DATA ANALYSIS DURING THE DEVELOPMENT OF FUEL-INJECTION-EQUIPMENT (FIE): THE IMPLEMENTATION OF DATA-CONSISTENT METHODS IN ORDER TO GENERATE INTERACTIVE DATA VISUALIZATIONS FOR FIE COMPONENT TEST.

By

Alexandra Lehner

Submitted to

Central European University

Department of Business and Economics

In partial fulfillment of the requirements for the degree of Master of Science

Supervisor: TOTH, Zoltan

Budapest, Hungary

Year 2019/2020

Introduction

The client is a leading global supplier of technology and services. The company employed roughly 400,000 associates worldwide and generated sales of approximately 78 billion euros in 2019. The project was conducted with one subsidiary in Austria, which focuses on the development of common rail injectors for commercial vehicles.

The client has been storing measurement data for the product development of common rail injectors within a Microsoft Access Database (called “Befundungsdatenbank” or “BDB”) in relational schemas. During the last years, the IT infrastructure underlid a massive change.

Since 2018, the measurement data management system (MDM-Tool) "smardGuide" has been developed and gradually introduced. In this web application, the metadata and measurements are stored in a predefined structure and can therefore be called up in a structured and long-term manner. The next planned steps for the client has been to set up common data queries on top of the old and new databases and then make visual analysis with Tableau.

This project concentrates on drift measurement data, which shows the difference of two measurement points observed during an intermediate or end investigation and a start investigation.

IT Architecture

Both data sources coexist and are used for tableau visualizations separately. For the visualization of BDB data, Tableau connects to the BDB and replicates all relations of the tables. The smardGuide internally uses a MongoDB as data storage and provides data via a GraphQL interface. The data is queried from the smardGuide using the GraphQL interface in a Docker Container that runs within an OpenShift cron job. All metadata and measurements are stored in parquet files directly on HDFS (Hadoop Distributed File System). Several HIVE tables are build that allow to further process the data, e.g. SQL queries or Tableau.

Schemas of both data sources

The BDB consists of multiple tables with a relational schema. The table “HydraulicMeasurement” and “MeasurementEvent” contain the measurement data, whereas the tables “Injector”, “GeneralDate” and “Administrative” contain metadata.

The smardGuide is a measurement data management software. The metadata and measurements are stored in a document-oriented structure in the smardGuide database. Those documents are flattened to be then available in a relational schema. Data from the smardGuide can be retrieved via HIVE.

Mapping of features

For a combined data visualization, it is necessary to know which variables and measurement data are needed for the Tableau dashboard. The following steps had to be undertaken:

1. Predefine all variables which have to be visualized in the Tableau dashboard
2. Find those predefined variables in both data sources
3. Match those predefined variables with each other

Step 1-3 has been a less complex task for the BDB as each important variable is stored in a separate column, which can easily be used within Tableau without preparatory work.

Step 1-3 has been more complex for the smardGuide. Mostly metadata is stored on a column-base whereas measurement data is stored on a row-base. This data structure cannot be processed within Tableau without any preparatory work. The preparatory work contains a transformation of the data from row-based to column-based. This process is called "pivoting". Additionally some of the metadata, which is necessary for the analysis, are stored in longer strings, which had to be broken apart in order to receive the required variables from the smardGuide. Also it has been found that the smardGuide was not storing any drift values. In order to retrieve these values, they had to be calculated using a complex SQL query.

Implementation

The clarity about the schemas and the mapping of both data sources has not yet answered the question of how to combine the data sources within Tableau. In the following paragraphs, the two most relevant approaches are described. The first approach describes how to join the data directly in Tableau and the reason why this approach failed. The second approach, which was then also implemented, describes how to join the data in the back-end and retrieve both data sources by one SQL query in Tableau.

Approach 1

Tableau offers the possibility to add multiple data sources. The intent is to extend the rows from the BDB by the data from the smardGuide. For the purpose of extending rows, Tableau offers two possibilities: the UNION-Function or a FULL OUTER JOIN with additional calculated fields. The additional calculated fields are necessary, because joining the data only extends columns, not rows. There are multiple disadvantages and difficulties implementing these solutions. Here are some examples, why theses solution were not efficient nor realizable and did not meet the requirements:

- The BDB would be added to Tableau with its full extend. For the analysis only a reduced amount of tables and variables are in scope. Tableau recommends to read only the necessary data into Tableau in order not to impair the performance of the dashboards.[1]
- When connecting to data with a Microsoft Access database, the FULL OUTER JOIN is not available due to limitations in the Microsoft Access database. UNION is not available with multiple connections.
- Besides of the technical limitations, also the complexity of the schemas and the mapping would have led to a highly difficult solution, which is hardly understandable for anyone who is going to work with this Tableau dashboard.

Approach 2 (implemented!):

The second approach was to combine the data sources already in the back-end. More precisely, the BDB was stored in the data lake directly. Two of the client's employees carried out this replication. By making the BDB available in HIVE, it is possible to retrieve both data sources with one single SQL query. This solution has the advantage that also other measurement data, which are not important for this capstone project, can be queried from HIVE at any time.

[1] https://www.tableau.com/about/blog/2016/1/5-tips-make-your-dashboards-more-performant-48574

Setting up SQL Queries

The representation of the entire SQL query would go beyond the scope of this document. For this reason, the main commands used in the query are listed exemplarily:

WITH, SELECT, CAST, CONCAT, SPLIT_PART, MAX, CASE, LEFT JOIN, INNER JOIN, WHERE, GROUP BY, UNION ALL

Both data queries are combined with the UNION ALL command, which combines the result set of two or more SELECT statements. For the UNION ALL command it is necessary to keep all columns from both results sets in the same order and with the same data structure.

Tableau dashboards

The client is interested in plotting the drift measurement data on the y-axis and the runtime of the investigation, in either km, h or mls, on the x-axis. The runtime in either h, km or mls are calculated in Tableau because investigations have different runtime units. In order to make the investigations comparable the runtime is converted with a fixed conversion rate of 50km/h into all three units. Multiple filter options (customer, injector generation, engine project, etc.) are available. Additionally, as we now plot data from two different data sources, the data source itself is added as a filter option.

Challenges

Authorizations: Due to an approval concept behind each role, approving roles and authorizations usually take a significant amount of time at the client's company. One insight for the client was, that the timely preparation of all necessary roles and authorizations is key.

Schemas: The final implementation of the HIVE tables and their relations were not yet finalized and continuously updated during the project period. Additionally the approach of recreating the smardGuide schema on my own was only partially possible without additional hints from the client's side. Only after multiple consultations with the client, a final structure of the smardGuide schema was then determined, with which the project was continued.

Mapping: Another challenge that took a significant amount of time was mapping the variables between the two databases. Since there was no predetermined mapping of the two databases, the process of finding all variables was a much longer process than originally planned. Variables in the BDB were identified relatively quickly. This process was much more complex in the smardGuide. Each of the approximately 20 variables was either found by manually searching the smardGuide or variables whose storage location could not be found manually were clarified within further meetings with the client.

SQL Query: *Row-based data storage*: The line-based data storage of the smardGuide led to more complex SQL queries. The PIVOT command not available in Impala, therefore a MAX(CASE) command was used. MAX(CASE) can cause errors if multiple entries exist. *Drift calculation*: The drift measurement values are not stored separately in the smardGuide and had to be calculated manually within the SQL query. *Logic of data storage*: The analysis-oriented storage of BDB differs from the memory-optimized storage of smardGuide. The client needs to clarify if further measurement data exists, which is available in BDB but not stored in smardGuide