A Roadmap for Leveraging Business Intelligence as a Growth Engine in Small and Medium Enterprises

IT4BI MSc Thesis

Student: Ivan Popov
Advisor: Patrick Fahrun
Supervisor: Oscar Romero

Master on Information Technologies for Business Intelligence
Universitat Politècnica de Catalunya

Barcelona
04.09.2014

A thesis presented by Ivan Popov
in partial fulfillment of the requirements for the MSc degree on
Information Technologies for Business Intelligence

"The best way to predict the future is to invent it."

- T. Hook

Dedicated to myself who made it despite of all that happened this year ...

Abstract

A rapid growth of the Business Intelligence (BI) market makes advanced analytics available for Small and Medium Enterprises (SME). The process of BI integration and adoption is quite difficult for SMEs because of their limited resources and other peculiarities. This study presents a roadmap for leveraging BI as a growth engine in SMEs. This "BI roadmap for SMEs" considers specification of BI solution requirements, development aspects such as OLAP data modelling and choosing BI software, while paying attention to success factors. Theoretical framework is based on the experience gained during the practical development of the BI solution at the company representing SME.

Contents

List of Figures

List of Tables

Abbreviations

IT	**I**nformation **T**echnology
BI	**B**usiness **I**ntelligence
CIO	**C**hief **I**nformation **O**fficer
SME	**S**mall and **M**edium **E**nterprises
EU	**E**uropean **U**nion
OS	**O**perating **S**ystem
OLAP	**O**nline **A**nalytical **P**rocessing
KPI	**K**ey **P**erformance **I**ndicator
DBMS	**D**ata **B**ase **M**anagement **S**ystem
PaaS	**P**latform **a**s **a** **S**ervice
PoA	**P**ower **o**f **A**ttorney
RoP	**R**equest **o**f **P**ayment
B2B	**B**usiness **T**o **B**usiness

Chapter 1

Introduction

1.1 Topic introduction

Nowadays more and more businesses, including not IT-related, turn their attention to BI as the new source of inspiration for strategic development. However this is not the ubiquitous case and still there are plenty of managers who either do not see the benefits BI can bring or doubt about its return on investment.

According to Gartner [4] the market of Analytics as Business Model, e.g. Business Activity Monitoring, Data Integration, Data Quality, CRM Analytics, Data Warehousing, etc., has grown from $57 billion in 2010 to $68 billion in 2012 and is expected to reach $136 billion in 2020. At the same time IT budgets remain fairly constraint revealing less than 5% growth during 2003 – 2013 timespan [4]. Furthermore, nowadays businesses require sophisticated analytics to stay competitive. Under such circumstances, CIOs have nothing to do except elevating Analytics and BI up to the first level priority. In 2012 31% of potential users of IT-systems were reached by Analytics. Out of these 31%, 25% were driven by traditional solutions of IT departments. By 2014 this number will be 50% and it is expected to reach 75% by 2020. Nevertheless, only 11% of organisations today do advanced analytics [4].

1.1.1 Organisational set-up

The thesis is written while doing an internship as a BI specialist at a company that belongs to the SME segment(due to the NDA signed, the company's name can not be disclosed in the thesis). In particular the work was executed at the Business Development department, which is the entry point of the company's BI adoption, under the supervision of its head who is entitled as an advisor of this thesis.

The work involved communication with an advisor as a representative of the business side and collaboration with the development team while applying and testing implementation steps of the thesis. The working place was allocated in the company's office hosting all departments including IT and Business Development ones.

1.1.2 Motivation

The statistics from the beginning of this chapter tells that further exploration and adoption of BI by businesses is their inevitable next step. The challenge of utilising BI as an internal growth engine is mainly introduced among SMEs for two major reasons:

1. SMEs are limited in budgets and sometimes cannot afford BI solutions
2. SMEs constitute an absolute majority, i.e. more than 99% of all businesses in EU [5]

Currently a variety of BI tools for SMEs is available [6, 7], including web-based, hosted services that do not require installation and maintenance [8]. They provide numerous sets of features as well as diverse pricing schemes.

Choosing the BI tool, SMEs try to find a happy medium between the functionality and the cost of the BI software and/or service. The common approach is trying them one by one which can be time-consuming and expensive. It becomes a cumbersome task to choose the one which will fit the company in the best way in terms of compromise between matching the required functionality and the cost. It is especially challenging in fast-growing start-ups as the solution does not only need to fulfil current requirements but also the requirements in the future which change rapidly as company's size increases in terms of revenue and staff.

As for the particular company where the work on the thesis was conducted, its recent development of the business with its consequent increase in the volumes of operational data produced the necessity of bringing the Advanced Analytics to a higher level. As it will be shown later, the BI usage was quite immature and it was creating a great potential for the BI solution to bring the tangible benefits.

1.1.3 Scope

The main concept of the realisation consists of implementing the goals of the thesis (listed in 1.1.4) at the operational environment of the existing business of the company

from SME segment. The main methodological approach is practical implementation of the planned activities over the IT-system of the company with further analysis and discovering of the general ideas that could be applied to almost any SME. For this purpose, the whole path towards the main objective is supposed to be considered from two angles: if this measure, feature, requirement, etc. is typical for concrete company or for all SMEs in general.

1.1.4 Objectives

The main goals of the thesis proposed was to investigate the real case of adoption of the BI solution within the context of the representative company which belongs to SME. The result of this investigation will bring the insights on how to discover properties and specificities of the business that should be taken into account while developing BI solution, how to define criteria of the successful BI solution and what are potential pitfalls and bottlenecks the business can meet on the way to leveraging BI. Therefore the main objectives of the work can be formulated as follows:

1. To develop a framework of using BI as a growth engine for SMEs. Experience of building BI solution for the specific business of the considered company should be taken into account while creating this framework. Finally it should present a set of steps required to build BI solution, so-called BI roadmap, so that any common-case SME-company could benefit from going through it.
2. To build a real-life BI solution for the SME-representative company. The obtained BI infrastructure should satisfy the company's business goals. In particular it should:
    - enable various departments to perform most of ad-hoc analysis without the support of the BI team
    - provide an "easy to maintain" dashboard that can be used by all company's employees

1.1.5 Initial planning

It was planned to achieve the main objectives in accordance to the gant chart presented at figure 1.1. The implementation steps are listed below in details:

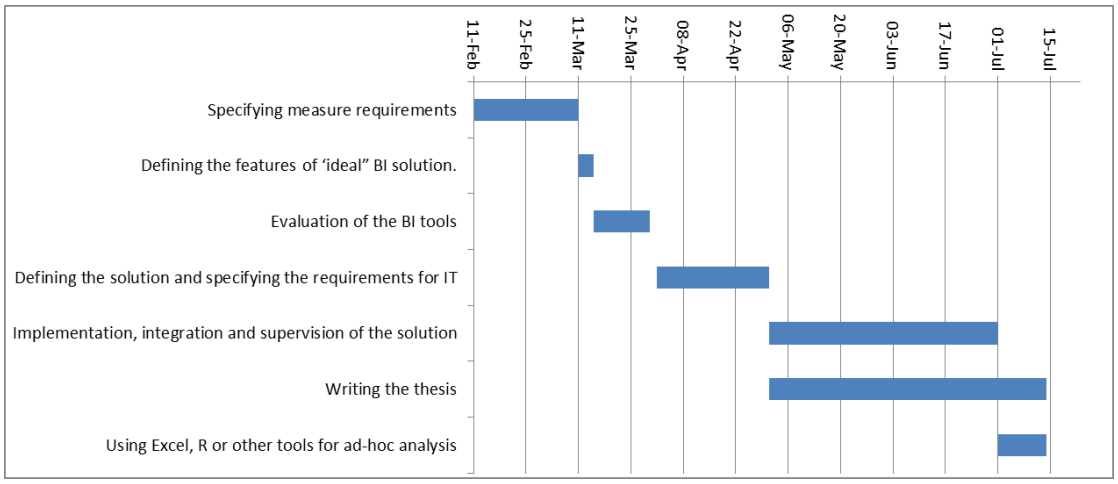

FIGURE 1.1: Initial planing

1. Specify measure requirements

   - At first the business domain of the company should be studied. The task here is to understand the business model, the main KPIs and to find out what are the current needs of the company, to think about BI features which can be a potential solution for these needs.

     Generalisations for SMEs:

     To analyse the connection between KPIs, business goals and BI features.

   - The next task is to detect any dependencies, recommendations or restrictions of developing BI solution based on the parameters of the company's business and IT characteristics, e.g. nature of data-sources, DBMS type and vendor, types of both used and desired analyses, rhythm and peculiarities of analytics, etc. For this purpose, the business of the company will be studied from the technical point of view, in particular: which DBMS is used to store data, how business entities and their relations are presented in the database, what the analytical queries are, what is the workload on what part of the database they produce, etc.

     Generalisations for SMEs:

     To define which of these dependencies, recommendations or restrictions are typical only for the considered company and which are common for SME-companies?

2. Define the features of "ideal" BI solution

   It should be clarified which features of BI solution it is possible to define and up to which extent for both the considered company and other SME-companies.

3. Evaluation of the BI tools

   - At first, several BI tools should be chosen for comparison among existing BI solutions. For instance web-powered BI tools mentioned by Grabova et al. [9]. Primarily price and declared functionality will be considered as comparative criteria. The functionality should be considered in terms of meeting the requirements specified in the first step of this implementation plan. Then the following factors we will be considered: easiness of integration into existing system, user-friendliness of the interface, maintenance efforts required, etc. At last practical evaluation of the tools performance and effectiveness should be conducted.

     Generalisations for SMEs:

     The relationship between the BI tool's functionality, its price and business requirements should be analysed. In particular it should be studied if it is possible

     - to assign weights to BI tool's features and, considering the price, calculate abstract measures such as price/feature, percentage of the tool utilisation, etc.
     - to develop some evaluation formula for the BI tool where some feature's coefficients are same for all SMEs and some vary from company to company
     - to introduce performance factor into this evaluation formula

   - Considering in-house implementation. This task is based on the results from the one above. In case none of the tools provides "ideal" solution, the in-house implementation of BI solution should be built.

     Generalisations for SMEs:

     The goal here is to define measures and factors when the company should consider in-house implementation of BI solution.

4. Defining the solution and specifying the requirements for IT

   At this step the requirements for IT-needs should be specified based on the work done by this moment according to the implementation plan.

   - At first an appropriate database should be selected where all the testing could be run.
   - Then it should be defined which data cubes are required, what are the dimension, measures and facts. Then respective data cubes should be created.
   - ETL-jobs considering the update rhythm of the data should be defined and developed.

- The respective testing should be executed

Generalisations for SMEs:

The goal here is to find the relations between the business requirements from the first step and technical requirements defined at current step.

5. Implementation, integration and supervision of the solution

   Having done the work specified above the implementation of the developed solution will be done in the collaboration with the development team. This mostly means the supervision of the integration and resolving the issues if they occur.

6. Using Excel, R or other tools for ad-hoc analysis

   At the conclusion it is planned to fulfill ad-hoc analysis by means of Excel, R or other tools. The performance and effectiveness of the solution should be measured. This final analytics should tell how successful the solution produced is.

   Generalisations for SMEs:

   At this step the measures of the success of the BI solution should be defined.

1.1.6 Structure of the document

The remainder of the document is organised as follows:

Section 1.2 of this chapter explains basic concepts and terminology used in this thesis and presents related work.

Chapter 2 describes BI roadmap for SMEs. It provides a framework for the developing BI solution as well as presents theoretical discussions on the topic. In particular how to gather and specify requirements, define architecture for BI solution, develop OLAP data model, choose BI software and which success factors to consider.

Chapter 3 presents the case study of practical application of the BI roadmap in particular company from SME segment.

Chapter 4 concludes the thesis and discusses future work directions.

1.2 Scientific Background

1.2.1 Basic concepts

Conceptual, logical, physical design is a commonly-used terminology for describing data modelling in data warehouse [10], [11]. In essence:

- Conceptual design defines data model on the level of entities and their relations.
- Logical design further defines attributes for the entities from conceptual design as well as resolves their relations by means of primary and foreign keys.
- Physical design concludes data modelling phase by defining table and attribute names, attribute types and other database implementation details.

While describing the design phase of the OLAP schema the following terminology will be used: *star schema, snowflake schema, fact table, dimension table.* These terms are widely used in dimensional schemas design and have been defined and described in many sources, for instance by Date [12]. The brief definitions are provided below:

- Dimension table - the table which elements represent the entities of (usually real-world) dimensions like place, time, product, person, e.g. objects and subjects of business processes.
- Fact table - the table which elements represent the combination and interaction of the objects and subjects of business processes(dimension entries) as well as some measurable results of these interactions which are represented by the attributes also called measures.
- Star schema - particular design of fact and dimension tables where the tables are denormalised and all dimension tables are directly related with the fact table.
- Snowflake schema - is normalised version of the start schema where some dimension tables are not directly related with the fact table, but through the other dimension tables.

Business entity is another concept that will be used further in this document mostly in the generalisation parts. It is introduced by the author of the thesis and it is defined as follows:

- Business entity - is a central, essential element of the business, its main result which is offered to the customer such as produced good or service, the main thing that differs companies of the same type of business, companies of the same market segment.

Power of Attorney (PoA) - authorisation given to the company in order to represent customers interests while requesting the compensation from the airline.

Accounting item - financial unit, representing any single money movement whether it is direct payment from one party to another or calculation of interest.

1.2.2 Related Work

BI adoption in SMEs is one of today's hot topics in BI that counts number of articles and reports published.

As it was said in section 1.1.2, low budgets of SMEs create the barriers for BI inception. However, compared to their larger counterparts in other aspects, SMEs are rather more inclined to use BI. Small sizes allow SMEs to be more agile in leveraging BI as a competitive advantage "because of the closeness of executives to the technology, business and customers" as Dresner [13] mentions in his research report. Guarda et al. [14] also sees BI as competitive advantage for SMEs.

Information/Data, *Technology*, *Intelligence* and *Implementation and Communication* are considered by Canes [15] as four critical areas of BI implementation in SME. Main barriers and benefits are also discussed by Canes [15], Suleymanov [16], Scholz et al. [17] and Baransel [18].

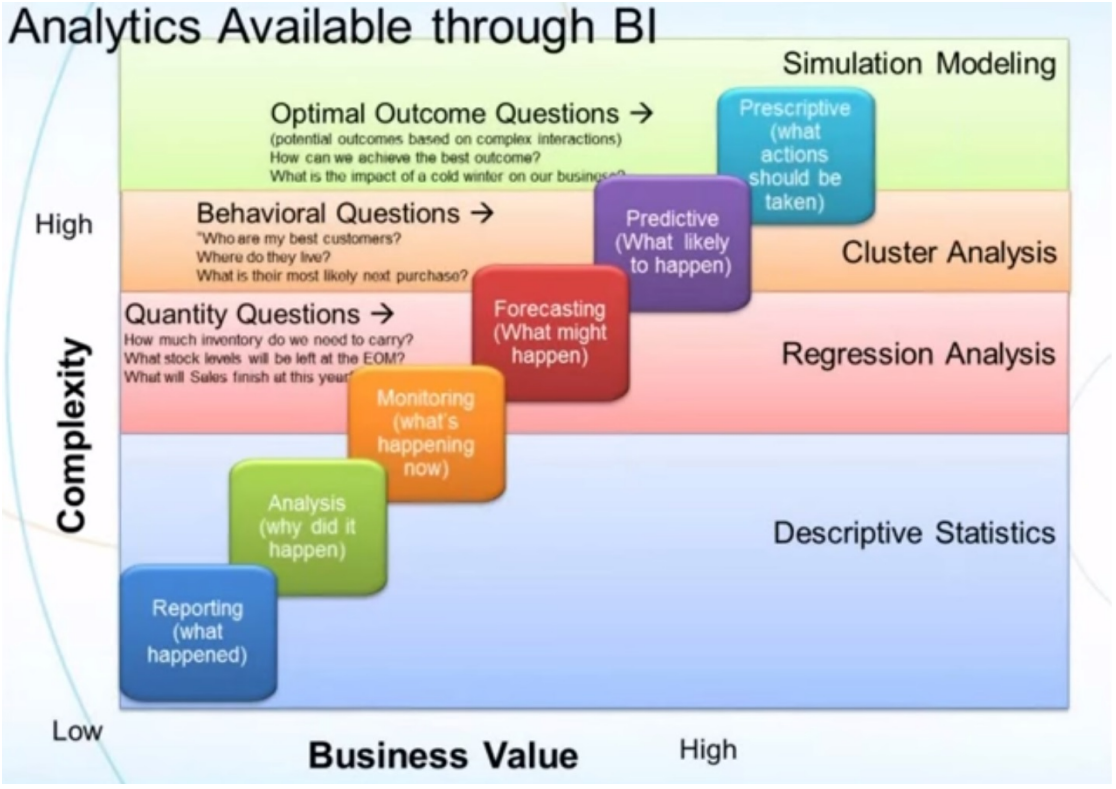

FIGURE 1.2: Analytics available through BI [1]

Currently organisations have a choice between several types of solutions to implement including in-house, open-source and cloud solutions that are discussed by Suleymanov [16]. Information about criteria for selecting BI solution can be easily found, but it is always too general. For instance, [19] mentions speed of implementation, flexibility, and ease of use criteria but doesn't explain how to measure them.

The trend of the recent years such as embedded BI is analysed in the report by Lock [20]. In particular it is mentioned that Best in Class companies conduct six-times the percentage of workforce on strong analytical activities.

Whether BI project will be successful or not depends on many factors. Those that may lead to a pitfall are discussed by Imhoff [21] and Wise [22]. Successful factors studies include those by Olszak and Ziemba [23] and Scholz et al. [17].

There are several studies on particular cases of BI implementation in SMEs. One, by Olszak and Ziemba [23] discusses determinants and barriers in BI adoption based on the results of interviews with representatives of SMEs located in Poland. Other by Scholz et al. [17], considers 214 German SMEs.

Figure 1.2 attempts to draw a whole picture of possible scenarios of analytics available through BI today. It shows six possible types of analytics that can be established depending on business goals and available resources.

Chapter 2

BI roadmap for SMEs

During the process of designing and developing BI solution for the company, at which placement the thesis was written, lots of discoveries has been made and many ideas were born on a topic of leveraging BI in SMEs. Some of them are relevant only in the context of the current company's business, the major part, though, can be relevant for many SMEs. Those from this major part are provided below in a form of BI roadmap for SMEs that summarised the experience gained. The roadmap is depicted in the diagram from the figure 2.1. Let's first consider the process-flow of the roadmap and describe its elements in details in subsequent sections of this chapter.

After the acquaintance with business model of the enterprise a further exploration is divided in two parallel directions: technical examination of the system and identification of business requirements. The former one continues until the final requirements are specified. As for the latter, it starts with the definition of the scope of BI solution, for instance departments. Next, strategic goals of targeted departments are identified with further specification of their KPIs. The means of KPIs monitoring are included in technical examination as well as the current problems in analytics. Besides identification of current problems in monitoring KPIs, new analytical perspectives to be used in the future are considered.

A dark rectangle in the background implies that directions of analytics, its current problems and future perspectives under both business and technical points of view are accumulated in iterative manner. First, depending on the scale of the business, it can be quite difficult to identify KPIs or problems of analytics all at once. Second, in some cases it can be more convenient to establish this procedure separately by departments. Anyway, whether to execute one or many iterations should be decided for each business individually.

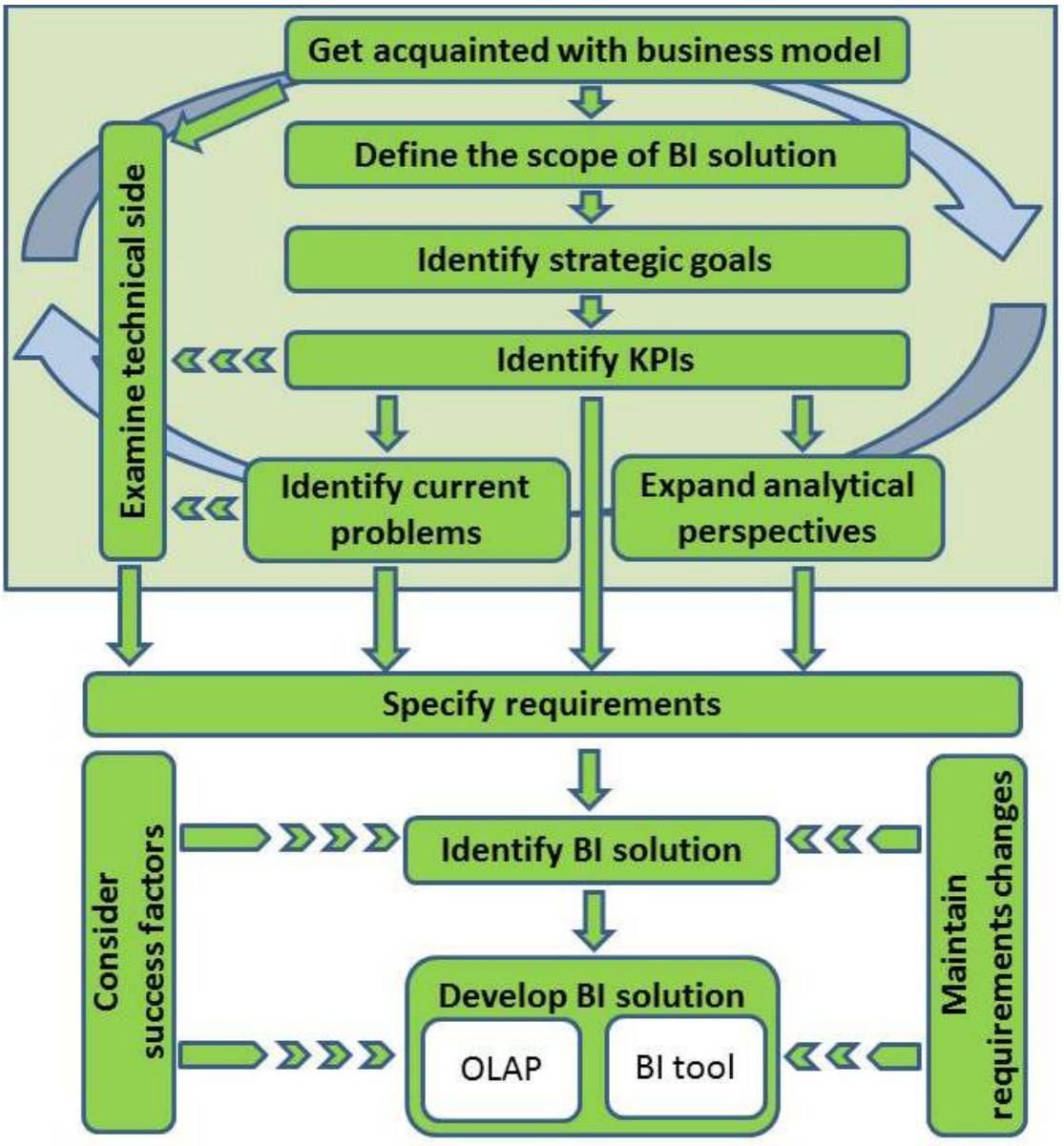

FIGURE 2.1: BI roadmap for SMEs

Based on the discovered aspects from above, the requirements for the solution are specified. Identification and development of the BI solution are naturally quite an individual for each business processes based on the requirements previously specified. However, general BI solution typically consists of at least two components such as OLAP system and some BI software platform. After-deployment maintenance of the minor and significant changes in requirements is reflected in the modification of the solution developed and in the modification of the architecture of the solution respectively. The final success of the BI solution is determined by success factors taken into account.

2.1 Getting acquainted with business model

The experience of BI integration gained while working on this thesis allows to formulate the following conclusion regarding the initial step of getting familiar with the business model. The person, responsible for enabling BI solution, must get very deep understanding of all business processes that make the company function. Ideally, he/she should know all the workflows that take place in all targeted departments. One of the common mistakes mentioned by Wise [22] is to concentrate on IT needs rather than on business objectives. Although the decisions that will follow up will have a technical nature, a good orientation in operational processes will play a key role in making these decisions.

Reading documentation is one of the ways to familiarise with the business, but the most important knowledge and understanding come from the discussions with people from key positions. BI goals demand to receive factual information, but people tend to present their operational experience freely formulated in the form of stories. While "5 best practices for telling great stories with data" [24] discusses how to bring life to data and facts, the task for the BI integrator at this step is to do a reverse engineering of this paper and to extract metric-oriented information from the stories told by people.

2.2 Scope of the solution and strategic goals

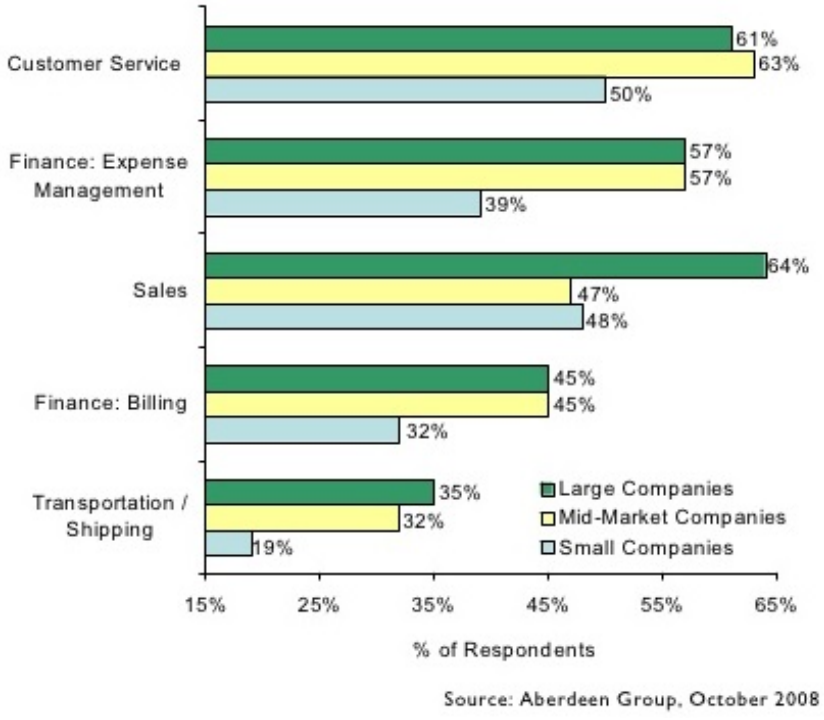

FIGURE 2.2: Top departments driving the need for BI [2]

"It is critically important to take a strategic view in defining critical success factors" says Canes [15] talking about the most important aspect of the BI project.

At first it is necessary to define the scope of BI solution in terms of departments that have a need for BI. The figure 2.2 from study by Aberdeen Group [2] lists departments in the biggest need for BI. As it is shown in section 3.2 *Customer Service* and *Finance* departments were also among BI-targeted departments in example of the considered company.

Strategic goals reveal something general by nature and, therefore, some of them such as "increase customer satisfaction of ..." or "identify ... that produce the biggest revenue" will sound similar for most common departments from one SME to another except for the part referencing to business entity (recall definition from subsection 1.2.1). Strategic goals will be specific for the departments directly related to the business entity. In fact, *Legal* department is a source of peculiarities in the studied business as it deals with legal case that is a business entity of the considered company. Therefore, except for legal department, all written in subsection 3.2 can be true for many other SME and the general strategic goals for SME's departments can be obtained by replacing the words 'legal case' with 'business entity'.

However, similar strategic goals are not an indication of the similar BI solution. Current problems and needs as well as current state of the IT and BI influence more on the final solution as it is demonstrated in case study in chapter 3.

2.2.1 KPIs

Strategic goals, having been identified, require quantitative indicators whether SME moves towards to or away from their achievements, for what KPIs are used. In the attempt to structure and identify KPIs there is a big temptation to classify all that is measured as KPIs. It can be the case for SME to keep all kinds of statistics, graphics, reports - all in one place, for instance in dashboards. Some of these dashboards could be just of no big importance or consist of tabular data representing some database views. One should bear in mind that KPI definition implies only *key* and only *performance* indicators to be assigned to this category. Jackson [25] provides a campaign against illiteracy about KPIs in his book. According to it there are two types of KPIs on a high level: *visionary* and *tactical*. The former are the most general ones reflecting what the company tries to achieve. The latter group is more related to daily operations and finally it forms the main focus of the BI solution. Jackson [25] also provides four attributes that characterise KPI:

- *Time period.* KPI indicates a trend over a time period.
- *Benchmark.* KPI is used to monitor deviation of metrics from certain control values.
- *Actor.* In case the metric deviates significantly, there is an analyst responsible for further analytical investigation of the reason.
- *Action.* In case the metric deviates significantly, some action should be taken.

If the metric doesn't have all four attributes than Jackson [25] defines it as *count* or *ratio*. Anyway it is not prohibited to use BI solution for monitoring such metrics, but the requirements specification should be done with an emphasis on KPIs. It is also important to properly identify KPIs as, later on, they will be used as a landmark for the BI solution development.

It is shown in case study that the meaning of some KPIs was not clear for employees. It led to the necessity to for them to appeal to IT-team. To avoid such problems, it is important to assure that KPIs in the BI solution are understandable not only for its developers but also for targeted audience. This obvious fact could be overlooked if BI solution is being developed without proper level of collaboration with its prospective users.

2.3 Current problems and new analytical perspectives

In general, current problems and future needs can be identified the same way through interviews with key employees and familiarising with workflows of operational processes. Actually the problems of current analytics and its new perspectives are interconnected and come together quite often. For instance, inability to measure something by a certain dimension can be classified as both: current problem and necessity of analytical expansion.

There is also a dependency between the degree of interactivity of analytics for end-users and the amount of problems and needs. If the solution is completely operated by IT-team providing a read-only analytics, the users will always have a requests for modification or creation of new elements of analytics like dashboards or reports. If end-users can not only consume, but modify and create their own analysis, a considerable part of the overhead related to communication between BI users and IT-team will be reduced.

2.4 Examine technical side

As it was mentioned in the introduction to this chapter, technical aspects of the business are studied parallel to the process of strategic goals, KPIs and current problems and needs identification.

Data sources

The technical examination of the business should be started with the identification and detailed study of the data sources. This can be done by searching for answers on the following questions:

- what kind of data is used in the system?
- what data is stored?
- where and how is the data stored?
- how is the data processed?

After getting the whole picture, it should be defined what data sources are required for prospective analytics.

Unintegrated data as typical state of IT-system in SMEs

Data integration is a typical source of technical issues for SMEs. "The information base of an SME involves a lot of unstructured data which are stored in e-mails, a high number of Excel Sheets and unconsolidated reports and query results from a number of OLTP systems", according to Baransel [18]. This can be caused by many reasons. For instance, IT-systems of SMEs often present extended version of the proof of concept since start-up times. Another reason is that limited resources of SMEs postpone system refactoring for "better times" that never come.

What data to integrate?

After data sources have been identified, it should be determined whether they should be integrated or not. In most cases, the answer will be positive as integrating data in a single data warehouse eliminates the problem of establishing uniform analytics in different operational areas as well as the problem of unintelligent, manual data processing. In seldom cases some data like Google Analytics [26] data may not require integration. We will see in example of current project that data about phone calls from Top-Link provider was integrated into the OLAP schema by common *User* dimension. The point is that Top-Link provides *agentID* of the employee from Customer Care department

who answered the call that can be matched with *agentID* attribute of *User* dimension. *Calls* fact table in its turn refers to *User* dimension thus making possible to connect phone call statistics with other analytical areas. Besides, as mentioned in section 3.3, the connection of Google Analytics with back-end data is in the future plans.

How to integrate?

In case data integration is required, the next question would be whether SME is able and willing to do it on their own as a preparation step of BI integration or to make it a part of a complex BI solution. The answer to this question defines whether ETL functionality will be a part of requirement for BI tool.

As it was mentioned in the BI roadmap diagram from figure 2.1, KPIs and current problems should also be examined from technical point of view. Finally, the most important outcome from this step of the BI roadmap is definition of the set of data sources to be supported by the BI tool.

