Celonis AP (Accounts Payable) Case Study

Welcome to the Celonis Accounts Payable Case Study! Throughout this case study, you will gain hands-on experience creating the data model for Account Payable from start to finish.
You will be using the experience you've obtained throughout the Data Engineer Procure-to-Pay training for this case study.

Tasks to complete during the Case Study:

1. Overview of Accounts Payable
2. Connect to the Source System
3. Set Up Extraction
4. Filter and Execute Extraction
5. Create Activity Table (Transformations)
6. Adding an Activity
7. Set Up the Initial Data Model
8. Extend the Data Model
9. Finalizing the Data Model

Before you begin the Case Study, please make sure you have general understanding of Vertica SQL.

Overview of Accounts Payable

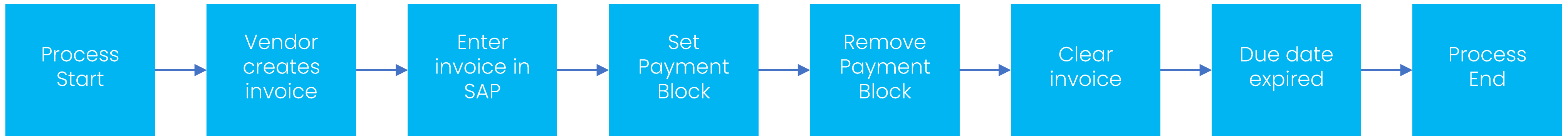
What is Accounts Payable?

Accounts Payable can be defined as a company's invoices that must be paid off within a given period to avoid a default from vendors. These invoices can be for any products or services that have already been delivered, but not immediately paid. Accounts Payable acts as a short-term loan that can typically be paid within 15, 30, or 45 days after the company receives the invoice. Note the following details:

- Accounts Payable can be found under "Current Liabilities" under a company's Balance Sheet.
- Companies will try to lengthen the amount of time to increase the amount of cash, while suppliers prefer a shorter amount of time.
- Suppliers sometimes will sell at a discount if the company can pay within a shorter time frame.
- Companies that tend to exceed the date expired are likely to result in poor relations with the supplier.

Although the Accounts Payable Process may differ in different organizations, we will be using the simplified example below throughout the case study.

Accounts Payable Process Model



Connect to the Source System

Objective of the Task
Before we start building any of the data connections, we must first connect to the source system.

Instructions
Connect to Source System by creating a New Data Pool in the Event Collection tab of Celonis. When setting up the New Data Pool, remember to do the following:

- Name the Data Pool accordingly
- Set up a New Data Connection to a Database with the credentials below

Credentials
Name: Data Scientist AP Case Study
Connection type: Direct
Type: Microsoft SQL Server (native)
Host: 3.120.99.40
Port: 1433
Database Name: celonis
Schema Name: CASESTUDY
Username: Training_CelonisDataEngineer
Password: Celonis123!

Set Up Data Extraction

Objective of the Task

When we work in Celonis IBC, we must extract the data from the source system. To do so, a data job must be initiated to extract the necessary data.

Instructions

After you've created a New Data Pool in the Event Collection tab of Celonis, you need to create a New Data Job to execute the data extraction. The following must be done to successfully complete the extraction:

- Name the Data Job accordingly (Accounts Payable)
- A New Extraction must be created and named accordingly within the Data Job (Extract AP Raw Data)
- Add all the Accounts Payable tables to the extraction

Continue to the next slide to properly set up the following for your extraction:

- Set any filters you may need prior to the extraction.
- Joining tables prior to the extraction.

Filter and Execute Extraction

Objective of the Task

We filter certain columns and rows in Celonis because not all the data from the source system is useful. Filtering will allow you to extract and use specific data, rather than include unnecessary or sensitive data.

Instructions

Let's extract only the invoices that were created after October 1st, 1994.

Filter and join the Data

BKPF: Filter on CPUDT(Creation Date) for 10/01/1994 12:00 AM until Now
BSEG: *Join to BKPF on the following*

- MANDT
- BUKRS
- BELNR
- GJAHR

And filter on joined table BKPF.CPUDT >= '01.10.1994'

**We must join the tables because there is no date column that can be filtered from the table BSEG.*

Once you have properly configured the Data Job, make sure to toggle **Reload all data** before **executing the Data Job**. Verify that there are no conflicts with the data load and proceed to creating the Activity Table.

