LB Dev

ALB APIs

ALB Architecture

Architecture files

File Name	Presentation
FY22 Q4 - Architecture Impact Analysis for ALB DataLab - IT11398 - Solvay Digital Labs	
FY22 Q3 - Architecture Impact Analysis for ALB - IT11398	

ALB Data Architecture - Schema/Models

This section aims to present the Data Architecture implemented for data flow in Lab Booster.

ELN Schemas

File Name	Data Model File
Agro	
Battery	
Coatings	
Seed Care	
Actizone	
HPC Flocculation	

ELN Spreadsheets design standards

Design is really important for user experience.

The user feedback about the first version of the ELN templates where more about the design than the content. They didn't really enjoy the "black and yellow" spreadsheets.

So for Coatings' Paint Formulation SS V2, we worked on defining design standards using Solvay's color palette as a base.

These standards are going to evolve according to the future needs.

Here are documentations about developed spreadsheet in PROD:

Agro		
Battery	Conductivity	
Battery	Mechanosynthesis	
Coatings	EP	
Seed Care	Formulation	
Seed Care	Results & Requests	

Lab Booster Data model

Overview

A **Data Model** represents the way data is structured in a dataset or a database, such as Lab Booster's data ocean.

The data model defines how the data lake or data ocean is connected to:

- The data input i.e. ELN, LIMS systems, connected instruments etc.

- The data output i.e. the WebApp DataLab in which users can access data

Context

As of mid-2023, each market in Lab Booster has its own data model i.e. its own way to structure data.

At each new project, connections to the data lake must be built again

Objective

Our aim is to have a common data model for all markets, to bring:

- Accelerated delivery of new projects
- Better performance
- Less maintenance

This page is divided two sections

1. Entity-Relationship Diagram (ERD), which served as a basis to design the data model
2. Data model

Entity-Relationship Diagram (ERD)

Data Models are generally based on a diagram or schema called **Entity-Relationship Diagram** defining

- Entities i.e. a definable object or concept within a system

- Relationships i.e. how entities are related to one another

Building the ERD is a preliminary step to designing the actual data model to ensure that all required entities and relationships are accurately defined and represented.