2.5 Specification of the BI requirements

As shown in the roadmap diagram, requirements for BI solution iteratively accumulate details of technical examination, identified KPIs as well as current problems and new analytical perspectives. Therefore they present quite a diverse set of parameters such as data sources and their characteristics, requirements to the BI software, currently used and required for further analytical measures and dimensions, etc. Though, the set of parameters is not limited and will vary from enterprise to enterprise, the following three characteristics of BI system are highlighted:

- *Rhythm of analytics.* It should be defined what is the average or most common frequency of update that satisfies analytical needs. It is also important to know if there is an analytical area that requires real-time data and if this area is major or minor part of the analytics. As far as real-world systems can handle the time only as discrete measure, in principal *real-time* can be an arbitrary interval. Whether *real-time* is a second, a minute, an hour or more should be defined be with regards to the needs of particular SME. At the end, rhythm of analytics may impact the whole architecture of the BI solution.
- *Depth of analytics.* This parameter defines the volume and dimension of multi-dimensional analysis. At this point all must-have dimensions as long as their hierarchical depths should be defined.

- *Time-frame of analytics.* Basically the choice should be made between creating analytics based on the whole operational period of the enterprise or some fixed interval. This may be extremely important for the enterprises that reached the scale of Big Data.

During development of the BI solution for the considered company several observations, that could be used disregarding of the object of analysis, were made.

- For KPIs connected with time dimension, the rhythm of update should at least match the level of granularity of the time dimension. The exception are KPIs related to the current/last day/month/year that require to be updated before the end of the current day/month/year.
- KPIs not connected with time dimension are required to be updated as frequent as possible.
- Work indicators such as *number of answered calls* or *emails* may require real-time monitoring.
- Due to the monthly-based nature of the financial reports, most of the KPIs for finance represent monthly statistics and therefore it makes sense to update them monthly.
- Customer care departments deal with clients via diverse means of communication and solve diverse problems, but the customer and operator (a person who answers phone or email) dimensions will be present in any kind of analysis.

2.6 Identification of the BI solution

When specifying BI requirements it is important to take into account that "implementing a BI system is not a simple activity entailing merely the purchase of a combination of software and hardware; rather, it is a complex undertaking requiring appropriate infrastructure and resources over a lengthy period" [23].

Figure 1.2 illustrates that the main challenge for SMEs is to find the happy medium between the cost and the value.

Naturally there is no uniform algorithm for identification of the solution that could be applied at all SMEs ubiquitously. Arriving to this point of the BI roadmap, each SME will be carrying its own set of requirements and problems identified previously and to be addresses by the BI solution. However, choosing directions at the following reference

points indicates the way of problem solving and leads as a lighthouse towards shaping and formalisation of the BI solution.

Business vs Technical requirements

The process of BI solution identification starts from reconsidering requirements specified. First, requirements can be splitted into business and technical ones. Olszak and Ziemba [23] mention that BI system may be analysed from technical and business perspectives which means that BI solution is able to address both technical and business issues. Case study implementation where OLAP schema provided users with better understanding and easier manipulation of the data, whereas BI tool provided technological access to it, can be an example of the above said.

IT-based solution vs Self-service BI

FIGURE 2.3: BI solution identification. Starting point

Starting from the most general things, probably the first question to be asked is whether the solution should be more of IT-based or more close to self-service BI. Splitting SMEs in two groups conventionally called "small" and "medium" enterprises, the former approach is more suitable for medium or technology-based enterprises that include pretty big and solid IT-departments. The latter approach happens to be one of the current trends in BI that is confirmed by the report of Econimist Intelligence Unit [27]; it is more suitable for small enterprises.

Software costs vs IT-specialist costs

However, the experience of the case study revealed a barrier on the path of self-service BI being established in SMEs, especially small ones. It can be described as follows: the closer solution to the self-service BI, the more intelligence in technologies it requires. This means - more functional, more complex, more reliable and, in general, more expensive software is required. The point here is that another fact can be left overlooked. It is a fact that on the path in the opposite direction, towards IT-based solution, there are thresholds after which the number IT-specialists required to support the solution increases. The cost of even one IT-specialist per year is many cases can be much more

than the cost of the most expensive software. Therefore besides the price of using the software for self-service BI, the price of "not using" it should also be considered.

Role and place for OLAP

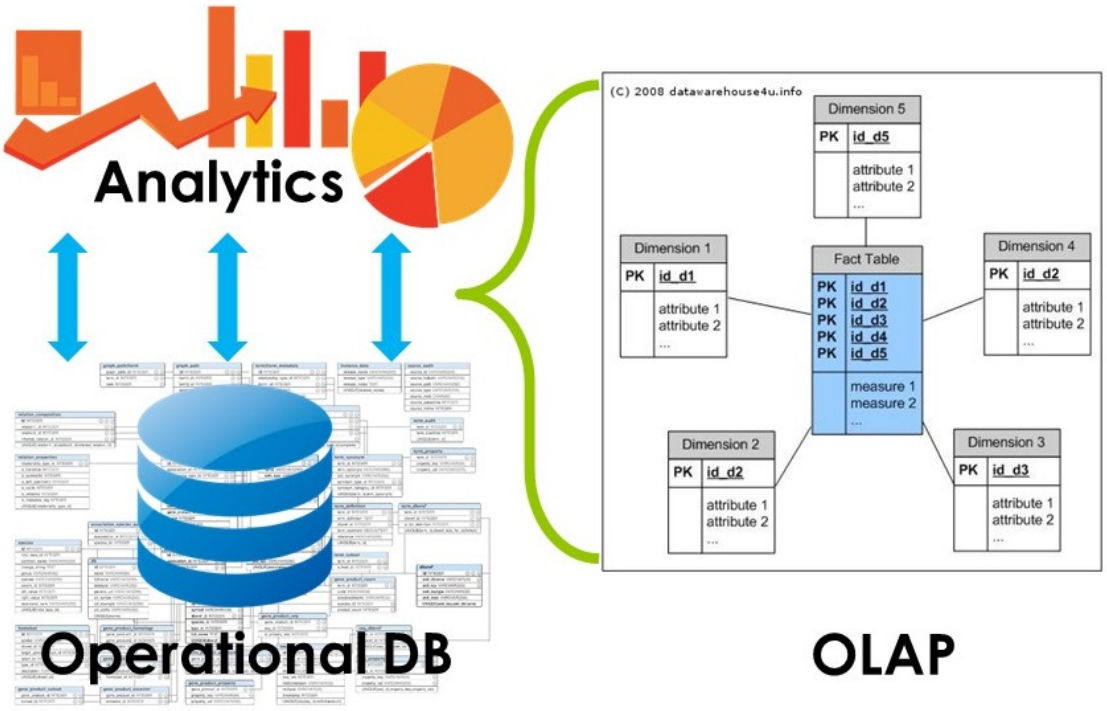

FIGURE 2.4: OLAP as intermediate logical layer

In the development of the self-service BI topic, it is worth to say that it doesn't appear by itself with the purchase of some software, although vendors promote it exactly in this way. Establishing self-service BI requires actions from IT side and an inception of intermediate logical layer between the Analytics and operational data is almost an inevitable step. This layer is supposed to hide typical complexities of operational data from end users, exclude redundant data and pre-calculate some metrics as described in subsection 3.6.1.

One of the solutions is developing OLAP star- or snowflake-schema that was also implemented at the considered company. This approach is also beneficial from the maintenance point of view. If all structural changes of operational data such as renaming, removal, etc. of tables and their attributes are followed by respective adjustments in the OLAP schema, then analytics remains stable and reliable without necessity to maintain each of its elements like dashboards or reports.

This intermediate logical layer can be either created manually by custom script or using ETL functionality of the BI tool. The choice depends on complexity of the system, in particular volumes of the data, operational data model and schema, rhythm of analytics, etc.

Embedded BI? At which level?

Another angle, BI solution should be looked under, is embedded BI that is also today's trend. First, it should be defined whether internal or external users of application will benefit from embedded BI and if yes - at which level it should be embedded. For instance, 5 levels of embedded BI from static reports up to advanced analytics against a data mart are described in Jaspersoft's report [28].

2.7 BI solution development. OLAP component

In most cases analytics will require to be established on top of some OLAP data layer. Denormalised state, poor structure and redundancy of operational data are just a few reasons to be mentioned. Therefore, creating a multidimensional model becomes an essential part in BI solution development. This topic has been well-studied and well-documented in scientific literature. First of all, the whole process can be presented as three consequent phases of design, commonly-referred to as *conceptual, logical,* and *physical design* [10]. Though these phases typically have significant overlaps, let us consider them one by one.

Conceptual design

The main outcome of the conceptual design is the set of entities and relations that will be used as facts and dimensions at the further design phases. Actually, the proper work at the previous steps should contain all required information. This information needs to be transformed from the requirements into a form of entities and their relations. One part of entities that will become dimensions in the future can be extracted from current and future analytical perspectives. For example analysis by *product* and *region* produces *product* and *region* entities and dimensions respectively. The rest part of entities (future facts) can be extracted from the events that happen according to the discovered business model.

Logical design

At this phase, attributes of entities are defined and their relations are resolved by means of primary and foreign keys. A methodology of building Dimensional Fact model from Entity/Relation scheme described by Golfarelli et al. [11], could be used, for instance. Alternatively, a *Bus Matrix* proposed by Kimball and Ross [29] can be created by defining a crossings between measures and dimensions. In order to identify measures one should recall specified requirements as well and in particular KPIs. In fact, most of the KPIs will be exactly the measures. The rows of the bus matrix will form what Kimball

and Ross [29] call *first-level data marts* that can be further organised in *consolidated data marts* representing final facts.

Physical design

Physical design concludes data modelling phase by defining table and attribute names, attribute types and other database implementation details.

Use existing analytical queries for building multi-dimensional model

The absence of consolidated BI solution at the enterprise being an object of BI integration doesn't mean that KPIs are not monitored. Most probably it is done via ad-hoc SQL-queries. Analysing these queries can be useful in building multi-dimensional model. Dimension table is typically created from the operational one by selecting required attributes, excluding the others and adding calculated fields. Its connection with *select* clause of the query is obviously straightforward. The entry of the fact table represents the crossings of several dimensions characterised by some measure(s) and therefore it is usually based on the join of several tables. The underlying SQL-query of the KPI can point on the fact table to be used for it. The experience shows that each fact table is based on the same type of join with the same set of tables.

Avoid additional overhead with simple dimensions

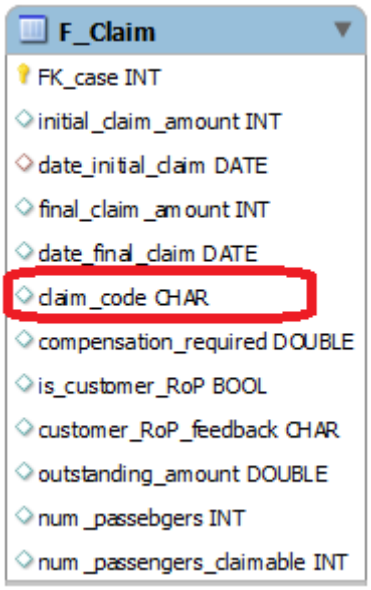

FIGURE 2.5: Embedded simple dimension

Dimensions without deep level of hierarchy or with only one attribute not counting ID attribute can be embedded into the fact table by this non-ID attribute that is demonstrated by figures 2.5 and 2.6. In the case study example attributes *unsuccessful_reason*

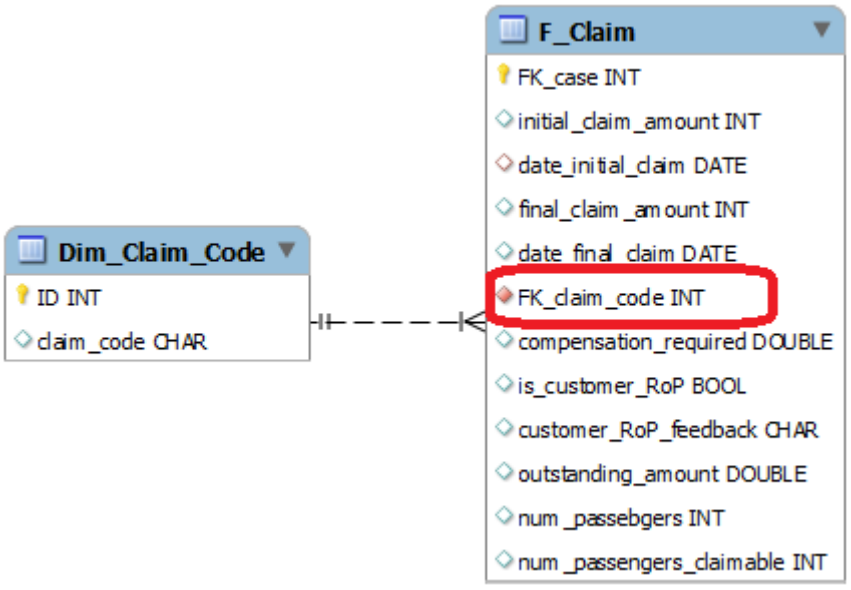

FIGURE 2.6: Simple dimension not embedded

and *legal_insurance* of Case dimension, *claim_code* of Claim fact, *transport_status*, *transport_assessment* and *transport_data_source* of Problem Flights fact could be designed as a separate dimensions. However this would produce just additional overhead because of two reasons:

1. These dimensions would have only two attributes: ID and the attribute with actual value. In this case instead of referring to the entry of dimension table by ID, it is possible to insert the actual value of this entry into the table which refers to the considered dimension. Thus the referring table would preserve the same amount of attributes and no additional dimension tables would be needed. For instance, *F_Claim* table could have an attribute containing IDs of entries from dimension table *Dim_Claim_Code*. Instead it has *claim_code* attribute directly containing the claim codes used in the system and thus the same result is provided in more efficient manner.

2. These dimensions would be used only in one place of the schema to the opposite of Country or Airline dimensions.

Therefore, for efficiency reasons all potential dimension tables that are supposed to be used in a single place of the schema and that have only one attribute, not counting ID, should not be created, while the respective values should be directly embedded into the referring tables. The benefit of such an approach is reduced overhead without any semantic loss for the analysis.

Use timestamps or historicise data?

The problem of analysis based on data without timestamps as well as its solution is described in the following sections of the case study: 3.4.1, 3.3 and 3.7.1.2. In essence, the lack of timestamps for some metrics forces to calculate them and store on a regular basis, for instance daily. In fact this is another problem that should better be solved at the operational level by storing timestamps of all possible data manipulations. From the other hand, if it is not possible and/or reasonable to store timestamps of everything or if the business already operates, having a separate table with historical data can be a solution.

Data quality matters

Data quality issues can come across as a side effect during design and development of the OLAP schema. Although it is possible to execute some data cleaning at this intermediate layer of the IT-architecture, it is better to do it at the level where the problem is located. As the experience showed, all data quality issues from not normalisation to wrong data type finally lead to the problems ranged from additional overhead up to incorrect analytics.

ETL vs custom script

Technically, OLAP component can be implemented by means of ETL functionality of some software or by custom script.

The former is of course an option if the enterprise already uses some BI tool that provides ETL functionality. It is also reasonable to use a tool if transformations are mostly related to matching operational tables on those of multi-dimensional model rather than to programming complex logic of extracting data from diverse tables, grouping it, filtering. etc.

Custom script can be an option if purchasing/installation/maintenance of some BI tool creates an overhead. It is useful when the final OLAP schema is not very complex but its internals such as facts require highly-specific logic for data extraction. It is important to run scripts populating fact and dimension tables in a single transaction. Otherwise there may be an issue when the fact being created refers to the dimension elements which did not exist when dimension tables were populated. This can be the case when operational data is being added while the script is running.

2.8 BI solution development. Tools evaluation

As it is mentioned by Canes [15], recently many pricely-affordable BI software appeared on the market offered as SaaS or using licensing models. However studies such as by Isik et al. [30], where only 21% of SMEs consider using software to support innovation process, show that main barriers belong to those of early adoption factors like *not awareness of existing software* and *not clear benefits* described by Rogers [31]. At the same time Canes [15] cites one of the market analysts about the growth limitations of currently affordable platforms and provides the drawbacks of so popular spreadshits. Eckerson [3] notices that BI software should be distinguished between tools and platforms that meet the needs of all users in organization presented on figure 2.7.

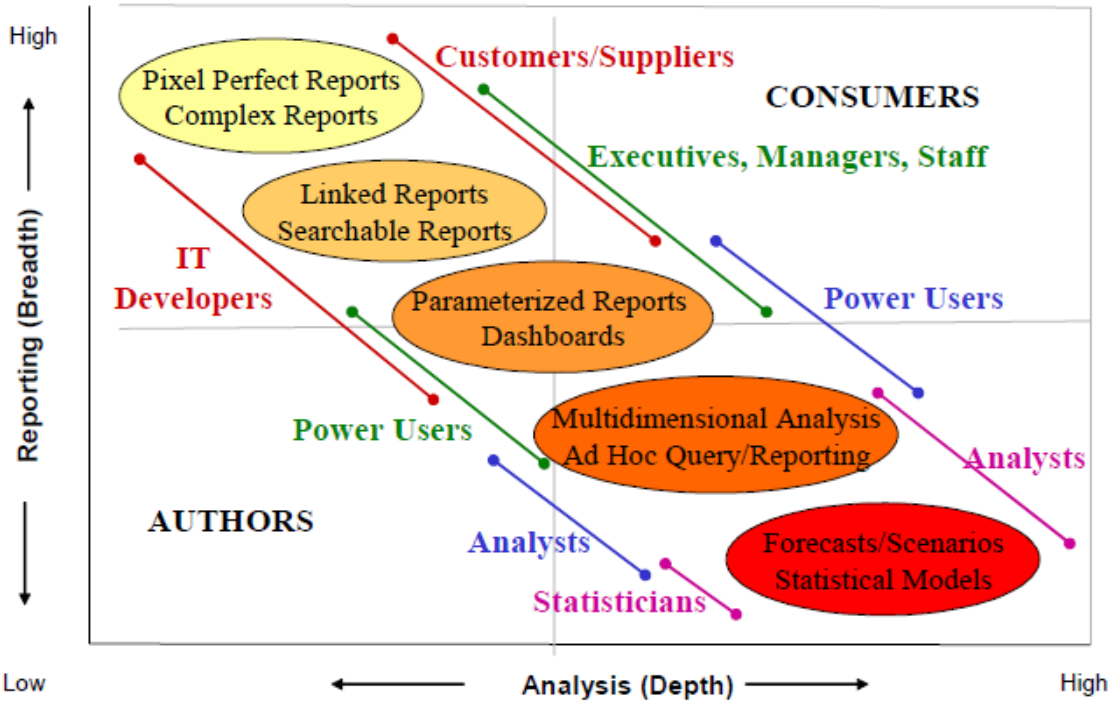

FIGURE 2.7: Mapping BI User Segments to BI Functionality [3]

Tools evaluation obstacles

The process of the BI software evaluation may turn out to be quite challenging and controversial because of the following reasons:

- *Lack of information.* The set of technical characteristics and main features presented on the official web sites varies from vendor to vendor. Some pieces of information may not be found even in the documentation and require the vendor to be asked directly.
- *Complicated and confusing deployment scenarios.* First, the models of usage vary from completely online to completely offline ones with many intermediate options.

Second, the tools offer standalone solutions as well as modular ones with different numbers and different names of these modules. This makes choosing the optimal deployment model quite a cumbersome task.

- *Not fixed prices.* Some vendors do not reveal the prices of their products or prices of individual components in order to involve the buyer into communication with their sales departments. This makes choosing the deployment option even more complex.
- *Time-consuming.* All the above factors lead to additional time of the evaluation. Actually the biggest "time consumers" may be not the factors related to the work with the tool but rather those related to the search of information and to the communication with the technical support and sales departments of the respective vendors.

The evaluation approach proposed by the roadmap consists of two major steps: preliminary and detailed evaluation. These steps can be thought of theoretical and practical evaluation where the latter has the biggest impact on the decision making. This is due to the fact that in most cases tools look amazing at their demos, and for those with similar characteristics the impression of which is the best and which is the worst is strongly influenced by the quality of presentation. Only applying these tools within the company's IT-infrastructure, to particular operational data reveals the true benefits and hidden drawbacks of leveraging them as a part of BI solution for the particular enterprise.

2.8.1 Preliminary evaluation

Preliminary evaluation is based on the tool's features and technical specifications obtained from the documentation as well as from samples and demos available on the vendor's official website. Its main goal is to execute a high level review of the big number of tools and to produce a short list of those for that worth to be evaluated further in details.

Choosing vendors

It is not possible and doesn't make sense to test all possible tools. It is proposed to define the filter that would reduce *all possible* to 10-20 candidates. The filter may be created based on the must have features and main requirements for the BI tool and BI solution in general. Articles with comparative overview such as Gartner's Magic Quadrant [32], enterprise's previous experience of working with BI tools or insights of the competitors and colleagues from industry may be used as a source of inspiration for the formation of list of tools.

Choosing evaluation parameters

Similarly to splitting requirements for the BI solution described in section 2.6, evaluation parameters can be splitted in two major groups: business and technical ones. The former group defines the business value that the tool is able to provide. The latter one considers technical requirements that are related to the BI tool such as support of certain databases. The set of parameters is defined for each SME and depends on previously specified requirements.

Does the price decide everything?

No questions, the price of the tool should be considered, but it is important not to focus on it. Actually, it can be pretty difficult task to find out the final price that may vary depending on number and type (read-only or full access) of users, type of deployment (cloud or on-premises). Therefore it is suggested to find out the order of magnitude and make it as initial threshold. Having chosen tools that satisfy this rough threshold it is suggested to calculate the final price after evaluation and communication with sales department of the vendor. First, if the tool is not satisfactory, the company won't be willing to purchase it even if it is very cheap. Second, if the tool meets the requirements but its price doesn't fit in the budget, communication with vendor's representative may bring agreements on price and licenses scheme beneficial for both sides. Unfortunately this logic may not work in some SMEs. The research by Estrin et al. [32] on 200 SMEs revealed that 80% of barriers to technology adoption are non-technical. This includes the managerial view on the software as on expenses rather than on the source of business advantage. This leads to the choice by the cost rather than by functionality.

2.8.2 Detailed evaluation

Detailed evaluation assumes practical evaluation of the candidates from the short list produced by the preliminary evaluation.

Real-life scenario

Practical evaluation is supposed to reveal the difference in impression left by the tool in vendor's demonstrations compared to the one from a real-life that can be quite a huge. The main reason of this difference lays in a fact that in most cases data from demonstrations is well-prepared beforehand and has simple structure. Experience of the case study confirms that the best way to evaluate BI tool - is to examine it in real-life scenario as nowadays all vendors offer a trial versions of their software. In most cases trial version is limited in time rather than functionality.

As far as the goal of the real life scenario is to get an idea about the tool's performance and efficiency within the business context of the enterprise, it should simulate some business process as much as possible. For instance, it can consist of building existing dashboard/report by means of examined tool or executing analysis that is typical for the enterprise.

For the sake of objectivity, the time-period for the whole evaluation process including gathering information from the documentation and demos should be the same for all tools.

Emphasis on important features

At the next step requirements for the BI tool should be reconsidered. Although the tools are supposed to be evaluated under many different angles, some pieces of functionality should be considered with an emphasis. The goal here is to decide what tool's features are crucial for the business of the enterprise. It can be derived from the definition of what type of analysis is the most important and/or the most common one. Finally this will result in a set of features that should be paid the biggest attention to during evaluation.

Measuring quantitatively

Whenever something is evaluated by several criteria there is always a demand to join results of evaluation by each criteria into a single measure that will identify best and worst objects of evaluation. As for BI tools, for instance, the following scheme of quantitative evaluation can be used. The scheme assumes assigning two levels of importance for each feature of BI tool. Actually, the features defined in the paragraph above can be considered as having higher level of importance. During evaluation positive and negative aspects should be collected and assigned to the feature or set of features. Features with higher level of importance receive +/- 2 points for each positive/negative aspect respectively. Other features receive +/- 1 point respectively. The final sum of these points defines the quantitative assessment of the tool.

Assumptions of self-service BI

The biggest conclusion for this section is related to the state of the art in BI software. At the moment so much desired by business and so much promoted by vendors self-service BI is not available as something that can be just purchased, installed once and used without any maintenance from the IT-side. Self-service BI that can be achieved includes involvement of IT-team in setting up the whole infrastructure from data connections up to data sources and maintaining them. It basically means that the end-user deals only with measures and dimensions with ability to access underlying data if needed. Business

users don't have skills, time and desire to deal with technical aspects of BI solution that is inevitable for any BI tool. Therefore it is necessary to have at least one IT-specialist to maintain BI solutions in setting up data connections, creating and modifying data sources, scheduling updates, etc. In this case end-users will be working with *prepeared* data and will be exploiting BI tool as efficiently as in vendor's demonstrations.

2.9 BI success. Factors and measures

BI success is related to the positive value an organisation obtains from its BI investment [33]. This positive value emerges from achievement of main objectives that are different for different organisations and, therefore, measures of BI success "differ across organisations and even across instances of BI within an organisation" [30]. However BI benefits can be classified in four groups according to Elbashir et al. [34]:

- *Business supplier/partner relation benefits* define improved coordination with business partner, improved responsiveness from/to supplier, reduced transaction costs.
- *Internal processes efficiency benefits* define increase of staff productivity and reduction of operational costs.
- *Customer intelligence benefits* define increase of marketing efficiency.
- *Organisational (strategic) performance* measures strategic enhancements such as increase of revenue or improved competitive advantage.

Important thing is that these benefits should be measured after some time since BI solution has been deployed. Imhoff [21] mentions that "your project may not truly end for several months, perhaps even years, after it has been organised". Time to implement is one of the first things that should be considered by management before starting BI project.

The figure 2.8 can be used as a landmark for timely comparison. Even for SMEs the discussion is going on about years, not months as the shortest age of BI deployment is specified as *up to 2 years* for 46% of SMEs while 38% of them require more than 2 years for BI deployment.

Besides numerous success factors there are many dimensions that these factors can be organized as Olszak and Ziemba [23] mentions. Ariyachandra and Watson [35] suggests to consider factors from process and infrastructure perspectives where the former is related to implementation and deployment of the BI solution and the latter is related

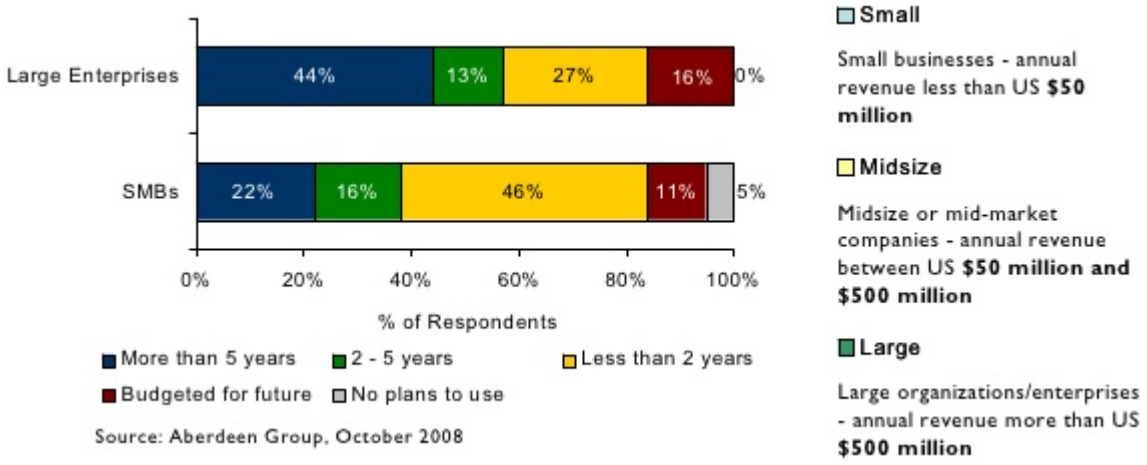

FIGURE 2.8: Age of BI deployment [2]

to the quality of the system obtained. Analogously, there is a division on organisational and technological ones, used, for instance by Isik et al. [30]. Let us consider success factors from these two perspectives.

2.9.1 Process perspective factors

Ease of use

According to Imhoff [21], "assuming the project is finished" is one of the common pitfalls in BI adoption. It always takes time for the solution to start working and for the end users to get used to it. Therefore, the factor of how long would it take the typical manager, customer or business analyst to learn how to use the solution will measure the business side of *Ease of use* criteria [19]. Technically it is defined by the amount and regularity of maintenance activities of BI solution.

User access

Isik et al. [30] defines this factor as an ability of BI solution to provide a differentiated access to functional areas of analytics for different types of users. The access may not only vary by its type from raw data to its visualised form in dashboards, but also by diverse means of access from desktop to mobile solutions. The more fine-grained user access is, the greater BI success is possible.

Skilled (qualified) sufficient staff/team/managers

The factor of personnel, that defines performance of the BI solution especially in its implementation phase, is also among the top ones in the study by Olszak and Ziemba [23]. Baransel [18] mentions that typically SMEs are understaffed in IT-personnel while

Williams [36] directly says that individuals responsible for implementation of the BI project "are overwhelmed by all of the activities they must do to get started."

A bigger team working on BI project leads to the development of more breadth-wise solution within the same amount of time. In case of a smaller team, the emphasis is made on a depth-wise approach where less mature but working right now solution is preferred to more functional but not integrated one.

2.9.2 Infrastructure perspective factors

Ubiquitous BI system

Integration between BI system and other systems was mentioned by all interviewed SMEs from the study by Olszak and Ziemba [23]. Consequences of absence of complex BI strategy are presented further in this document. Starting a BI campaign it is important to think about complex solution that can be integrated with all pieces of existing system as well as to make sure that all departments see their operation as a part of big ultimate solution. This strategy of total BI involvement guarantees better performance and bigger success of the BI solution.

Data Quality

Data quality, as one of the main success factors for business intelligence systems [37], [38], plays also a big role in case of SME as Olszak and Ziemba [23] shows. The best time and place to conduct data cleaning activities are before starting implementation of BI solution and at operational level. Common misconception regarding this fact is commented by Wise [22]: "Some organizations are misguided and think that their BI solution will provide the tools to fix their data problems." Therefore data quality factor acquires bigger importance as it influences not only business operations directly but also impacts derived analytics and further decision-making processes.

Data Management

The shortage of data and meta-data management can decrease the performance of the final solution. For instance, terminology, not consistently used across departments, may create barriers in understanding of analytics between its creators and consumers.

Chapter 3

Deploying the BI Roadmap in a Real Use Case

According to the BI roadmap, the first steps of BI integration at the considered company were acquaintance with business model and KPIs, identification of current problems and needs and, finally, specification of the requirements for the BI solution.

This information was mainly collected from interviews with heads of departments that are considered for leveraging BI in the future. Besides, the workflow processes of employees from these departments were studied with further matching of these operations on technical internals of the business such as database schemas, existing analytics, etc.

The aforementioned steps are detailed in the subsequent sections of this chapter.

3.1 Acquaintance with business model

The business of the company is based on offering administrative, procedural and legal help to the people who want to claim the airline because of the flight delay. The service is monetized by a success fee which is certain percentage of the final compensation of the customer.

The interaction with the customer happens via the web-site, emails, phone calls. The information system stores and processes the data related to flights, customers, claims, compensations and interactions with lawyers and court.

Departments-wise this looks as follows:

- *Customer Care department* is responsible for interaction with customers via phone and emails.
- *Finance department* monitors and processes all financial data and transactions.
- *Legal department* provides a legal assessment of the case, whether it is eligible for claiming. It is also responsible for handing the case over to the lawyer when necessary.
- *Marketing department* executes typical marketing research and monitoring.
- *Business development department* searches for an opportunities to increase performance of each department as well as the effectiveness of the whole business. It is an initiator of leveraging BI.

3.2 Scope of the solution and strategic goals

Initially it was agreed that the scope of the project will not aim:

- *IT department* develops and maintains technological component of the company. It is not involved in direct BI use as it doesn't monitor any KPI, doesn't require to manage complex ETL process and doesn't predict a necessity for integration of the operational data in the nearest future.
- *HR department* simply deals with 30-35 employees of the company and hires 1-2 persons in quarter. The company's scale doesn't require HR to use BI.
- *Sales department* communicates with few business partners in order to get additional customers. It's only representative utilises SalesForce [39] CRM that covers all current needs. As the company doesn't sell any goods or even a service in its common sense, but generate revenue from successful juridical cases, 'sales' monitoring is executed by other departments.

