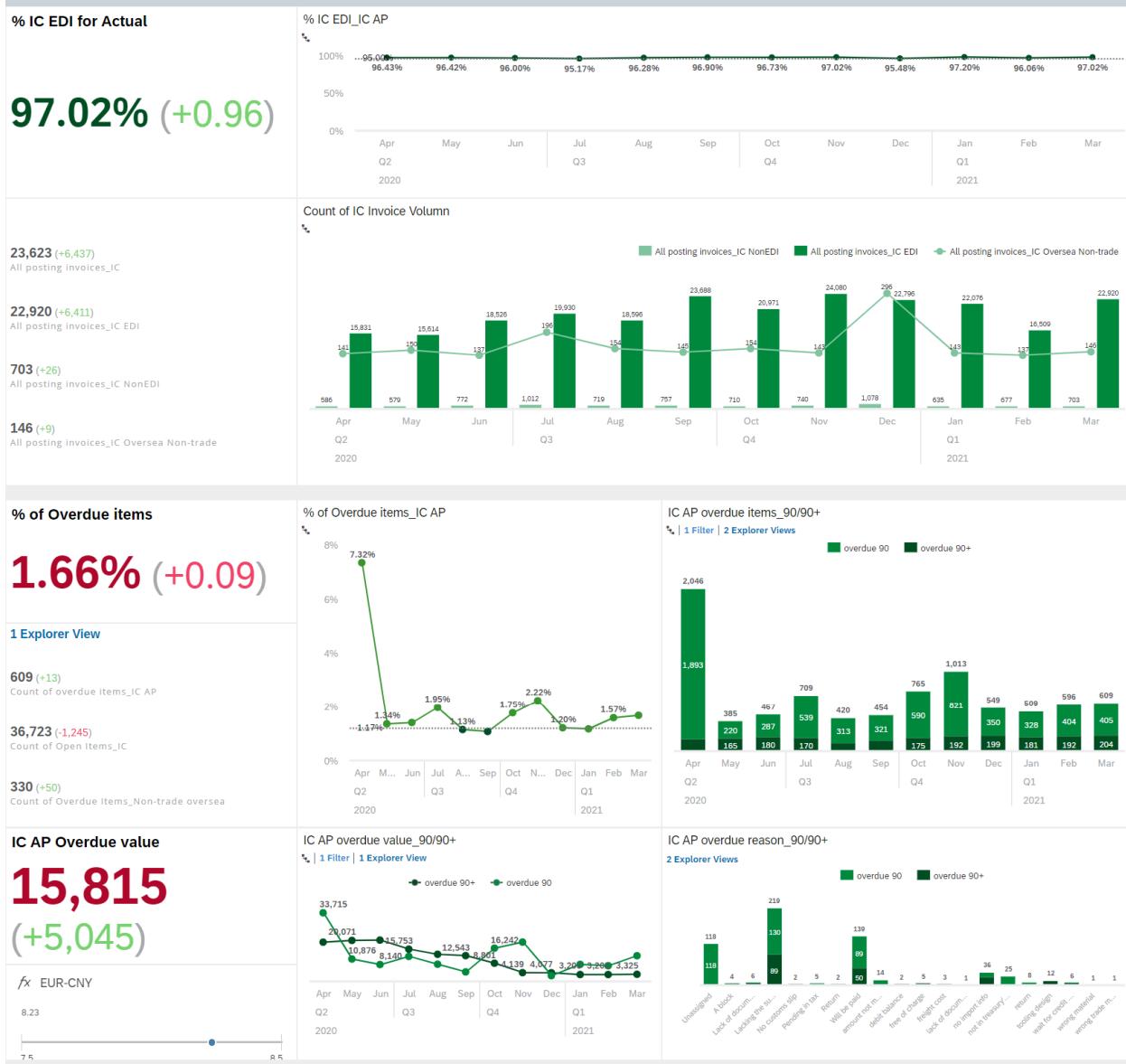
Target Improvement to AP aging

0.02

0 1

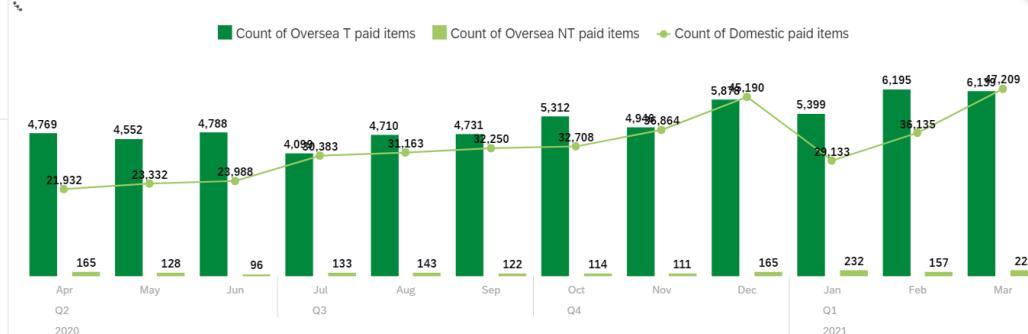


## Intercompany & Payments



**Payment Volume****53,572 (+11,085)**47,209 (+11,074)  
Count of Domestic paid items6,139 (-56)  
Count of Oversea T paid items224 (+67)  
Count of Oversea NT paid items

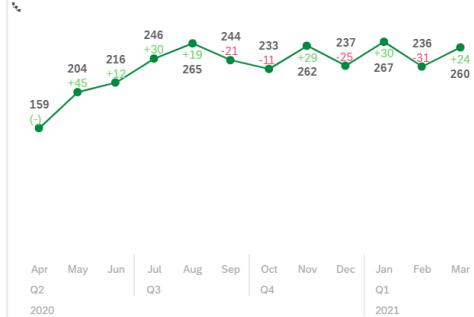
## Payment volume

**% of Manual payment runs****64.52%****% of Normal payment runs****28.54%****% of Special payment runs****6.95%**

## Payment Runs\_Special &amp; Normal



## Payment Runs\_Manual

**% Overdue IC AR****1.81%**

## % Overdue items\_IC AR

**1 Explorer View**

494

Count of overdue items\_IC AR

27,346

Count of open items\_IC AR

14,655.35

ABS overdue value of ICAR in TEUR

## overdue items\_IC AR

2 Explorer Views

● overdue 90+ ● overdue 90



## IC AR overdue reason\_90/90+

1 Filter

7

Unsigned

Amount small co...

Business argue

Check with custo...

Credit note will b...

The bank missed...

Treasury no pay...

236

172

56

4

**Payment Volume**

1 Filter | 1 Explorer View

**53,572**

Count of paid items

**47,209**

Count of Domestic paid items

**6,139**

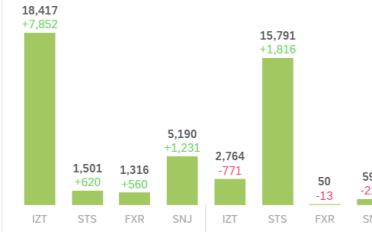
Count of Oversea T paid items

**224**

Count of Oversea NT paid items

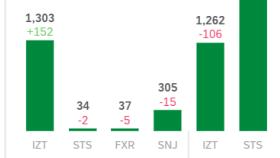
## Domestic paid items by LE

2 Filters | 1 Variance



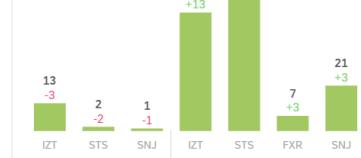
## Oversea T paid items by LE

2 Filters | 1 Variance



## Oversea NT paid items by LE

2 Filters | 1 Variance

**Payment Runs**

1 Filter | 2 Explorer Views

**403**

Count of payment runs\_Total

**28**

Count of special payment run

**115**

Count of normal payment run

**260**

Count of manual payment runs

## Special payment runs\_3RD

1 Filter | Top 5



## Manual payment runs\_3RD

Top 10

339324 普恩信(杭州)机械部...	<b>16</b>
253050 上海宝井铜材加工配送...	<b>11</b>
277367 上海戈冉泊精模科技有...	<b>11</b>
290081 苏州星诺奇科技股份有...	<b>9</b>
277903 江阴市精成数控有限公司	<b>8</b>
279202 浙江辛子精工机械有限...	<b>7</b>
277398 宁波海亚特滚子有限公司	<b>5</b>
282926 浙江西密克轴承股份有...	<b>4</b>
288731 江苏苏美达德隆汽车部...	<b>4</b>
356726 江苏省第一建筑安装集...	<b>4</b>

Credit Management



**83.74 (+0.57)**

Δ 2021 Feb

### Overdue Rate

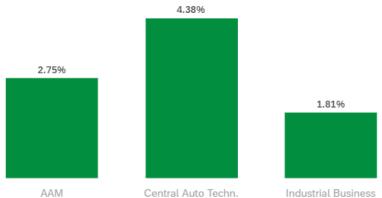
| 1 Variance

**3.77% (+0.42)**

Δ 2021 Feb

### Overdue Rate by Division

in % | 2 Filters



### CLR Approval Days

**0.17 (+0.02)**

CLR Approval Days (Δ 2021 Feb)