This section is split in two parts

1. Entity-Relationship Diagram design
2. ERD mapping with R&I workflows

Entity-Relationship Diagram design

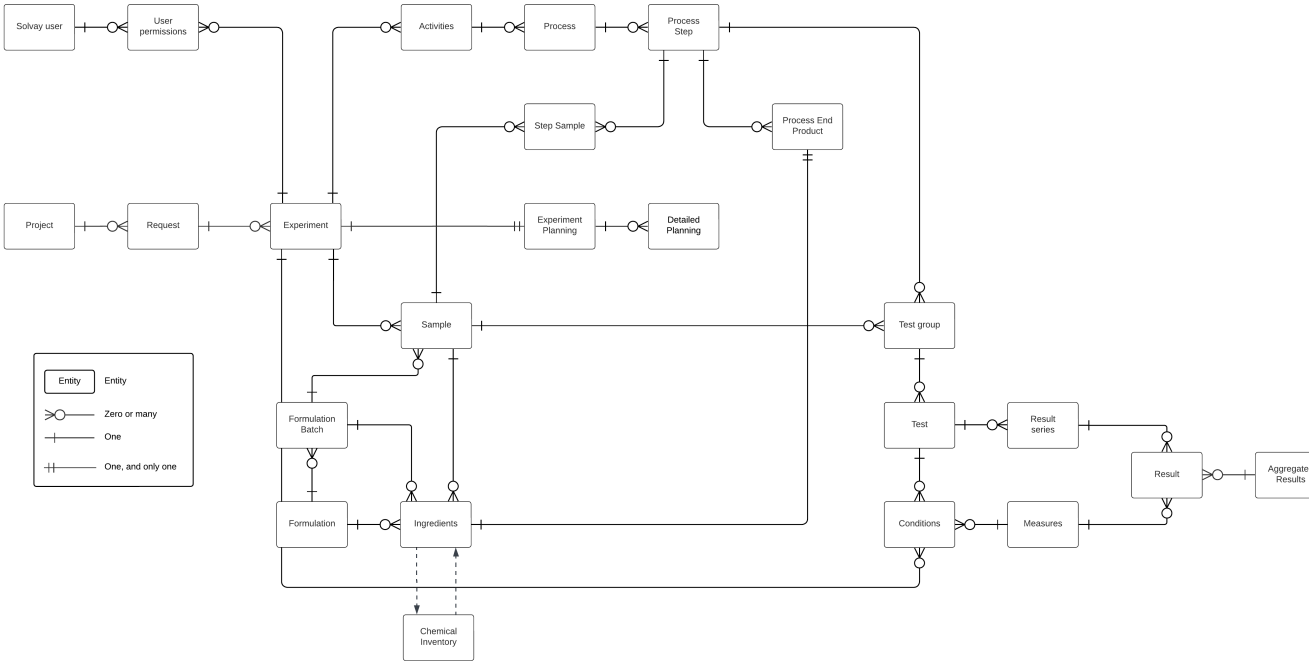
Entity dictionary

Entity	Definition	Example(s)
Experiment	A recording of a workflow performed in the lab by an operator at a given date to achieve an objective An ***Experiment*** includes • *Activities* • *Samples* • *Tests* • *Request* • *Planning*	***Experiments*** created and recorded in ELN IDBS, LIMS Labware LIMS Agilab...
Solvay User	A recording of the user that created the Experiment, including Solvay ID and email	
User Permissions	A setting determining what application options the user has access to	
Request	A recording of information provided by user requesting an *Experiment* A ***Request*** includes • ***Request*** date • *Sample* information • Information on user making requests	***Requests*** for BioMatTech - Biodegradability testing include • Request name • Requestor name • Request date • Priority • Status • Test method required • Sample name • Sample ID • Sample status • etc.
Planning	A recording of when the *Experiment* is supposed to be performed A ***Planning*** includes • *Tests* or *Activities* expected date • *Results* availability date	***Planning*** in Novecare - Méréville Request & Results includes • Expected application (of slurries & powders on seed) date • Operator performing the application
Activity	A group of *Process*es performed in the lab in a specific order	In Novecare - Méréville Request & Results, two ***Activities*** are found, Application and Testing
Process	A group of *Process Steps* performed in the lab in a specific order	In BatMat -Mecanosynthesis, the Mecanosynthesis ***Process*** is defined by several successive *Process Steps* 1. Jar Preparation 2. Milling 3. Drying 4. Calcination 5. Finishing

Process Step	A recording of tasks performed in the lab, defined by its name and date A ***Process Step*** includes • *Conditions* in which it is carried out • Input and output *Step Samples* • *Tests* performed during ***Process Step*** • *Process End Product* A ***Process Step*** follows a Standard Operating Procedure (SOP)	In Aroma - Fermentation the Growth **Process Step** is defined by the date on which it is performed and includes • *Conditions* - Scale, Temperature, pH... • Input *Step Samples* - Starter media and Substrate • Output *Step Samples* - Sample #, Date and Time • *Tests* - Optical Density and Glucose analysis • *Process End Product* - Growth media
Process End Product	The chemical output of a *Process*, defined by its name and date ***Process End Product*** characteristics include composition, aspect, mass and/or volume... A ***Process End Product*** can be registered as a new *Ingredient* for other *Formulation (Batch)* or *Process Steps*	In Aroma - Fermentation, the ***Process End Product*** of the *Process Step* "Bioconversion" is vanillin In Novecare - Méréville Formulation Recipe, the ***Process End Product*** of the Formulation *Process Step* is a formulation In BatMat - Mecanosynthesis Jar Slurries, Amorphous Precursors and Raw Calcined Products are ***Process End Products***
Ingredient	A chemical product, defined by its name and unique ID and recorded in an inventory ***Ingredient*** characteristics include date, batch number, supplier, physical state (liquid/solid), density, color... An ***Ingredient*** can be: • A *Formulation Batch* • A *Sample* • A *Process End-Product*	In Aroma - Fermentation, the substrate Ferulic acid is an ***Ingredient*** In Novecare - Méréville Request & Results, Slurries and Powders are ***Ingredients*** In BatMat - Mecanosynthesis Jar Precursors, Slurries, Amorphous Precursors and Raw Calcined Products are ***Ingredients***
Formulation	A combination of chemical products defined by the *Ingredients*, the *Ingredients* target proportions and its name ***Formulation*** characteristics include total number of chemical products, target concentration, target volume, calculated density...	In Novecare - Méréville Request & Results, a Recipe is a ***Formulation*** and is defined by name, ID and label. Characteristics include Number of products, Products, Recipe unit, Recipe Price, Calculated Recipe Density...
Formulation Batch	A combination of chemical products defined by the *Ingredients*, the *Ingredients* actual proportions, its name, unique ID and date ***Formulation Batch*** characteristics include total number of chemical products, actual concentration, total volume, density, container (vessel, jar, bottle)... A ***Formulation Batch*** is a *Formulation* that has been created in the lab	In Novecare - Méréville Request & Results, a Batch of Recipe is a ***Formulation Batch*** and is defined by name, ID and label Characteristics include Recipe selection, Actual Weight (of Products)
Sample	A part of a substance or component that is taken from the whole substance or component, defined by its name, unique ID and date A ***Sample*** can come from • An *Ingredient* • A *Formulation Batch* • A *Process End-Product* • A *Request* A ***Sample*** can be used for • A *Test* • A *Process Step* See *Step Sample* for ***Samples*** taken during a *Process Step*	***Samples*** come from • An *Ingredient* : Inoculum in BioMatTech - Biodegradability • A *Formulation Batch* : Batch of Recipe in Novecare - Méréville Formulation • A *Process End-Product* : Finished Product in BatMat - Mecanosynthesis ***Samples*** are used for • A *Test*: Batch of Recipe to characterize at t0 in Novecare - Méréville Formulation • A *Process Step:* Growth mass used in Bioconversion *Process Step* in Aroma - Fermentation