Therefore strategic goals were identified for the following departments:

- *Customer Care department.*

  The main goal is to improve customer satisfaction in term of communication with the company. In particular it means to reduce the number of not answered phone calls, not answered emails, not processed inbound documents, e.g. all pieces of interaction with customers that need to be processed. It also includes reducing the waiting time of the phone calls.

- *Finance department.*

  The strategic goal is to identify legal cases that produce the biggest revenue in the shortest time in order to target the business for cases with similar characteristics. The other goal, financially, is to track whether the company experiences economic growth or decline by monitoring the money turnover.

- *Legal department.*

  The legal department is directly related to the service the company provides, which means the better it operates, the more successful the business is. Therefore the strategic goal is to minimise the time of legal case processing as well as to increase the percentage of legal cases closed successfully, e.g. those where the claim has been satisfied and customer received a compensation from the airline.

- *Marketing department.*

  The marketing goals of the current business are typical for marketing departments of almost any business - to increase the conversion rate of the web-site visitors as well as to explore new ways of getting customers.

- *Business development department.*

  The goals of this department can be listed as follows:

  - To keep operational data up-to-date in order to support operational processes as well as real-time analysis.
  - To identify completeness and consistency problems of the data.
  - To reduce the complexity of the current analytics as well as to make it more available for not IT-users, shifting it towards self-service BI.

3.2.1 KPIs

In principal every dashboard could have been associated with certain KPI and this part of work could have been done. However there were around hundred of dashboards and it was possible to systematise them only after having held the aforementioned interviews with heads of departments. During the interviews, the people were asked to point which dashboards they use, which of them can be considered as KPI and how they interpret the values from these dashboards. They were also asked to prioritise them if possible.

At first the interviews revealed some at least interesting if not unexpected results:

- Several dashboards were not mentioned by anybody.
- The meaning of some of them was not clear at all or to some extent.

Second, people were not eager to provide a priority for dashboards and the first answer was that all of them are important. At the end it was possible to obtain two levels of priority:

- 'must-have' - important ones, necessary for daily operations
- 'useful' - the ones used from time to time.

Group	N of dash-boards	Used by departments	Description
Emails	5	CC, Legal	Statistics about the number of unanswered emails and their age (period of time since receiving).
Documents	4	CC, Legal	Same statistics as for emails, but regarding unprocessed documents.
Calls	7	CC	Statistics of the customer phone calls such as N and % of (un)answered calls, N of calls outside service hours, average and total waiting/speaking/ holding time, etc.
Process	29	BD, Legal	Statistics relevant to the business workflow such as distribution of legal cases in certain statuses or process steps, timely analysis of coming from one step to another, etc.
Finance	16	Finance, BD	All financial measures like sum of payments by airlines, N of cases to be paid, average claim amount, % of claims enforced(those received compensation), etc.
Marketing	17	Marketing, Finance	Marketing indicators either based on the data from Google Analytics or operational data such as frequency of new legal cases openings or legal cases distribution by promotion partner.

TABLE 3.1: Main dashboards groups

Another part of result was that KPIs were grouped by their semantics. These grouping became a starting point in designing a snowflake-schema as a first step of the implementation of the BI solution. The main six groups are listed in the table 3.1 (*CC*- customer care, *BD* - business development).

Though it may be noticed that dashboards from the same group are used by several departments, the vast majority of the dashboards were used by a single department and therefore it was not possible to outline clear leaders and outsiders. This concluded that most of the KPIs should be considered while designing the BI solution.

3.3 Current problems and new analytical perspectives

During the interviews with heads of departments current problems and needs were revealed. Let us consider them by separately by each department.

- *Customer Care department.*

  The main request was connected to providing more diversity to the existing dashboards. For phone calls, for instance, measures of *sum* and *count* in addition to *average* and *roll up* by month in addition to *drill down* by days were requested. Similar statistics was requested for emails age and period of answering.

  As for the problems, the main one was related to the lack of easy way to manage phone calls statistics. As it is described in 3.4.2 the data from Top-Link was received via emails in bad visual format and required some workaround efforts from IT-department to format it in Excel, then transfer to Google drive in order to be later displayed on the dashboard.

- *Finance department.*

  The current needs were also connected to receiving more detailed statistics related to already existing in current dashboards. For instance it was desired to introduce more dimensions for analysis such as airline, country, advocate, user (operator from company's side).

  The problem of finance department was the fact that some useful KPIs and statistics were not presented by dashboards or were presented not in visual-friendly way.

- *Legal department.*

  The requirements from legal departments were mostly connected with providing additional types of analysis. For instance process-related analysis showing how long it takes for airlines to pay, percentage of claim amounts paid by airlines, distribution of legal cases by statuses and age and some other.

  The lack of statistics advocate-wise or airline-wise as above prevented the department to optimise its workflow. For instance, knowing that particular airline doesn't pay compensation by request in 99% of cases, the case could be handed over to the advocate immediately without default waiting for two weeks.

- *Marketing.*

  Talking about the problems of marketing it requires to start with the reason why most of the metrics from Google Analytics were monitored with Klipfolio rather

than by internal means of the service. The point is that Gogle Analytics provide overwhelming variety of metrics with great level of detalisation which that makes it easy to overlook some global trend. An interesting fact is that Google indirectly admits this statement by providing "Intelligence events" module. This module tracks which measures have underwent significant changes so that the user could focus its attention on the main trends. In addition the visualisation capabilities of the service are pretty limited comparing to BI tools for dashboarding.

Besides, the main marketing challenge is also related to Google Analytics. At the moment there is no connection between the analysis provided by Google Analytics and the back-end data, e.g. it's possible to monitor user's behaviour on the web-site and user's actions reflected in the operational database but these analyses are not bounded, the user from web is not mapped to the user from back-end. Such kind of mapping could bring new insights for marketing strategy.

- *Business development department.*

  The needs of this department are connected with enhancements in the performance analysis, case-specific statistics and so-called case portfolio.

  The first one is about measuring percentage of claims enforced, distribution of court decisions or the reason of unsuccessful case closures, the timestamps between various case processing steps, etc.

  Case-specific statistics includes distribution of cases by the delay timestamps, delay reason, airline, airport, etc.

  Case-portfolio is the quantitative distribution of cases by their processing steps.

  The main problem of the department is the current complexity of building any type of analysis which requires writing complex SQL queries.

3.4 Business from technical point of view

3.4.1 Identification of data sources

The following types of data have been identified:

- *Legal case data, customer's data, emails body, advocate-related data, finance data.*

  Partly produced by web-application, partly filled in manually by customers as well as by operators on company's side, it is stored in DBMS. In the first half of the project MySQL was used as a working DBMS engine. It was replaced with PostgreSQL because of the better performance of the latter.

- *Data about airlines, flights and airports.*

  This data is received from OAG flights data provider [40] and is stored in DBMS as a local copy.

- *Statistical data about customer calls.*

  The data is received and stored in form of emails from telecommunication provider Top-Link [41].

- *Analytical data.*

  Most of the marketing-related data is not stored but obtained from Google Analytics. Other analytical data is produced by analytical SQL-queries. A very small part of it is stored in Google Drive and historicised reports, the rest is not stored but generated with each dashboard refresh.

- *Electronic copies of the customers' documents.*

  The scanned documents are received from IAB provider [42] and are stored on the FTP server.

Except for the last one, all of them were required by analytics.

3.4.2 Characteristics of data sources and their usage

Some specific details regarding accessing and processing data are documented below.

The data stored in the DBMS is accessed by the web-application and by the analytical queries. The data from Top-link being received in emails was manually copied into Excel, underwent several manipulations and was stored in Google Drive as a data source for some Klipfolio dashboards.

Some data-sources were manual accessed by the employees of the legal department. For example the weather at the airports was checked at the WU portal [43] which provides a free access to the historical weather data.

Regarding data organisation in DBMS, all application back-end data is store in one database. The local copy of OAG is stored in another database.

The system architecture is based on a single-node solution with the respective mirroring. At the moment the system does not face any concurrent access issues. The first reason is that the web application accesses the production server (master) while analytical queries are run on its replica (slave). The second reason is the scale of the business. The number of users (the order of magnitude is thousands per month) is not so high to generate the

traffic to overload the server. The user interaction with application is very limited – only registration of the case and editing the data.

PaaS cloud solutions from the Inter.net [44] and Domain factory [45] providers are used to maintain DBMS.

At the moment there is only one regular update of the local copy of OAG DB which happens daily.

There are several views in the database used for analytical reasons. None of them are materialised, but this does not influence the performance due to the low analytical workload.

Regarding the time-span, all the data staring from the first days of operation is stored in the DB without archiving.The total volume of the databases is around 200GB (61GB OAG database + 133 GB back-end database).

3.4.3 Data integration

Analytical data sources could be briefly listed as

1. PostgreSQL DBMS
2. Top-Link emails
3. Google Analytics
4. Resulting datasets of daily-run custom SQL-queries manually copied in Google Drive

First, it was decided that data from emails can be integrated into DBMS later as a separate project by company's own efforts. Therefore BI solution should be developed considering this data presents in the DBMS. Second, lack of the timestamps for some metrics remained the necessity to run custom SQL-queries daily. It was proposed to historicise their results in a special table in DBMS. That was later implemented as a part of BI solution, replacing manual copy-paste of this data with automatic append to the *Historical_Data* table.

Thus, the set of data sources was reduced to two: PostgreSQL DBMS and Google Analytics.

3.4.4 KPIs

At the moment when current project was started almost all business metrics were monitored by means of dashboards created with BI tool Klipfolio [46]. Regarding the data sources:

- The vast majority (around 90%) of dashboards had datasets from operational database as data sources.
- Around 8% were connected to Google Analytics platform, and built based on the data from Google Analytics.
- Few metrics were monitored directly via Google Analytics web-interface.
- 1-2% of metrics used data from Google Drive that was a manual copy of the resulting datasets from daily-run custom SQL-queries.

3.4.5 Current problems

In general there was a huge problem with managing and updating dashboards caused by several reasons:

- In Klipfolio, almost all dashboards were build using its own predefined data source. Only few of the data sources were shared. From time to time, the configuration, in particular IP address of the server, changed what required to reconfigure all data sources for all dashboards.
- All the dashboards, except those based on the Google Analytics data, were based on SQL statements querying operational database. Each time some attribute of the table was renamed, deleted or moved to another table, dashboards' underlying SQL queries referring to this attribute had to be adapted to this change.
- As far as there was no centralised BI politics, IT-team functioned not concerning of any BI solution like dashboards and changed database schema without any notice or warning what brought the problem from the bullet above.
- There were several metrics that could not be obtained for the not current day. For instance 'Number of unanswered emails' was calculated using boolean value of the respective table attribute stating that email is either *answered* or *unanswered*. There was no timestamp associated with this attribute, thus it was not possible to get the date when the attribute's value was changed. Therefore it was obtained

daily by means of SQL-query and stored in Google Drive. The SQL-query was run manually and it produced missing data if it was not run at some day for some reason. Such kind of metrics produced that only part of analytical data referred in subsection 3.4.1 that was stored.

These obstacles influenced that almost permanently some of the dashboards were out-of-date.

3.5 Specification of the BI requirements

Technical requirements were summarised in a following way. Mostly desirable from the BI software functionality was providing ad-hoc analysis, followed by powerful dashboarding and reporting capabilities. The support of the PostgreSQL and Google Analytics as data sources was also required. As far as data integration issue was not huge, it was defined to be solved by means of the company, therefore no ETL functionality was required.

Currently used and required for further analytical measures and dimensions were identified in sections 3.3, 3.2.1.

The last part of the interviews was dedicated to the desired rhythm, depth and timeframe of analytics. The answers didn't vary too much across the departments, so the summarised version is provided below:

3.5.1 Rhythm of analytics

For KPIs connected with time dimension, the rhythm of update equal to the granularity of the time dimension was sufficient. It means that statistics per month could be updated monthly, etc. KPIs not connected with time dimension were desired to be updated daily. Few KPIs, mostly those connected with the work being performed at the current moment such as number of documents to be processed, were required to have real-time update. Due to the nature of financial operations most of the finance-related KPIs were required to be updated monthly.

Summarising, daily update was sufficient for vast majority of the metrics.

3.5.2 Depth of analysis

For bigger part of the dashboards the users were satisfied with the current level of granularity. As it was mentioned in 3.3 for some KPIs additional drill down or roll up was required.

As for dimensions, at least the following ones were required: advocate, airline, country, case status, promotion code, operator, delay reason, time. In marketing, specifically marketing channel, keywords and device.

3.5.3 Time-frame

The analysis was required to be done based on data within the following time frame: between 6 and 12 months.

3.6 Identification of the BI solution

Before starting the actual developing of BI solution, a revision of problems and needs identified in the previous step was conducted. As a result the high-level requirements and main issues to be addressed were summarised as follows:

- to find a suitable BI tool able to provide
  - ad-hoc analysis
  - powerful dashboards
  - manageable reporting
- to reduce the complexity of current analytics
- to establish self-service BI as much as possible

3.6.1 Business requirements

As it was mentioned in 3.3 the current state of analytics experienced problems connected with changes in operational database schema, consistency of the data sources for dashboards, necessity to manually run some SQL-queries and several others.

Recalling strategic goals from 3.2, one of them was to enable not IT-people (at least heads of departments) to be active users of analytics, i.e. to execute ad-hoc analysis

as well as to create dashboards themselves. The main reason why self-service BI could not be established at that time was the complexity of the analytical model. It can be highlighted as follows:

- Operational database totalled 129 tables and 45 views that were referred by underlying SQL-queries of the dashboards. Some tables revealed the source of complexity themselves as they consisted of dozens of attributes. Modifying existing or building new dashboard required deep knowledge and understanding of data model: how business data is stored in database (which tables, which data-types), what is the schema (how the tables are related).
- Many SQL expressions either contained up to 10 joins of tables or counted tens of lines of code containing embedded MySQL functions or both. This fact required deep knowledge of SQL language and its MySQL specification in addition.
- SQL expressions ubiquitously contained references to various values (usually statuses or entity types) via their IDs or abbreviations, references to conditional expressions with NULLs or empty values and other syntactical nuances. That was making SQL expressions not human-readable, and thus not understandable for not IT-users.
- Combinations of attributes were used as calculated fields especially in financial metrics. At the same time many attributes were never used in analytics and just produced redundancy that complicated the matter.

To summarise, business requirements included providing end-users with some instrument to execute ad-hoc analysis build dashboards and create reports. In general, it is requirement to make analytics more available for business users.

3.6.2 Technical requirements

Bearing in mind the findings about a lie of the land from sections 3.1 - 3.5, it was confirmed from the company's side that there is no need for BI tool to provide data warehousing features such as ETL functionality. Therefore the role of the BI tool in the final solution was determined as an instrument of visualisation for analytics by means of dashboards, reports and ad-hoc analysis. Naturally, these three components should be not worse than that of Klipfolio, BI tool in use of that time. Support of PostgreSQL and Google Analytics as data sources was a must-have characteristics for the BI tool.

3.6.3 IT-based solution vs Self-service BI

Since the beginning it was desired to establish self-service BI as much as possible.

3.6.4 Software costs vs IT-specialist costs

Naturally, the ultimate goal of the company was to spend minimum money on software. However, later on it was practically demonstrated that powerful software can bring real benefits and return its investments.

3.6.5 Role and place for OLAP

Because of the complexity of analytics issue detailed in section 3.6.1, it was decided to design OLAP schema upon operational database and then to evaluate BI tools with produced OLAP schema as a data source. Operational database was quite denormalised and required quite sophisticated extraction of data and organising it into star-schema. Besides, there was no requirement of ETL functionality for data integration. Therefore it was decided to create OLAP schema with a custom PL\SQL script.

3.6.6 Embedded BI

Minimum second level of engagement, according to [28] was required. Users needed to share interactive reports and dashboards that could be regularly updated.

3.6.7 Structuring BI solution

A certain BI tool was supposed to provide visualisation options for dashboards, reports and ad-hoc analysis with price to quality ratio acceptable by the company. It was also noticed that problems of complexity of analytics will most probably not be solved by any BI tool or at least will require a lot of work in data pre-processing. Instead, it was proposed to create OLAP star or snowflake schema so that further analytics would be established on top of it by means of the BI tool chosen. In fact of it, it was reasonable to evaluate BI tool after creating OLAP schema. This approach could guarantee the desired outcome of the final BI solution.

Therefore it was decided that the problem of the complexity of the analytics should be resolved before the problem of dashboarding, reporting and ad-hoc analysis.

3.7 BI solution development. OLAP schema

The design of the OLAP schema has come through the conceptual, logical and physical steps of data modelling defined in 1.2.1. Although sometimes these design phases do not have unambiguous borders and overflow one into each, for better understandability, they are described in separate sections below.

3.7.1 Conceptual design

Here the identified facts and dimensions are listed without specifying particular attributes of their respective fact and dimension tables what will be done in the Logical design section.

3.7.1.1 Dimensions

The identification of dimensions was done by analysing the gathered requirements as well as currently used KPIs. The OLAP schema included both - dimensions from required analysis and currently used dimensions. They are considered one by one below.

- *Date/Time* dimension probably presents in all OLAP schemas. As far as all DBMS and other BI tools provide variety of features date/time-oriented, at least functionality to extract year, month, day, hour, etc., it was decided, for simplicity, not to create respective tables for date/time dimensions, but store date/time attributes directly in fact tables. However, to demonstrate all possible types of analysis, in further pictures, date and time dimensions are shown as separate tables, connected by foreign keys.
- *User* dimension represents the person on the company's side who performs any type of business workflow operations whether it is interaction with customer or setting some status to the case.
- *Caller* dimension is used to distinguish between people calling to Customer Care department. It includes customers as well as other inquiring people who can be considered as potential customers.
- *Customer* dimension directly represents customers whose legal cases are being processed.
- *Country* dimension indicates the country of the business entity such as customer or airport.

- *Case* dimension is the central and most important one. It represents the main business entity - legal case. Formally speaking, this business entity could also be classified as a fact because it encompasses several attributes which can be classified as measures, as you will see later in logical design section. However it was decided to consider legal case as a dimension for two main reasons. First, from business point of view legal case represents single real-world entity. Second, in most analyses it is treated as a dimension. At the end, the correctness of this classification is determined by a simple comparison: which choice will lead to more effective BI solution and thus will be more beneficial for business. From now on, for convenience legal case will be referred as 'case' without adjective 'legal'.
- *Case status* dimension reflects a variety of statuses that can be set for a case.
- *Advocate* dimension determines an advocate which is involved into the claiming process of particular case.
- *Airport* dimension title speaks for itself. It is used for referring both departure and arrival airport.
- *Airline* dimension can be utilised for both marketing carrier (airline that sold the ticket) and operating carrier (airline that operates the flight).
- *Delay reason* dimension represents hierarchical structure of reasons of flight delays.

3.7.1.2 Facts

Recalling the table 3.1, it was mentioned that KPI groups were a starting point in developing entities that reveal some facts and thus the bases of fact tables. Except Marketing group where most of KPIs were related to Google Analytics, each group can be associated with at least one fact table. Let us consider the facts first (after the name of the fact, the name of underlying KPI group is provided in brackets).

- *Calls* (calls) element provides statistics for each phone call to the Customer Care department (recall TopLink data source from 3.4.1).
- *Documents* (documents) element represents time and processing details of each sent/received document.
- *Emails* (emails) includes almost the same information as Documents fact with respect to the emails.

- *Case History* (process) fact defines the time and type of the operations that was applied to the case. It is expressed by storing all statuses that the case ever had including an indication of the current status of the case.
- *Advocate Authorization* (process) fact happens when the customer authorises an advocate to represent him in the court. This additional jurisdictional assistance from the advocate is required in about 60% of the cases.

Before providing the definition of following two fact it is worth to say that the financial part of the business workflow was the most difficult one to understand. Though financial model of the business is not the central part of this thesis, providing its three main peculiarities will highlight the problems and tasks of analytics related to this business area.

1. The payments for claim can be received by the company, advocate or customer directly.
2. Each case can receive claim money in one or several payments.
3. Airline can pay separately for each case or in one transaction for a number of cases.
4. Success fee is either deducted from the amount received by advocate/company or requested from the customer in case he received the money directly.

With this in mind let us consider the three consequent facts.

- *Case Payments* (finance) fact represents payments assigned to the case differentiating by recepient.
- *Case Accounting* (finance) fact is related to the incoming payment with differentiation by its type whether it is amount to be paid to the customer, success fee, advocate fee, court fee, interest, taxes, etc.
- *Claim* (finance) fact depicts the details of claim and compensation.
- *Problem Flights* (process) fact represents particular flight that had delay, cancellation or other claimable problem.
- *Historical Data* (calls, documents, emails, finance). As it was mentioned in 3.4.1 and 3.3, some data without timestamps was calculated and stored in Google Drive on a daily basis. Historical data fact contains the values of all such metrics as well as the date of their calculation.

3.7.2 Logical design

As soon as the dimensions and facts have been identified at the conceptual level it was the time to define attributes of the respective dimension and fact tables as well as to name them. Again, current types of analysis as well as required one were taken into account. Already mentioned methodologies by Golfarelli et al. [11] and by Kimball and Ross [29] consider creating multi-dimensional model from scratch. However, existing KPIs presented in dashboards were based on analytical queries that could already provide the insights on formation of dimensions and facts, as explained in 2.7. For each KPI, these insights were summarised in the table 3.2 with one entry as an example.

KPI Name	KPI group	Description	Operational tables used	Prospective facts	Prospective dimensions
Advocate commissioning	Process	Monthly N of cases by agents where commissioning started after "2013-09-01"	legal_case_history	Case History	Case, User, Case Status, Date/Time
...	...	...	...	...	...

TABLE 3.2: Insights on multi-dimensional model from SQL-queries for KPIs

Each attribute can be classified if it is a native one (copied from operational table without modifications) or a calculated field (it value is based on values of several attributes from the same or different tables). For clarity reasons, calculated fields are marked with CF abbreviation in brackets after the attribute name. Below each table is considered one by one with the naming scheme as it exists in the system.

3.7.2.1 Dimension tables

Dim_Date/Time

As it was mentioned before, the tables for date/time dimensions displayed on logical were not created physically. All date/time related analytics is leveraged by embedded functionality of BI tools.

Dim_User

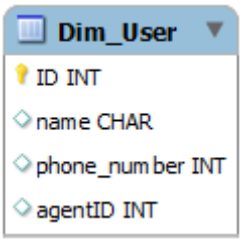

FIGURE 3.1: Table Dim_User

- *ID* - identificator.
- *name* (CF) - the employee's first and last names.
- *phone_number* - employee's working phone number. Applicable only for employees from Customer Care department.
- *agentID* - identificator of the employee responding to phone call. AgentID is provided by TopLink along with statistics about phone calls. Applicable only for employees from Customer Care department.

Dim_Caller

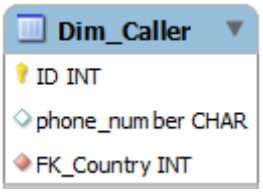

FIGURE 3.2: Table Dim_Caller

- *ID* - identificator.
- *phone_number* - phone number of the calling person.
- *FK_country* (CF) - country of the phone number.

Dim_Customer

- *ID* - identificator.
- *FK_country* (CF) - customer's country of residence.
- *gender* - customer's gender.

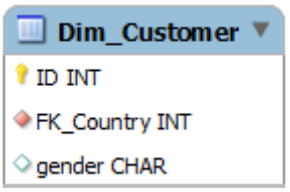

FIGURE 3.3: Table Dim_Customer

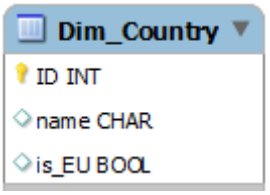

FIGURE 3.4: Table Dim_Country

Dim_Country

- *ID* - identificator.
- *name* - name of the country
- *is_EU* - flag, whether the country belongs to EU or not.

Dim_Case

- *ID* - identificator.
- *date_created_at* - the timestamp when the case was created.
- *handling_status* (CF) - for not closed cases shows whether the case is handled by the company or advocate.
- *date_closed_at* - the timestamp when the case was closed
- *where_known* - the source of customer's knowledge about the company's service. It is obtained from the respective field of the user interface.
- *promotion_code* (CF) - the code of the promotion partner, for instance the website selling flight tickets.
- *AGB_version* - version of the terms and conditions agreed upon by customer. AGB - abbreviation of terms and conditions in German language.
- *unsuccessful_reason* - juridical reason of unsuccess.
- *FK_customer* - customer of the case, the person which is entitled to receive compensation in case of success.

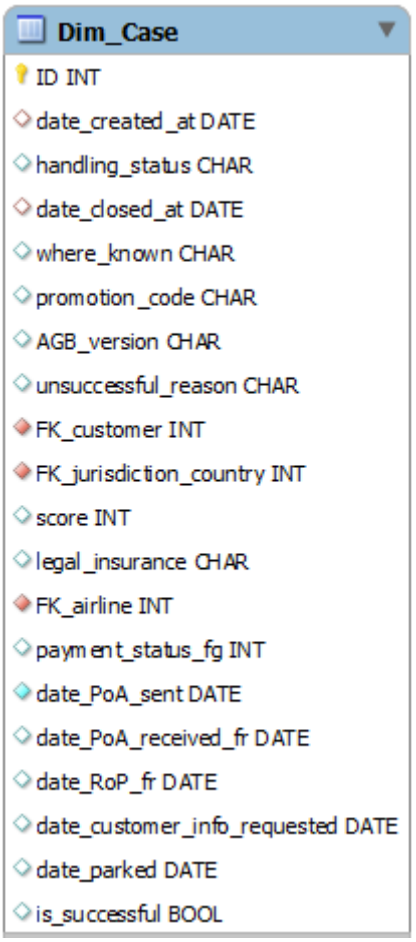

FIGURE 3.5: Table Dim_Case

- *FK_jurisdiction_country* - the country in which legal field the compensation is being claimed.
- *score* - the measure of probability of the success of the case from the the legal perspective.
- *legal_insurance* - the insurance against flight delays or cancellation if purchased in addition to the flight ticket.
- *FK_airline* - the airline which is being claimed. Not necessarily the same as operating airline.
- *payment_status_fg* - indicator whether the company owes the compensation money to customer or the customer owes the success fee to the company or nobody owes nothing.
- *date_PoA_sent* (CF) - date when PoA was sent to the customer.
- *date_PoA_received_FR* (CF) - date when PoA was received by the company

- *date_RoP_FR* (CF) - date when request of payment was sent by the company to the airline.
- *date_customer_info _requested* (CF) - date when some information like boarding pass was requested from the customer.
- *date_parked* (CF) - date when the case was parked (postponed).
- *is_successful* (CF) - flag, whether the case was successful or not.

Dim_Case_Status

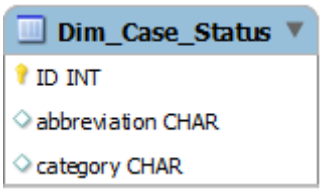

FIGURE 3.6: Table Dim_Case_Status

- *ID* - identificator.
- *abbreviation* - abbreviation of the case status used in the company.
- *category* (CF) - category of case status, formed as a generalisation of the above-mentioned abbreviation attribute. For instance, statuses representing successful closure of the case (in the court, by advocate or simply by the company without appealing to advocate and/or court) are joined into 'Closure_Succ' category. Formally speaking this categorisation is nothing more than creating hierarchy in this dimension.

Dim_Advocate

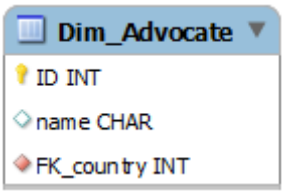

FIGURE 3.7: Table Dim_Advocate

- *ID* - identificator.
- *name* - the name of the advocate that has been authorized by the customer.
- *FK_country* (CF) - the country of the advocate's professional occupation.

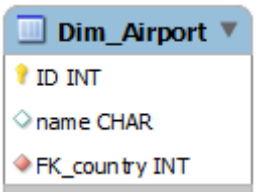

FIGURE 3.8: Table Dim_Airport

Dim_Airport

- *ID* - identificator.
- *name* - name of the airport.
- *FK_country* - the country of the airport.

Dim_Airline

FIGURE 3.9: Table Dim_Airline

- *ID* - identificator.
- *name* - name of the airline.
- *language* - the language used in communications with airline. At the moment only German and English are used.
- *is_EU* - flag, whether the airline is based in EU or not.

Dim_Delay_Reason

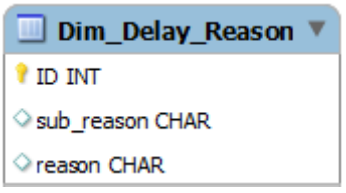

FIGURE 3.10: Table Dim_Delay_Reason

- *ID* - identificator.

- *sub_reason* - detailed reason of delay.
- *reason* - general reason of delay. The delay hierarchy is created here similar to case status dimension. The difference is that the most general level of it was not created from scratch. The source of this dimension table were two tables containing delay reason and sub-reason respectively, connected with one-to-many relation.

3.7.2.2 Fact tables

F_Calls

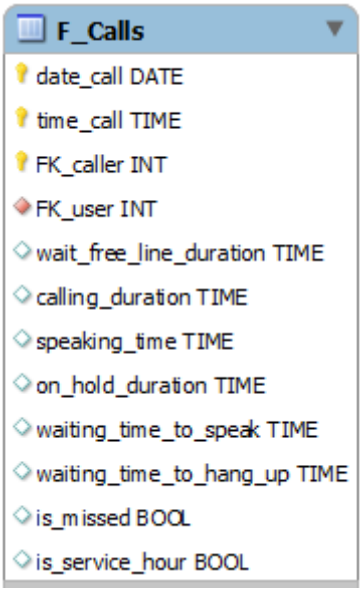

FIGURE 3.11: Table F_Calls

- *date_call* - date of the customer's call.
- *time_call* - time of the customer's call.
- *FK_caller* - the person who called.
- *FK_user* (CF) - the employee who responded the call.
- *wait_free_line_duration* - period of time in waiting for free line.
- *calling_duration* - period of time between the beginning of the phone ringing and time when someone picked up the phone or the caller hung up.
- *speaking_time* - speaking time.
- *on_hold_duration* - the period of time when the caller was put on hold.

- *waiting_time_to_speak* (CF) = wait_free_line_duration + calling_duration. Applied to answered calls.
- *waiting_time_to_hang_up* (CF) = wait_free_line_duration + calling_duration. Applied to missed calls.
- *is_missed* (CF) - flag, whether the call was missed.
- *is_service_hour* (CF) - flag, whether the call was made during service hour.

F_Documents

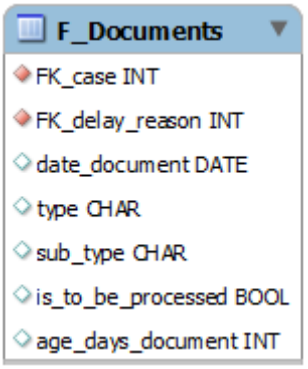

FIGURE 3.12: Table F_Documents

- *FK_case* - the case to which the document is related.
- *FK_delay_reason* - delay reason of the associated case.
- *date_document* (CF) - date when the document was created/received.
- *type* - type of the document.
- *sub_type* - subtype of the document.
- *is_to_be_processed* (CF) - flag, whether the document has been already processed.
- *age_days_document* (CF) - time period in working days since the document was received.

F_Emails

- *FK_case* - the case associated with email.
- *date_received_at* - timestamp when the email was received.
- *is_unanswered* (CF) - flag, whether an email was answered.

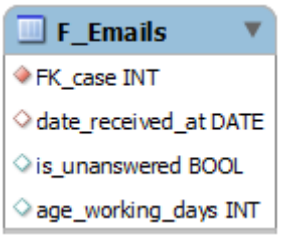

FIGURE 3.13: Table F_Emails

- *age_working_days* (CF) - time period in working days since the email was received.