### CLR Approval Days Trend

0.13 → 0.15 → 0.17

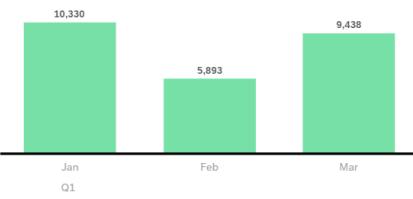
Jan  
Q1  
2021

### DN Release in Minutes

**28.88 (+2.62)**

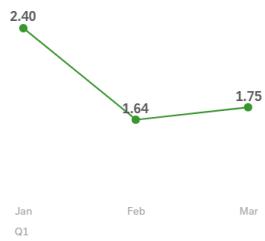
Average DN Release Time (Δ 2021 Feb)

### # of Released DN



Jan  
Q1  
2021

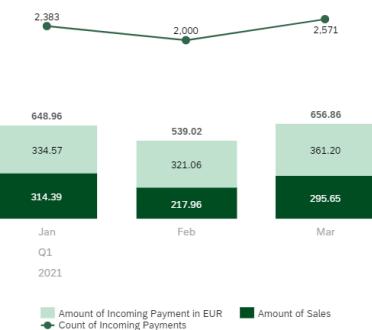
### Bank Receiving to Clearing Days



Jan  
Q1  
2021

### # of Incoming Payments

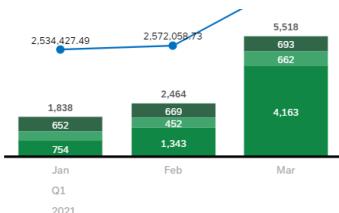
in Million, --



Amount of Incoming Payment in EUR  
Count of Incoming Payments

### Number of Overdue Items

3,160,070.09



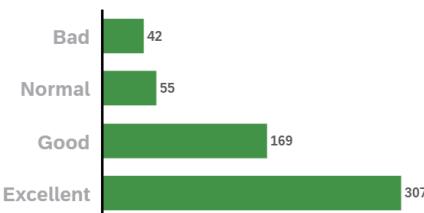
90+/Count of Open AR Items  
31-90/Count of Open AR Items  
1-30/Count of Open AR Items  
90+ Days Overdue Amount in EUR

SCHAEFFLER

## Payment Behavior Analysis

10:45 April 16, 2021

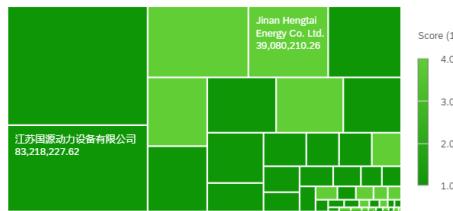
### Payment Behavior Overview by Credit Account



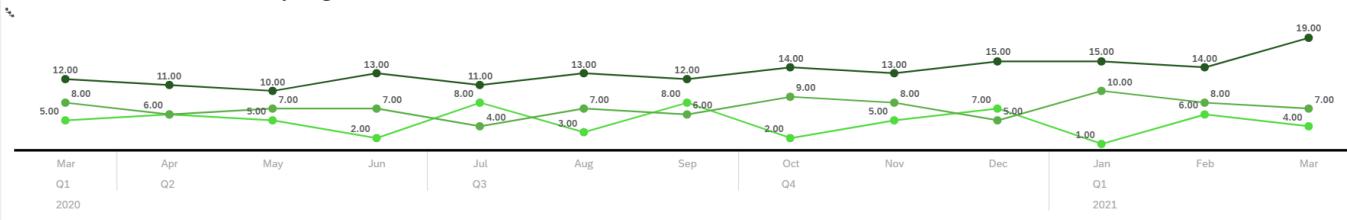
### Payment Behavior - BAD

Bad  
**42**

### Overview (Bad Customer)



### Trend for overdue credit account by range



### Overdue % Excluded

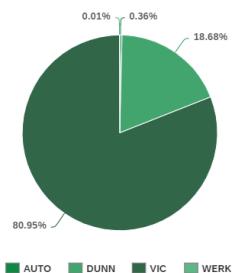


### Bad customer details

3 Filters | 2 sorts applied

credit_account	Payme...	1-30	31-90	90+	sales_by_ca	1-30 overdr...	31-90 overdr...	90+ overdr...	count items for ran...	count items for ra...	count items for ra...	Score by ...
100709 浙江吉润春晓汽车部件有限公司	Bad	567,478.36	1,134,956.72	972,071.84	1,800,028.30	31.53 %	63.05 %	54.00 %	1.00	2.00	4.00	1.00
22086 Opel Automobile GmbH	Bad	224,078.40	293,219.06	704,884.12	-226,286.55	-	-	-	0.00	1.00	1.00	1.00
24505 STV Slovakia a.s.	Bad	74,591.62	33,517.49	40,860.22	447,549.72	16.67 %	-	-	2.00	1.00	2.00	1.00
32872 Honda of the UK Manufacturing Ltd.	Bad	4,415,493.00	8,517,789.99	22,854,304.29	17,418,947.51	25.35 %	48.90 %	131.20 %	6.00	7.00	10.00	1.00
33455 泰盈汽车技术中心有限公司	Bad	739,117.64	1,636,435.28	3,326,029.38	11.11 %	22.22 %	49.20 %	2.00	3.00	5.00	1.00	
44452 江苏国源动力设备有限公司	Bad	13,500.00	-	83,204,727.62	288,606,840.11	-	-	28.83 %	0.00	0.00	13.00	1.00
46021 Honda Motor Europe Logistics NV Procu...	Bad	170,615.97	230,279.27	85,715.77	1,970,664.15	-	11.69 %	-	3.00	4.00	1.00	1.00
51680 Perusahaan Otomobil Nasional Bernad	Bad	902,235.53	2,115,910.28	1,130,169.28	26,811,674.50	-	10.13 %	-	1.00	2.00	3.00	1.00
51704 Valeo Sistemas Electricos S.A. de C.V.	Bad	119,347.12	89,574.14	161,767.31	1,345,677.53	-	-	12.02 %	2.00	2.00	2.00	1.00
52210 国铁集团北京局集团有限公司物资供应段	Bad	-	-	1,327,750.00	46,400,295.04	-	-	-	0.00	0.00	1.00	1.00
52213 天津铁路股份有限公司(奥马)车辆段	Bad	95,198.99	107,808.19	1,630,603.00	-	-	-	-	0.00	0.00	0.00	1.00
53432 甘肃西安车务有限公司	Bad	-	-	64,927.20	-	-	-	-	-	-	0.00	1.00
58649 上汽通用五菱汽车股份有限公司	Bad	28,223,817.97	12,697,780.43	8,561,703.40	870,210,331.62	-	-	-	2.00	1.00	1.00	1.00
59514 AP Automotive Products S.r.l. Societa U...	Bad	309,989.25	-	-	-	-	-	-	0.00	-	-	1.00
60307 国电联合动力技术(保定)有限公司	Bad	1,890,064.94	3,780,129.88	1,890,064.94	11,340,389.64	16.67 %	33.33 %	16.67 %	1.00	2.00	1.00	1.00
61398 广汽菲亚特克莱斯勒汽车有限公司	Bad	13,684,437.41	9,617,723.54	2,592,864.48	307,024,643.05	-	-	-	2.00	2.00	1.00	1.00
61514 KOMTEK Spolka z ograniczoną odpowid...	Bad	162,843.78	840,600.86	-	6,814,710.01	-	12.34 %	-	1.00	4.00	0.00	1.00
61853 大秦铁路股份有限公司(湖东)车辆段	Bad	45,000.00	90,000.00	180,000.00	-	-	-	-	0.00	0.00	0.00	1.00
61862 广汽乘用车有限公司	Bad	2,399,221.75	3,314,911.51	3,928,329.22	35,227,254.76	-	-	11.15 %	3.00	6.00	6.00	1.00
66526 广汽三菱汽车有限公司	Bad	31,665.31	67,736.14	72,141.66	3,301,390.48	-	-	-	1.00	2.00	2.00	1.00
68877 Setco Automotive Limited	Bad	97,762.26	368,323.20	1,107,362.52	-	-	-	-	0.00	0.00	0.00	1.00
74861 FCA US	Bad	-	-	569,812.04	-	-	-	-	-	-	0.00	1.00
75231 国电联合动力技术(连云港)有限公司	Bad	3,622,624.47	5,040,173.18	2,520,086.59	20,160,692.72	17.97 %	25.00 %	12.50 %	2.00	2.00	1.00	1.00
76128 General Motors do Brasil Ltda.	Bad	12,235,534.25	9,354,772.48	7,096,389.44	284,619,444.78	-	-	-	2.00	2.00	2.00	1.00
79711 宁波吉利罗佑发动机零部件有限公司	Bad	1,538,049.78	3,076,099.56	14,768,568.48	428,352.86	35.06 %	718.12 %	3,447.75 %	1.00	2.00	13.00	1.00
8 华晨宝马集团沈阳有限公司	Bad	-	-	6,630,700.00	-	-	-	-	-	-	0.00	1.00
82344 广汽上中下自动变速器有限公司	Bad	820,844.21	1,910,186.40	24,191,353.94	2,474,940.74	33.17 %	77.18 %	977.45 %	0.00	0.00	1.00	1.00
83533 宁波吉利汽车研究开发有限公司	Bad	68,589,313.01	33,575,094.00	13,162,797.01	184,521,991.99	37.17 %	18.20 %	-	4.00	2.00	3.00	1.00