Creating the Activity Table (Transformations)

Objective of the Task

We will now create the **Activity Table (New Transformation)** so that we can fill in our activities throughout the case study. The activities are based on what was provided in the process model (see slide 2).

Remember to name the Activity Table accordingly

_CEL_AP_ACTIVITIES

Make sure the Activity Table contains *at least* the following columns

CASE_KEY, ACTIVITY, EVENTTIME

Optional fields in the table

SORTING

Provided SQL Script to Create a new Activity Table

```
CREATE TABLE _CEL_AP_ACTIVITIES(
          "_CASE_KEY" VARCHAR (50),
          "ACTIVITY" VARCHAR (50),
          "_EVENTTIME" DATETIME,
          "_SORTING" INT
);
```

Once you have executed the SQL code and created the Activity Table, start adding activities onto the table with the information on the next few slides.

Note: We recommend creating a New Transformation for each new activity to easily organize your activity table. Remember to highlight the SQL code before selecting the Execute button.

Adding an Activity - Overview

Objective of the Task

Think about the following questions when creating the activities on the following slides:

Where can I locate the case key?
This can be found under 'Creating a Case Key'

Where can the timestamps be found?
This can be found under 'Creating the Event Time'

What sorting does the activity have?
This can be found under 'Creating the Sorting'

What tables are needed? (FROM)
This can be found under 'SAP Tables needed'

How are these tables connected? (JOIN)
You will have to use your knowledge of the tables to connect them.

Are there any restrictions needed? (WHERE)
This can be found under 'Table filters needed'

Adding an Activity – Vendor Creates Invoice

Objective of the Task

We will now be adding activities onto the activity table following the next few slides.

Activity Name (ACTIVITY_EN)	'Vendor creates invoice'
Timestamp	BKPF.BLDAT
Activity Description	Date which can be found on the invoice document

TASK	ACTIVITY TABLE COLUMN	ACTIVITY TABLE VALUE
Creating the Activity Name	ACTIVITY	Vendor creates invoice
Creating the Case Key	_CASE_KEY	Concatenate the following: • BSEG.MANDT • BSEG.BUKRS • BSEG.BELNR • BSEG.GJAHR • BSEG.BUZEI
Creating the Event Time	_EVENTTIME	BKPF.BLDAT
Creating the Sorting	_SORTING	0

TABLE FILTERS NEEDED

- BSEG.BSCHL = '31'

BSCHL (Posting Key) 31 refers to vendors invoices

- AND BKPF.BLDAT IS NOT NULL

Excludes rows that are missing a document date.

SAP TABLES NEEDED

http://leanx.eu/en/sap/table/search

- BSEG
- BKPF

Columns Affected/Result	85,951

Adding an Activity – Vendor Creates Invoice (SQL Code Example)

Explanation

Provided is the SQL Script for the first activity! Comments in orange will recap on what the SQL code does. You will be responsible for adding rest of the activities into the activity table.

SQL Script for Vendor Creates Invoice

INSERT INTO _CEL_AP_ACTIVITIES(“_CASE_KEY”, “_ACTIVITY”, “_EVENTTIME”, “_SORTING”) *Inserting the new activities into the activity table*
SELECT DISTINCT
  “BSEG”.“MANDT” || “BSEG”.”BUKRS” || “BSEG”.”BELNR” || “BSEG”.”GJAHR” || “BSEG”.”BUZEI” AS “_CASE_KEY”, *Concatenation of the different values to set them as the Case Key*
  ‘Vendor creates invoice’ AS “_ACTIVITY”, *Activities will be marked as “Vendor Creates Invoice”*
  “BKPF”.”BLDAT” AS “_EVENTTIME”, *Set as the Event Time*
  0 AS “_SORTING” *Because the data only contains event times at the day level, we want to make sure this activity is selected as the first to occur.*
FROM “BSEG”
JOIN “BKPF” ON *We have to join on the two tables with the following foreign keys listed below*
“BSEG”.”MANDT”=“BKPF”.”MANDT”
  AND “BSEG”.”BUKRS” = “BKPF”.”MANDT”
  AND “BSEG”.”BELNR” = “BKPF”.”BELNR”
  AND “BSEG”.”GJAHR” = “BKPF”.”GJAHR”
WHERE BSEG.BSCHL = ’31’
  AND BKPF.BLDAT IS NOT NULL; *Applying the filter using the Where statement*

Optional Step:
After creating the first activity, you can go set up the Data Model and create an Analysis with a Process Explorer to verify that the activity table was successfully created. See the slide “Set up the Data Model” for more details.