F_Case_History

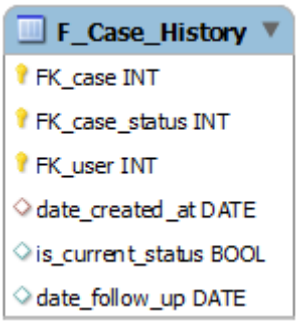

FIGURE 3.14: Table F_Case_History

- *FK_case* - the case.
- *FK_case_status* - the status of the case.
- *FK_user* - the company's employee who set this status for the case.
- *date_created_at* - the timestamps when the status for the case was set.
- *is_current_status* (CF) - flag, whether this status is current.
- *date_follow_up* - date, when the action relevant to the particular status should take place.

F_Advocate_Authorization

- *FK_case* - the case.
- *FK_advocate* - the advocate being authorised by the customer to represent his interests in the court.
- *hiring_status* (CF) - reflection of the hiring process. Briefly, the customer has to authorize advocate electronically via user interface and sign and send respective paper.

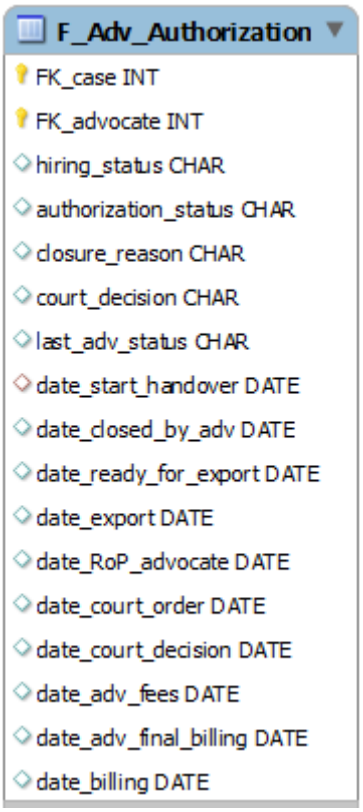

FIGURE 3.15: Table F_Advocate_Authorization

- *authorization_status* (CF) - shows the phase of the case dealing by advocate. After starting authorization, the advocate can be hired, then the case can be exported, then it can be closed by advocate. This process can also be paused. This status shows in which phase of this process the case is.
- *closure_reason* - the reason of closing the case.
- *court_decision* - the decision of the court.
- *last_adv_status* (CF) - last status of the case related to the advocate.
- *date_start_handover* (CF) - date when the process of advocate authorization (handing over to advocate) was started.
- *date_closed_by_adv* (CF) - date when the case was closed by advocate.
- *date_ready_for_export* (CF) - date when the advocate was authorized and the case was ready for being exported to the advocate.
- *date_export* - date when the case was exported to the advocate.
- *date_RoP_advocate* (CF) - date when the request of payment was made by the advocate.

- *date_court_order* (CF) - date when the case was delivered to the court.
- *date_court_decision* (CF) - date of the court decision.
- *date_adv_fees* (CF) - date when the advocate requested the court fees to be paid.
- *date_adv_final_billing* (CF) - date when the advocate sent the final bill.
- *date_billing* (CF) - date when the advocate bill was paid.

F_Case_Payments

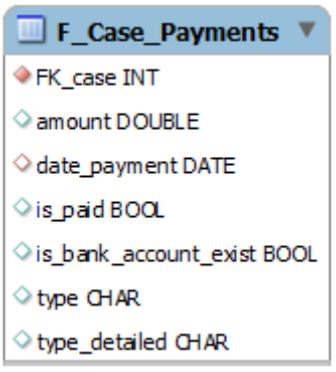

FIGURE 3.16: Table F_Case_Payments

- *FK_case* - the case to which the payment assigned.
- *amount* - the amount of payment.
- *date_payment* (CF) - the date when the payment was assigned to the case.
- *is_paid* (CF) - flag, whether the the payment is done or to be done.
- *is_bank_account_exists* (CF) - flag, whether the customer has provided his bank details.
- *type* (CF) - type of the payment.
- *type_detailed* (CF) - subtype of payment.

F_Case_Accounting

- *FK_case* - the case to which this accounting item belongs.
- *date_accounting* - date of this accounting item.
- *amount* - amount of accounting item.

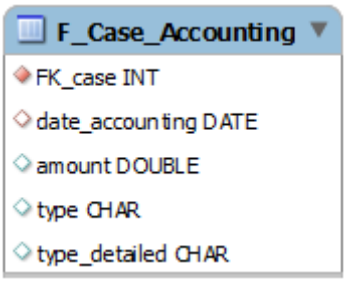

FIGURE 3.17: Table F_Case_Accounting

- *type* (CF) - type of the accounting item.
- *type_detailed* (CF) - subtype of accounting item.

F_Claim

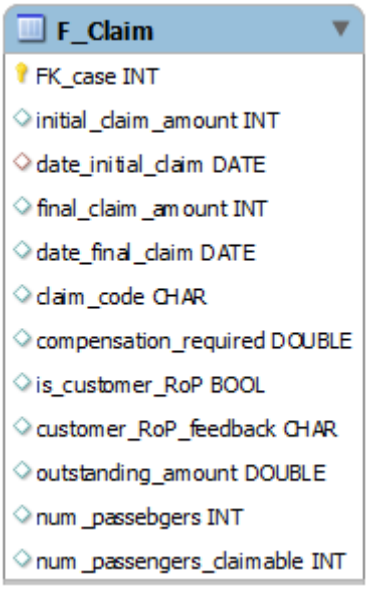

FIGURE 3.18: Table F_Claim

- *FK_case* - the case associated with the claim.
- *initial_claim_amount* (CF) - the amount of claim initiated by customer.
- *date_initial_claim* (CF) - date of initial claim.
- *final_claim_amount* (CF) - the final amount which was requested from airline.
- *date_final_claim* (CF) - the date when the final claim was calculated.
- *claim_code* - type of the claim

- *compensation_required* (CF) - total compensation required. Calculated as claim amount multiplied by the number of claimable passengers minus discounts or any types of vouchers used while buying a ticket.
- *is_customer_RoP* - flag, whether customer requested compensation himself.
- *customer_RoP_feedback* - the response to the customer's request for compensation.
- *outstanding_amount* - the amount left to pay by the airline.
- *num_passengers* - number of passengers of the claim.
- *num_passengers_claimable* - number of passengers that are eligible for compensation (for instance, excluding children in certain age)

F_Problem_Flights

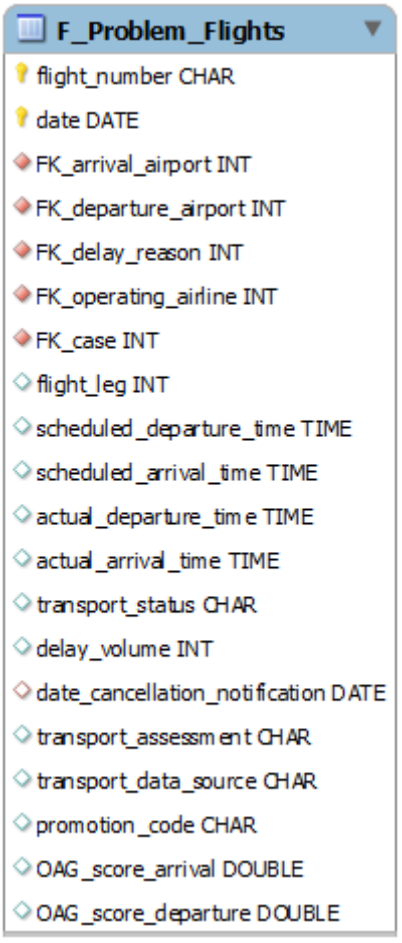

FIGURE 3.19: Table F_Problem_Flights

- *flight_number* - unique flight number.
- *date* - date of the flight.

- *FK_arrival_airport* - arrival airport of problem flight.
- *FK_departure_airport* - departure airport of problem flight.
- *FK_operating_airline* - airline that operates the flight. Not to be mixed up with airline from Case dimension which is a claimable opponent.
- *FK_case* - the case associated.
- *flight_leg* - leg of the flight for indirect flights. For direct flight the value is always equal to 1.
- *scheduled_departure_time*
- *scheduled_arrival_time*
- *actual_departure_time*
- *actual_arrival_time*
- *transport_status* - indication if it was delay, cancellation or other problem.
- *delay_volume* - the flight delay in minutes.
- *FK_delay_reason* - the reason of the delay.
- *date_cancellation_notification* - date when the customer was notified about flight cancellation.
- *transport_assessment* - assessment of the aircraft problem
- *transport_data_source* - source of the information about the problem flight, whether it is taken from the customer or checked in flights database like OAG.
- *promotion_code* (CF) - the code of the promotion partner, for instance the website selling flight tickets.
- *OAG_score_arrival* - airport score from 0 to 1 representing the probability that the flights to arrival airport at given time could have delays or be cancelled, for instance because of the weather.
- *OAG_score_departure* - airport score from 0 to 1 representing the probability that the flights from departure airport at given time could have delays or be cancelled, for instance because of the technical problems in the airport.

F_Historical_Data

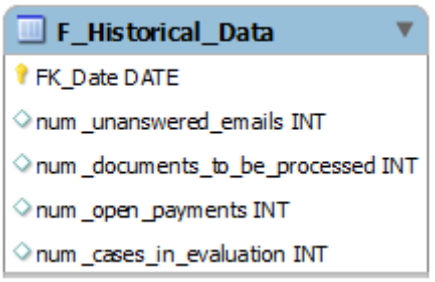

FIGURE 3.20: Table F_Historical_Data

- *date* - date when the measures of this fact table are historicised.
- *num_unanswered_emails* (CF) - number of emails that are not answered at the moment.
- *num_documents_to_be_processed* (CF) - number of incoming documents that are not processed at the moment.
- *num_open_payments* (CF) - number of payments that have to be paid.
- *num_cases_in_evaluation* (CF) - number of cases that are in the evaluation process of legal department.

3.7.3 Physical design

Figures from 3.1 to 3.20 also reveal the data types of the attributes in dimension and fact tables. In most cases the data types match those from operational database except for calculated fields. Primary key and foreign key attributes are marked with yellow keys and filled with red diamonds respectively.

Bearing in mind the complexity issue (recall 3.6.1) the naming notation has also been considered. First of all this reflected in particular values of calculated fields describing type or status. For instance 'HandelingLawyer' value of *handling_status* attribute of Case dimension is calculated using the following expression '*advocateHiredIP is not null and datais Verified=true and handoverpauseddate is null and isAuthorized=true*'. Second, naming notation provides prefixes *date_*, *is_* and *FK_* for attributes of date type(not containing time), boolean attributes and foreign keys respectively.

3.7.4 Enabling OLAP schema

The choice of not using an ETL tool for implementing OLAP schema was done because of the additional overhead in terms of time and effort that would be required for installation

and maintenance activities. There were also no necessity to manage big data. Creating of some of the simple dimensions like Airline was a straightforward selection of the required attributes from operational table. However most of the facts and several dimensions required complex logic for data extraction.

Let us consider the part of the script that populates Case dimension. Its listing is presented in figure 1 in Appendix. Besides the main operational table *legal_case* it contains:

- 2 *case when...then...end* expressions
- *if* expression
- string formatting
- 3 left joins with other tables
- left join with sub-query that contains
  - 2 left joins with other tables
  - grouping with various aggregations
  - 8 *if* expressions
  - 2 *case when...then...end* expressions

Thus, the main efforts were related with expressing such complex logic rather than with establishing the whole ETL process. Therefore, three design phases described above ware realized by a PL/SQL script consisting of two parts and splitted into two files.

The first part of the script creates the OLAP schema from scratch. In particular it deletes and creates tables with all the attributes of respective data types as well as relations between these tables according to their design. This script is supposed to be run each time the schema undergoes any modifications. The scope of the script covers all dimension and fact tables except *F_Historical_Data* which was created once and has never been deleted since then. New attributes can be added to this table, but this is not included in the considered script. The reason of this exception follows from the semantics of this table which holds historical data that can not be reproduced.

The second part of the script complements the first one as it truncates the content of the tables and populates them from scratch. This script is supposed to run on a regular basis as soon as it satisfies the rhythm of analytics. At the moment it runs on an hourly-basis which matches the requirements from 3.5.1 and even satisfies those few metrics which require more frequent than daily updates. In case more frequent than hourly-based

rhythm of analytics is required the separate ad-hoc solution should be provided what is discussed in more details in the final chapter of the thesis. The scope of the script covers all tables with the same exception for *F_Historical_Data*. This table is not truncated but only extended with new data during each run of the script.

Both parts of the script use the principle of cascade dropping and truncating tables as there are foreign key dependencies between the tables. These dependencies as well as consistency considerations require the script to be run as a single transaction. The point is that the script's execution time counts several minutes and during this time new entries could be added to operational database that could damage the integrity of the data.

3.7.5 OLAP schema in action

In order to demonstrate the developed schema in real-life scenario, its parts has been considered in terms of addressing the existing analytics as well as required one. The examples provided below demonstrate are only an excerpt of the analytics that can be established with this OLAP schema.

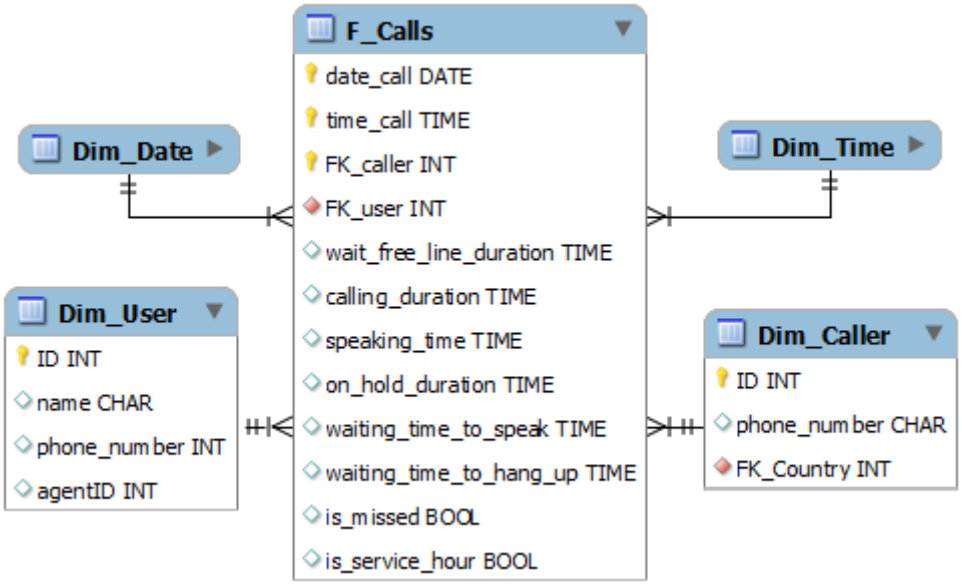

FIGURE 3.21: Calls analysis

Calls. Current analysis

- Phone reachability during service hours.
- Hourly distribution of percentage of answered calls. Daily and monthly statistics.

- Hourly distribution of number of calls outside service hours. Daily and monthly statistics.
- Average phone waiting time per month.

Calls. Required and further analysis

- Total number of answered calls. Daily and monthly statistics.
- Percentage of answered calls for the whole day. Daily and monthly statistics.
- Average phone waiting time per day.

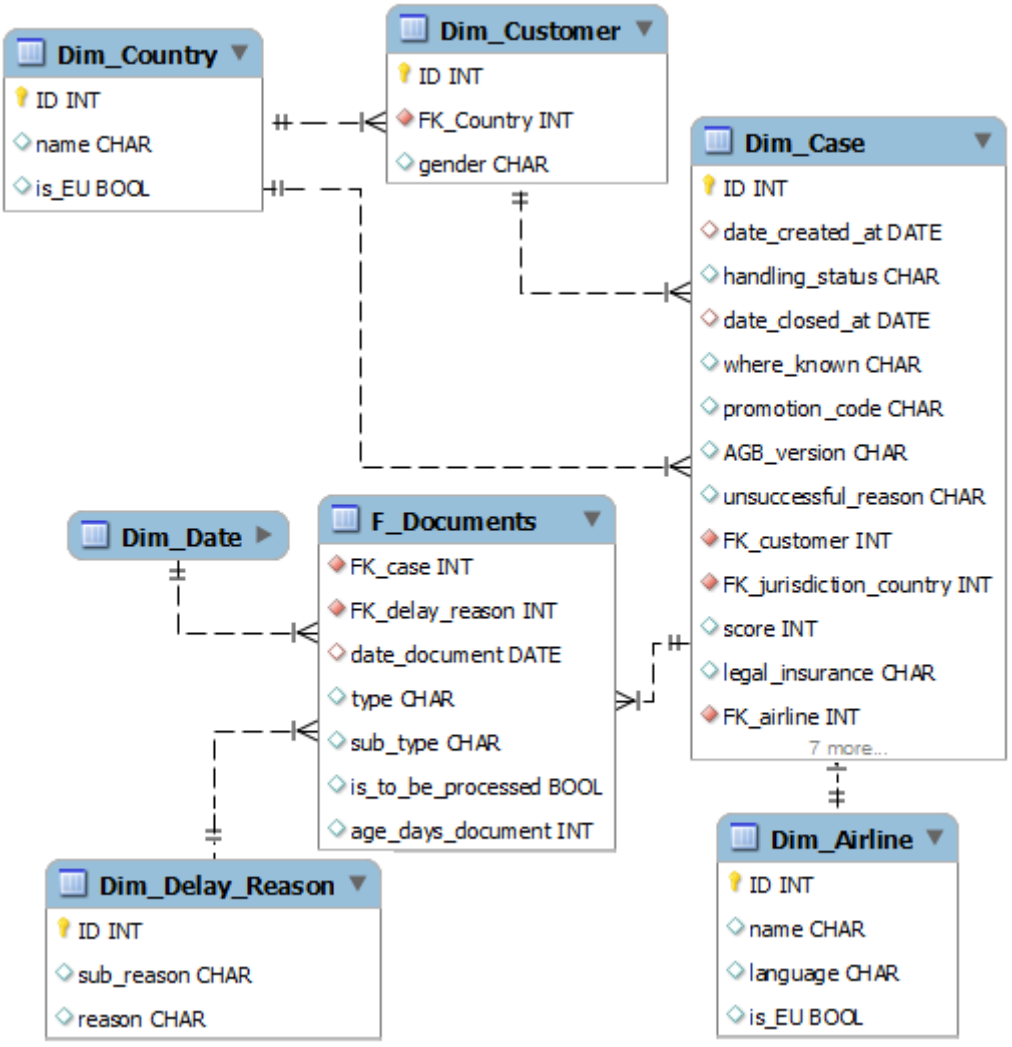

FIGURE 3.22: Documents analysis

Documents. Current analysis

- Documents received from IAB provider (document of particular type). Monthly statistics.

- Documents age distribution of those needed to be processed.
- Number of advocate invoices to be processed.
- Total number of invoices per month.

Documents. Required and further analysis

- Average age of the documents to be processed.
- Number of documents for cases of particular airline, customer, country, customer country, etc.
- Distribution of the documents by delay reason.
- Time statistics for documents of any type.

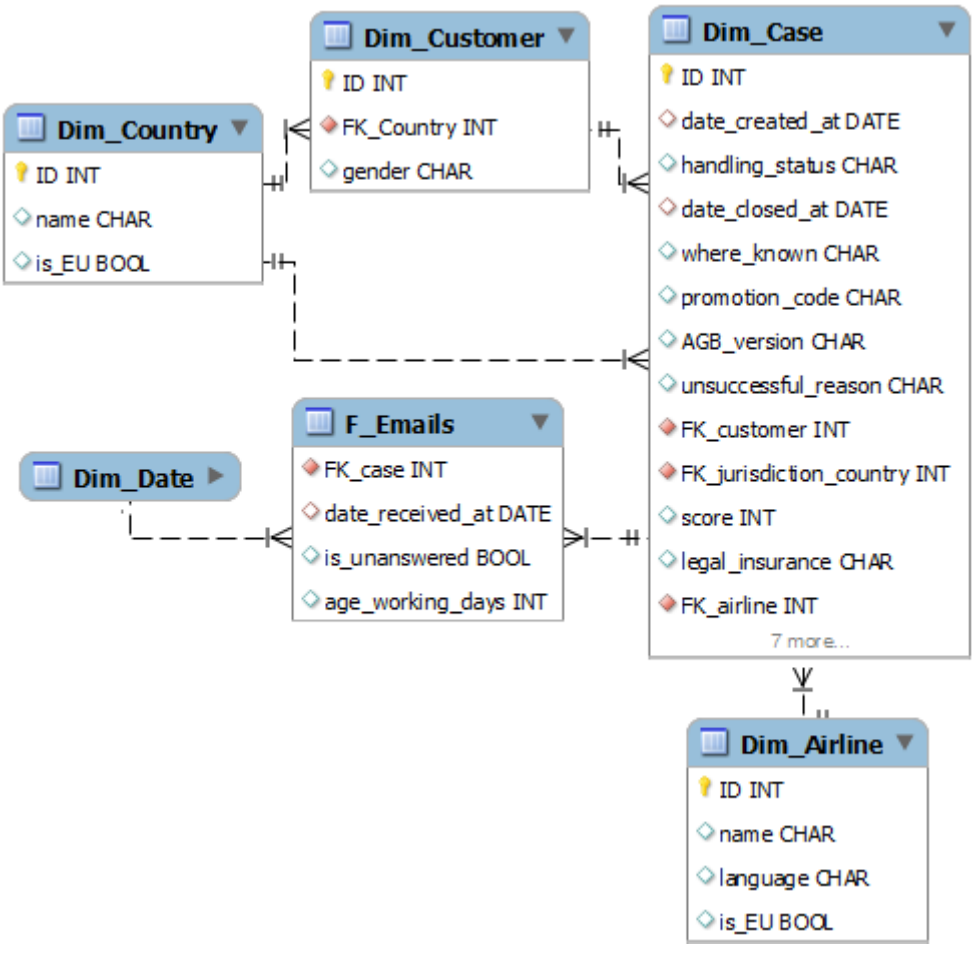

FIGURE 3.23: Emails analysis

Emails. Current analysis

- Number of unanswered emails.
- Age of oldest unanswered email in working days.
- Daily distribution of oldest email.

Emails. Required and further analysis

- Number of documents for cases of particular airline, customer, country, and other case attributes.
- Average age of the email.
- Percentage of unanswered emails per day, week, month, etc.

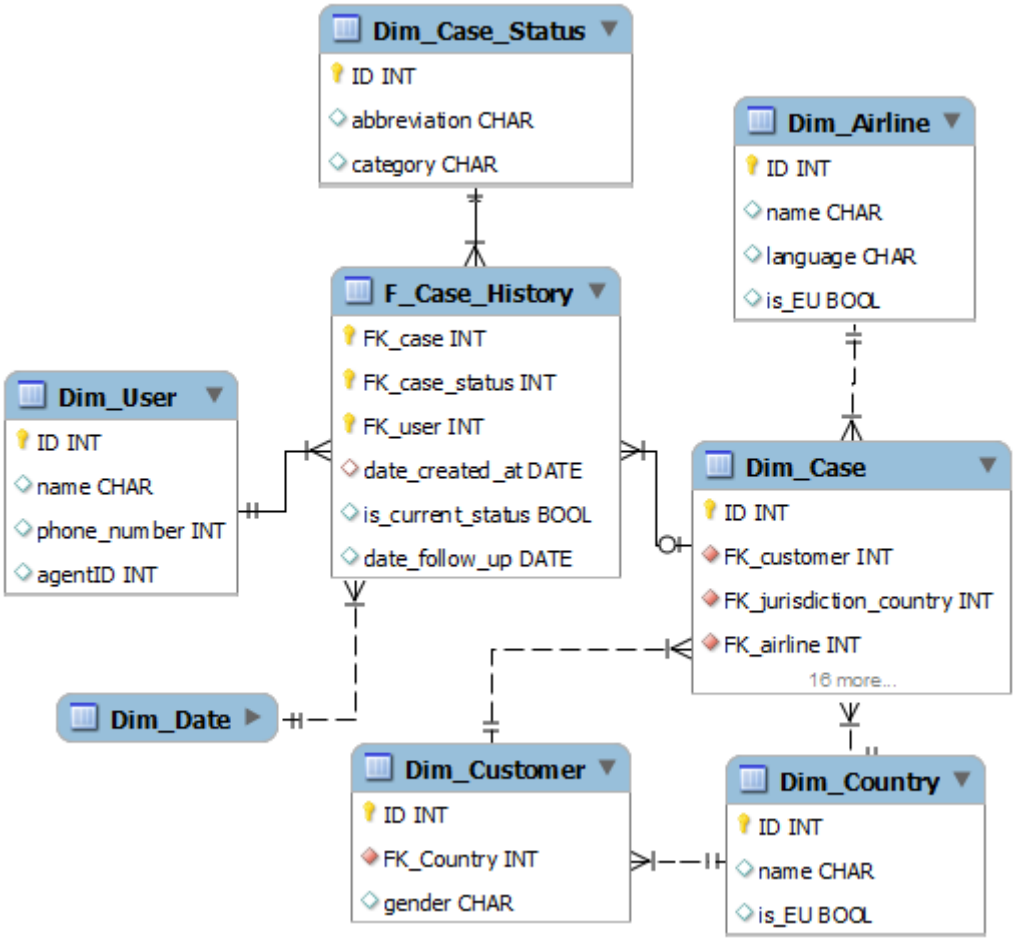

FIGURE 3.24: Case History analysis

Case History. Current analysis

- Number of cases with certain status, in particular:
  - requesting from client (status *category* is one of 'Request_Client_Boarding_pass', 'Request_Client_Booking_confirmation', 'Request_Client_Booking_number'...).
  - parked cases (status *category* is 'Parked').
  - cases delivered to court (status *category* is 'Court_Order_Filed').
  - cases where advocate has sent request of payment to Airline (status *category* is 'RoP_Advocate').
- distribution of cases closure type (distribution of status *abbreviation* for status *category* in 'Closed_by_adv', 'Closed_by_FR').
- user-wise distribution of cases with status *category* 'Starting_commissioning', 'Parked', 'Closed_by_FR' as well as total number of cases.

Case History. Required and further analysis

- Statistics about current statuses of the cases.
- Number of certain statuses set by user(s).
- All kinds of time-based statistics related to statuses.
- The above types of analysis sliced by case dimension(cases of particular airline, customer, country, successful/unsuccessful, etc)

Advocate Authorization. Current analysis

- Number of cases exported to advocate with system name 'MSR'.
- Number or cases closed/not closed by advocate 'MSR'.
- Distribution of *last_adv_status* attribute for cases exported to advocate.
- Cohort analysis of case closure for cases exported to 'MSR' advocate up to 8 months ago.
- Number of cases exported to 'MSR' advocate weekly.
- Distribution of *authorization_status* for 'MSR' and 'KSP' advocates.

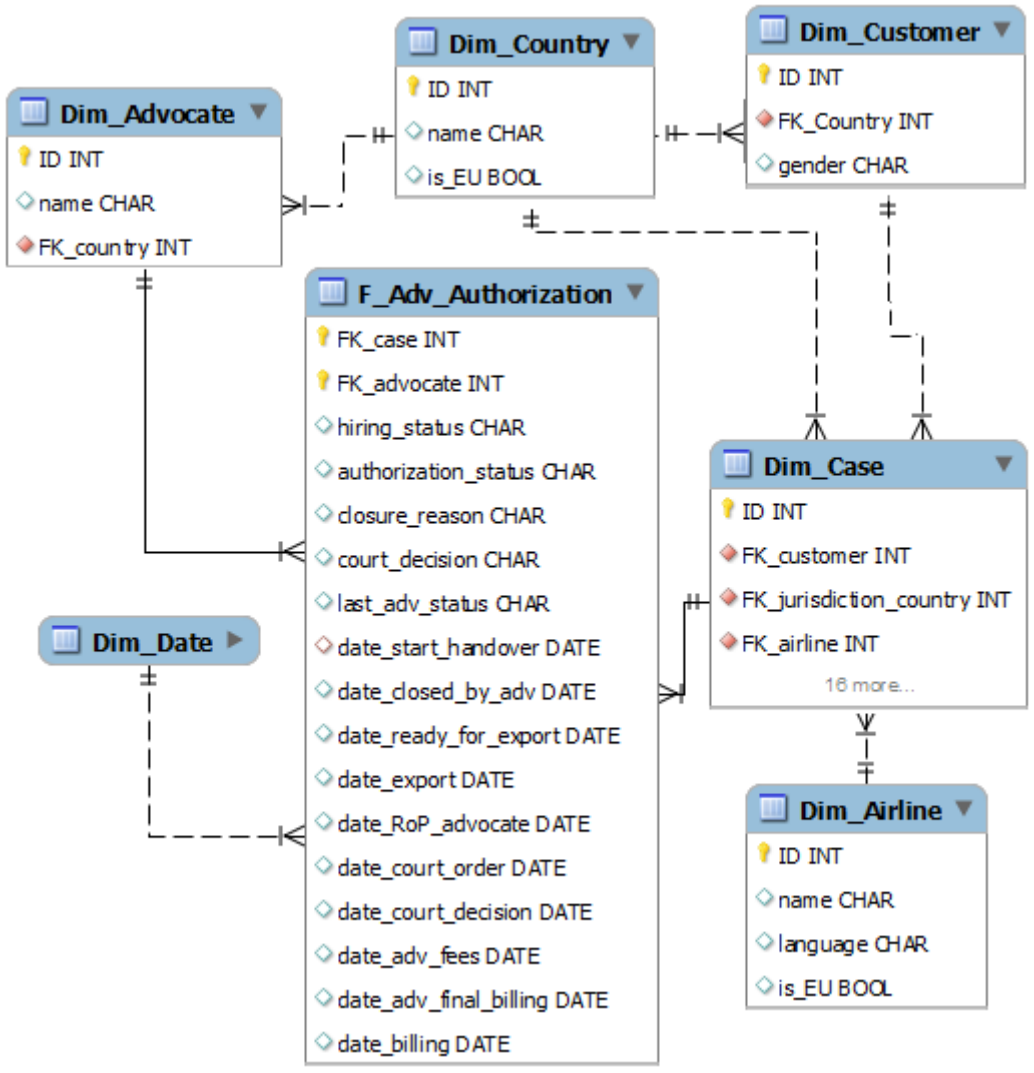

FIGURE 3.25: Advocate Authorization analysis

- Authorization problems analysis (*authorization_status* equals 'Only_Authorized', 'Only_Hired', 'Not_Hired_Not_Authorized') for 'MSR' and 'KSP' advocates.

Advocate Authorization. Required and further analysis

- All the statistics related to 'MSR' advocate is now available for any advocate.
- Cohort analysis of authorization process using attributes with *date_* prefix that represent a timestamp of a particular step in advocate authorization.
- Again, all possible analysis involving Case, Advocate and Date dimensions.