## Case Type Overview



## VIC by processor

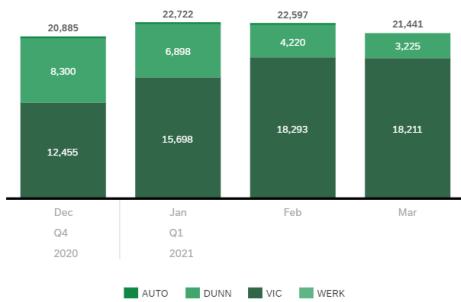
VIC

FANHAJ	VIC	1,202
SHIYTI	VIC	424
WANGYFN	VIC	719
FANGYU	VIC	569
WUYO	VIC	559
GUYIW	VIC	449
CHENXQ	VIC	391
ZHANGGOH	VIC	383
ZHANGYTN	VIC	383
WTUTIN	VIC	347
ZHANGYDA	VIC	339
ZHANGA	VIC	302
HOIJUN	VIC	280
CHANGYUA	VIC	268
HUJIA	VIC	260
YUJYA	VIC	247
SHICHI	VIC	218
YAOJIE	VIC	215

## VIC by customer

33493 重庆长安汽车股份有限公司	2,224
33264 吉奥汽车股份有限公司	1,666
99073 菲亚特克莱斯勒动力总成有限公司	1,391
99523 绵阳埃泰克传动科技有限公司(衡水)有限公司	680
77491 浙江汽轮机(集团)有限公司	439
59338 工业泵测试(天津)有限公司	410
33214 上海汽车变速器有限公司	383
81692 上汽通用汽车有限公司武汉分公司	307
79972 长城汽车股份有限公司衡水分公司	306
33063 江铃汽车股份有限公司	294
99813 马来西亚(江苏)有限公司	291
99478 广州精盛动力总成有限公司	275
45124 东风汽车有限公司东风日产商用公司	247
57563 大众汽车自动变速器(天津)有限公司	230
79962 安徽江淮汽车集团股份有限公司发动机分	226
33062 东风悦达起亚汽车有限公司	213
62117 陕西飞德利新材料有限公司	185
60494 华晨宝马汽车有限公司	160

## Trend for the past 4 months



## DUNN by processor

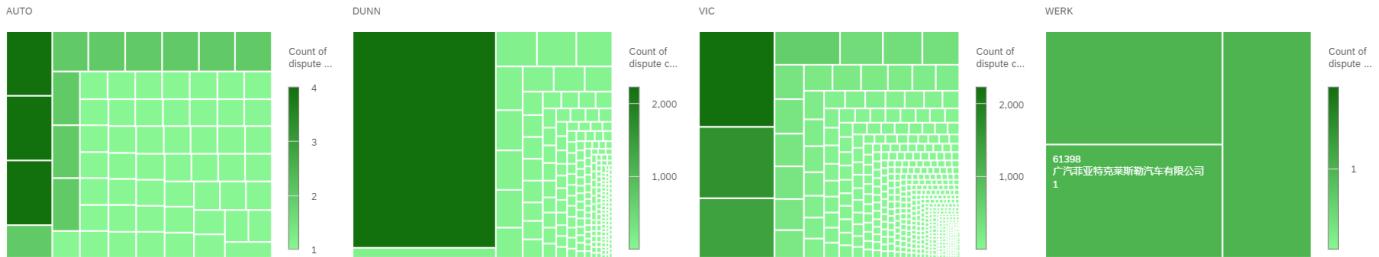
DUNN

LUJNH	DUNN	1,308
YANGJGT	DUNN	1,260
SHIYTI	DUNN	758
CHENNN	DUNN	359
ZHANGGOH	DUNN	203
LUYO	DUNN	109
GAOJAH	DUNN	74
WANGYFN	DUNN	57
CHANGX	DUNN	33
ZHANQA	DUNN	17
FANHAJ	DUNN	14
ZHANLA	DUNN	5
DINGDOG	DUNN	3
HOIJUN	DUNN	3
XIAOSHE	DUNN	3
CHEFAG	DUNN	2
LIQ	DUNN	2
LIUXME	DUNN	2
ZHUI	DUNN	2
CHENZHII	DUNN	1
GUYIW	DUNN	1

## DUNN by customer

99073 菲亚特克莱斯勒动力总成有限公司	2,218
99525 镇江传动机械技术有限公司	107
33062 东风悦达起亚汽车有限公司	106
33214 上海汽车变速器有限公司	99
62117 陕西飞德利新材料有限公司	93
99813 马来西亚(江苏)有限公司	84
83353 宁波吉利汽车动力总成有限公司	78
81010 江铃汽车股份有限公司(盖州分公司)	75
99813 马来西亚(江苏)有限公司	74
33264 吉奥汽车有限公司	58
100763 菲亚特传动系统(江苏)有限公司	56
61062 广州爱卡汽车有限公司	56
55593 大众汽车零部件有限公司	49
59680 广州广汽吉奥电动汽车有限公司	43
100471 四川吉利汽车制造有限公司	31
53442 长城汽车股份有限公司	31
93480 吉利汽车研究院(宁波)有限公司	31
93669 吉奥汽车有限公司	31
82977 浙江吉利汽车研究院有限公司宁波分院	27
33064 一汽-大众汽车有限公司	26
61398 广汽菲亚特克莱斯勒汽车有限公司	26
79985 大众汽车自动驾驶器(天津)有限公司	26

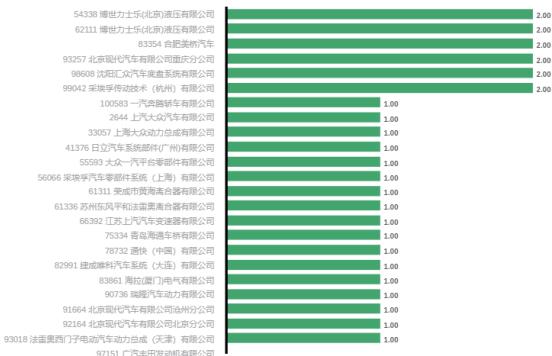
## Detail for each case type



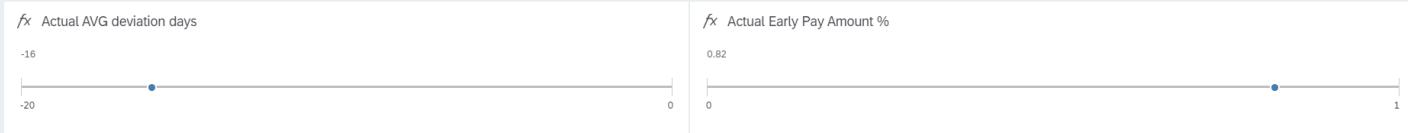
## VIC Setting Rules Checking



### Top customers in past 3 months



### Payment Term Checking



TOP 20 2020. DEC.



TOP 20 2021.JAN.



TOP 20 2021.FEB



## Glossary:

1. VIC AVG deviation days (以VIC为基准计算得出的平均付款差异天数)
  2. Actual AVG deviation days (以实际开票日为基准计算得出的平均付款差异天数)
  3. VIC early pay amount % (以VIC为基准计算得出提前付款占付款金额的比重)
  4. Actual early pay amount % (以实际开票日为基准计算得出提前付款占付款金额的比重)

**Customer Deviation Of Pay Days=All Cleared BillingΣ[ABS(Billing Amt)×(clearing date- net due date)]/Σ[ABS(Billing Amt)]**

# Monitoring

With Azure Data Factory it's straightforward to keep an overview of all the data pipeline runs in the solution space. You can quickly see which pipelines ran over a period of time and how it was triggered.