Step Sample	A part of a substance or component that is taken from the whole substance or component in relation to a *Process Step*, defined by its name and date A ***Step Sample*** can be • An input for the *Process Step* • An output of the *Process Step*	In Aroma - Fermentation, ***Step Samples*** are taken throughout the three *Process Steps* to monitor the chemical reactions
Sample Test Plan	A *Planning* defined for a set of *Samples*, defined by its name and the timing The ***Sample Test Plan*** characteristics include total number of *Samples*, *Tests* to perform ... A ***Sample Test Plan*** can apply in the context of • A *Process Step* • A *Request* • A *Planning*	In Novecare - Méréville Formulation the ***Sample Test Plan*** defines when Samples should be taken during an ageing *Process Step* It is defined by • Protocol name • Initial storage date • Number of *Samples*
Test Group	A group of *Tests* performed on the same *Sample*	Characterization tests (OD manual, OD dencytee and Glucose) performed during the Growth *Process Step* in Aroma - Fermentation for a ***Test Group***
Test	A measure of *Sample* behavior when a procedure is carried out	***Tests*** performed in BatMat - Mecanosynthesis include Particle size test, SEM test, Lumisizer test, H NMR test, P31 NMR test, Li7 NMR test, Discrete value test
Measure	A property that can be measured A ***Measure*** can serve both a *Condition* and/or a *Result*	pH is a *Condition* in Aroma - Fermentation and a *Result* in BioMatTech - Biodegradability
Conditions	A variable or setting defined by the operator for • A *Test* and affecting its *Result* • A *Process Step*	In BioMatTech - Biodegradability, ***Conditions*** for the Dry matter *Test* include Empty aluminium cup weight In Aroma - Fermentation, ***Conditions*** of the Growth **Process Step** include Scale, Temperature, pH...
Results	The outcome of a *Test* performed on a *Sample* in specified *Conditions* ***Results*** can take the form of • A numerical value • A set of numerical values (i.e. curve) • A non numerical value (i.e. observations)	A pH value is a ***Result*** of a biodegradability *Test* in BioMatTech - Biodegradability A conductivity curve is a ***Result*** of a conductivity *Test* in BatMat - Conductivity Observations are a ***Result*** of a Look after Attrition *Test* in Novecare - Méréville Request & Results
Results Series	A set of *Results*, obtained at different time intervals, for a *Test* performed in the same *Conditions* on the same *Sample*	
Aggregated Result	A *Result* obtained by aggregating *Results* from several *Tests*	In Aroma - Fermentation, the maximum amount of vanilin produced during the Bioconversion *Process Step* is an ***Aggregated Result*** as it aggregates several vanilin concentration measure *Results* In Novecare - Méréville Request & Results, averages calculated from two different *Test Results* are ***Aggregated Results***

Entity-Relationship Diagram



ERD mapping with R&I workflows (WIP)

Three types of R&I workflows were identified

- Formulation workflows
- Synthesis workflows
- Analysis workflows

This was done in order to ensure that the ERD defined accomodates all types of R&I workflows.

The mapping done for different workflows is summarized in the table below.

GBU/F- R&I	Workflow name	Workflow type	Mapping status	Link to mapping	Documentation - Data capture
Novecare GBU	Seed Care Formulation	Formulation	Done	Seed Care mapping	ELN template
Novecare GBU	Seed Care Request & Results	Formulation	Done	Seed Care mapping	ELN template
Battery Platform	Mecanosynthesis	Synthesis	Done	Mecanosynthesis mapping	ELN template
Aroma Performance GBU	Fermentation	Synthesis	Done	Fermentation mapping	ELN spreadsheet mockup
BioMatTech Platform	Biodegradability	Analysis	Done	Biodegradability mapping	LIMS spreadsheet mockup
Specialty Polymers GBU	Aging, Mechanical, Thermal	Analysis	Ongoing		
Specialty Polymers GBU		Synthesis	To do		
Novecare GBU	Agro	Formulation	To do		
Novecare GBU	EP Coatings	Synthesis	To do		
Novecare GBU	Paint Coatings	Formulation	To do		
Corporate R&I	Solvent platform - Solubilization		To do		
Corporate R&I		Analysis	To do		

Green Hydrogen Platform	Conductivity	Analysis	To do		

BigQuery

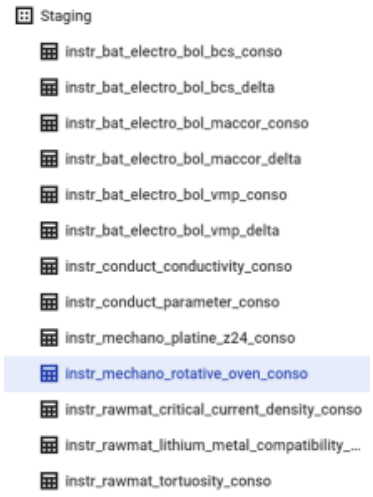
New Data Model of ALB Data Mart (Exposition layer): https://app.genmymodel.com/api/projects/_k07o4IBOEe29ie0vpi-P5A/diagrams/_k07o4oBOEe29ie0vpi-P5A/svg

Data Mapping to Data Mart:

The following BigQuery datasets are all staging as per the data convention explained previously.

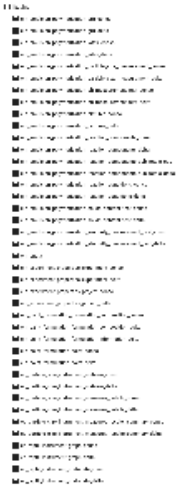
For more ETL (extraction, transformation, loading) details, please refers to: App Lab Booster (ALB) - Data

Batteries



Materials

Coatings



Data mapping

File Name	Data Mapping File
Agro	
Coatings	Version 2:
	Version 1:
Battery	
Materials	
Seed Care	

ALB - Performance Optimization Solutions

Data Compression

Introduction

In order to optimize Tableau' performances, I just tried to apply an algorithm for compressing data to be plotted on charts, in order to suppress "useless" points, and maintain the trends as is!