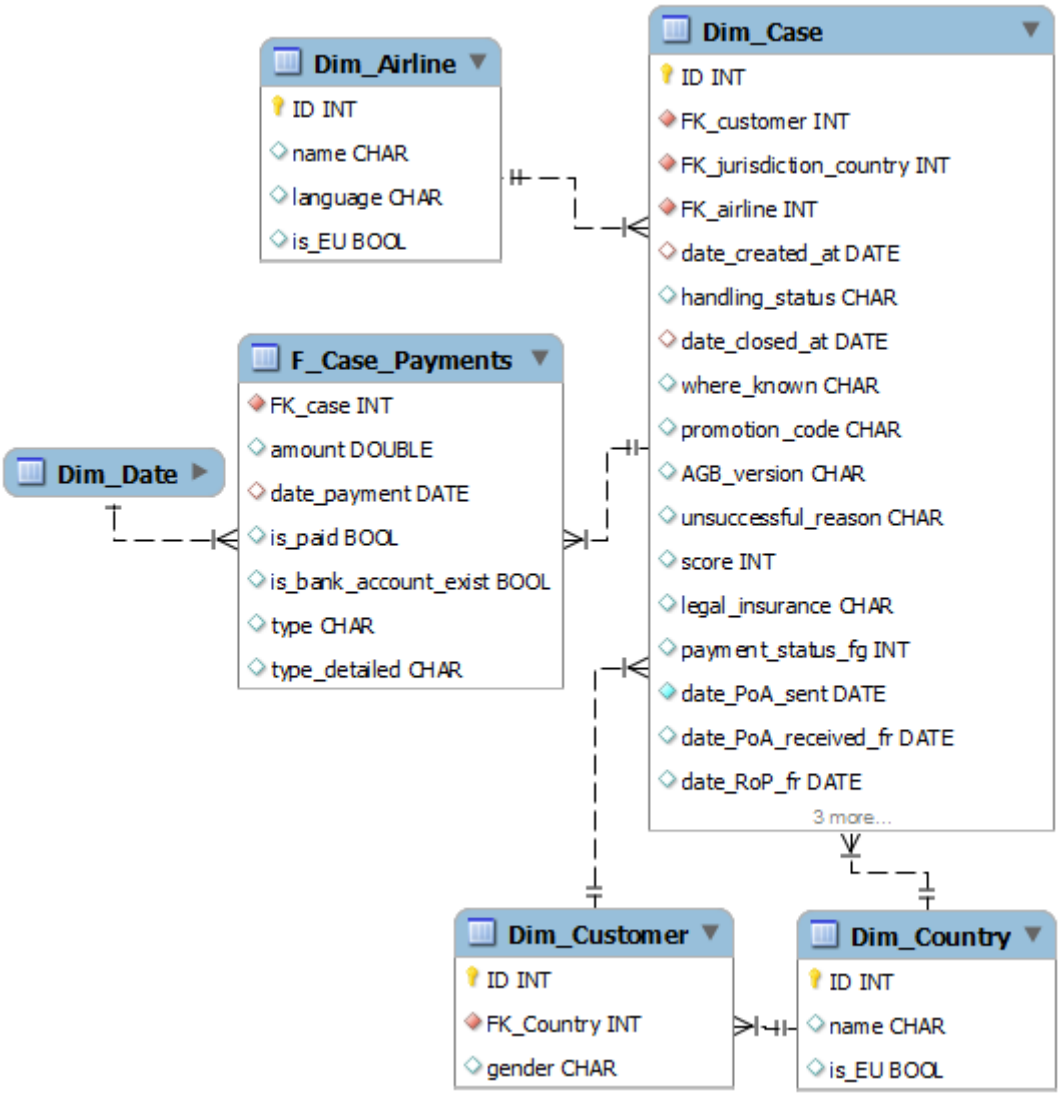

FIGURE 3.26: Case Payments analysis

Case Payments. Current analysis

- Distribution of cases that received some payments by *handling_status* ('Handling_By_FR', 'Handling_By_Lawyer', 'Closed'). Monthly statistics.
- Time delay between request of payment (*date_RoP_FR* in Dim_Case) and date of the payment (*date_payment*)
- Number of cases where the payment should be made (*is_paid* is false).
- Number of cases to be paid with/without bank account details (*is_bank_account_exists*).
- Percentage of claims that enforced payments per month.
- Amount of payments enforced in last 90 days.

Case Payments. Required and further analysis

- All possible analysis involving Case, Date Airline and Customer dimensions.
- Statistical measures of payments such as maximum, minimum, average diaggregating by the connected dimensions.
- Distribution of payments by their *type* and *type_detailed* attributes.

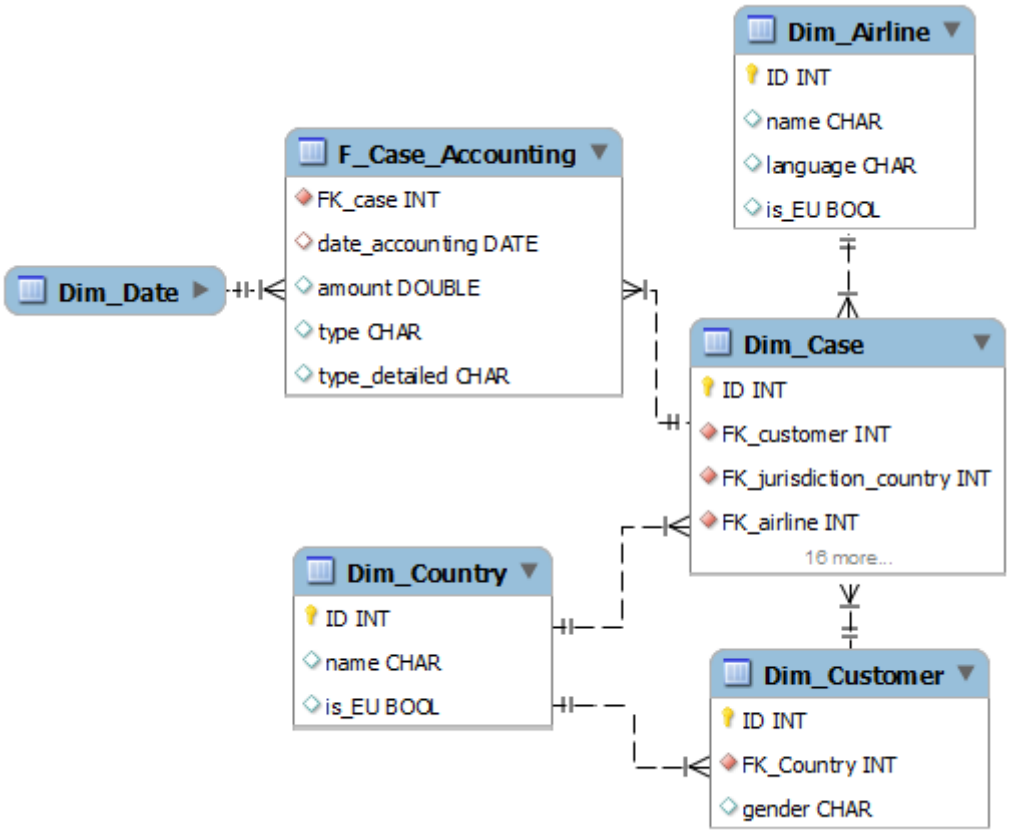

FIGURE 3.27: Case Accounting analysis

Case Accounting. Current analysis

- Revenue(*type*='Revenue') per month.
- Unsettled amount per case (*type_detailed* in ('Success_Fee', 'Success_Fee_VAT', 'Service_Fee', 'Service_Fee_VAT', 'Interest', 'Interest_VAT', 'Cancellation_Fee', 'Cancellation_Fee_VAT'))

Case Accounting. Required and further analysis

- Analysis, similar to that of Case Payments.

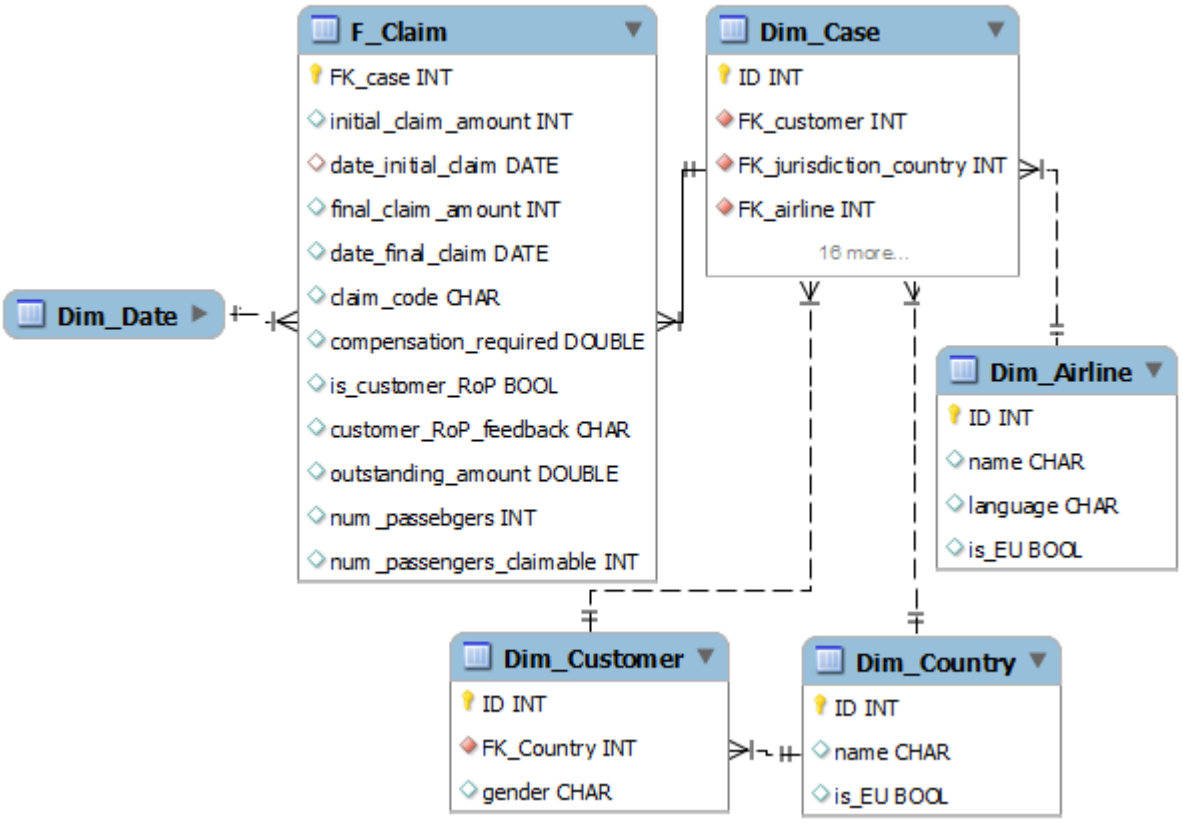

FIGURE 3.28: Claim analysis

Claim. Current analysis

- Distribution of cases by claim code.
- Monthly statistics about amount of claim.
- Percentage of claims that enforced payments per month.
- Amount of payments enforced in last 90 days.

Claim. Required and further analysis

- Statistics connected with various attributes of Case dimension
- Claim volume evolution using *initial_claim_amount* and *final_claim_amount*
- Average claim amount per passenger.

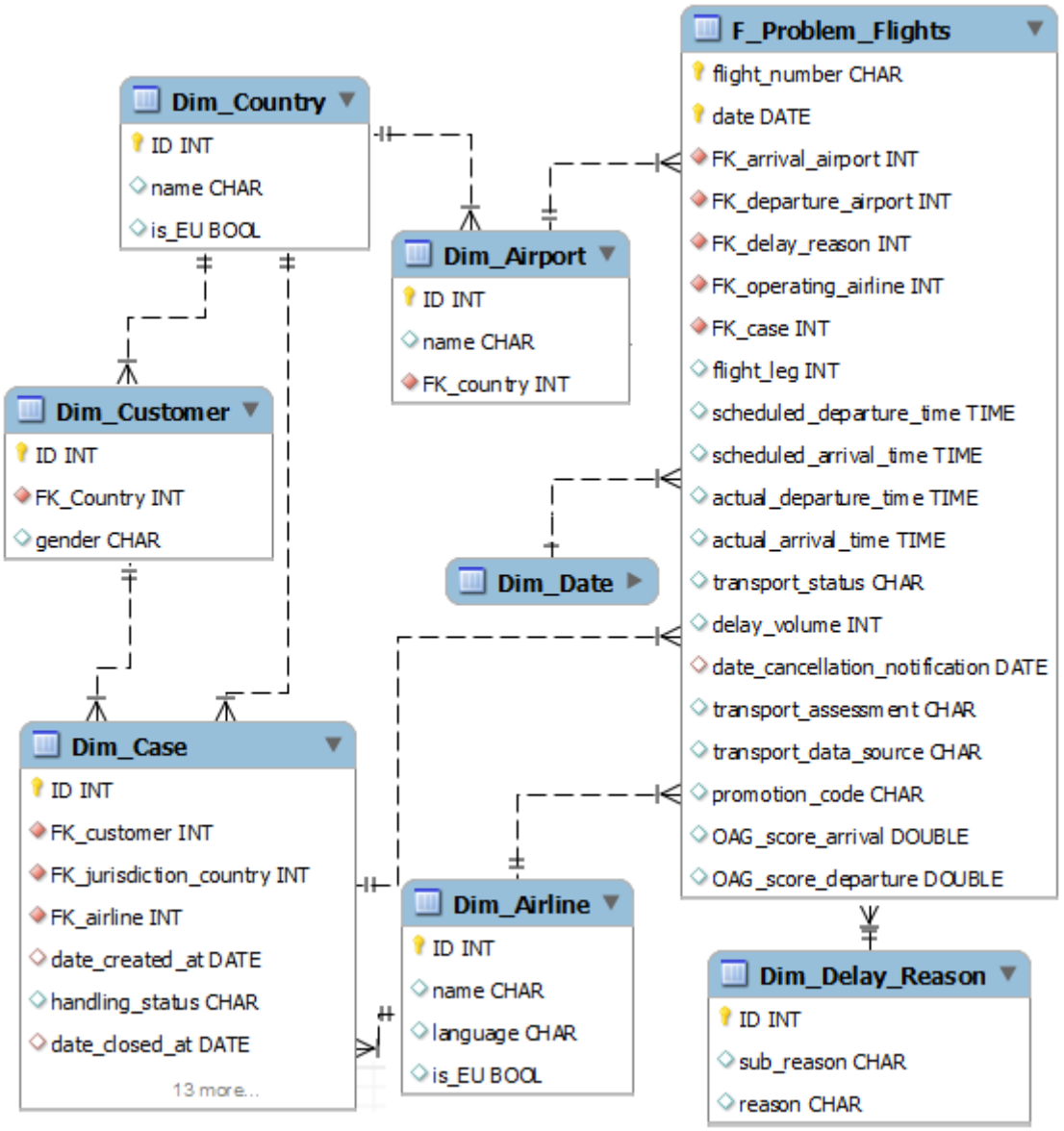

FIGURE 3.29: Problem Flights analysis

- Claim amount per airline/customer/country.

At the time of BI solution development the area of analytics related to Problem Flights was the one with the fewest KPIs established. Therefore this part of the OLAP schema is mostly an option for a potentially further analisis.

Problem Flights. Current analysis

- Distribution of the delay reasons and sub-reasons per month.
- Parameters of the cases without delay reason.

Problem Flights. Required and further analysis

- Average delay per airline, airport, airport country, etc.
- Distribution of OAG scores for departure and arrival airports per country.
- Number of delayed flights at particular day of the year (for instance Christmas), week, month, etc.
- Correlation between the delay volume and delay reason on one side and airline and airport on the other side.

3.7.5.1 Ad-hoc analysis

Besides the considered scenarious based on typical facts, the developed OLAP schema can be used in arbitrary ad-hoc analysis. The practical example which demonstrates how the complexity of analytics has been reduced is provided below.

The dashboard from figure 3.30 displays the statistics about airlines with respect to number of cases (total and in the last month), volumes of claims (total and in the last month), number of closed cases, including those closed successfully, share of claims volume and number of cases, percentage of cases handed over to advocate as well as percentage of those cases that went to the court.

Top Airlines - Number of Cases and Claim Volume (pgBE)

Airline	N of cases	N of cases la...	Claim volume	Claim volume...	N closed cases	N cases clos...	Claim share	Cases share	Handed over ...	Court ordered...
Air Berlin	9574	385	3492825	148475	7722	5703	18.6	20.0	32.9	13.6
Condor	8669	395	3893050	179200	4093	3281	20.8	18.1	107.0	58.3
Lufthansa	4654	218	1739025	83875	3803	1969	9.3	9.7	16.4	20.0
TUIfly	1945	132	743550	51875	1188	843	4.0	4.1	40.2	34.8
Germanwi...	1872	167	517350	44000	914	567	2.8	3.9	80.4	52.7
Ryanair	1734	109	514850	32700	774	278	2.7	3.6	97.8	39.8
Air France	1519	74	664250	33575	880	430	3.5	3.2	39.7	20.9
easyJet	1254	102	338075	28550	711	376	1.8	2.6	63.0	33.5

FIGURE 3.30: Top Airlines dashboard

Parameter	Old query	New query	Complexity reduction factor
N of joins	31	2	15.5 (94%)
N of tables used	6	3	2.0 (50%)
N of attributes used	23	11	2.1 (52%)
N of lines of query code	112	25	4.5 (78%)

TABLE 3.3: Reducing of complexity with OLAP schema

This quite a diverse statistics was obtained by the complex SQL query which listing is provided at figures 2 and 3 in Appendix. The quantitative complexity of this long query consists of receiving data from many different tables and therefore of utilisation of multiple joins. The qualitative complexity encompasses an access of some entities by

referring to the numeric values of their identificators as well as embedding the code for calculated fields.

After implementation of OLAP schema and enabling it into the system, absolutely the same result was received with the query provided in figure 4 in Appendix. Both quantitative and qualitative factors of complexity were decreased and the query itself became more readable for the person who is not familiar with the database structure. The details of the complexity reduction are summarised in the table 3.3

3.8 BI solution development. Tools evaluation

3.8.1 Preliminary evaluation

3.8.1.1 Choosing vendors

The requirements for BI tools to provide powerful dashboards, manageable reporting and ad-hoc analysis formulated in Chapter 3.6 were used as a main filter. The formation of the list of tools was also inspired by articles with comparative overview such as Gartner's Magic Quadrant [47] as well as by the company's previous experience of working with BI tools. It was also influenced by the insights of the colleagues from other SMEs, that, for instance, recommended some tools from the review by Huefner [48].

Finally, the following products were chosen for evaluation: *Ducksboard* [49], *Cyfe* [50], *Sumall* [51], *QlikView* [52], *Microstrategy* [53], *Jaspersoft* [54], *SpagoBI* [55], *Tableau* [56], *Statsmix* [57] and *HaloBI* [58].

3.8.1.2 Choosing evaluation parameters

As far as the general and ultimate goal was to establish self-service BI, all the process of evaluation was executed though the prism of self-service solution. This approach was especially important when considering business parameters that were defined as follows:

- *Ad-hoc analysis provided.* Given that all necessary database connections established and data sources created, how easy it is for the end user to create his own analysis. How intuitive it is for him to operate with dimensions and measures at pivot analysis principle, execute roll-up and drill down operations, slice and dice using filter functionality, etc. The statistical analytics including trending/forecasting is also considered here.

- *Dashboarding capabilities.* This parameter indicate how powerful the created dashboards can be: diversity of possible graphs and charts, level of interactivity for the end user, data navigation and manipulation within the data source, etc.
- *Reporting features.* What are the possibilities for scheduling and customisation.

The following technical parameters were defined as meaningful and important for the considered company:

- *Data sources.* As it was mentioned in subsection 3.4.4, 90% of KPIs were based on the operational data which was managed by PostgreSQL at the time of the tools evaluation process. Therefore PostgreSQL support was determined to be a must for a BI tool while having Google Analytics among its data sources was classified as desirable.
- *Data capacity.* The size of operational database was around 200GB, the OLAP schema used only a set of tables and their attributes and thus should have been considerably less. In addition, as it was shown in section 3.7.5 a typical scenario would be to load only a part of the schema. Therefore, a support of data sources starting from Gigabytes should have been given a try.
- *Users/Roles.* It was required to be able to provide different access rights to the users.
- *Performance.* During the preliminary evaluation this parameter rather meant technical basis than real performance that was evaluated at the next step of detailed evaluation.
- *Multiplatform support(mobile).*
- *Installation.*
- *Operating system.*

As for financial aspects, the main goal was to discover the pricing scheme and estimate the order of magnitude of the final price.

It is worth to say that all these parameters were also considered in terms of comparison with current BI tool in use, Klipfolio. This tool provided more or less satisfactory dashboarding facilities, but didn't allow to execute an ad-hoc analysis. It had a limitation of 10Mb per dataset and therefore required to pre-define the data by means of SQL-query which contradicted the direction of self-service BI.

3.8.1.3 Creating a short list

	QlikView	**Microstrategy**	**Jaspersoft**	**Tableau**
Ad-hoc analysis	Provides new data exploration model. When the data is selected in one table, data in all tables is immediately shown as related or not related to the selected one. Trending is provided. Search box for data navigation	Pivot analysis by drag-and-drop. Integration with R. Trending is provided	Pivot analysis by drag-and drop. Trending at least in dashboards. Hierarchical sliding along dimensions.	Pivot analysis by drag-and-drop. Powerful data navigation with filters. Integration with R. Trending and forecasting is provided.
Dashboards	High level of layout customisation. Zooming functionality drills down the data. Sufficient diversity of graphs. Interactive dashboards.	Colour and size visualisation of dimension values. Sufficient diversity of graphs. Interactive dashboards.	Zooming functionality drills down the data. Sufficient diversity of graphs. Interactive dashboards.	Interconnected filters on panes of dashboards. Colour and size visualization of dimension values. Sufficient diversity of graphs. Interactive dashboards.
Reporting	Report editor for custom reports. Exporting in image, PDF format.	Exporting in PDF, email or Microstrategy fil format. Customised, pixel-perfect reporting as well as scheduling is available in Express edition	Drag-and-drop report designer using a standard browser or mobile device. Scheduling and pixel-perfect for printing. Interactive reports. Export in PDF, Excel, XLSX, DOCX, CSV, RTF, ODT, ODS	Export in image, PDF, email formats. Sharing by email or weblink as well as internally among server users.

Table 3.4: Business parameters

As a result of preliminary evaluation, the tools have been splitted into three following groups:

1. Lightweight, online, inexpensive tools with limited functionality, not able to connect to on-premises databases. These tools are mainly suitable for marketing purposes, to monitor social networks or other online activities associated with website.

	QlikView	**Microstrategy**	**Jaspersoft**	**Tableau**
Data sources	All relational including PostgreSQL, NoSQl.	All relational including PostgreSQL, NoSQL.	All relational including PostgreSQL, NoSQL, flat files.	All relational including PostgreSQL, NoSQL, flat files, Google Analytics
Capacity		Datasets up to 5 GB	Terabyte scale In-Memory engine	Gygabytes to Petabytes
Users/Roles	yes	yes, in Express edition	yes	Supports Microsoft Active Directory, in addition to the built-in Tableau user system
Performance	In-memory of QlikView server	In-memory, sub-second performance on terabytes	In-memory analysis engine	Data Engine is designed with "live connection". Not limited by "all in memory"
Platforms	Multiplatform	Browser version - free, additional fee for other platforms	Multiplatform	Multiplatform
Installation	Standalone application	The server is installed on-premises and accessed from the browser.	The server is either instaled on-premises and accessed from the browser or accessed online.	The server is either installed on-premises or accessed online. Both accessed via desktop version
OS	Windows-based	Windows-based	Windows-based	Windows-based

TABLE 3.5: Technical parameters

These tools are Ducksboard, Cyfe and Sumall.

2. Powerful tools providing variety of options for dashboards and ad-hoc analysis but at the same time requiring programmatic approach to many of its features. For instance Statsmix offered its functionality through API calls and by the way didn't support PostgreSQL. As for SpagoBI and HaloBI, they required elements of coding in data cubes specification and some other procedures. Besides they both provided complex user-interface flooded with menus and settings that could not be quite user-friendly for people without IT-background. Therefore these tools could not be considered as candidates for establishing self-service BI.

3. The rest of the tools - QlikView, Microstrategy, Jaspersoft and Tableau - revealed a good potential for self-service BI. They offered a solid functionality in dashboarding and reporting and, most importantly, provided an easy and intuitive way to build ad-hoc analysis.

	QlikView	**Microstrategy**	**Jaspersoft**	**Tableau**
Cost	Express edition is free for 1 year. Even the range of further price was not disclosed by email	$1,350 per named user, $35,000 per server	Annual, subscription. Online server: <$1/h, no data and no user limits. $762/year with annual plan or ∼$1000 with 8h per day.	$2000/user for Desktop Professional edition, $1000/user (disclosed at the phone call) for on-premises server, $500/user/year for online server

Table 3.6: Financial parameters

An interesting fact discovered during evaluation is related to the data sources. The tools from the first group didn't support the connection to the on-premises databases, while allowed to connect to Google Analytics. At the same time, more powerful tools didn't provide a connection to Google Analytics except for Tableau.

Another thing is that discovering the complete price schema considering type of deployment, number of users, etc. required communication with vendors from emails up to face-to-face meeting (insisted on by Microstrategy, but declined at this point of time).

Summarising, the first group of tools had technical limitation of inability to connect to PostgreSQL database, the second group was too complex in use for non-IT users, so only the tools from the third group could be considered for establishing self-service BI. As far as third group tools' prices were of the same order of magnitude, the company expressed the general willingness to spend several thousands of dollars for BI tool if it meets its needs. Therefore, it was decided to include the tools from the third group into a short list and to come to the financial question after their detailed evaluation.

Results of preliminary evaluation with respect to business, technical and financial parameters are provided for the tools from the short list in tables 3.4, 3.5 and 3.6 respectively. Details of other tools evaluation are not revealed as they became not important for further evaluation after the tools were excluded from the candidates for the short list.

3.8.2 Detailed evaluation

3.8.2.1 Real-life scenario

In order to test the efficiency of the BI tool within the scope of the company's business it was decided to select one of the current dashboards and try to build it by means of the tools from the short list. The selected dashboard should have contained most typical elements of analysis and thus could have been thought of as the "average" dashboard of the company.

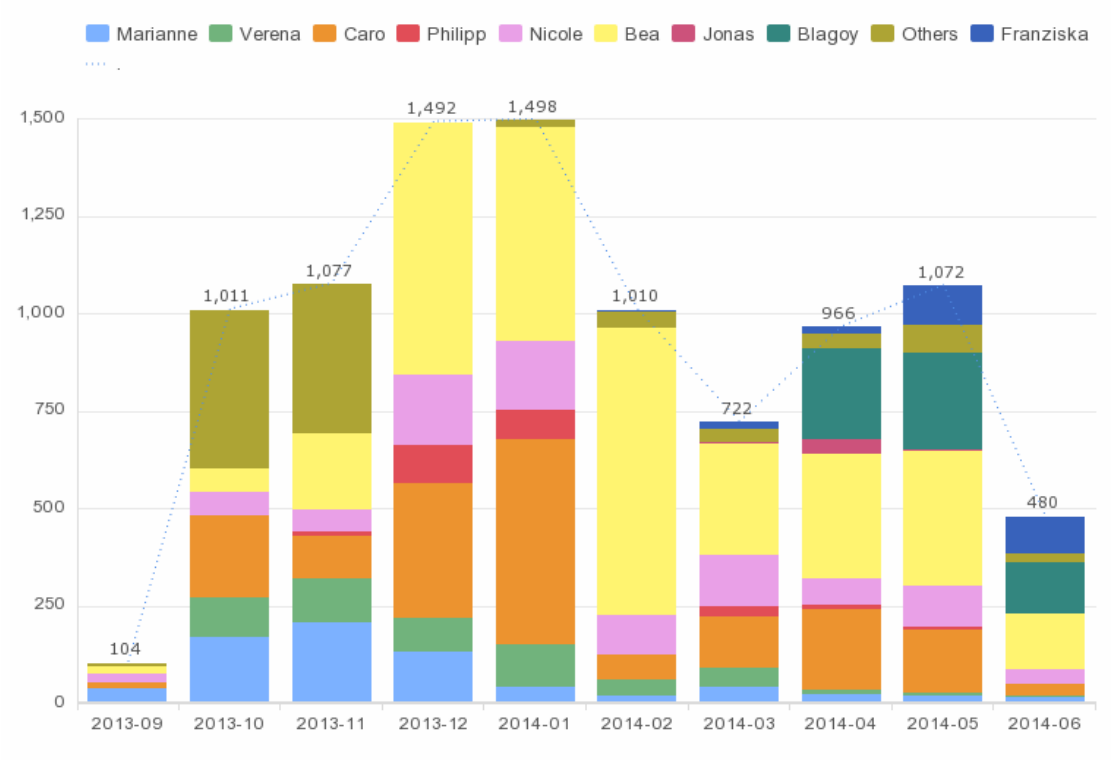

FIGURE 3.31: Test dashboard. Klipfolio representation.

The chosen dashboard (illustrated in the figure 3.31), let us call it *Test dashboard*, represents number of cases handed over to the advocate with distribution of the employees who initiated this operation in the system. This analysis is based on three tables from OLAP schema: *F_Case_History*, *Dim_Case_Status* and *Dim_User*. The tables are joined by ID attributes in dimension tables and respective foreign keys in fact table. The dashboard is used by the Legal department and contains the following analytical parts, typical for the company:

- *Utilisation of Date/Time dimension.* In particular aggregation by time periods (in this case by month) and filtering at the data source level (in this case only statistics since September 2013).

- *Slicing by certain case status.* In current dashboard the cases are sliced by 'Starting_commisioning' value of *category* attribute in Case Status dimension.
- *Distribution of measure values by User dimension.* It is also typical to present only certain dimension values while grouping the rest ones in the so-called 'Others' value. In this case exact statistics is shown only for nine employees, while all others are grouped in 'Others'.

3.8.2.2 Emphasis on important features

Self-service BI

Test dashboard developed in Klipfolio (illustrated in the figure 3.31.) was built upon the dataset returned by pre-defined SQL query. Roughly speaking Klipfolio just displayed already prepeared values for rows and columns with only differentiation of users. As far as self-service BI was an ultimate goal, the task for candidate tools was more complex. Given the dataset of the three tables joined as above, the tools were tested in their ability to provide user with instruments to build the same dashboard without coding.

Date/Time dimension

The current analytics was scrutinised in terms of most common and most important dimensions, measures and data manipulation operations. Date/Time dimension was defined to be the one. It was counted that Date/Time dimension is involved in 91% of current dashboards and it is manipulated in variety of ways including filtering at data source and data presentation level, slicing, formatting, cohort analysis, etc. Therefore it was decided to pay special attention to how tools deal with Data/Time dimension.

Update and sharing

Another aspect of crucial importance in detailed evaluation was data refresh and dashboards sharing between users. In order to test BI tools in terms of mechanism of keeping dashboards up to date, a special indicative dashboard displaying the number of records in *legal_case_history* table was created as well. This was done for convenience as adding records to this table was a matter from seconds to minutes.

3.8.3 Evaluation results

For the sake of objectivity, each BI tool was evaluated for the same time period. The period constituted of five working days and included first introduction to the tool via

web-site, watching educational demos and tutorials, reading documentation and naturally tool's installation, set up and building the test dashboard. Besides objectivity this approach was aimed to give an answer which of the tools is the closest one to self-service BI by comparison what was achieved during the evaluation time and how easy it was. The limited time period also plays a role of the disclaimer to the statements from the next sections as author assumes that some problems identified there could be resolved within the bigger time frame. However this is again an indication of level of manageability, user-friendliness and self-service BI.

The BI tools were evaluated according to the same scheme. At first, connection to the database was established and the required tables were loaded into the data source. Second, while the dashboard was being built, the Date/Time dimension functionality was examined along with another ad-hoc features. Third, reporting functionality was tested along with options of sharing the dashboard.

In the following sub-sections the tools are considered in the order from least to most suitable one for the BI solution, that is - QlikView, Microstrategy, Jaspersoft, Tableau.