### Pipeline runs

Triggered   Debug   Run   Cancel   Refresh   Edit columns   List   Gantt

Search by run ID or name   Beijing, Chongqing, ... : Last 30 days   Pipeline name : All   Status : All   Runs : Latest runs  

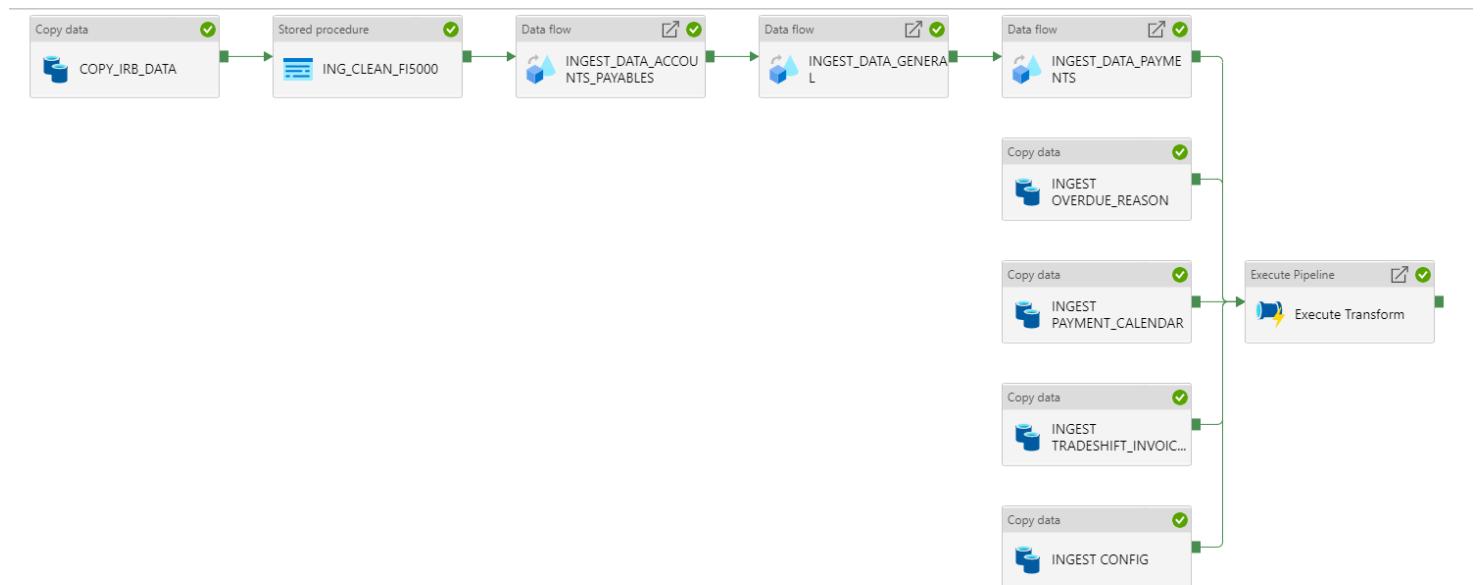
Showing 1 - 100 items

Pipeline name	Run start ↑↓	Run end	Duration	Triggered by	Status	Run	Parameters	Annotations
PIP_P2P_INGEST	3/19/21, 6:00:00 AM	3/19/21, 10:15:51 AM	04:15:50	Weekly_Run	Succeeded	Original		
PIP_P2P_TRANSFORMA...	3/19/21, 7:18:30 AM	3/19/21, 10:15:47 AM	02:57:17	101b359b-1d0d-4352-a7	Succeeded	Original		
PIP_P2P_OUTPUT	3/19/21, 10:14:45 AM	3/19/21, 10:15:46 AM	00:01:00	c2a64405-0d03-4c25-aa:	Succeeded	Original		
PIP_P2P_OUTPUT	3/19/21, 1:57:43 PM	3/19/21, 1:58:46 PM	00:01:03	Manual trigger	Failed	Original		
PIP_TEST	3/19/21, 2:23:00 PM	3/19/21, 2:28:54 PM	00:05:54	Test	Failed	Original		
PIP_TEST	3/19/21, 2:38:00 PM	3/19/21, 2:53:41 PM	00:15:40	Test	Failed	Original		
PIP_TEST	3/19/21, 2:53:00 PM	3/19/21, 3:08:28 PM	00:15:28	Test	Failed	Original		
PIP_TEST	3/19/21, 3:08:01 PM	3/19/21, 3:18:40 PM	00:10:39	Test	Failed	Original		
PIP_TEST	3/19/21, 3:23:00 PM	3/19/21, 3:37:23 PM	00:14:22	Test	Failed	Original		
PIP_TEST	3/19/21, 3:38:00 PM	3/19/21, 3:41:27 PM	00:03:27	Test	Failed	Original		
> PIP_TEST	3/19/21, 3:43:58 PM	3/19/21, 3:53:13 PM	00:09:14	Manual trigger	Failed	Rerun (Latest)		
PIP_TEST	3/19/21, 3:53:01 PM	3/19/21, 4:30:12 PM	00:37:11	Test	Succeeded	Original		
PIP_TEST	3/19/21, 4:08:00 PM	3/19/21, 4:24:40 PM	00:16:39	Test	Failed	Original		
PIP_TEST	3/19/21, 4:23:00 PM	3/19/21, 4:38:37 PM	00:15:36	Test	Failed	Original		
PIP_TEST	3/19/21, 4:38:00 PM	3/19/21, 4:53:54 PM	00:15:54	Test	Failed	Original		
PIP_TEST	3/19/21, 4:53:00 PM	3/19/21, 5:24:26 PM	00:31:26	Test	Succeeded	Original		
PIP_TEST	3/19/21, 5:08:00 PM	3/19/21, 5:48:38 PM	00:40:38	Test	Succeeded	Original		
PIP_TEST	3/19/21, 5:23:00 PM	3/19/21, 5:24:46 PM	00:01:45	Test	Failed	Original		
PIP_TEST	3/19/21, 5:38:00 PM	3/19/21, 5:54:27 PM	00:16:27	Test	Failed	Original		

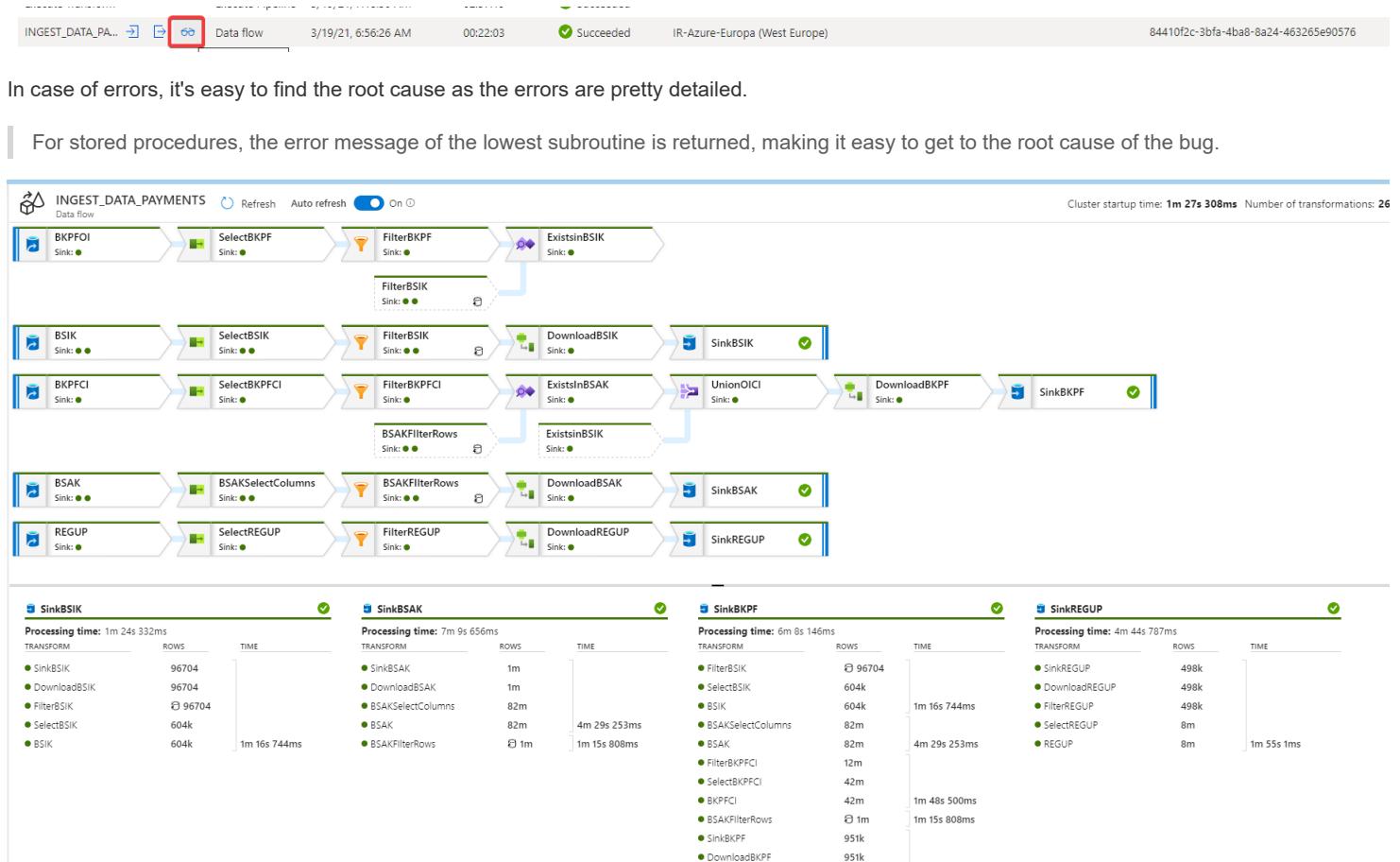
  

Pipeline name	Run start ↑↓	Run end	Duration	Triggered by	Status	Run	Parameter:
PIP_P2P_INGEST	3/19/21, 6:00:00 AM	3/19/21, 10:15:51 AM	04:15:50	Weekly_Run	Succeeded	Original	
PIP_P2P_TRANSFORMA...	3/19/21, 7:18:30 AM	3/19/21, 10:15:47 AM	02:57:17	101b359b-1d0d-4352-a7	Succeeded	Original	
PIP_P2P_OUTPUT	3/19/21, 10:14:45 AM	3/19/21, 10:15:46 AM	00:01:00	c2a64405-0d03-4c25-aa:	Succeeded	Original	

Every pipeline is set up in a modular fashion. The monitoring function gives you an accessible overview of which modules were running successfully or which one caused the pipeline to stop.