For example, for an experiment we could have more than 9000 lines in our table named "ODS.raw_data_conso" (~2000 lines for each specimen):

```
1  SELECT Count(*) FROM `prj-labbooster-materials-prod.ODS.raw_data_conso`
2  WHERE test_number = '26888881'
```

Press Alt+F1 for Accessibility Options

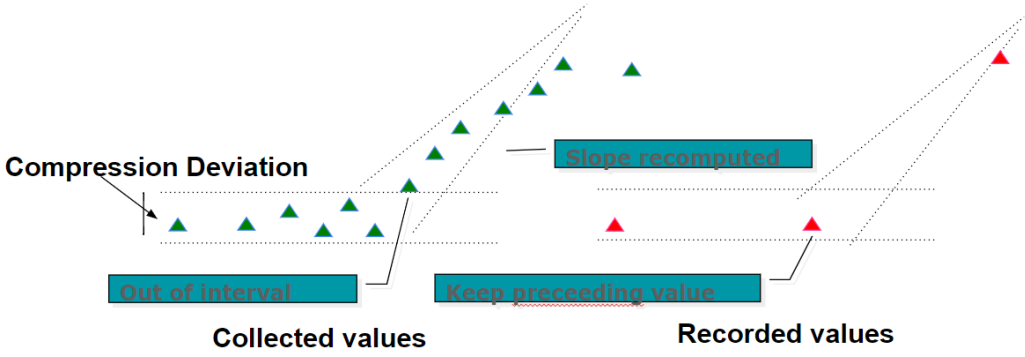
Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	f0_
1	9132

The chart, as plotted on Tableau, is this one (540 values for each trend):

The idea consisted in applying a MES domain' compression algorithm (named Swinging Doors) for minimizing the storage needed on hard disks, and preserving the trend' characteristics!



Normally, this algorithm is applied to only one trend, but with a little modification it's possible to extend it to multiple series at the same time!

Accepting an error of 0.1% (evaluated on the data span of each series), the rows decrease from 540 to... only 36, and the trend is preserved!

In this case the data compression factor is the following

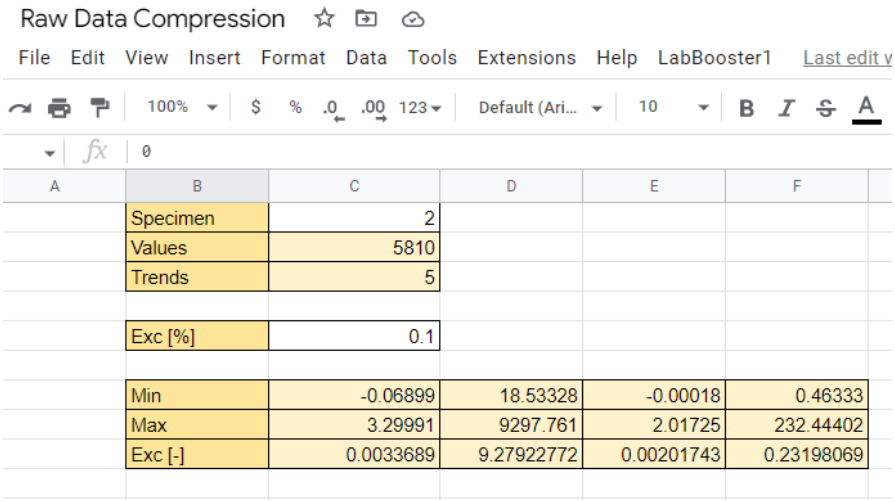
Ratio = 1 - [Final rows] / [Original rows] = 93.33%!

With an error of 0.5%, we could drop to 17 rows, but the trend is not so smooth (Comp Ratio = 96.8%):

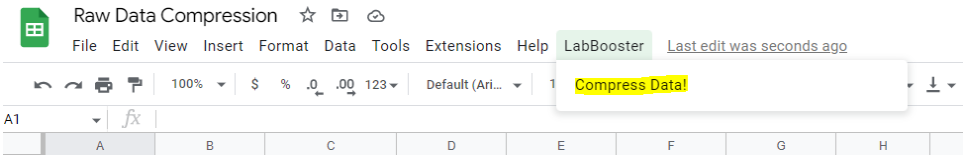
You can find here the test I've made, on Google Sheet:

Raw Data Compression

- on the tab "Original Data", there are the values, as stored in GBQ
- on the tab "Data Table", you can find the "real" table, pivoted by the column "component_name", aggregated by time, and filtered by a specimen (cell C1, white background)
- the maximum error on each variable is evaluated starting from the relative error (cell C5)



- In order to launch the algorithm, it's possible clicking on the custom menu "LabBooster":