3.8.3.1 QlikView

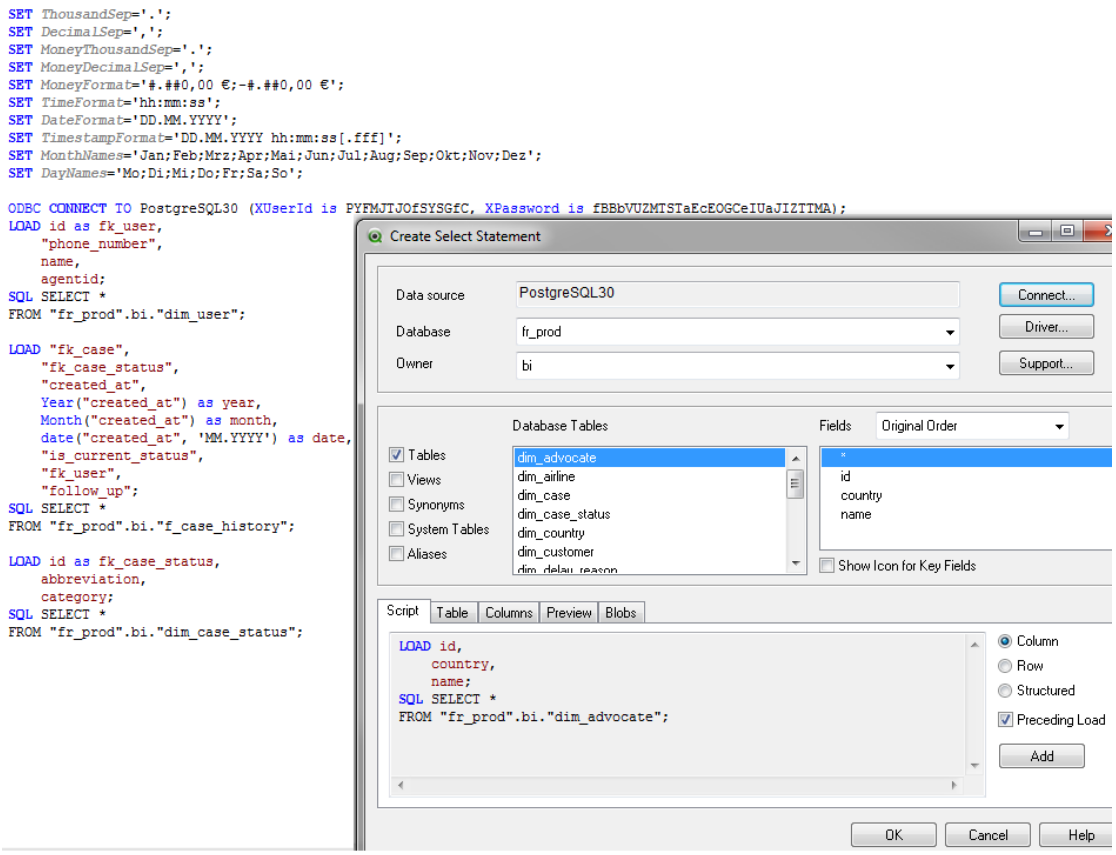

FIGURE 3.32: Script Editor in QlikView

Creating data source and loading data

Current aspect of QlikView was definitely the most complex and the farthest from self-service BI among all tools considered. The point is that starting from database connection up to selection of particular attributes - everything had to be done by means of Script Editor shown at figure 3.32. For instance creating calculated fields had also to be done via script modification inside of the editor.

Furthermore, in order to join tables, QlikView requires to name respective key attributes with the same name. It can also be noticed at figure 3.32 where script contains the following piece of code: *LOAD id as fk_user*.

Building a dashboard. Working with Date/Time dimension

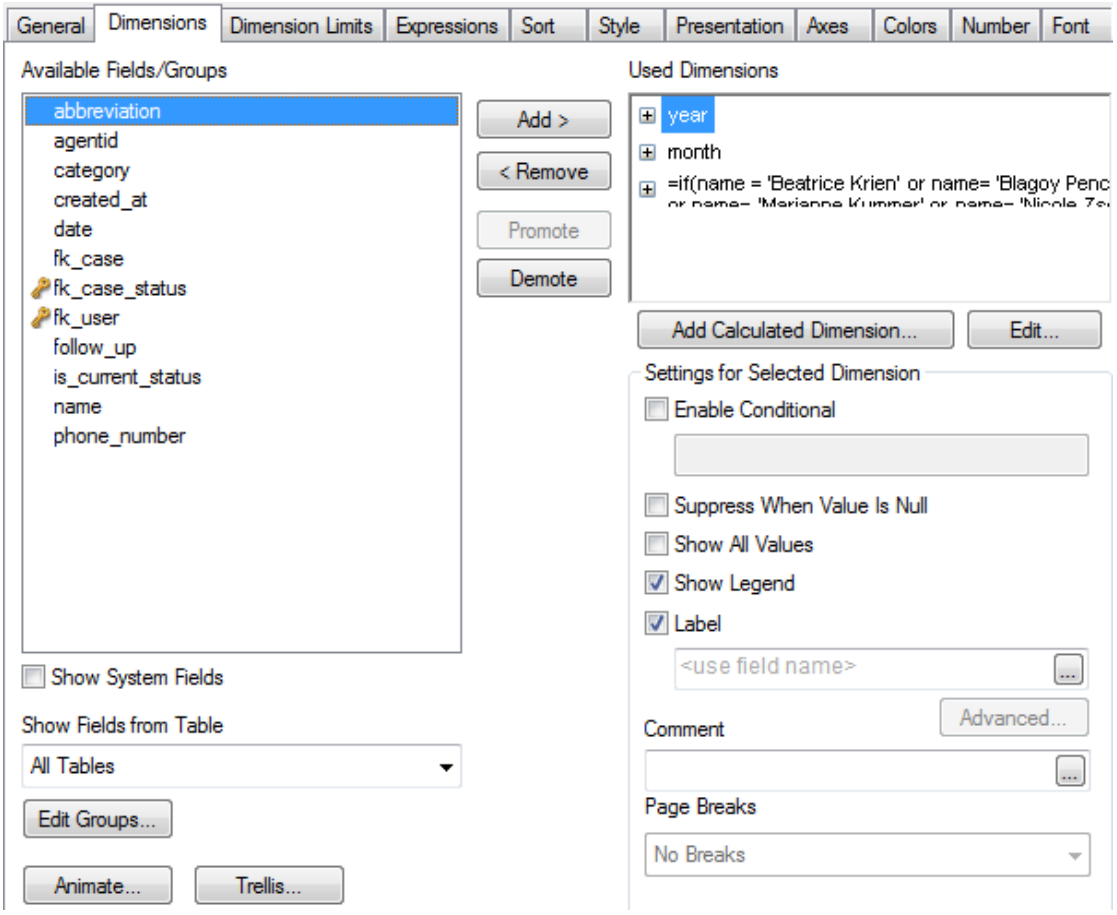

FIGURE 3.33: Dashboard development in QlikView

The process of building a dashboards also starts with quite a programmatic and not intuitive approach. To select measures, dimensions, filter data, specify scale of axis, format visual representation, in other words - to execute any manipulation - one should do it by means of a huge dialogue box with tens of tabs and hundreds of settings, as shown in figure 3.33.

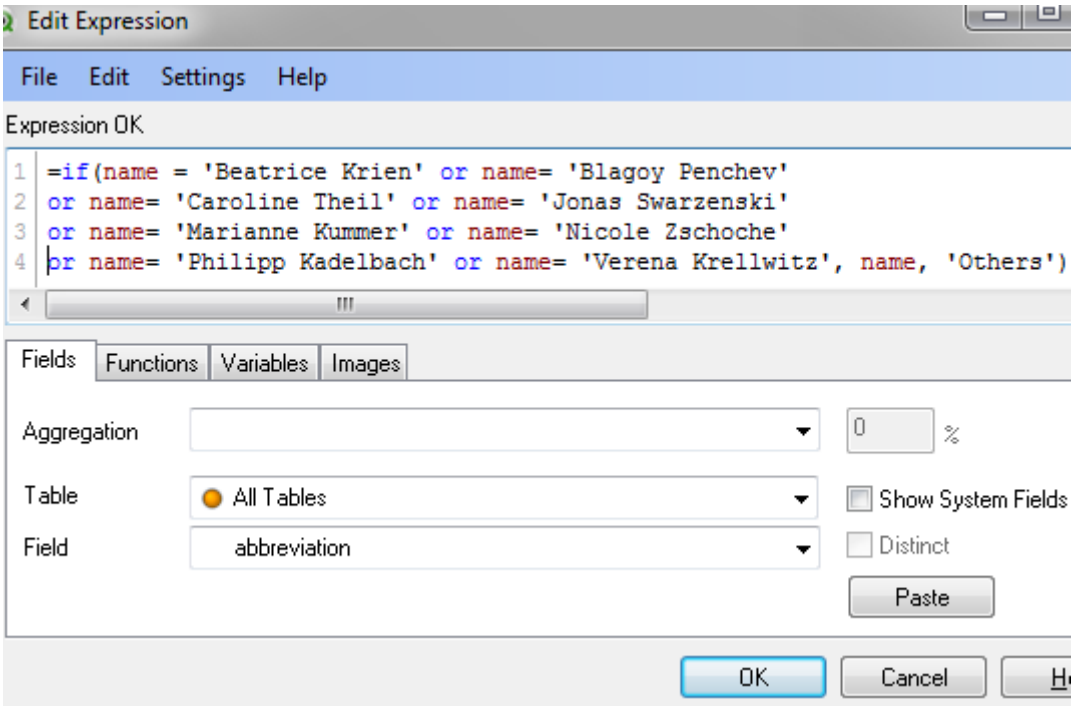

FIGURE 3.34: Grouping users in QlikView

As it was mentioned in section 3.8.2.1, only few users required separate statistics (distinguished by colour), while all others could be joined in one group called 'Others'. However, grouping required to write a script as well (see figure 3.34).

Although Qlikview style was quite far from self-service BI, sometimes it provided a great level of details that could not be obtained by simple *point click done* approach. For instance single *count* measure was provided in many options, besides distinct count: missing count, null text count and others.

Talking about positive things, the main thing should be mentioned is unique data exploration model that was probably the main reason for considering this tool. The main idea is presented in figure 3.35. The green colour indicates selected values, white - related values and grey - not related values. Therefore the user can select a value of any attribute and see how the related data in all other tables without any SQL code.

The second positive moment is that QlikView provides great facilities in specifying layout of the dashboard. It is already shown in figure 3.35 how each attribute's values can be explored. Besides, there are several useful elements that can also be exploited. The figure 3.36 demonstrates *current selection* element along with *search box* element that searches within all attributes of the data source.

The last thing of the pros is zooming functionality that allows to drill down along the selected dimension.

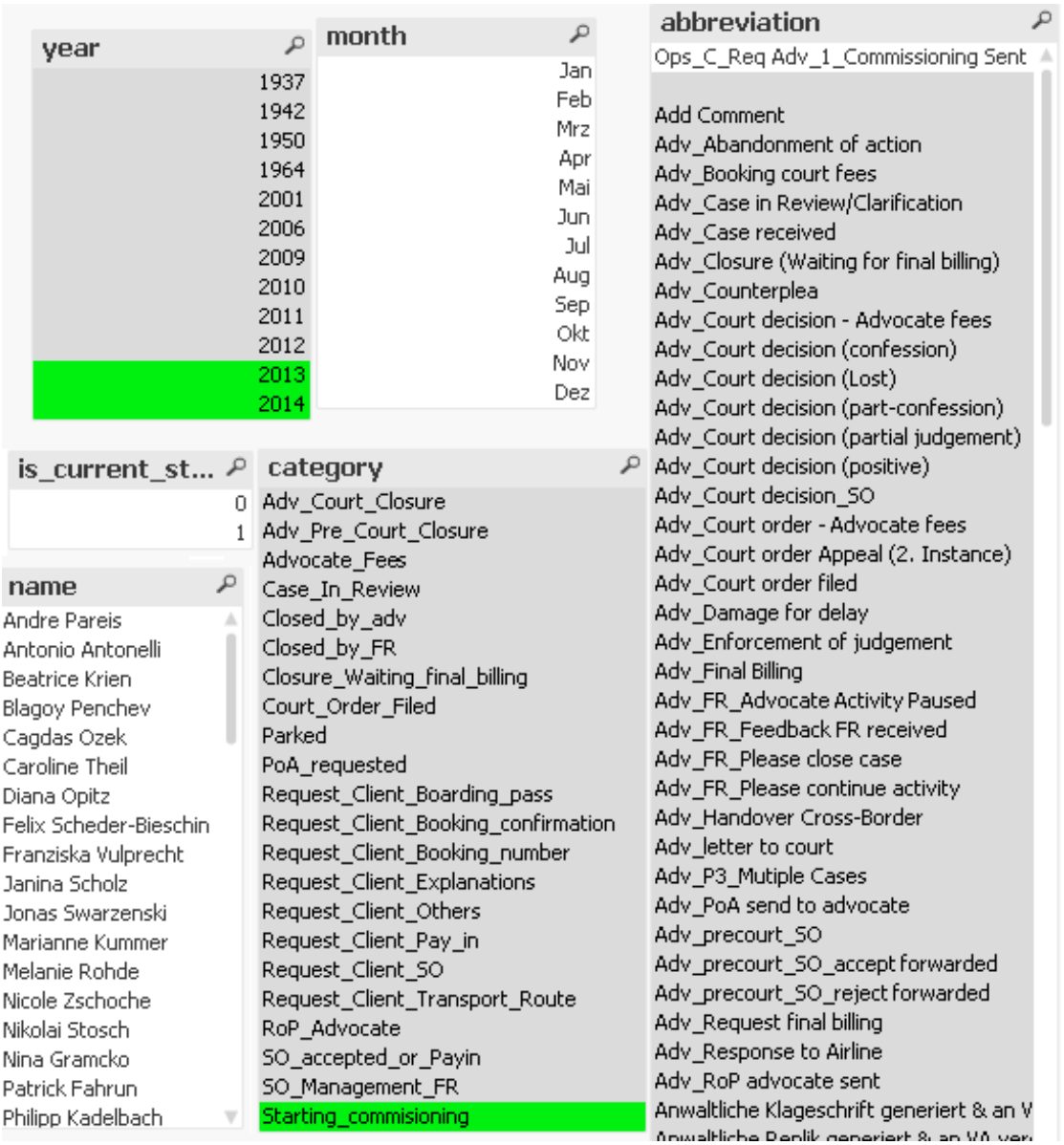

FIGURE 3.35: Data exploration model in QlikView

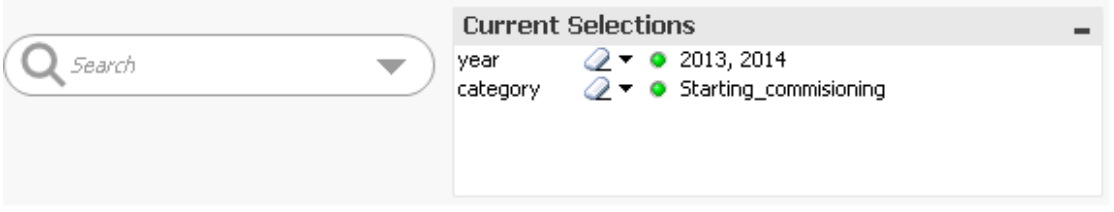

FIGURE 3.36: Layout elements in QlikView

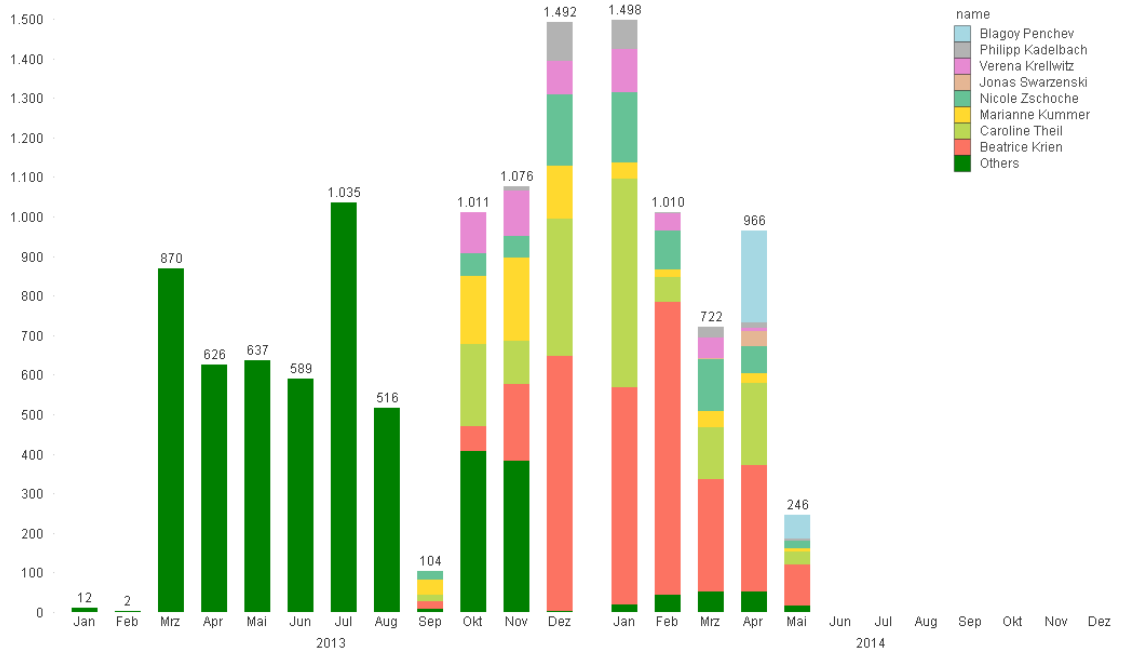

FIGURE 3.37: Test dashboard built in QlikView

Coming back to drawbacks, another one was inability to view underlying data. However the main functional problems were related to Date/Time dimension. The dashboard built is presented in figure 3.37.

First, test dashboard requires to consider only the records with attribute *created_at* from *F_Case_History* table to be not "older" than September 2013. Second, the statistics should be provided with monthly aggregation for all months after the mentioned one. The former requirement could only be solved by inserting respective condition into the script of the data source. It can not be done with dashboard designer as it allows to select only particular values that would require to select thousands of values of *created_at* attribute.

The main source of problems with data in QlikView is that the tool doesn't recognise date data type as a special one. For this reason sorting functionality, for instance, allows to sort numeric or text values, but not date. It requires a workaround of using special *Dual* function that associate date values with numeric ones. Finally this peculiarity reflected in the dashboard that doesn't look similar to its reference point from Klipfolio. Excluding months from January till August as not relevant for 2013 would exclude the same months in 2014, therefore the dashboard contain all 24 months for these years, including redundant, not relevant ones.

Setting up updates and testing report functionality

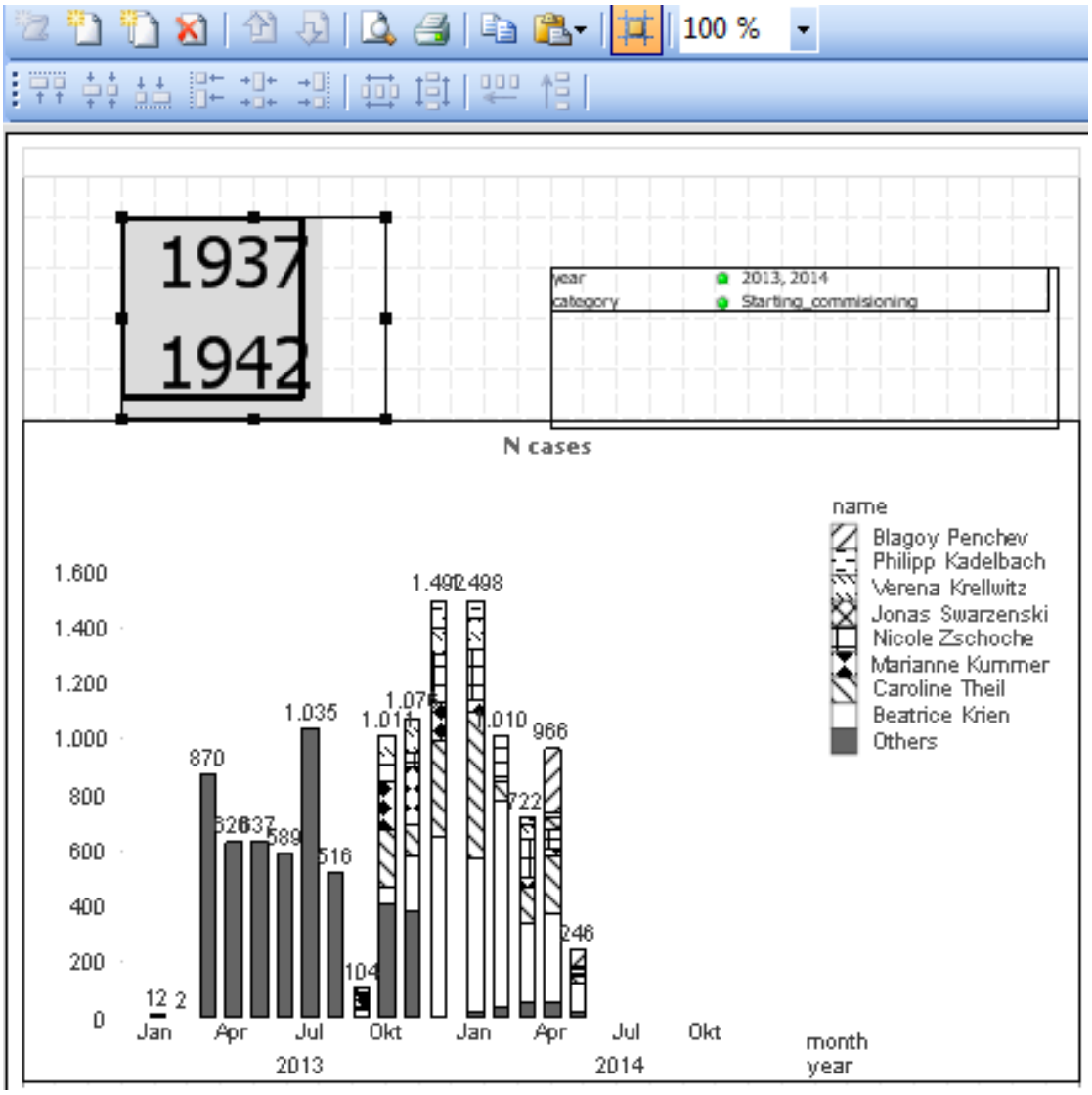

FIGURE 3.38: Report editor in QlikView

The report editor, shown in figure 3.38 in similar fashion allows to specify the layout in details and to place the elements from the dashboard. However this is provided in the same complex approach overwhelmed by dozens of settings.

There are a few sharing options for dashboard: by image or in email in native format. At the same time there is an option that has not been seen in other tools such as export of the underlying data to Excel.

Regarding the data refreshments, QlikView provided the worst functionality. No scheduling and only reload of the data into data sources.

3.8.3.2 Microstrategy

Creating data source and loading data

The first drawback was discovered pretty early, while creating the data source for a dashboard. It was not possible to rename attributes. This led to the situation that there were *Id* and *Id (1)* attributes in the dashboard designer presenting identificators from User and Case Status dimensions. This problem was complicated by the fact that all selected attributes could not be associated with their origin tables as can be seen from the print-screen presented in figure 3.39.

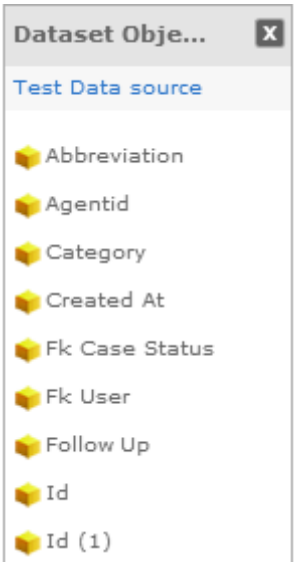

FIGURE 3.39: Data source representation in Microstrategy

The next drawback is related to the general stability. An attempt to edit already created data source resulted in redirection to its initial state.

Building a dashboard. Working with Date/Time dimension

The process of building a dashboard allowed to talk about self-service BI, though it left several questions unanswered. However a few positive things should be mentioned:

- Ad-hoc analysis is provided in up-to-date drag&drop style.
- Performance was quite good during dragging and dropping measures and dimensions on rows and column
- It was possible to access underlying data for a quick view.

The dashboard built with Microstrategy is displayed on figure 3.40.

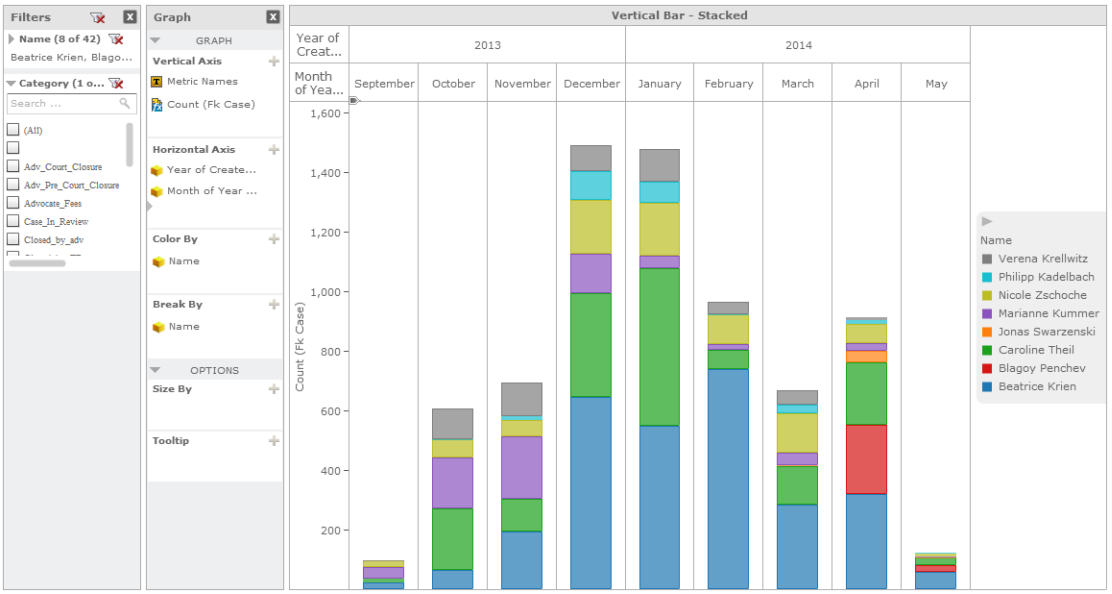

FIGURE 3.40: Test dashboard built in Microstrategy

The biggest limitation is related to inability to group values. The grouping functionality was not discovered and therefore the 'Other' group is not presented in the dashboard.

Another problem was related to Date/Time dimension. The first requirement of Test dashboard to filter out the data that is older than September 2013 has not been completely solved within a reasonable amount of time. Probably the main reason is not intuitive programmatic approach of such filters and poor documentation on the topic. The findings allow to say that the solving of this problem is just a matter of time.

As for monthly aggregation, the tool automatically creates fields 'Month of...', 'Year of...' based on the attributes that have date/time data type. It can be seen in the picture that these fields were used in 'Horizontal axis' section that allowed to create required aggregation.

Setting up updates and testing report functionality

The Express version of the tool provides a bunch of functionality for data refreshment and dashboard updates.

The update of the data source can be scheduled with daily updates as the highest frequency.

The reporting and dashboards sharing is possible via pdf, web-link, Microstrategy file or embedding into email. Pdf that is shared is not static, it is interactive given that it resides on a machine that has an access to Microstrategy server. Emails can be scheduled the same way as data refreshments.

3.8.3.3 Jaspersoft

Creating data source and loading data

The whole installation procedure appeared to be a bit complex as it required to modify some settings in the configuration files of the Jaspersoft.

As for the data connection and loading, the process was quite straightforward and didn't have some flaws as BI tools evaluated above. For instance, Jaspersoft didn't require tables being joined to have attributes with the same name. The tool revealed an intelligence and correctly joined *Id* from *Dim_User* table with *fk_user* from *F_Case_History* table.

Besides, good types recognition was also the case with this tool as it allowed to specify the display format for suggested measures and dimensions.

The main drawback at this step is related to calculated fields. Grouping employees in group 'Others' (required for Test dashboard) was more programmatic as in QlikView rather than based on intuitive selection.

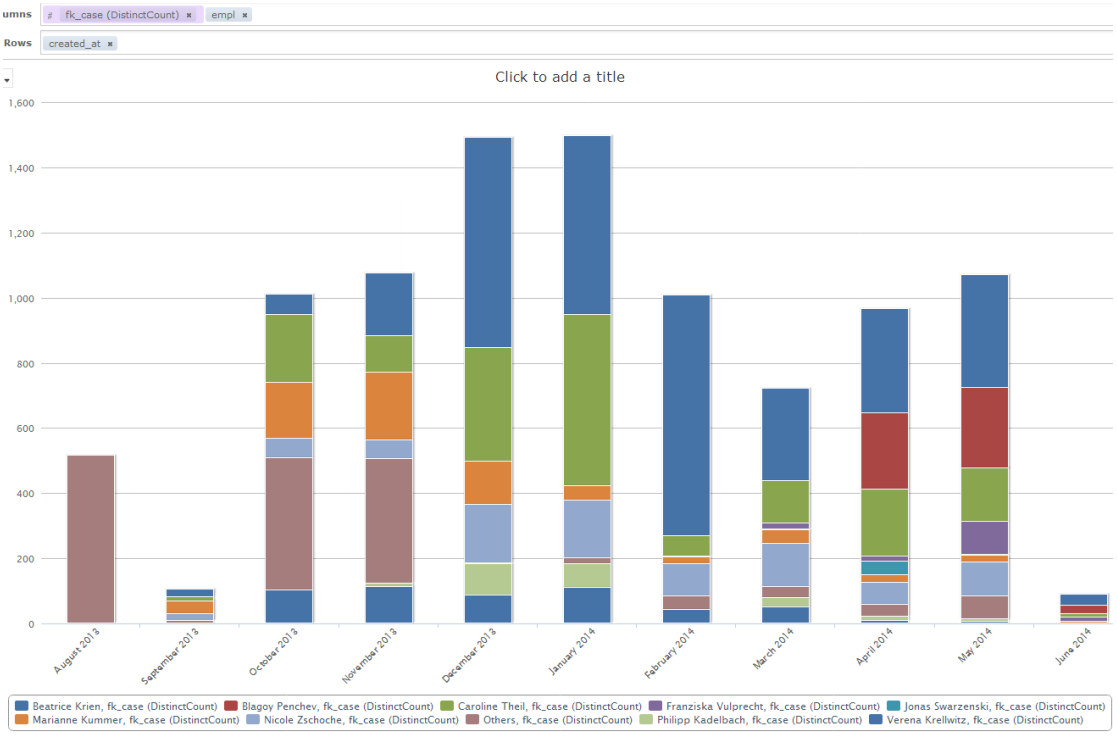

FIGURE 3.41: Test dashboard built in Jaspersoft

Building a dashboard. Working with Date/Time dimension

Continuing on calculated fields, it would be much more convenient to be able to define them not only at the step of data source creation but at the dashboard developing step

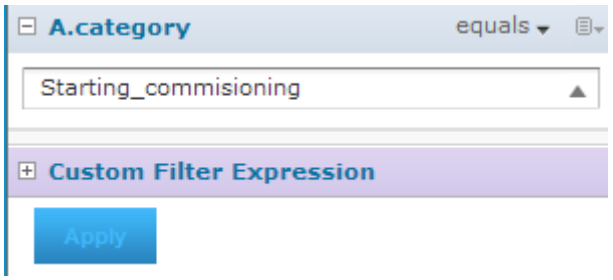

FIGURE 3.42: Filter of ad-hoc designer in Jaspersoft

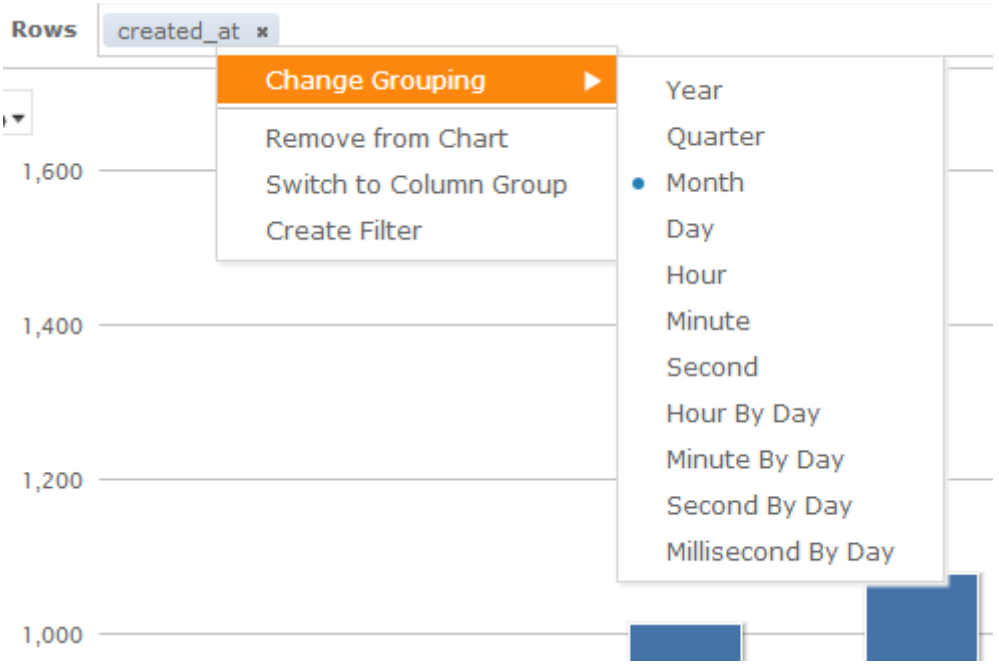

FIGURE 3.43: Date/Time aggregation in Jaspersoft

as well. Unlike the other tools Jaspersoft doesn't provide it and get its first negative point related to dashboarding.