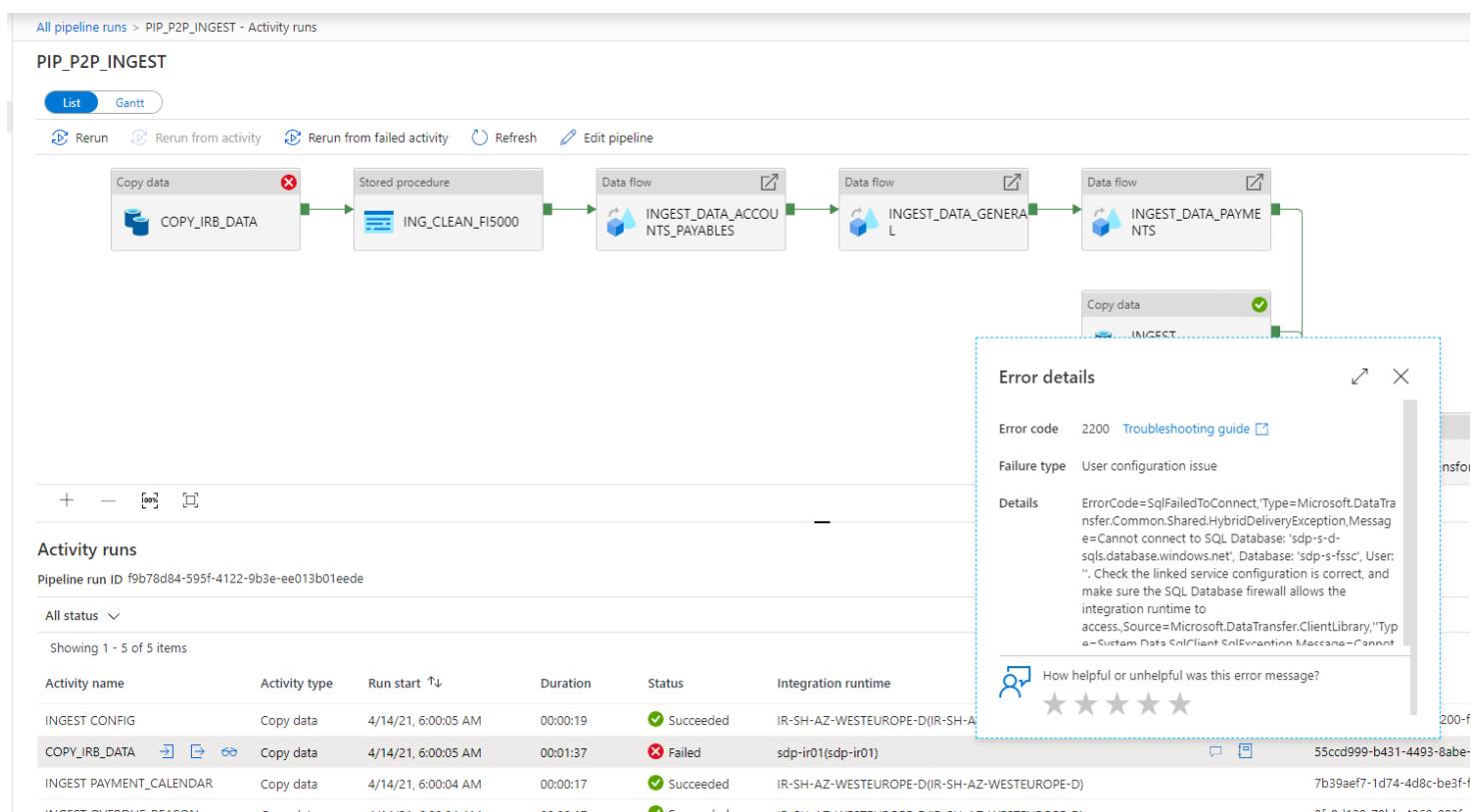


For data flows, it's even possible to see how each action performed and how many rows were processed, etc.



In case of errors, it's easy to find the root cause as the errors are pretty detailed.

For stored procedures, the error message of the lowest subroutine is returned, making it easy to get to the root cause of the bug.



PIP\_O2C\_TRANSFORM

List Gantt Rerun Rerun from failed activity Refresh Edit pipeline

Error details

Error code: 2402  
Failure type: User configuration issue  
Details: Execution fail against sql server. Sql error number: 2801. Error Message: The definition of object 'p\_tp30\_all\_cust\_items' has changed since it was compiled.  
Source: Pipeline PIP\_O2C\_TRANSFORM

Activity runs

Pipeline run ID: 1cd07371-05f6-4ae4-8d53-660e2e7c6ac0

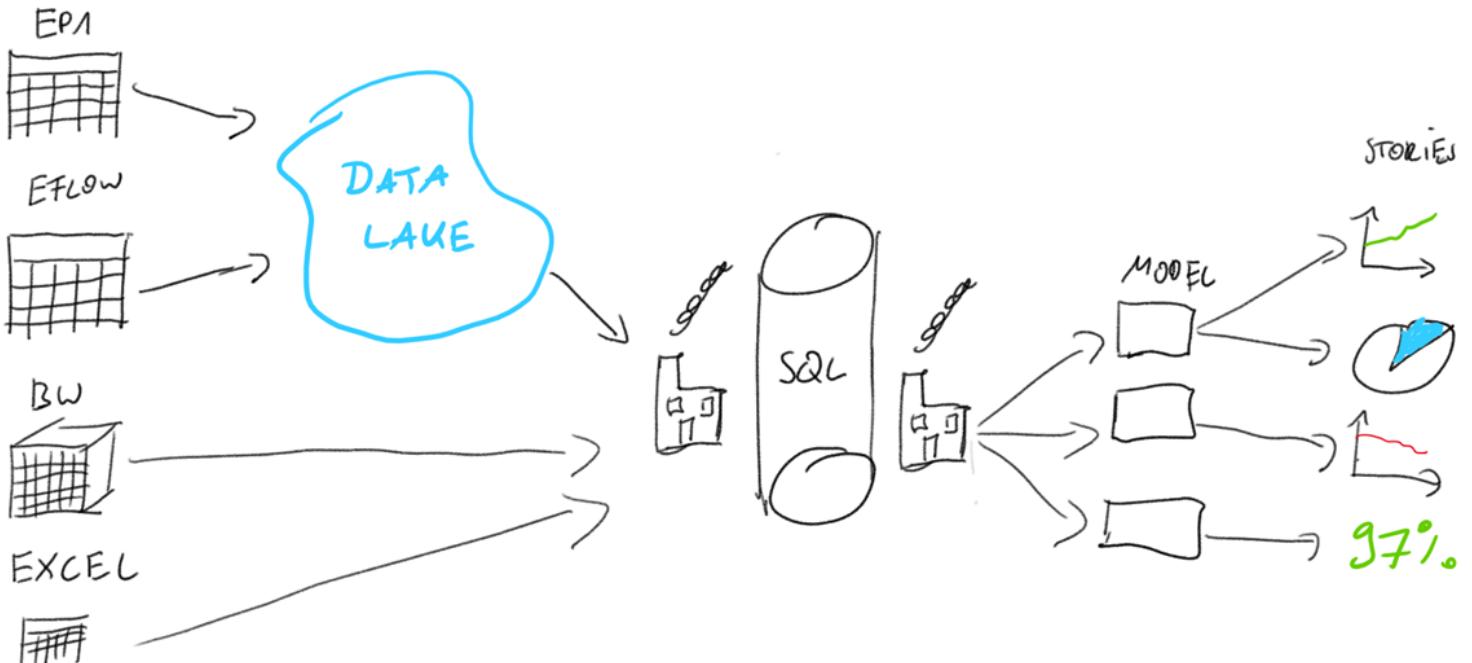
All status ▾

Showing 1 - 4 of 4 items

Activity name	Activity type	Run start ↑	Duration	Status	Integration runtime	
TP3_EXEC	Stored procedure	3/18/21, 10:36:27 AM	00:37:58	✖ Failed	DefaultIntegrationRuntime (West Europe)	c0f87ff6-309a-4731-933b-220
TP2_EXEC	Stored procedure	3/18/21, 10:35:29 AM	00:00:57	✔ Succeeded	DefaultIntegrationRuntime (West Europe)	5b231d80-6a95-4483-8963-21
TP1_EXEC	Stored procedure	3/18/21, 10:34:14 AM	00:01:15	✔ Succeeded	DefaultIntegrationRuntime (West Europe)	8242c429-9140-4a9c-80c4-55
CLN_EXEC	Stored procedure	3/18/21, 10:28:27 AM	00:05:46	✔ Succeeded	DefaultIntegrationRuntime (West Europe)	79357c62-190d-413d-9976-8t