Adding an Activity – Enter Invoice in SAP

Objective of the Task

We will now be adding activities onto the activity table following the next few slides.

Activity Name (ACTIVITY_EN)	'Enter invoice in SAP'
Timestamp	BKPF.CPUDT, BKPF.CPUTM
Activity Description	Timestamp on which the document was entered into the SAP system

TASK	ACTIVITY TABLE COLUMN	ACTIVITY TABLE VALUE
Creating the Activity Name	ACTIVITY	Enter invoice in SAP
Creating the Case Key	_CASE_KEY	Concatenate the following: • BSEG.MANDT • BSEG.BUKRS • BSEG.BELNR • BSEG.GJAHR • BSEG.BUZEI
Creating the Event Time	_EVENTTIME	BKPF.CPUDT, BKPF.CPUTM
Creating the Sorting	_SORTING	10

TABLE FILTERS NEEDED

- BSEG.BSCHL = '31'
- AND BKPF.CPUDT IS NOT NULL

Excludes rows that are missing the account document date

- AND BKPF.CPUTM IS NOT NULL

Excludes rows that are missing the posting date

SAP TABLES NEEDED
http://leanx.eu/en/sap/table/search

- BSEG
- BKPF

Columns Affected/Result	85,950

Adding an Activity – Set or Remove Payment Block

Note:
Unlike the previous two activities you've created, this is slightly more complicated.
Keep in mind that the SQL Script you create will generate two different activities.

Activity Name (ACTIVITY_EN)	Use a CASE WHEN statement: If CDPOS.VALUE_NEW is empty, set 'Remove payment block'. If CDPOS.VALUE_OLD is empty, set 'Set payment block'
Timestamp	CDHDR.UDATE, CDHDR.UTIME
Activity Description	Timestamp when a payment block has been entered into an invoice Timestamp when a payment block has been removed from an invoice

TASK	ACTIVITY TABLE COLUMN	ACTIVITY TABLE VALUE
Creating the Activity Name	ACTIVITY	Remove payment block Set payment block
Creating the Case Key	_CASE_KEY	Concatenate the following: • BSEG.MANDT • BSEG.BUKRS • BSEG.BELNR • BSEG.GJAHR • BSEG.BUZEI
Creating the Event Time	_EVENTTIME	CDHDR.UDATE, CDHDR.UTIME
Creating the Sorting	_SORTING	Use a CASE WHEN statement: For 'Set payment block', set sorting to 20. For 'Remove payment block', set sorting to 30.

TABLE FILTERS NEEDED

- CDPOS.VALUE_NEW IS NULL OR CDPOS.VALUE_OLD IS NULL

Excludes rows that are missing a new or old changed field

- AND BSEG.BSCHL = '31'

- AND CDPOS.FNAME = 'ZLSPR'

Only includes rows that have a payment block

SAP TABLES NEEDED
http://leanx.eu/en/sap/table/search

- BSEG
- BKPF
- CDPOS
- CDHDR

Columns Affected/Result (Remove Payment Block)	1,639
Columns Affected/Result (Set Payment Block)	14

Adding an Activity – Clear Invoice

Activity Name (ACTIVITY_EN)	'Clear invoice'
Timestamp	BSEG.AUGDT
Activity Description	Date on which the invoice has been cleared (paid)

TASK	ACTIVITY TABLE COLUMN	ACTIVITY TABLE VALUE
Creating the Activity Name	ACTIVITY	Clear invoice
Creating the Case Key	_CASE_KEY	Concatenate the following: • BSEG.MANDT • BSEG.BUKRS • BSEG.BELNR • BSEG.GJAHR • BSEG.BUZEI
Creating the Event Time	_EVENTTIME	BSEG.AUGDT
Creating the Sorting	_SORTING	40