The second point is lack of the functionality to see underlying data.

The last portion of negative is caused by especially important Date/Time dimension. For some reason date filter didn't work properly. It worked correctly with exact dates, but didn't work when the range condition (dates after September 2013) was introduced. Furthermore, it led to the software crash with the necessity to restart the host machine.

The continuation of discussion on Date/Time dimension will start the topic of positive findings. At first, in Jaspersoft it is possible to filter data at the step of the data source creation as well as at the dashboard development step. For test reasons, the dates were filtered in the data source to start from August 2013. This can be noticed in figure 3.41 that displays the ad-hoc designer. It was supposed that the date can also be filtered

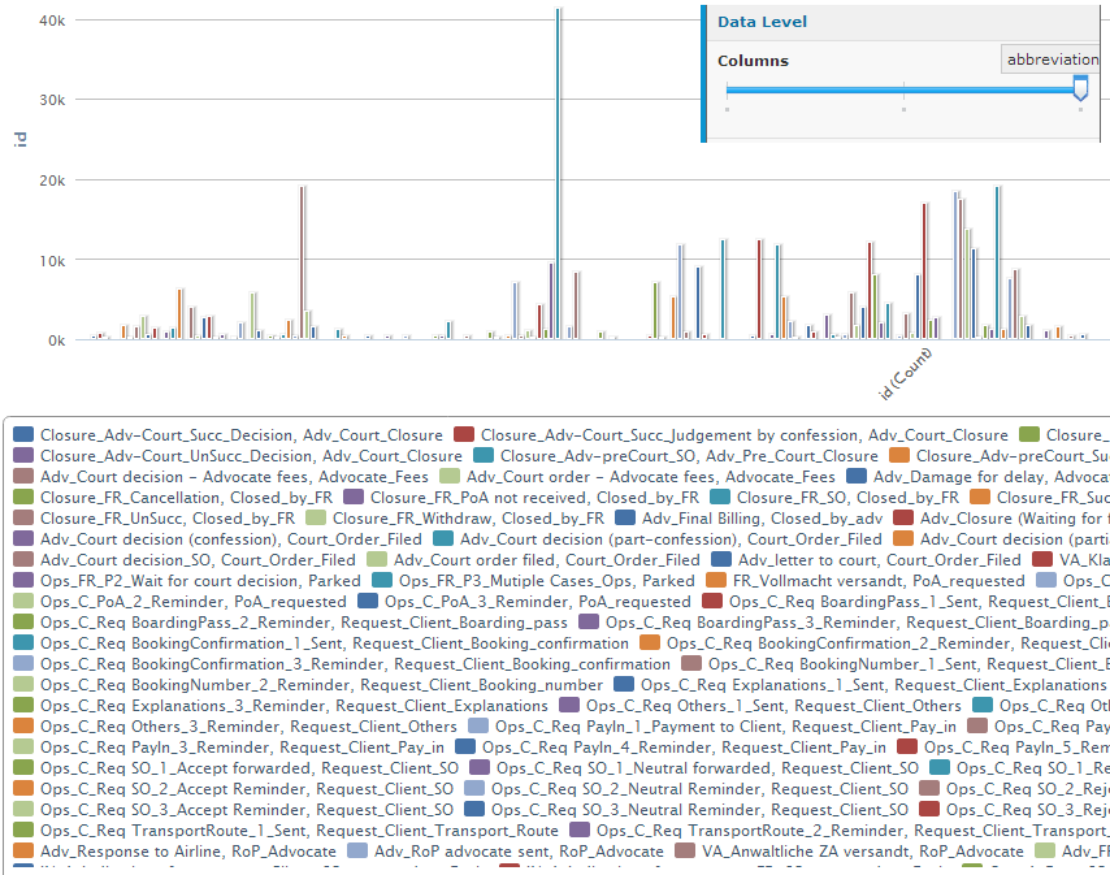

FIGURE 3.44: Lowest hierarchical level of Case Status dimension

in ad-hoc designer to start from September 2013 the same way as *category* attribute is filtered (see figure 3.42).

The main positive feedback with respect to Date/Time dimension is how easy and efficient Jaspersoft deals with time aggregations. The required for Test dashboard monthly aggregation can be done just by selecting respective item in context menu of the targeted attribute. This is demonstrated in figure 3.43. Speaking of self-service BI, it worth to be mentioned, that it took quite a while to reach the same result with previous tools, while it was the matter of minutes in Jaspersoft.

Aggregation functions for measures can be manipulated in the same on-the-fly manner from context menu of the respective attribute.

Zooming functionality, same as in QlikView, is also provided. Besides, Jaspersoft is the only tool from the short list that allows to slide along the hierarchy of dimension attributes. For better understanding let us consider another dashboard uses the number of records in *F_Case_History* table as a measure with *category* and *abbreviation* hierarchical attributes from Case Status dimension. As we know from subsection 3.7.1.1, *abbreviation* is the lowest level of hierarchy while its aggregation *category* is the highest

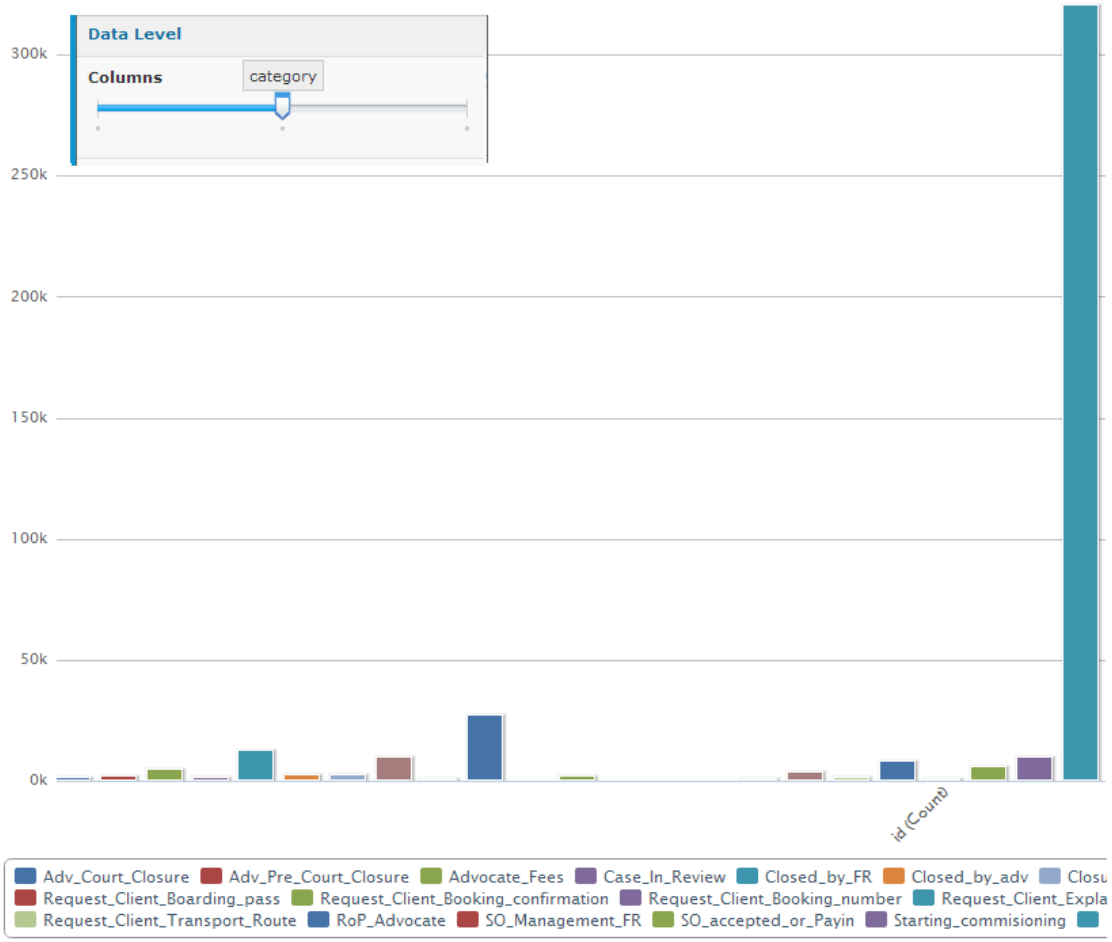

FIGURE 3.45: Middle hierarchical level of Case Status dimension

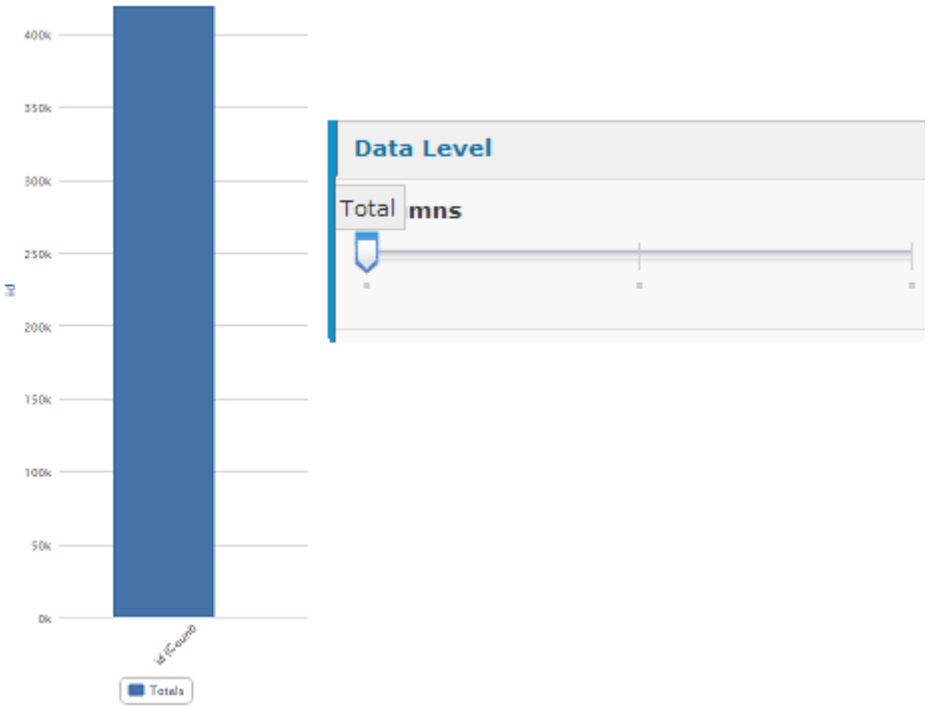

FIGURE 3.46: Highest hierarchical level of Case Status dimension

one before the root. By moving a data level slider from lowest hierarchical level to highest one we can see the number of records distribution at different levels from *abbreviation* to the total number. This process is illustrated in figures 3.44, 3.45, 3.46.

This functionality would be especially useful for dimensions with deep hierarchical level.

In general ad-hoc analysis and creation of dashboards is executed in user-friendly drag-and-drop manner and can be called self-service BI with good level of confidence.

Setting up updates and testing report functionality

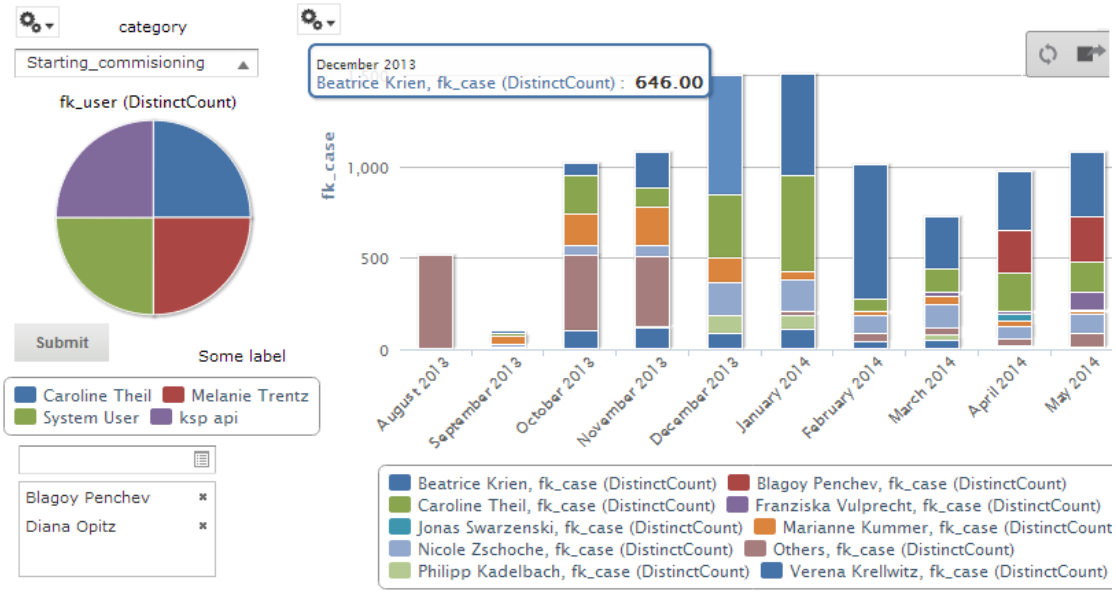

FIGURE 3.47: Dashboard update options in Jaspersoft

Reporting functionality is presented pretty good in Jaspersoft. First of all the report can be shared in variety of formats: pdf, Excel, CSV, DOCX, RTF, ODT, ODS, XLSX. Second, reports as well as dashboards are not static but rather interactive.

Besides, refreshment scheme of the reports allows to not only to set any possible schedule of updates but also to send various notifications by email. As for the dashboards, separate elements of dashboard can be refreshed either independently or all together, using *Submit* button of the dashboard designer. This is illustrated by the figure 3.47 where the arbitrary pie chart is added to the bar chart of the Test dashboard for demonstration purposes. It can be noticed that there is refresh button in the upper right corner of the chart from Test dashboard as well as *Submit* button on the left.

The small nuance is related to the data source update that should be established in addition to the report or dashboard update.

3.8.3.4 Tableau

Creating data source and loading data

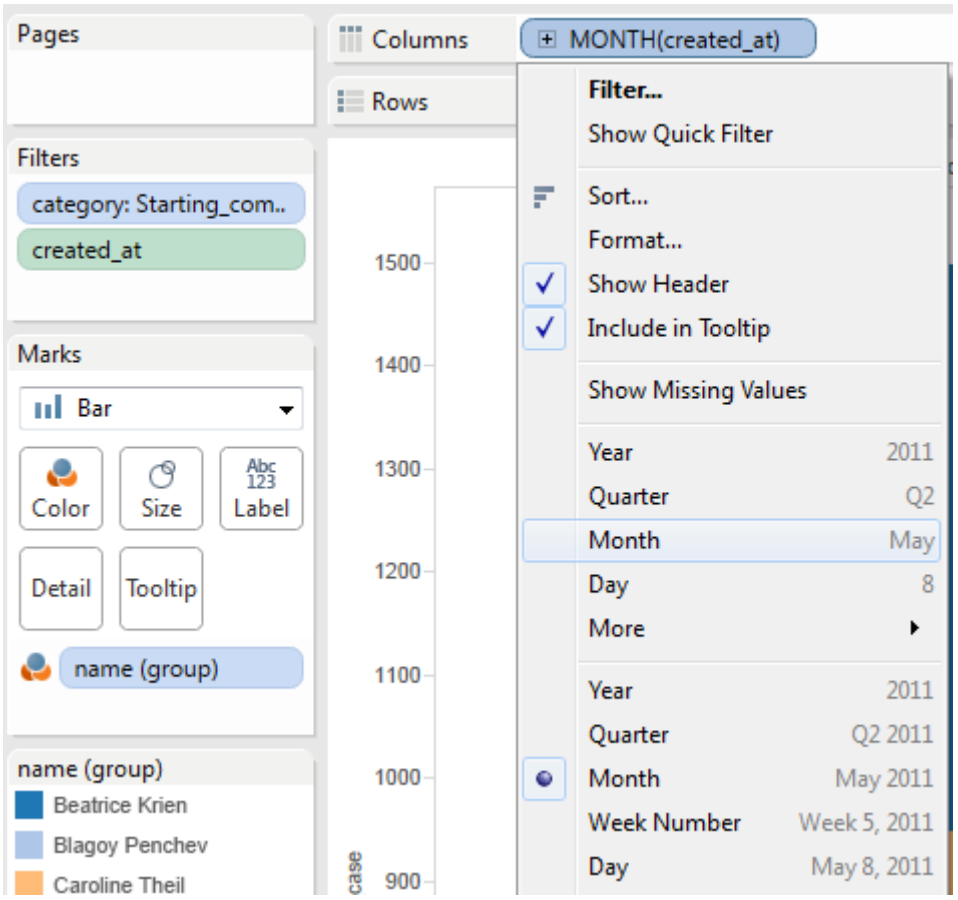

FIGURE 3.48: Date/Time management in Tableau

The first step of the Tableau didn't have any drawbacks. The required tables were easily selected and joined. Tableau suggested way of joining tables was also intelligent enough to determine that *Id* attribute from *Dim_User* table should be joined with *fk_user* from *F_Case_History* table.

Moreover, the tool provides two options for data connection: to connect *live* or to extract the whole data or part of it and work with local copy.

Another important thing is that Tableau is the only tool from the short list that provides connection to Google Analytics.

Building a dashboard. Working with Date/Time dimension

First of all, data type recognition allowed user to format or adjust data type of the attribute in the dashboard designer. This eliminates the necessity, for instance, to use date/time functions at the step of data source creation.

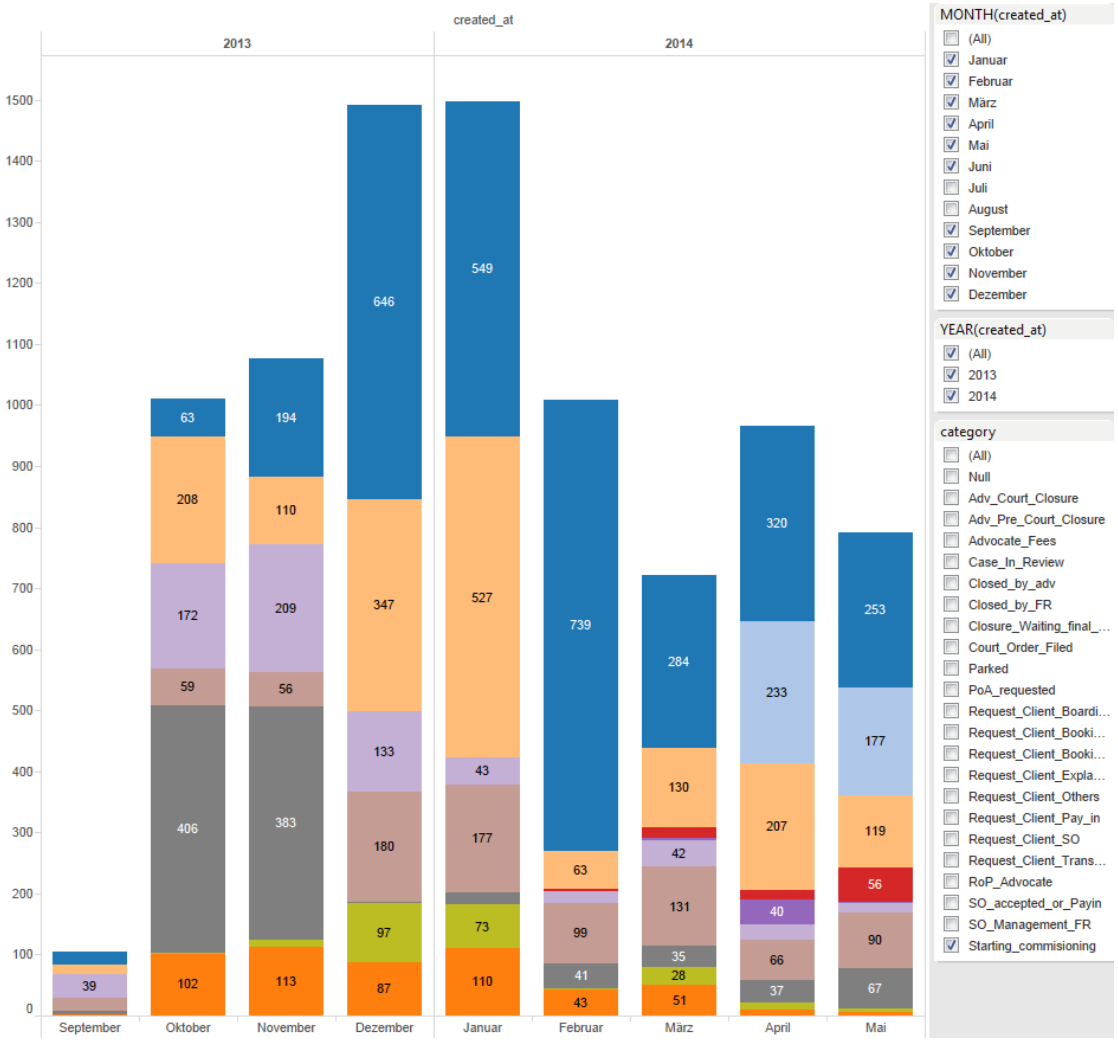

FIGURE 3.49: First approach to Date/Time management in Tableau

Building a dashboard in Tableau was in general was pretty easy. Option to view the underlying data, drag-and-drop style of pivot analysis, intuitive way to match dimension values with colour or size and many other features brought the tool directly to the self-service BI.

Providing a good functionality in all dashboarding aspects, Tableau's biggest advantage over the other tools was revealed in dealing with Date/Time dimension.

First, it offered two ways of working with dates. The first one used month and year of the particular date so to say separately, similarly to the approach from QlikView. The second way allowed to consider month and a year together, similarly to the approach from Jaspersoft. The figure 3.48 demonstrates the context menu which 8th to 12th items from top are related to the first way and 13th to 17th are related to second.

The first approach, shown in figure 3.49, is not suitable for particular Test dashboard as it does not allow to select/unselect a month of a particular year. For instance, if

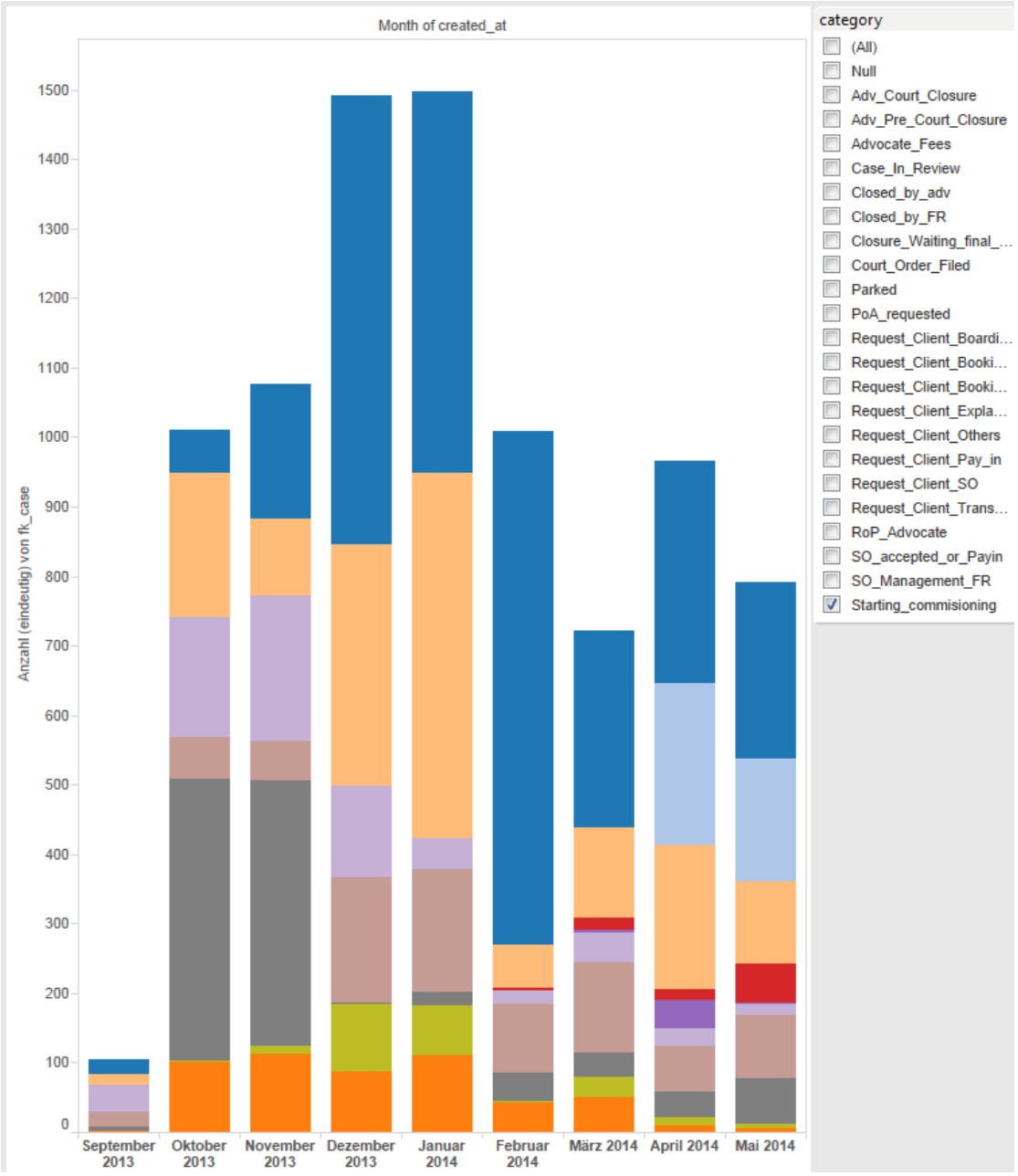

FIGURE 3.50: Second approach to Date/Time management in Tableau

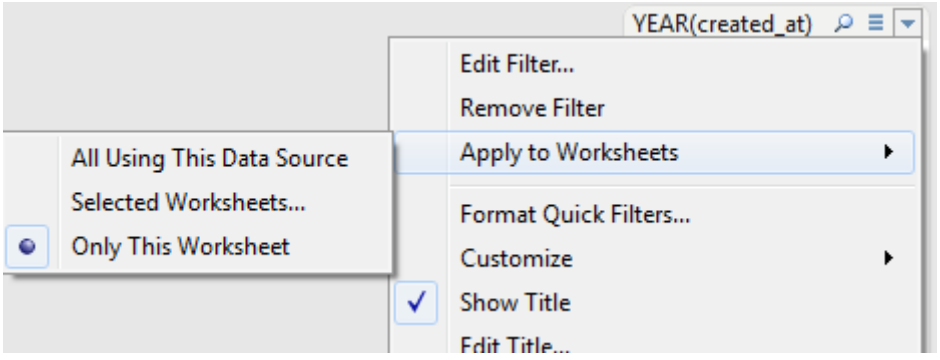

FIGURE 3.51: Filter application in Tableau

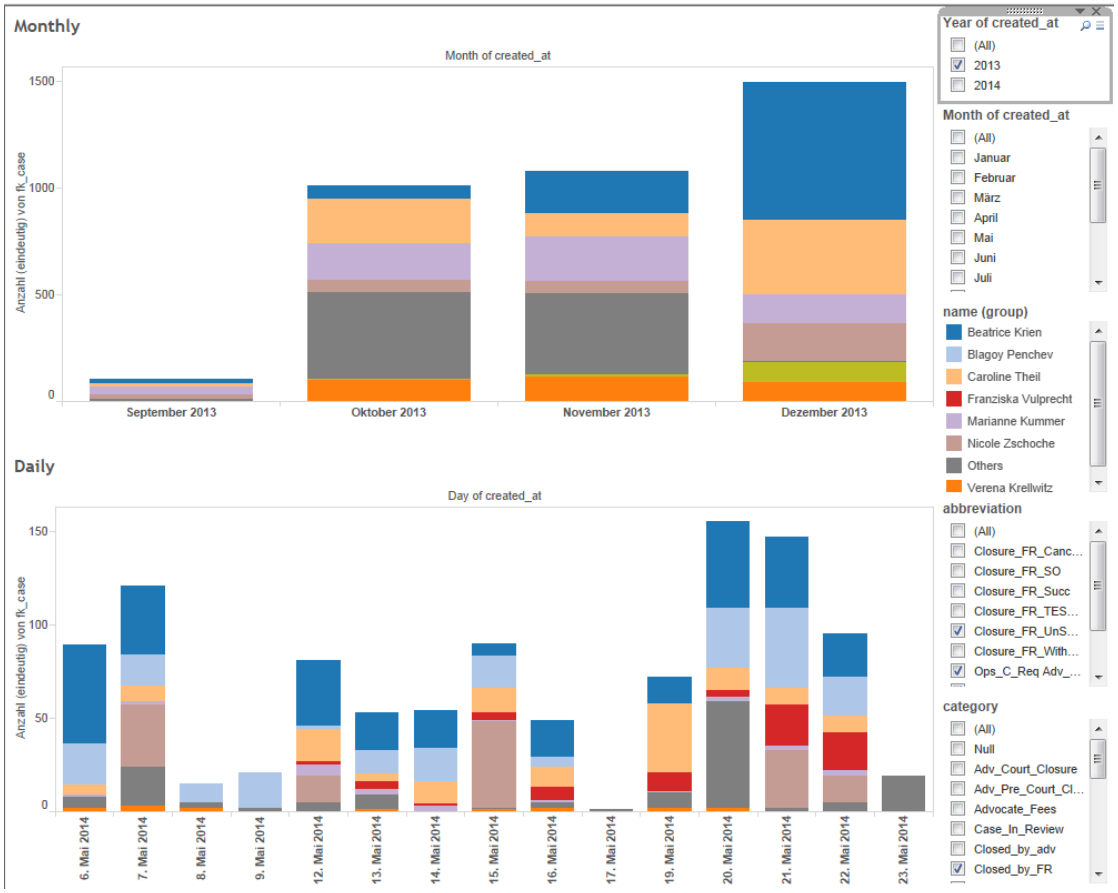

FIGURE 3.52: Filter application in Tableau

the month January is selected/unselected than it is selected/unselected in all years. The second approach, shown in 3.50, allows to select/unselect a particular month of the year. These details create strong arguments for Tableau as a tool for BI solution as Date/Time dimension was determined as one of the most important things for consideration.

Another thing worth to be mentioned separately is filtering. To finish the Date/Time topic, the targeted dimension can be filtered using relative date, range of dates including all possible intervals, by individual dates and so on.

First useful thing about filters is that besides filtering itself it is possible to choose whether to show so-called quick filters on the dashboard or not. Quick filters has been already shown in 3.49 and 3.50 in form of check-boxes which is not the only possible form of quick filters.

The second important moment is shown in figure 3.51 - filters can be applied only to the current dashboard or all other dashboards using the same data source. The resulting dashboard is demonstrated in figure 3.52. Here we can see that quick filter of the year is only applied to the upper chart and unselection of the year 2014 doesn't affect the bottom chart.

Setting up updates and testing report functionality

Tableau provides several options for data update, though the scheme is not straightforward and the most of the updates should be managed on server side, not from the desktop application.

First of all refresh can be applied to the dashboard or to the underlying data source that is shown on figure 3.53. However the update of the dashboard make only sense when *live* connection is used. In case of extracted data source, it should be updated separately.

It is also possible to update data source incrementally by choosing a time limit or threshold value for some attribute.