## Data Flow

The pipelines are currently running on a weekly basis through below architecture:



Step	Time	Contact Person
Export Data from EP1	Sundays	Regin, Norbert / Horvat, Radu Claudiu
Importing data from EP1 and EFLOW to data lake	Tuesdays	Mandeep, Kaur
Running Pipeline in Data Factory	Thursday	Wang, Yanhu
Importing data to SAC	Thursday	Shen, Jie / Bao, Jianxin

# Flow Date Tracking

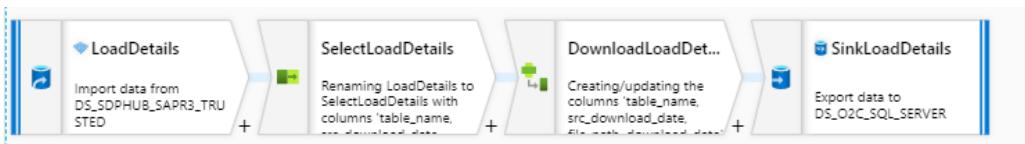
## Export date from EP1

This is the most important date as business needs to know when we exported the SAP data from EP1. Therefore it is tracked through the whole pipeline.

There is a table called e-dlk-uebersicht-ladedatum tracking the export time from SAP.

Objekt	Tabelle	Quelle	Erstelldatum_in_DEV_DATA
EP1_KEKO	EP1	EP1	2021-04-11T06:19:40.58
EP1_PROJ	EP1	EP1	2021-04-09T19:42:39.067
EP1_TCLA	EP1	EP1	2021-04-09T19:42:10.91

This table is part of the dataflows for both o2c and p2p.



It is then saved in the ing\_load\_details tables in dbo and o2c schema.

	table_name	src_download_date
1	KEKO	2021-04-04 04:31:49.1370000
2	PROJ	2021-04-02 19:56:50.9300000
3	TCLA	2021-04-02 19:56:07.6630000
4	KONH	2021-04-04 00:13:07.7770000
5	VTFA	2021-04-05 02:16:47.2600000
6	EIPA	2021-04-03 02:40:24.4370000
7	EKKO_Selection_jTrack	2021-04-05 20:20:12.3670000
8	LIK_P_Selection_jTrack	2021-04-05 21:29:46.5030000
9	LIPS_Selection_jTrack	2021-04-06 01:58:35.7630000
10	VEKP	2021-04-05 08:20:52.0670000
11	VBDK_Selection_jTrack	2021-04-06 05:24:01.5600000
12	VBRK_Selection_jTrack	2021-04-06 02:02:25.0300000

In every CLN\_\* procedure it is then updated for each table:

```
alter procedure [dbo].[p_cln_bsak] as
begin
    update cln_bsak
    set cln_bsak.src_download_date = src.src_download_date
    from cln_load_details as src
    where table_name = 'bsak'
```

In the SAP tables, it is then populated finally as src\_download\_date and loaded into SAC:

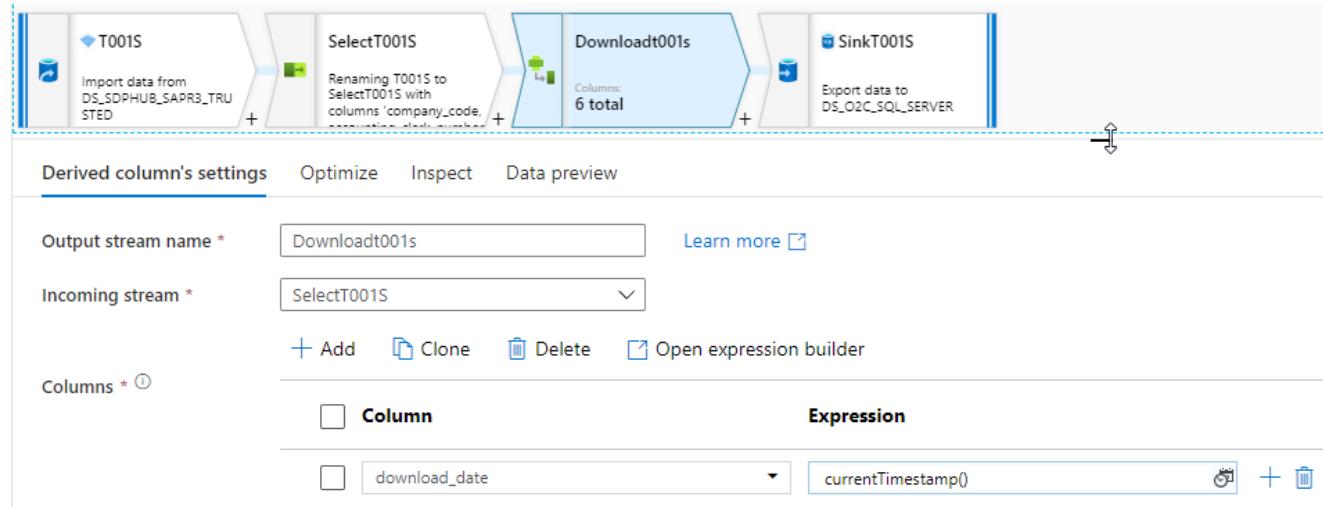
The screenshot shows the SAP Data Modeler interface. The top navigation bar says "Files / Open Items New AR Model". Below it, there are tabs for "Model" and "Data Management", with "Model" being the active tab. A search bar is present. The main area displays a hierarchical tree of data objects. The "Account" node has several children: "Account", "Date", and several date-related objects like "clearing\_date", "posting\_date", etc. One object, "src\_download\_date", is highlighted with a red rectangular box.

## Import data from EP1 and EFLOW to the data lake

We can track this date in the table e-dlk-uebersicht-ladedatum, too, but we don't populate or check it.

## Running Pipeline in Data Factory

Every Data Flow populates the download date as the last step:



For BW reports, the download date is added in the copy data step:

General Source Sink Mapping Settings User properties

Source dataset \* DS\_BW\_FI5000 Open New Preview data Learn more

Use query Table Query

Query SELECT \* FROM connection\_layer\_udc...

Query timeout (minutes) 120

Additional columns + New Delete

NAME	VALUE
src_download_date	@utcnow()

This way, we can check in the SQL server when we last downloaded the data in the table:

CLIENT	COMPANY_CODE	COMPANY_NAME	CITY	FILE_PATH	DOWNLOAD_DATE	COMPANY_NAME_SHORT	src_download_date
100	0083	Schaeffler China Co. Ltd.	Taicang	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	IZT	2021-04-02 18:18:34.563
100	0189	Schaeffler Holding(China)	Shanghai	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	SAM	2021-04-02 18:18:34.563
100	0199	Schaeffler T. (Shanghai)	Shanghai	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	STS	2021-04-02 18:18:34.563
100	0289	Schaeffler (Ningxia) Co.	Ningxia	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	FXR	2021-04-02 18:18:34.563
100	0369	Schaeffler Fric. Products	Suzhou	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	LFC	2021-04-02 18:18:34.563
100	0371	Schaeffler Aerospace B.Co	Taicang	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	SAB	2021-04-02 18:18:34.563
100	0377	Schaeffler (Nanjing) Co.	Nanjing	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	SNJ	2021-04-02 18:18:34.563
100	0404	Schaeffler (Xiangtan)	Xiangtan	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	SXT	2021-04-02 18:18:34.563
100	0426	Schaeffler IDT (Changsha)	Changsha	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	SIC	2021-04-02 18:18:34.563
100	0429	ETC Trading	Shanghai	/sapr3/ep1-cdm/t001/undefined/sapr3_ep1-cdm_t001...	2021-04-12 06:19:09.8940000	STE	2021-04-02 18:18:34.563

It is then populated along with the src\_download date and can be found in SAC too:

Account

Date

clearing\_date

posting\_date

document\_date

entry\_date

baseline\_date

dunning\_date\_last

download\_date

src\_download\_date

last\_internal\_review

dispute\_pro\_411b2t3k1s

dispute\_pla\_5e5d2a02r6

## Importing data to SAC

For every Model, an import job from the file server is set up weekly:

The job can be refreshed manually as well by hitting the Refresh button. We can change the scheduling by clicking the calendar button.