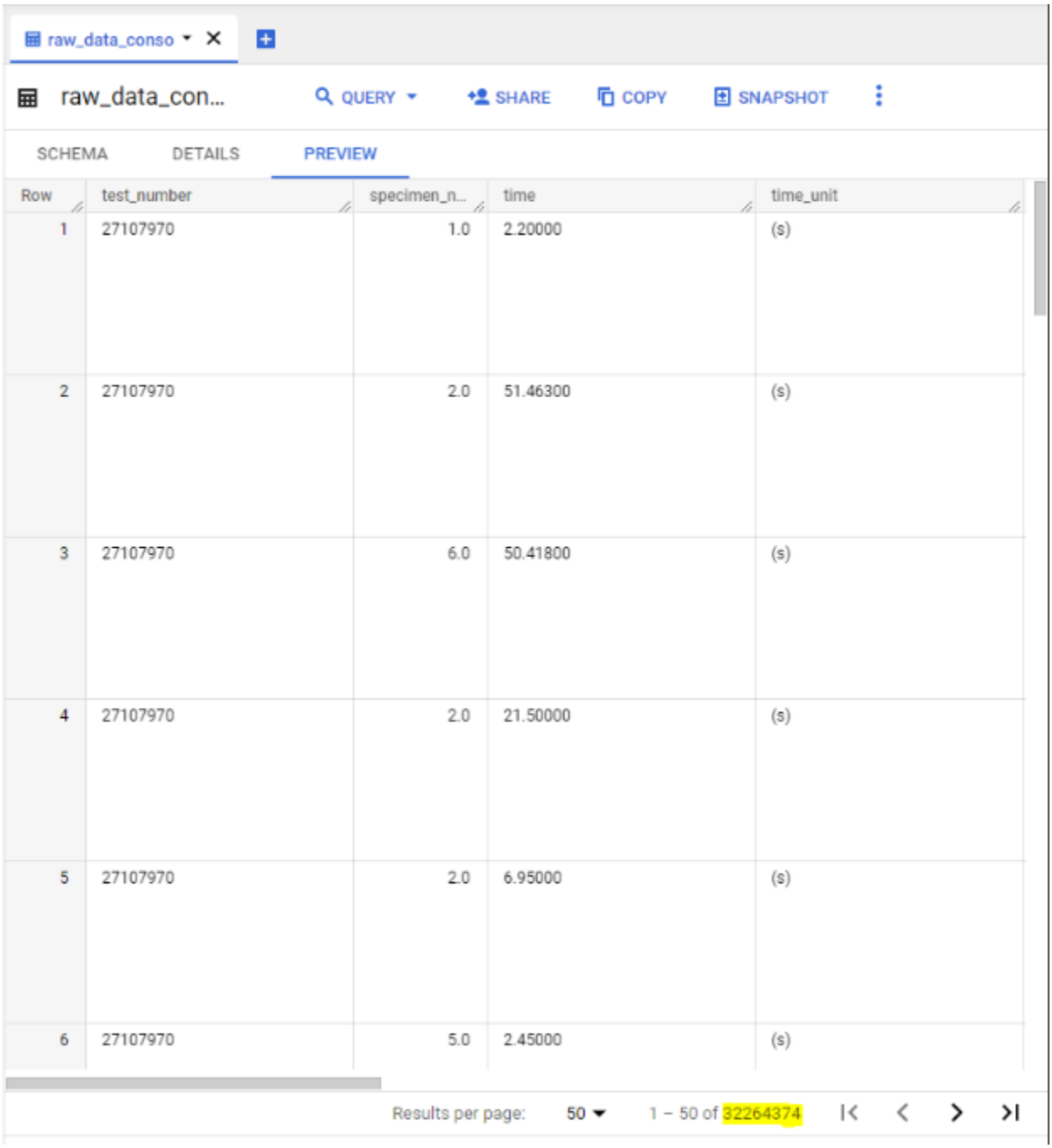


- The results are plotted in the "Results" tab;
- The Compression ratio is evaluated in the "KPIs" tab.

More insights

For example, for Lab Booster - Materials, we have 572 tests within our database:

Its raw data is stored in a single table, that has 32 millions rows!



In order to present this result to Tableau, we're facing some issues, since Tableau has to copy the entire dataset from Google Cloud Platform to the Tableau Server itself, and this process takes almost 2 hours to complete, preventing us to work with the same update frequency of Talend (Talend - refresh scheduled every hour; Tableau - refresh scheduled every 4 hours).

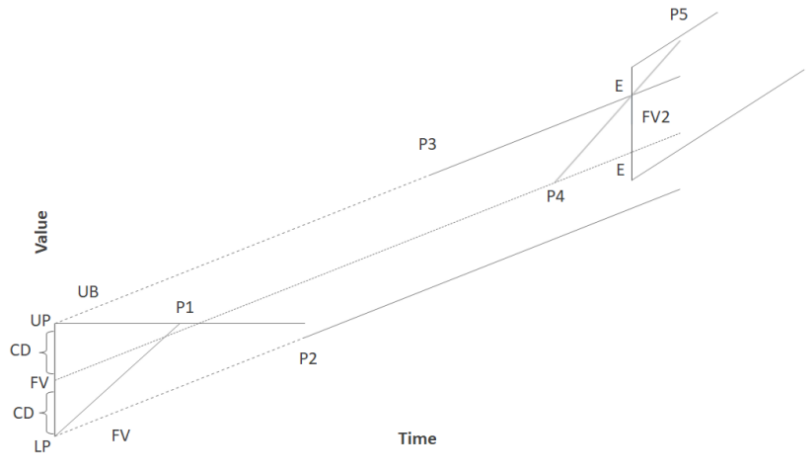
This will be the case also of DSC Thermal analyses, for example: ADC/Labs database is not designed to host very large tables of raw data (as far as I understood from the tech lead of ACD). For a single DCS analysis, we should save the data of three ramps, for having the info of time/temperature /heat flow. In each ramp, the time is recorded every 0.01 s, and we could have several values (I analyzed the 1st heating of a test in Bollate):

To store this data within a database' table, we need 3420 rows (more than 10k for storing the three ramps). So, if we have 100 analyses to store within the database, we need more than 1 million rows.