TABLE FILTERS NEEDED

- BSEG.BSCHL = '31
- BSEG.AUGDT IS NOT NULL

Excludes rows that are missing a clearing date

SAP TABLES NEEDED
http://leanx.eu/en/sap/table/search

- BSEG
- BKPF

Columns Affected/Result	59,865

Adding an Activity – Due Date Expired

Activity Name (ACTIVITY_EN)	'Due date expired'
Timestamp	BSEG.ZFBDT + (BSEG.ZBD1T or BSEG. ZBD2T or BSEG. ZBD3T)
Activity Description	Last date on which the invoice should be/have been cleared

TASK	ACTIVITY TABLE COLUMN	ACTIVITY TABLE VALUE
Creating the Activity Name	ACTIVITY	Vendor creates invoice
Creating the Case Key	_CASE_KEY	Concatenate the following: • BSEG.MANDT • BSEG.BUKRS • BSEG.BELNR • BSEG.GJAHR • BSEG.BUZEI
Creating the Event Time	_EVENTTIME	Use a CASE WHEN statement: If BSEG.ZBD3T >0, add BSEG.ZBD3T to the base date BSEG.ZFBDT. Else, if BSEG.ZBD2T >0, add BSEG.ZBD2T to the base date BSEG.ZFBDT. Else, if BSEG.ZBD1T >0, add BSEG.ZBD1T to the base date BSEG.ZFBDT. If both BSEG.ZBD1T, BSEG.ZBD2T and BSEG.ZBD3T are 0, use BSEG.ZFBDT.
Creating the Sorting	_SORTING	50

TABLE FILTERS NEEDED

- BSEG.BSCHL = '31'
- AND BSEG.ZFBDT IS NOT NULL

Excludes rows that are missing a due date.

SAP TABLES NEEDED
http://leanx.eu/en/sap/table/search

- BSEG
- BKPF

Columns Affected/Result	85,914

Set up the Initial Data Model

Objective of the Task

After creating and adding into the activity table, it's time to set up the initial **Data Model**.
This can be done in the Process Data Model's tab under the Event Collection

Instructions

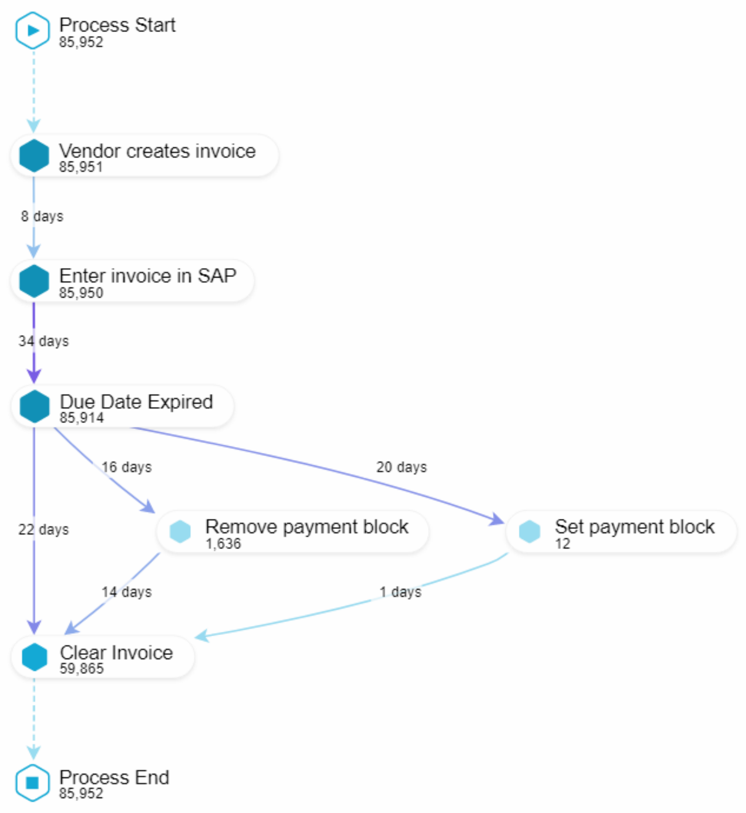
Create a New Data Model and name it "Accounts Payable". Once it's been created, add the activity tables you created on the previous exercise onto the list and select the Celonis Activity Table (_CEL_AP_ACTIVITIES). Make sure to properly configure the following columns for the activity table:

- Case ID
- Activity
- Timestamp
- Sorting
- End Timestamp (Optional – You may skip this step!)