On the right side, the last imports are displayed. If there are two files merged, it is also showing here:

In this case, the OPEN ITEMS NEW AR MODEL contains the main file (STA\_OPEN\_CUST\_ITEMS) and an additional file from the Excel Upload folder:

Name	Date modified	Type	Size
FF	4/9/2021 3:49 PM	File folder	
AP Data Input.xlsx	4/14/2021 9:58 AM	Microsoft Excel Worksheet	
AD KPI Document.xlsx	3/28/2021 3:55 PM	Microsoft Excel Worksheet	
<b>Business Division.XLSX</b>	4/9/2021 4:51 PM	Microsoft Excel Worksheet	
TU24.XLSX	3/24/2021 11:03 AM	Microsoft Excel Worksheet	

We can check the mapping of the model from the base file here:

## Mapping Summary for sta\_open\_cust\_items\_1618198371110\_QueryKNE1J292TDV3W

### Mapping Summary

Mapping Target	Attribute	Mapped Column
Account	Account	-
company_code	Dimension ID	company_code
document_number	Dimension ID	document_number
fiscal_year	Dimension ID	fiscal_year
line_item	Dimension ID	line_item
customer_number	Dimension ID	customer_number
	Description	customer_name_chinese
special_gl_indicator	Dimension ID	special_gl_indicator
assignment_number	Dimension ID	assignment_number
currency	Dimension ID	currency
reference	Dimension ID	reference
document_type	Dimension ID	document_type
posting_key	Dimension ID	posting_key
debit_credit	Dimension ID	debit_credit

### Action Log

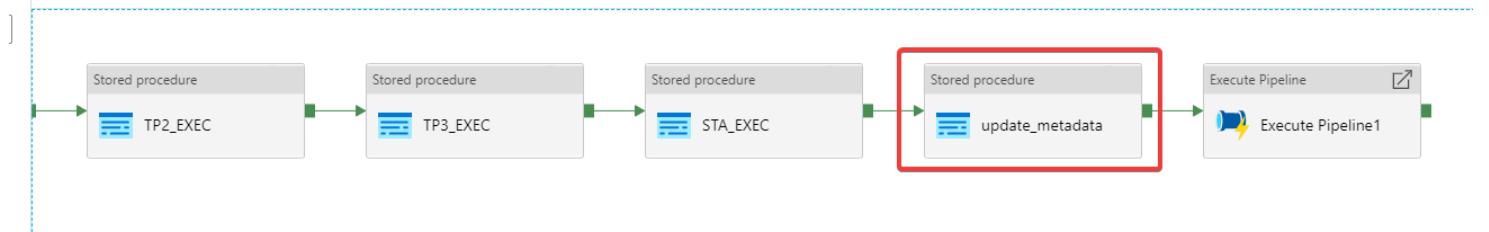
**Combine**  
 sta\_open\_cust\_items\_1618198371110\_Query  
 KNE1J292TDV3W with  
 Business\_Division\_1618198521544\_QueryKN  
 E1MABSPD1A7

Additional calculations that are done in the model can be checked here as well. They will appear on the right side of the screen.

## Compare Q-D System

Changes should be implemented in the test system first and then transported to the P-System. We need a way to quickly track how the row counts in the SQL Server changed to verify if the development is successful after the change.

To do this, the metadata (column count and row count) for each table is written down in another table called metadata:



	tablename	rowscount	colcount
1	dbo.CLN_DRDC	43112	7
2	o2c.cln_adrc	43112	7
3	dbo.CLN_BKPF	1013701	7
4	o2c.cln_bkpf	625458	7
5	o2c.cln_bsad	598801	42
6	dbo.CLN_BSAK	1121861	30
7	o2c.cln_bsad	79961	42
8	dbo.CLN_BSIK	85374	30
9	dbo.CLN_EBAN	138024	7
10	o2c.cln_eflowdn	521652	4
11	o2c.cln_eflowtask	148148	11
12	dbo.CLN_EKBE	3854931	17

Because the d-system and p-system are completely different SQL Servers, it's not easy to directly compare the metadata tables from Q and P-System. We built a small Power BI Dashboard that can automatically pull data from both Q and P for comparison:

tablename	D_ROW_CNT	P_ROW_CNT	D_COLCNT	P_COLCNT	ROWMATCH	COLMATCH
DBO.CLN_DRDC	43112	43112	7	7	█	█
DBO.CLN_BKPF	1013701	1013701	7	7	█	█
DBO.CLN_BSAK	1121861	1121861	30	30	█	█
DBO.CLN_BSIK	85374	85374	30	30	█	█
DBO.CLN_EBAN	138024	138617	7	7	█	█
DBO.CLN_EKBE	3854931	3857582	17	17	█	█
DBO.CLN_EKKO	52	52	8	8	█	█
DBO.CLN_FI5000	280624	281795	40	40	█	█
DBO.CLN_LOAD_DETAILS	199	199	4	4	█	█
DBO.CLN_OCRLOG	244289	244291	7	7	█	█
DBO.CLN_OVERDUE_REASON	7205	7205	8	8	█	█
DBO.CLN_PAYMENT_CALENDAR	732	732	6	6	█	█
DBO.CLN_REGUP	68845	68845	10	10	█	█
DBO.CLN_T001	10	10	8	8	█	█
DBO.CLN_T001S	4818	4818	8	8	█	█
DBO.CLN_T024	2972	2972	5	5	█	█
DBO.CLN_TRADESHIFT_INVOICES	267622	242428	12	12	█	█
DBO.CLN_VF_KRED	14740	14740	14	14	█	█
DBO.CONFIG	133	133	6	6	█	█

This way, it's easy to double-check whether the change had an unwanted impact on other tables.

## Pipeline runs

we can easily track in the overall pipeline runs

# Frequent Questions / Issues / Requests

## Requests

### We (Business) want to add a new column to model X

There are three options to do this:

1. Create a new SAC Model and Link the data in the story

The screenshot shows the SAP Analytics Cloud (SAC) interface. At the top, there's a navigation bar with icons for search, lightbulb, notifications (51), and help. Below it is a toolbar with various icons for display settings and data management. The main area is titled 'Builder' and shows a 'Data Source' section. A data source named 'PoC\_Invoices\_IRB' is listed, and its details are expanded, showing it's linked to 'AP\_Data\_Input 2'. A red box highlights this expanded section. A warning message at the bottom right of the builder panel says: '⚠ This data source is optimized for story building and does not automatically refresh. Read More'. At the bottom of the screen, there's a chart structure panel with tabs for Comparison, Trend, and Distribution.

- Business can create the model and link themselves
- Can link data from another context (e.g., Actual model and Target model)
- Functionality for calculations limited
- Clunky /Strange errors sometimes
- Business has to maintain the data manually

## 2. Add the data in the SAC Model via the Upload Excel folder on the SAC Fileserver

The screenshot shows the Data Timeline interface. On the left, there's a sidebar with navigation icons and a "Last Updated" section showing "Apr 15, 2021 8:55:09". The main area is titled "Data Timeline" and shows a "Next scheduled refresh" with a checkmark. Below it, a log entry for a query refresh is displayed:

- sta\_all\_cust\_items\_1618212537062\_QueryKNE...
- Apr 15, 2021 at 8:47:15 (Duration: 00:07:54)
- sta\_all\_cu... refreshed successfully
- Business\_D... refreshed successfully

### 2.1 Business needs to add a new column to model.

The screenshot shows the SAP Modeler interface. The top navigation bar includes "Files / All Items New AR Model", "Model" (selected), and "Data Management". The main area displays a list of columns on the left and a data preview table on the right. A tooltip "Add new Dimension" is shown over a button in the toolbar, which is highlighted with a red box. The data preview table shows the following rows:

Account	clearing_date	posting_date	
amount_document	2020-05-29	2020-05-29	2020
amount_document	2021-02-24	2021-02-24	2021
amount_document	2021-02-25	2021-02-25	2021

### 2.2 Run the SCHEMA pipeline to print only the first 500 rows of the model

The screenshot shows the Microsoft Data Factory interface. On the left, under 'Factory Resources', there is a tree view of pipelines:

- Pipelines (9 items):
  - PIP\_TEST
  - O2C (4 items):
    - PIP\_O2C\_INGEST
    - PIP\_O2C\_OUTPUT
    - PIP\_O2C\_OUTPUT\_SCHEMA** (highlighted with a red box)
    - PIP\_O2C\_TRANSFORM
  - P2P (4 items):
    - PIP\_P2P\_INGEST



2.3 Business will import the data and do a new mapping for the changed import file

2.4 Run the OUTPUT pipeline to print the entire data model to the SAC Fileserver

2.5 Business now imports the entire data and double checks the change

++ Business can do the change themselves  
++ Has full functionality of SAC later on