The compression approach is data analytics-friendly: instead of storing all the data, this approach simplifies the data itself, in order to not store data that is "correlated", with an adaptive sampling step!

- the first step consists in defining a maximum error, named deviation. A reasonable value is 0.1% of the trend span (*i.e.* the difference between the maximum and the minimum of the trend itself)
- At every step, the algorithm evaluates the derivative between the previous and the current point.
- An isosceles parallelogram is defined:

    - its "oblique side" is the value of the Y variable, with the gap of the deviation error (for the current point, which coordinates are (Xc, Yc), it's the segment between Yc-Error and Yc+Error)
    - its lower base is the line that passes from the point (Xc, Yc-Error) and and has an angular coefficient equal to the evaluated derivative
    - its upper base is the line that passes from the point (Xc, Yc+Error) and and has an angular coefficient equal to the evaluated derivative

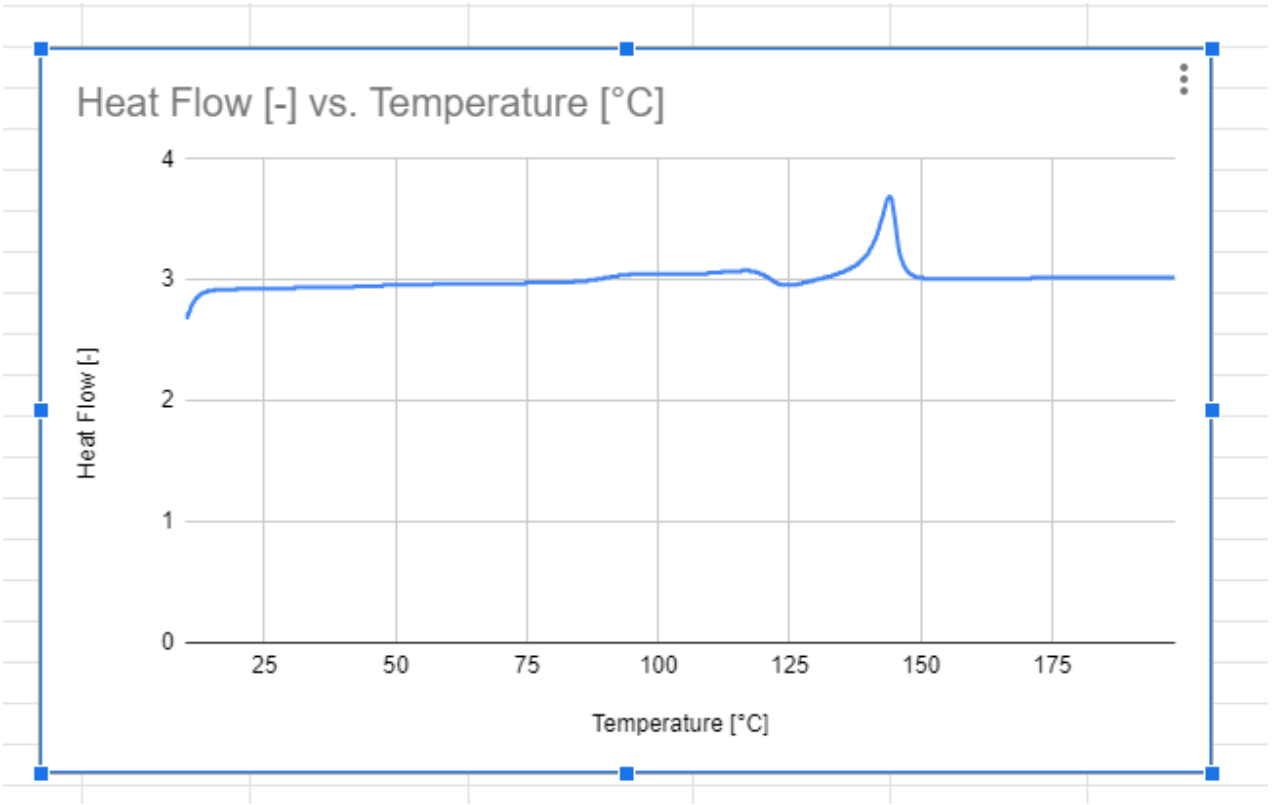


- If the Y of the next point is included within the parallelogram:
    - it means that, if we reconstruct this point through a linear interpolation, we'll commit an error of less than 0.1%
    - the current point is not saved
- In the opposite case, the current point is saved.