We will revisit adding additional tables and connections in the next slide
Force a complete reload of the activity table and make sure no critical errors show.
You may receive a warning error stating missing fields from the case table.

Verifying your Activity Model

Create a new Analysis with a Process Explorer. Add more activities and verify that your model looks like the one on the right.



Extend the Data Model

Objective of the Task

Just having the activity table is not enough for your Analyst to create appealing analyses! We will be adding 3 additional tables onto the Data Model.

Remember that you will have to create a new transformation for every additional table you would like to add.

The following tables should be created:

BSEG, BKPF, LFA1

Notes:

- When creating each of these tables, remember to make them AP_'TABLENAME'
- Make sure to use SELECT DISTINCT from the tables
- Filter should be BSEG.BSCHL = '31'

Example SQL Script for BSEG Table

```
CREATE TABLE AP_BSEG AS (
SELECT DISTINCT
          “BSEG”.*
          , “BSEG”.“MANDT” || “BSEG”.”BUKRS” || “BSEG”.”BELNR” || “BSEG”.”GJAHR” || “BSEG”.”BUZEI” AS “_CASE_KEY”
FROM “BSEG”
JOIN “BKPF” ON
          “BSEG”.”MANDT” = “BKPF”.”MANDT”
          AND “BSEG”.”BUKRS” = “BKPF”.”BUKRS”
          AND “BSEG”.”BELNR” = “BKPF”.”BELNR”
          AND “BSEG”.”GJAHR” = “BKPF”.”GJAHR”
WHERE “BSEG”.”BSCHL” = ’31’
)
```

You can verify the table by using SELECT COUNT(*) FROM AP_BSEG

Finalizing Data Model

Objective of the Task

Now that you've created the 3 additional tables, we have to define the relationship between the tables in the **Data Model**. Remember to connect the tables by the corresponding foreign key.

BSEG to BKPF:

Connect MANDT, BUKRS, BELNR, GJAHR

BSEG to LFA1:

Connect MANDT, LIFNR

BSEG to _CELONIS_AP_ACTIVITY:

Connect _CASE_KEY

Why did we have to create the AP_BSEG instead of using the imported BSEG?

We created this view "AP_BSEG" because we needed an activity column to connect with the Celonis Activity Table

Setting Up Aliases

Once all the tables have been properly connected, remember to create the proper alias for each of the AP tables:

- AP_BSEG -> BSEG
- AP_BKPF -> BKPF
- AP_LFA1 -> LFA1

Don't forget to force a complete reload!

Finalizing Data Model - Continued

Objective of the Task

The last step to this case study is to apply name mapping to your data so that your analyst will understand the information they are working with.

What views are needed for name mapping?

Table Name Mapping	DD02T
Technical names column	TABNAME
Pretty names column	DDTEXT
Language key column	DDLANGUAGE

Column Name Mapping	DD03M
Table names column	TABNAME
Technical names column	FIELDNAME
Pretty names column	SCRTEXT_M
Language key column	DDLANGUAGE

Create your analysis

Finally, return to your previously created analysis to make sure that the model you created makes sense!

Finishing up the Case Study

Objective of the Task

Provided are some review questions you can answer to make sure that your Analyst has the details they would need to create complex visuals!

1. Determine the % and number of cases where the process ends with “Due date expired”.
2. How many invoice items (cases) have been booked in per fiscal year (BKPF.GJAHR)? What is the year with the highest number of invoice items?
3. Calculate the percentage of cases that have not been cleared yet.
4. Select the first two variants on the Variant Explorer. What do you notice changes on the process flow?
5. What is the average throughput time (in days) between “Enter Invoice in SAP” and “Due Date Expired”?