-- More complexity in the model  
-- Business has to maintain data manually

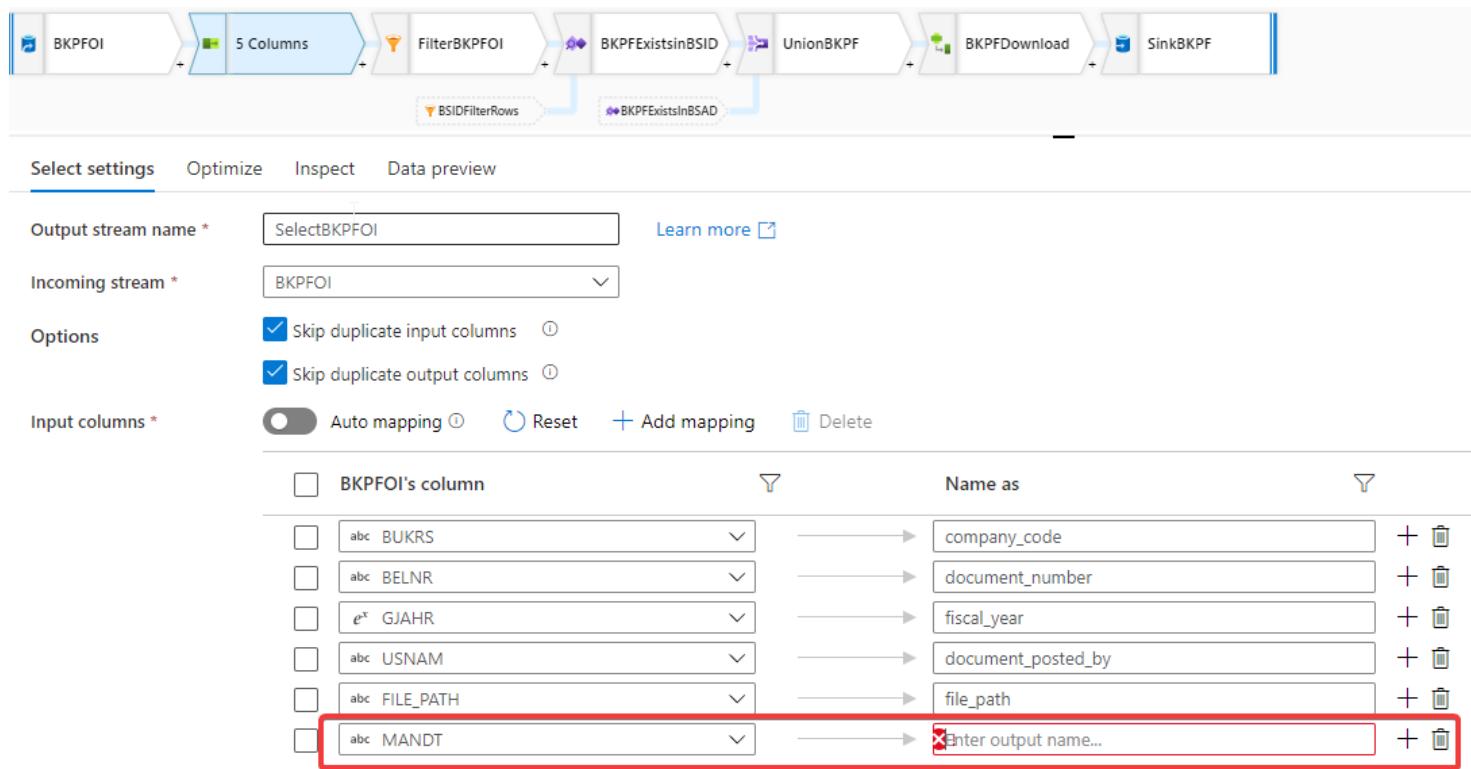
### 3. Add the column into SQL Server

I would always recommend doing option 1 or 2 first as a blueprint until the business case is clear and then automate on the SQL server afterward, avoiding much friction between IT and Business.

3.1 check with business where the information is stored and then start from there.

If the information is not on data lake yet, we need to request it from HQ first and schedule it to be loaded.

3.2 After this is done or if it is already available, the ingest pipeline must be adjusted e.g.



Here we are adding a new column from the BKPF table to be downloaded to SQL Server

### 3.3 Add a new column in the SQL tables of D-SQL-Server

Some tables are generated automatically here. We don't need to add additional columns. Some tables need to be created statically, like the STA\_\* tables and indexed tables. Here we need to add the coding.

Best way is to run the pipeline, see where it is interrupted, and then add the coding. For STA\_\* tables, we always need to add the new column manually as this will change the output to SAP and requires new mapping by business.

### 3.4 Business needs to add a new column to model.

Account	clearing_date	posting_date	
amount_document	2020-05-29	2020-05-29	2020
amount_document	2021-02-24	2021-02-24	2021
amount_document	2021-02-25	2021-02-25	2021

### 3.5 Run the SCHEMA pipeline to print only the first 500 rows of the model

3.6 Business will import the data and do a new mapping for the changed import file

3.7 Run the OUTPUT pipeline to print the whole data model to the SAC Fileserver

3.8 Business now imports the entire data and double checks the change

3.9 Transfer the coding to P-System. Check the chapters for transports above.

-- Fully Automated

-- Can fulfill all requirements

-- IT effort

## Add new BW cube to Solution Space

As described previously, it's advisable to ask business to first do a check by downloading the BW report to Excel and build a dashboard to understand their requirements by themselves. (type1 change)

Once the requirements are clarified, we can continue with the initiation of the type2 change.

In that case, a Denodo base view has to be created by HQ. This can be done by contacting Mandeep Kaur.

## Issues

**We (Business) want to add a new calculated dimension, but in SAC, we get the following error**

Story Data

Current Month Oct, 2020 Supplier Number (All) Accountant User Name (All)

DASHBOARD IC AP ANALYSIS IC AP EDI% ANALYSIS IC AP Overd... DASHBOARD IC AP

% IC EDI, All posting invoices_IC and others for Actual	% of Overdue value for Actual	% of
<b>1 Filter   1 Variance</b>  Unable to retrieve data from the data source.  Error: [Caught exception : exception 42556: Extended dimension processing error : Can't read view '26710169-7611-4589-8972-142769040036' ]	<b>1 Filter   1 Variance</b>  <b>1.87%</b> (+0.52)  14,799 ABS IC AP Overdue amount in TEUR 790,607 ABS IC AP amount in TEUR  fx EUR-CNY 7.83	<b>1 Filter   1 Variance</b>  <b>1.</b>  520 Cour Over  44,04 Cour

SAC seems to have problems if the calculated dimensions are created on dimensions with many different member values. e.g., this error will appear when creating calculated dimensions on the text column as every row has a different value in the field. It will most likely not appear for a calculated dimension on the company code as there are only ten different members.

There are two options to solve this.

1. Add the calculation into the model

## Mapping Summary

Mapping Target	Attribute	Mapped Column
Account	Account	-
OCR_supply_1y1s4g01j3	Dimension ID	OCR_supply_1y1s4g01j3
OCR_invoice_correction	Dimension ID	OCR_invoice_correction
Transaction_key	Dimension ID	Transaction_key
Activity_status	Dimension ID	Activity_status
	Description	Activity_status_Description
Legal_Entity	Dimension ID	Legal_Entity
	Description	Legal_Entity_Description
MMdoc_no	Dimension ID	MMdoc_no
Reference	Dimension ID	Reference
Invoice_state	Dimension ID	Invoice_state
	Description	Invoice_state_Description
Invoice_input_channel	Dimension ID	Invoice_input_channel
Reconciliat_1k164s05l2	Dimension ID	Reconciliat_1k164s05l2
Calendar_year	Dimension ID	Calendar_year
ctry_of_origin_REB	Dimension ID	ctry_of_origin_REB

## Action Log

Set formula of [GR/IR Status] to  
 "IF(ISNULL([Goods\_Receipt]), "SAP no GR",  
 IF([Goods\_Receipt]-[Invoice\_Receipt]>0,  
 "GR>IR", IF([Goods\_Receipt]-  
 [Invoice\_Receipt]=0, "GR=IR",  
 IF([Goods\_Receipt]-[Invoice\_Receipt]<0,  
 "GR<IR", "Other" ))))""

## Combine

STA\_IRB\_FULL\_1616995768802\_QueryKMU  
 5J4DE5PQDC with  
 T024\_1616997472918\_QueryKMU6JN9YBG  
 XSH

1.1 Business needs to add a new column to the model.

1.2 Run the SCHEMA pipeline to print only the first 500 rows of the model

1.3 Business will import the data and add the new calculation

1.4 Run the OUTPUT pipeline to print the entire data model to the SAC Fileserver

1.5 Business now imports the entire data and double checks the change

++ Change can be done by business themselves

-- Increases complexity of the model

2. Add the calculation into the SQL Server

2.1 add the calculation in the TP3\_\* stage on the SQL server. There are already many examples:

```
update o2c.tp3_all_cust_items set vat_issued = 'VAT issued' where left(REFERENCE,3) = 'INV'
```

2.2 Add the new column to the STA\_\* tables needed

2.3 Follow steps 1.1 - 1.5 from above

++ can fulfill all business requirements

-- IT effort

Again, it makes sense to do option one first and then move all the additional calculations to the SQL server once there are a couple of them accumulated in the SAC model (type1 change first)

## Business thinks that the data is wrong

At the start of the implementation phase, this seemed almost always to be correct. With the great help of our key users, we were able to find many bugs during the testing phase.

At the end of the project, however, the error was not in the solution in almost all cases but in the Excel comparison.

In case business reports that the data is wrong, we can have a quick check as follows:

1. Ask for a sample set with rows that should be included but are not / or excluded but should be included.
2. Traceback into STA / TP3 / TP2 / TP1 / CLN / ING tables to find the root cause.
3. If a calculation like 'arrears after net' is wrong, check directly in the EP1 system.