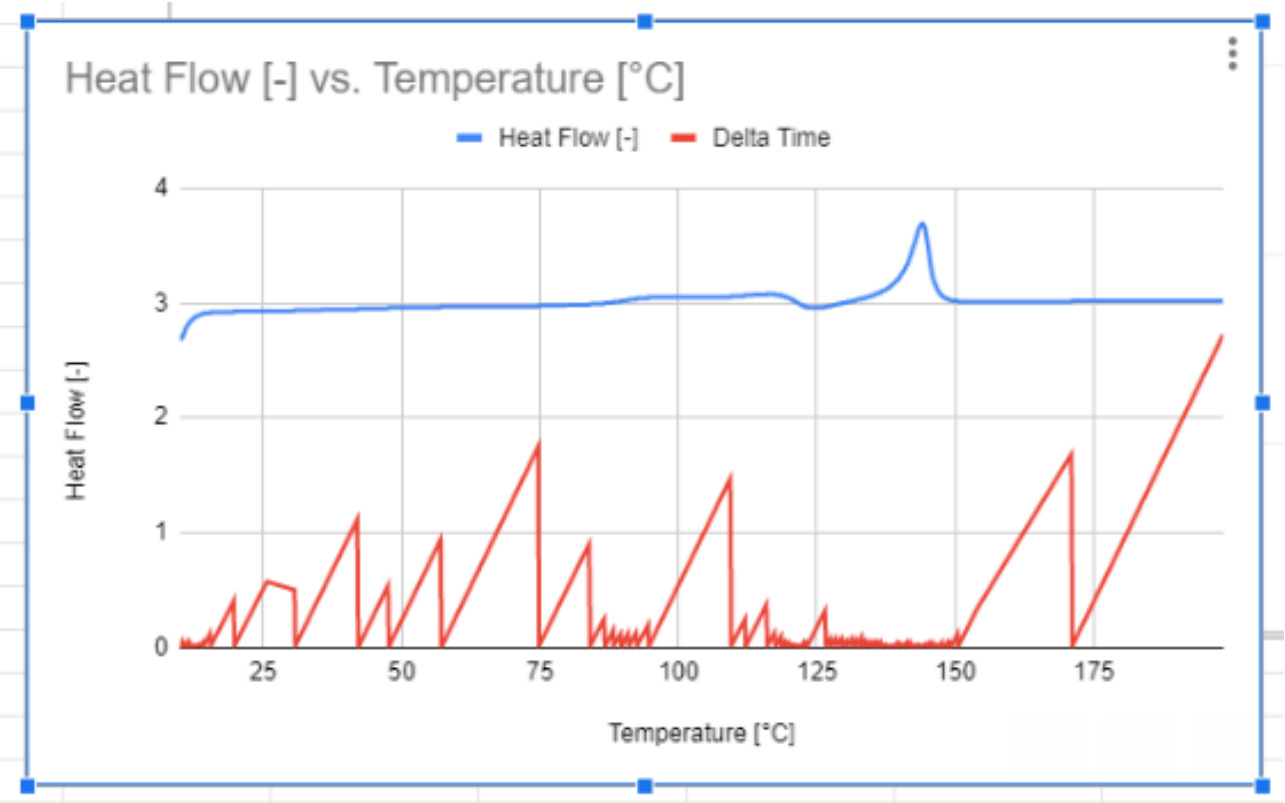
In literature, it was possible to find a compression algorithm that works for only 1 trend, but I modified the algorithm for compressing more trends (*i.e.*, the table columns) simultaneously. The solution consisted in defining a vector of the independend variable's compressed values (*e.g.*, time), and using an interpolation algorithm for retrieveing the values of the dependent variables, at the "compressed" indepenent variables' values.

This approach preserves all the points where there's a significant change in derivative (so, when it's happening a chemical/physical phenomenon), and simplifies the points where the derivative is almost constant. All the maximums and minimums are preserved, and this is the fundamental information for data analytics!

Applying this approach to a DCS analysis, we have this trend for the original values:



and this one for the compressed values (with this approach, we have the maximum definition on the "curve portions", and a "less-dense" point definition when the derivative is constant): the red line shows the time span between the actual compressed value, and the previous one!



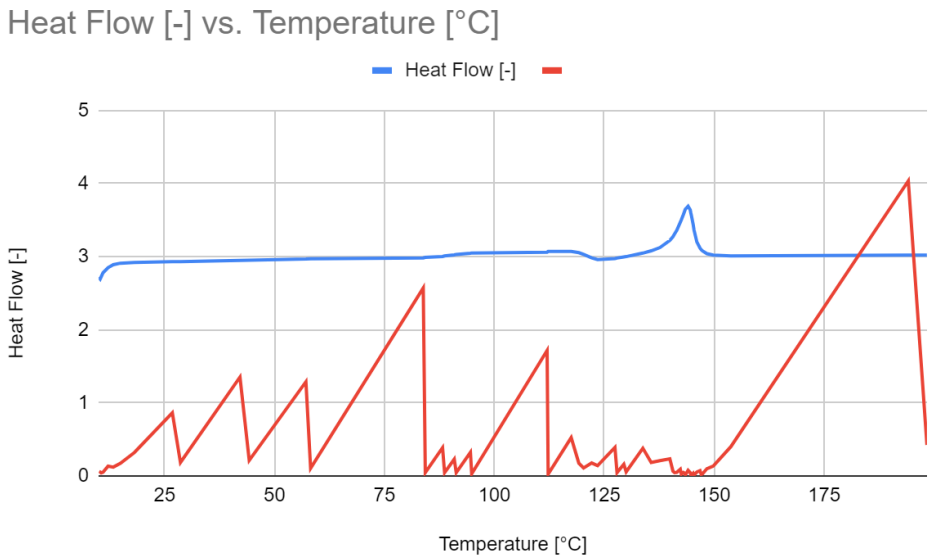
The compressed trend is created with 6 times less the original points, and it preserve all the information for the 1st/2nd grade derivatives:

Original Values	3420
Compressed Values	552
Reduction	83.86%

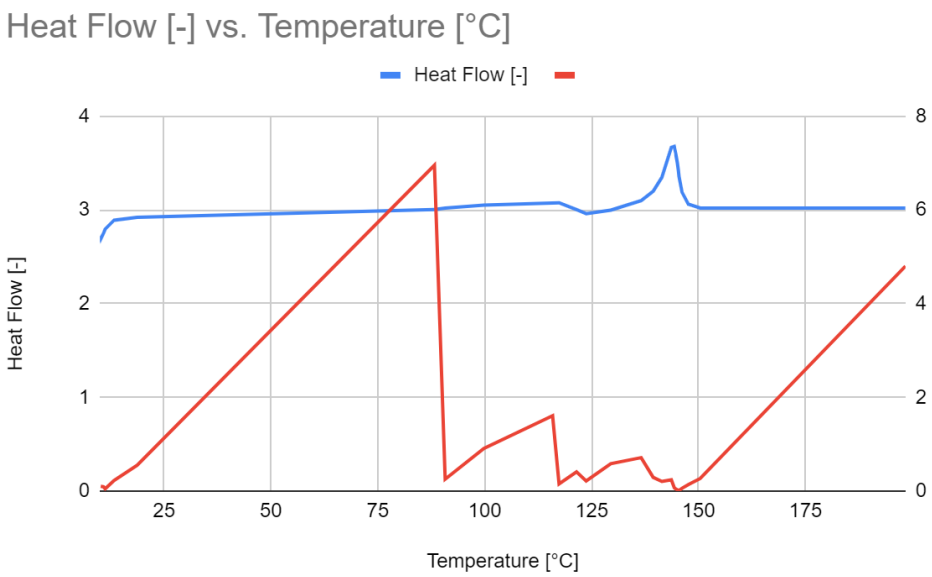
The "missing" points could be reconstructed through a linear interpolation approach, committing an error smaller than the 0.1% of the series' span!

If we vary the derivative error limit, here you can find some results (in red the step size):

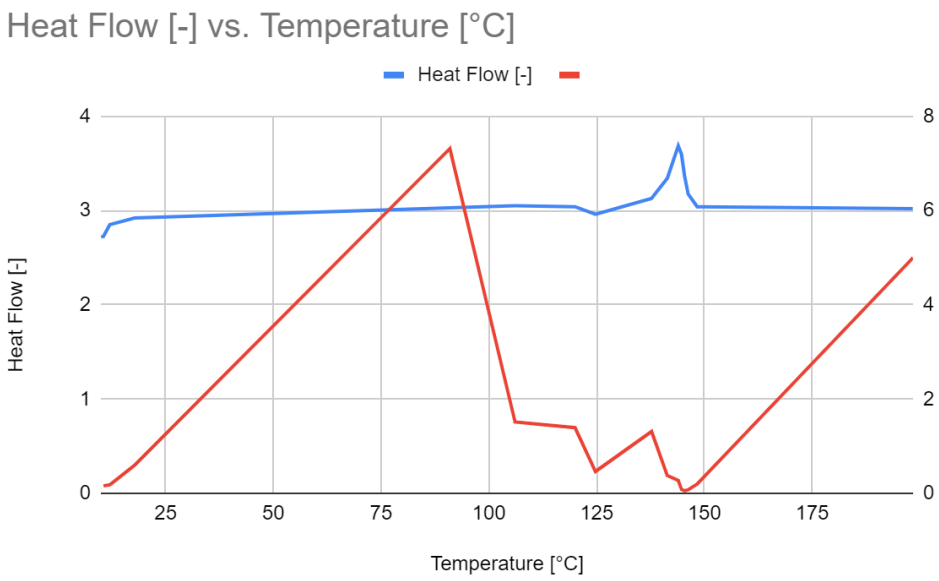
- Error = 0.5% --> 243 Values - Compr. Ratio = 93%



-
- Error = 1% --> 107 Values - Compr. Ratio = 97% (we start losing some information; the axis for the time step is on the right)



- Error = 2% --> 63 Values - Compr. Ratio > 98% (we've lost the info regarding 2nd grade derivative)



In my opinion, we could have two possibilities:

- applying this algorithm for capturing data, defining a small derivative error (= 0.1%), and saving between 80% and 90% of table rows
- applying this algorithm for visualization purposes only: there will be two "raw data tables" - original and compressed - and we expose to Tableau only the compressed ones.

The implemented algorithm

Input: a plain table, that has in 1st column the independent variable, and in the others the dependent variables;

Output: the compressed table, with the same structure, but with much less rows!