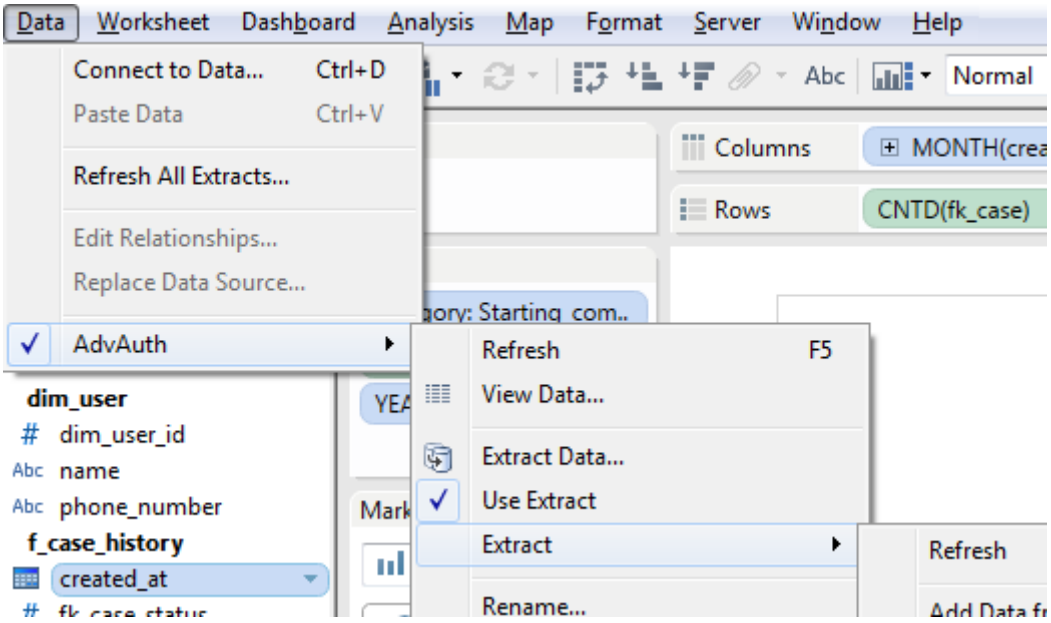

FIGURE 3.53: Update management in Tableau

Scheduling of the updates can be made only on the server. It can be done for both dashboards and data sources published to the server. The server can also be used to manage dashboards including basic edition operations. However to create and fully edit the dashboards is only possible with the desktop application.

As for the sharing - the dashboard can be used in the collaboration with other users who have access to the server. The basic reporting facilities include image, pdf format, direct embedding into an email and sharing by web-link.

3.8.4 Measuring quantitatively.

The evaluation described above brought interesting results. First of all, the tools considered appeared to be at different distances from self-service BI, though they all position

	Description	Mark
Q	- Zooming into dimension - Useful data exploration model - Programmatic approach allows to set up analysis in details	Pos
	- Underlying data is not available for viewing - Programmatic, not intuitive approach of building a dashboard	Neg
M	- Ad-hoc analysis is provided in drag&drop style - Underlying data is available for viewing	Pos
	- Attributes are not grouped by origin tables in the dashboard designer - Inability to group dimension values	Neg
J	- Ad-hoc analysis is provided in drag&drop style - It is possible to filter data at both steps: data source creation and dashboard development - Zooming into dimension - Aggregation functions for measures are easily accessible - Ability to slide along the hierarchy of dimension attributes	Pos
	- Grouping dimension values is more programmatic rather than based on intuitive selection - Calculated fields can only be specified while creating data source - Underlying data is not available for viewing	Neg
T	- Filters can be applied to any chosen dashboard using the same data source - Ad-hoc analysis is provided in drag&drop style - Underlying data is available for viewing	Pos

Table 3.7: Pros and cons. Self-service BI

	Description	Mark
Q	- Range time filtering is only possible via scripting	Neg
M	- Automatically created fields 'Month of...', 'Year of...'	Pos
	- Complex approach in date filtering	Neg
J	- Easy and efficient date/time manipulation	Pos
	- Date filter doesn't work properly	Neg
T	- Two ways of working with dates - Comprehensive date/time filtering	Pos

Table 3.8: Pros and cons. Date/Time dimension

themselves in this way. The overall impression placed QlikView out of the self-service BI horizon, categorised Microstrategy as not well developed for self-service BI and defined Jaspersoft and Tableau as self-service BI tools, while leaving a room for improvement for Jaspersoft and placing Tableau almost on absolute BI position.

However, not to be misled by subjective impression, all pros and cons of the considered tools have been analysed. Recall, that the tools were evaluated in terms of self-service BI, date/time dimension management and data update with dashboard sharing functionality. The positive and negative findings were grouped in the respective three groups: *Self-service BI*, *Date/Time*, *Update and Sharing*. The fourth group called *Others* included notions that could not be assigned to the first three groups.

	Description	Mark
Q	- Only manual data update - Few option for sharing dashboard, only image, pdf, embedded in email	Neg
M	- Dashboard data updates as well as email reports can be scheduled	Pos
	- Highest frequency of refresh is daily update	Neg
J	- Data updates for dashboards and reports can be scheduled with any possible interval	Pos
T	- Data updates for dashboards and reports can be scheduled with interval starting from 15 minutes - Dashboards can be accessed via web-link	Pos

TABLE 3.9: Pros and cons. Update and Sharing

	Description	Mark
Q	- Comprehensive dashboard layout management - Comprehensive report layout management - Export of the underlying data to Excel	Pos
	- Need to build script to create data source - Need to rename joining attributes with the same name - Complex approach in report construction	Neg
M	- Data source editing fails	Neg
J	- Doesn't require tables being joined to have attributes with the same name - Good data type recognition	Pos
T	- Intelligent join of the tables - Two options for data connection: live or local copy of the data - Good data type recognition - Google Analytics among data sources	Pos

TABLE 3.10: Pros and cons. General aspects

Tables 3.7 - 3.10 contain positive or negative ('Pos' or 'Neg' in *Mark* column) moments found during evaluation. The leftmost column contains the first letters of considered tools: Q - Qlikview, M - Microstrategy, J - Jaspersoft, T - Tableau. The first group of pros and cons related to self-service BI are listed in table 3.7. Pros and cons related to Date/Time dimension are summarised in table 3.8. Pros and cons related to Update and Sharing are summarised in table 3.9. Table 3.10 summarises other aspects of the evaluation.

It can be noticed from the tables the that the ratio of positive to negative aspects raises in the following sequence: QlikView, Microstrategy, Jaspersoft, Tableau. The quantitative evaluation scheme proposed in subsection 2.8.2 produces the following results, displayed in table 3.11 and confirms the general impression from the beginning of this section about how far or close to self-service BI the tools are. It can be noticed that Tableau has an advantage of minimum two points over each other tool in each aspect.

As for the opinion from the company's side, it was stated that only Tableau provides a self-service BI at acceptable level and that other tools would demand too much efforts

	Self-service BI	Date/Time dimension	Update and sharing	General aspects	Total
Q	+2	-4	-4	0	-6
M	0	0	0	-1	-1
J	+4	0	+2	+2	+8
T	**+6**	**+4**	**+4**	**+4**	**+18**

TABLE 3.11: Quantitative evaluation

from the personnel. In addition having Google Analytics among Tableau's data sources could help to build a solution with single entry-point for all types of analytics.

Though the software costs reasonable amount of money, the management decided to start with purchasing 5-6 licenses for the Professional Desktop editions and around 20 user licenses for the server. This was dictated by the necessity to have 5-6 users (heads of departments) that will be be creating and editing dashboards and around 20 users to share the dashboards with.

3.9 BI success. Factors and measures

Though, ideally, BI success should be evaluated after some time since its deployment, in case of the current project it was done right after the end of implementation because of the following two reasons:

- *Time scope of the thesis.* Initial planning (recall figure 1.1) assumed to have 2 months for implementation, integration and supervision of this project. That meant around 1-3 weeks for getting the feedback on the deployed solution.
- *Additional time for B2B communication on licensing model.* In addition to the lack of time mentioned above, after Tableau has been defined as the best choice for BI solution, both Tableau vendor and the considered company started a discussion about the purchase details. The point is that Tableau licensing model requires to pay for each user that is able to access dashboards on the server disregarding of read-only or full access mode. The company in its turn would like to have just a few users able to create and modify dashboards, while all other just to have a read-only access.

First of all time of BI solution deployment could not be measured traditionally because of the reasons above. In case of this thesis project the time was limited to around 4 months for practical work from getting acquainted with the business up to proposing and implementing a solution. Considering the figure 2.8 as a landmark for timely comparison where 38% of SMEs require more than 2 years for BI deployment, 4 months look like

incredible result. However, it should be taken into account that current solution is kind of a skeleton for the final one that has rooms for extensions and enhancements mentioned in future work section 3.10.1. At the same time, it is estimated that it requires 6-12 month to fully complete implementation of the solution. In this case it will fall under 46% of SMEs with the shortest age of BI deployment that is *up to 2 years*.

Despite of the time uncertainty, the solution can be analysed in terms of bringing BI benefits and achievement of main objectives. Before considering these aspects, let us first summarise the state of the project at this point of time.

Some parts of the project were fully implemented and deployed while others were just implemented and ready to be deployed. The former was OLAP schema that was regularly updated and that was already in action as data sources of 27 dashboards and ad-hoc reports were migrated from operational database to the schema. The latter was the part related to the BI software as Tableau has been selected as BI tool but not integrated, waiting for the agreement on licensing model. Therefore BI success evaluation was executed practically for the aspects related to the OLAP schema and theoretically for BI tool aspect. Practically speaking, OLAP schema already worked and brought the benefits of reduced complexity even with the old approach of using Klipfolio as visualisation tool. Subsection 3.7.5.1 has also demonstrated benefits of the schema with ad-hoc analysis. Theoretically, Tableau deployment is supposed to bring additional benefits to the table and finally deliver self-service BI.

BI benefits

Considering four types of BI benefits presented in section 2.9, current BI solution provides:

- *Business supplier/partner relation benefits* are expressed in better monitoring of claiming airlines process and legal case processing by lawyers. The former includes statistics about volumes and speed of paying compensation by airlines. The latter defines efficiency of collaboration with the lawyer for cases handed over to the court.
- *Internal processes efficiency benefits* are expressed in ad-hoc analysis of the whole process of legal case management from its opening until its closure. This also includes more elaborated ways of staff productivity monitoring.
- *Customer intelligence benefits* lay in the area of customer activity and behaviour improved monitoring. This includes mitigation of the visualisation shortages of Google Analytics.

- *Organisational (strategic) performance* is a subject of later measurements as strategic enhancements require time to emerge.

Achievement of main objectives

- *Self-service BI.* The ultimate idea of establishing self-service BI was to provide an intuitively easy way for business users to build required analytics by themselves. Assuming that all technical aspects like setting up data connections and some data modelling aspects like creating data sources are managed by IT and BI departments, it can be stated that self-service BI is available with current solution.
- *Better BI tool.* The goal of finding a suitable BI tool able to provide powerful dashboards, manageable reporting and ad-hoc analysis was also fulfilled. Tableau software confirmed its high user rating [47] and revealed its efficiency being evaluated on company's data and the OLAP schema of current solution. In addition having Google Analytics among Tableau's data sources helped to build a solution with single entry-point for all types of analytics.
- *Reduce complexity of analytics.* The issue of complexity of analytics was addressed by the OLAP schema that introduced clear naming and calculated fields, hid joins into fact tables and eliminated redundant fields. Section 3.7.5 and it subsection 3.7.5.1 have demonstrated this fact. Current BI solution also increased the level of atomisation by setting up a procedure to calculate historical data that was previously stored daily in Google drive, as described in 3.4.1 and 3.3.

3.9.1 Process perspective factors

Ease of use

From technical point of view, once deployed the solution doesn't require any maintenance efforts except ensuring regular execution of the schema-populating script. This is provided by the fact that OLAP schema addresses not only existing analytics but current requirements and even proposed requirements of the nearest future. This also requires collaborative actions from IT-team such as, for instance, reflection of operational changes into OLAP schema.

As for the business users, time and efforts invested in getting acquainted with new software will be paid back as soon as they begin to create their own analytics. It is important to note that end users need to familiarise themselves only with basic features such as drag-and-drop of measures and dimensions on a canvas (similar to known pivot

tables in Excel), drill-down, slicing, filtering, grouping values operations and few others. The solution supposes IT and BI department to manage all its technical aspects like setting up data connections and some data modelling aspects like creating data sources, so that the end user could leverage the power of BI software to work with well-prepared, clearly-understandable data in intuitively-easy fashion.

User access

Presented solution addresses the needs of all five targeted departments. In particular OLAP schema covers required analysis for Customer Care, Legal and Business Development departments completely, most of the analysis for Finance department and some part for Marketing department. Remaining part of analysis for the two latter departments is based on the data from Google Analytics. Due to the Tableau software that provides data connection to Google Analytics it is possible to cover analytical requirements from all departments by single solution. In addition, analysis based on Google Analytics data can be covered by OLAP schema as soon as this data is integrated into the system that is among the future work directions discussed in 3.10.1.

Section 3.7.5 has provided examples of current and required analysis for all departments including prospective analytics on flights presented by *Problem Flights* fact table.

Skilled (qualified) sufficient staff/team/managers

Not counting the head of Business Development department that defined the direction towards business goals, the author of this thesis was de-facto a program manager, designer and implementer in one face. Having a small BI team working on this project it would be possible to produce more breadth-wise solution within the same amount of time. Instead the emphasis was made on a depth-wise approach where less mature but working right now solution is preferred to more functional but not integrated one.

3.9.2 Infrastructure perspective factors

Ubiquitous BI system

At the considered business, inconsistencies were found at both back-end and front-end levels of BI system: operational changes by IT-team influenced OLAP schema, while miscommunication between departments on BI topic led to erroneous perception or complete misunderstanding of some analytical metrics.

Data Quality

Minor data quality issues have been discovered (inconsistencies with case statuses) and even some data cleaning has been executed (redundant case statuses have been removed)

during the implementation of the BI solution at the considered company. However the benefits of earlier data cleaning at operational level were practically confirmed by additional overhead and incorrect analytics resulting from data quality issues.

Data Management

At the considered company some terminology mostly related to *Legal* department was not used consistently. Sometimes it created barriers in understanding of some analytics as analytics was created by IT-team whereas consumed by other departments. During the implementation of this project the naming convention was considered and it was decided to be introduced to the personnel along with BI solution.

3.10 Case study conclusions

The considered case study showed an example of how BI roadmap can be applied in SME. Most importantly, it has demonstrated typical problems that may be found on the way of BI adoption in SMEs. In particular employees may not present their operational experience in metric-oriented way, such as formulation of KPIs. Users are biased by the existing tools and solutions that makes for them hard to discuss about new analytical perspectives. The study has shown an importance of discovering information about business operations, proper identification of KPIs, business requirements and technical characteristics. The fact that management tends to choose cheaper software rather than those, that satisfies business requirements more, may imply hidden costs of maintenance. Finally, its important that BI adoption initiative are spread throughout the whole enterprise.

3.10.1 Future work

Discussing fixed amount of time for implementation of the thesis project it was stated that current solution presents a certain prototype that can be extended and enhanced in the future. Below the future work directions are discussed.

OLAP schema

Current OLAP schema has a room for extensions. First of all, major directions of development, related to creating new dimension and fact tables, are connected with importing Google Analytics data and merging it with other back-end data that is discussed below in *Data integration* paragraph.

Fact tables can be supplemented with more dimensional attributes. For instance F_Documents can potentially have foreign keys to Dim_User, Dim_Advocate, Dim_Airline and Dim_Customer as the underlying table has respective fields, though containing null values at the moment.

Daily calculated metric *phone reachability during service hours* can be included in F_Historical_Data fact table. This and other metrics based on the data from Top-Link provider can be included as soon as the data from Top-Link is integrated into the system.

Data integration

Data integration is another direction of future enhancements. To recall, already mentioned Top-Link data is currently received in form of emails. The ETL task of extracting required data from these emails and placing it into respective database is among the tasks of IT-team for the nearest future.

The topic of integration Google Analytics data with other back-end data of the system has been raised several times in this thesis. The point is that currently to monitor user's behavior on the web-site the analysis provided by Google Analytics is used. At the same time user's actions are reflected in the operational database but these two analyses are not bounded, the user from web is not mapped to the user from back-end. Such kind of mapping could bring new insights for marketing strategy. The industry claims that it is possible to import Google Analytics data programmatically that is explained in details by Cutroni [59]. This method is also taken into account by BI team and management staff.

Other directions

As it is described in section 3.7.4 OLAP schema is regularly populated by the respective scripts. At the moment the script generates the schema from the scratch that can be potentially switched to incremental update. However this is not a trivial task as incremental update can not be established only based on the attributes of date/time type but requires more complex logic.

In principle, current scale of the business is not approaching big data level. However its future growth along with more data in terms of both volume and detalisation can at some point reach technical limitations of traditional relational DBMS. Some analytical queries take minutes to execute and operate with Gigabytes datasets. For instance delayed flights analysis deals with tens to hundreds of millions of records in respective tables. In case the business scales up to big data level, it will of course require not only to switch BI platform to the large-scale environment but all the system as a whole.

Chapter 4

Conclusions and future work

Current chapter concludes the thesis by summarising its main findings and contributions and discusses directions for future work.

4.1 Conclusions

Presented work has considered the importance as well as hurdles of exploiting BI in SMEs. A roadmap for leveraging BI as a growth engine was developed and proposed to be used in SMEs. The pivotal moments of this roadmap are listed below:

- *Get to know business from all sides.* Before specifying requirements for the BI solution, it is required to have a good understanding of business model, to define the scope of the BI project, to properly identify strategic goals, more specifically KPIs, to detect current problems as well as to expand analytical perspectives. Examination of the technical side from identification of analytical data sources up to decision on their integration should also be a part of this process that can be executed in a repetitive fashion across different organisational instances such as departments.
- *Specify BI requirements.* Results of the work done previously are used as an input for the specification requirements step. The output of this roadmap point consists of sets of business and technical requirements for BI solution including rhythm of analytics, depths of analysis, time-frame of analysis, etc.
- *On BI solution identification.* Considering both business and technical requirements, main questions about the architecture of prospective solution should be

answered. In particular, what is the place of the desired solution on the axis between self-service BI and IT-based solution, what is role and place for OLAP, at which level should BI be embedded, etc.

- *On development of OLAP component of BI solution .* Intermediate logical layer provides significant benefits at low cost. Existing analytical queries can be used for insights regarding design of the OLAP schema. Embedding simple dimensions into fact tables allows to reduce additional overhead. It is also preferable to execute data cleaning and to consider the usage of timestamps before the start of BI implementation.

- *BI tools evaluation.* A suggested approach requires to execute evaluation in two major steps. First, to choose a wide range of tools and evaluate them theoretically creating a short list of 3-5 tools for detailed evaluation. To get real impression about efficiency of the BI software it is necessary to execute detailed evaluation on operational data in real-life scenario. A list of features, that should be paid the biggest attention to, should be set. BI tools can be compared quantitatively (using the scheme from section 2.8.2) with respect to the list of features from above. Considering the price of BI software, the cost of additional IT-specialist should not be overlooked. This is also related to the nowadays trend of self-service BI, that can not be considered in a *purchase-install-use* way.

- *Success factors.* BI success should be measured after some time since the deployment of the BI solution. Among many factors impacting final success, factors of the following two categories are highlighted by the author of the thesis and SMEs' representatives from the research studies referenced in the document. Process perspective factors: ease of use, user access and skilled (qualified) sufficient staff/team/managers. Infrastructure perspective factors: ubiquitous BI system, data quality and data management.

Besides a theoretical framework of BI adoption in SMEs, the case study of its practical application has been provided as well. The case study presents an example of BI integration in real-life company that belongs to SME segment. It demonstrates the reflection of general ideas on a particular business as well as the influence of business peculiarities on BI solution development.

Summarising, nowadays SMEs have a great potential in using BI as a competitive advantage although it requires thorough consideration of pros and cons of cutting-edge technologies.

4.2 Future work

As any theory requires examples from practice to confirm or disprove it, BI roadmap for SMEs would also benefit from executing more case studies. Acquired feedback would be useful for the roadmap enhancements, detalisation and clarification of some of its parts, probably correction of some conclusions and its final shaping. In particular the following areas could be adjusted, detailed and enlightened:

- preparation activities related to the whole process of BI adoption
- participation of the management staff in definition of strategic goals
- influence of the Big Data level of business scale on the final solution
- the question of compatibility of different BI tools and differentiated BI solutions
- process of personnel education and training to use BI solution
- influence of personnel's experience and skills on this process
- maintenance of BI solution and its adaptivity to new requirements
- etc...

Appendix

```
select lc.id, lc.createdat,
CASE  WHEN lc.closedAt is not null THEN 'Closed'
  WHEN  (aa.advocateHiredIP is not null and aa.dataisVerified=true
  and aa.handoverpauseddate is null and aa.isAuthorized=true) then 'HandelingLawyer'
  ELSE 'HandlingByFR'
END as HandlingStatus,
lc.closedAt, lc.whereknown, if(lc.promotionCode like 'partner_ufd%', 'fluege.de',
lc.promotionCode), lc.agbVersion, ur.name as UnsuccessfulReason, lc.user_id,
lc.placeofjurisdictioncountry_id, lc.score, li.name, lc.claimOpponent_id,
substring(countrydomain from 2),
case lc.paymentstatusfg
  When 0 Then '0_balanced'
  When 1 Then '1_FR ows Customer'
  When 2 Then '2_Customer ows Flightright'
end,
CaseDates.date_PoA_Sent, CaseDates.date_PoA_received_FR, CaseDates.date_RoP_FR,
CaseDates.date_customer_Info_Requested, CaseDates.date_Parked, CaseDates.is_Successful,
CaseDates.closure_status
from legal_case lc
left join  unsuccessful_reason ur on lc.unsuccessfulReason_id=ur.id
left join  legal_insurance li on lc.legalInsurance_id=li.id
left join advocate_authorization aa on lc.advocateauthorization_id=aa.id
left join
(
  select lch.legalcase_id,
  date(min(if(lch.legalcasestatus_id in (102,108), lch.createdat, NULL))) as date_PoA_Sent,
  date(max(if(lch.legalcasestatus_id=1101, lch.createdat, if((documenttype_id=11 and d.isdeleted<>true),
    if(d.documentdate is  not null, d.documentdate, d.createdat), null)))) as date_PoA_received_FR,
  date(min(if(lch.legalcasestatus_id in (109, 111, 10013), lch.createdat,
    if(documenttype_id=1 and d.isdeleted<>true, d.createdat, NULL)))) as date_RoP_FR,
  date(min(if(statuses_id is not null and lch.legalcasestatus_id in (201, 202, 203, 206, 207, 208, 211,
    212, 213, 216, 217, 218, 221, 222, 223, 226, 227, 228,232,233,234,235,236, 301, 302, 303, 306, 307,
    308, 311, 312, 313), lch.createdat, null))) as date_customer_Info_Requested,
  date(min(if(lch.legalcasestatus_id in (701, 702, 703) and statuses_id is not null,
    lch.createdat, NULL))) as date_Parked,
  bool_and(case  when lch.legalcasestatus_id in(1002, 1003, 1007, 1008, 1010, 1011, 1012)
      and statuses_id is not null then true
      when lch.legalcasestatus_id in(1001, 1004, 1005, 1006, 1009, 1013, 1014) and statuses_id
        is not null then false
      else null end) as is_Successful,
  max( case  when lch.legalcasestatus_id between 1001 and 1014 then
    lcs.abbreviation else null end) as closure_status
  from legal_case_history lch
  left join document d on lch.legalcase_id=d.legalcase_id -- where lc.id>=13483
  left join legal_case_status lcs on lch.legalcasestatus_id=lcs.id
  group by lch.legalcase_id
) as CaseDates on lc.id=CaseDates.legalcase_id
```

FIGURE 1: Case dimension in OLAP schema script

```sql
select a.Airline, a.NumberCases as 'N of cases', b.NumberCases as 'N of cases last month', a.Claim as 'Claims volume',
b.claim as 'Claims volume last month', c.ClosedCases as 'N closed cases', c.ClosedSuccessfully as 'Closed successfully',
a.Claim/cl.TotalClaim*100 as 'Claim Share', a.NumberCases/tc22.TotalNCases*100 as 'Cases Share',
c.N_HandedOver/c.ClosedCases*100 as 'Handed Over Share',
e.CourtOrdered/c.N_HandedOver*100 as 'Court Ordered Share',
f.CFC_ClosedSuccessfully/e.CourtOrdered*100 as 'COF closed successfully'
from
(/* a - Total Number Cases and Claim per Airline*/
select legalnameshort as Airline, count(legalcase_id) as NumberCases, sum(claim) as Claim
from (/* y - detailed report*/
SELECT  lc.id as legalcase_id, a.legalnameshort,
sum((ifnull(c.claimAmount,0)*ifnull(c.claimable,0)+ifnull(lc.ticketprice,0)*ifnull(c.ticketPriceClaimable,0)-
ifnull(co.sumVoucher,0) - ifnull(co.sumCash,0))*c.numberPassengerClaimable) as Claim
FROM    claim c
        INNER JOIN
        (   SELECT  legalcase_id, min(createdAt) as createdAt
            FROM    claim
            GROUP   BY legalcase_id
        ) iclaim ON  (c.legalcase_id=iclaim.legalcase_id and c.createdAt=iclaim.createdAt)
left join legal_case lc on lc.id=c.legalcase_id
left join compensation co on lc.compensation_id=co.id
left join airline a on lc.claimOpponent_id=a.id
group by lc.id) as y
group by legalnameshort order by NumberCases desc limit 100) as a
left join
(/* b - Total Number Cases and Claim per Airline last week*/
select legalnameshort as Airline, count(legalcase_id) as NumberCases, sum(claim) as Claim
from (/* y - detailed report*/
SELECT  lc.id as legalcase_id, a.legalnameshort,
sum((ifnull(c.claimAmount,0)*ifnull(c.claimable,0)+ifnull(lc.ticketprice,0)*ifnull(c.ticketPriceClaimable,0)-
ifnull(co.sumVoucher,0) - ifnull(co.sumCash,0))*c.numberPassengerClaimable) as Claim
FROM    claim c
        INNER JOIN
        (   SELECT  legalcase_id, min(createdAt) as createdAt
            FROM    claim
            GROUP   BY legalcase_id
        ) iclaim ON  (c.legalcase_id=iclaim.legalcase_id and c.createdAt=iclaim.createdAt)
left join legal_case lc on lc.id=c.legalcase_id
left join compensation co on lc.compensation_id=co.id
left join airline a on lc.claimOpponent_id=a.id
and date_format(lc.createdat, "%Y-%m")=date_format(date_sub(current_date(), interval 1 month), "%Y-%m")
group by lc.id) as y
group by legalnameshort) as b on a.Airline=b.Airline
left join
(/*c - Number closed successfully per airline*/
SELECT  a.legalnameshort as Airline,  count(distinct if(lc.closedat is not null, lc.id, NULL)) as ClosedCases,
count(distinct if(lc.closedat is not null and lc.advocateAuthorization_id is not null and aa.exportDate
is not null, lc.id, NULL)) as N_HandedOver,
count(distinct if(legalcasestatus_id in (1002, 1003, 1007, 1008, 1010, 1011, 1012), lch.legalcase_id, NULL))
as ClosedSuccessfully
FROM    legal_case lc
left join airline a on lc.claimOpponent_id=a.id
left join legal_case_history lch on lc.id=lch.legalcase_id
left join advocate_authorization aa on lc.advocateAuthorization_id=aa.id
where lc.problemtransport_id is not null and statuses_id is not null
group by a.legalnameshort) as c on a.Airline=c.Airline
join
(
select sum(t.Claim) as TotalClaim
from (
SELECT  lc.id as legalcase_id,
sum((ifnull(c.claimAmount,0)*ifnull(c.claimable,0)+ifnull(lc.ticketprice,0)*ifnull(c.ticketPriceClaimable,0)-
ifnull(co.sumVoucher,0) - ifnull(co.sumCash,0))*c.numberPassengerClaimable) as Claim
FROM    claim c
        INNER JOIN
        (   SELECT  legalcase_id, min(createdAt) as createdAt
```

FIGURE 2: Top Airlines. Initial SQL-query listing. Part 1

```
            FROM    claim
            GROUP   BY legalcase_id
        ) iclaim ON  (c.legalcase_id=iclaim.legalcase_id and c.createdAt=iclaim.createdAt)
 left join legal_case lc on lc.id=c.legalcase_id
 left join compensation co on lc.compensation_id=co.id
 group by lc.id) as t ) as cl
 join
 (
 select count(*) as TotalNCases from legal_case) as tc22
 left join (
  SELECT  count(lc.advocateAuthorization_id) as N_HandedOver, a.legalnameshort as Airline -- , aa.exportDate
  FROM    legal_case lc
  left join airline a on lc.claimOpponent_id=a.id
  left join advocate_authorization aa on lc.advocateAuthorization_id=aa.id
  where  lc.advocateAuthorization_id is not null and aa.exportDate is not null
  group by a.legalnameshort) as d on a.Airline=d.Airline
 left join
 (
 SELECT  a.legalnameshort as Airline,
  count(distinct if(legalcasestatus_id=603, lch.legalcase_id, NULL)) as CourtOrdered
 FROM    legal_case lc
 left join airline a on lc.claimOpponent_id=a.id
 left join legal_case_history lch on lc.id=lch.legalcase_id
 left join advocate_authorization aa on lc.advocateAuthorization_id=aa.id
 where  lc.advocateAuthorization_id is not null and aa.exportDate is not null and lc.closedat is not null
 group by a.legalnameshort) as e on a.Airline=e.Airline
 left join
 (
 SELECT  a.legalnameshort as Airline,
 count(distinct if(legalcasestatus_id in (1002, 1003, 1007, 1008, 1010, 1011, 1012), lch.legalcase_id, NULL))
 as CFC_ClosedSuccessfully
 FROM    legal_case lc
 left join airline a on lc.claimOpponent_id=a.id
 left join legal_case_history lch on lc.id=lch.legalcase_id
 left join advocate_authorization aa on lc.advocateAuthorization_id=aa.id
 where lc.problemtransport_id is not null and statuses_id is not null
 and lc.id in
 (
 SELECT  distinct lc.id
 FROM    legal_case lc
 left join legal_case_history lch on lc.id=lch.legalcase_id
 left join advocate_authorization aa on lc.advocateAuthorization_id=aa.id
 where  lc.advocateAuthorization_id is not null and aa.exportDate is not null and lc.closedat is not null
 and legalcasestatus_id=603)
 group by a.legalnameshort
 ) as f on a.Airline=f.Airline
```

FIGURE 3: Top Airlines. Initial SQL-query listing. Part 2

```
select Airline, N_of_cases, N_of_cases_last_month, Claims_volume, Claims_volume_last_month, N_closed_cases, N_cases_closed_succ,
round((Claims_volume/TotalClaim*100)::numeric, 1) as claim_share, round((N_of_cases::double precision/TotalNCases::
double precision*100)::numeric,1) as cases_share, case when N_closed_cases=0 then 0 else round((handed_over::
double precision/N_closed_cases::double precision*100)::numeric, 1) end as handed_over_share, case when handed_over=0 then 0
else round((court_ordered::double precision/handed_over::double precision*100)::numeric, 1) end as court_ordered_share
from
(
select da.name as Airline, count(dc.id) as N_of_cases, sum(if(age(dc.created_at)<= interval '1 month', 1, 0))
as N_of_cases_last_month, sum(dc.initial_claim_amount) as Claims_volume, sum(if(age(dc.created_at)<= interval '1 month',
dc.initial_claim_amount, 0::double precision)) as Claims_volume_last_month, sum(if(dc.Handling_Status='Closed', 1, 0))
as N_closed_cases, sum(if(dc.is_successful, 1, 0)) as N_cases_closed_succ, sum(if(hand_over.handed_over, 1, 0))
as handed_over, sum(if(hand_over.court_ordered, 1, 0)) as court_ordered
from bi.Dim_Case dc
left join
(
    select FK_Case, true as handed_over, if(date_Court_Order is null, false, true) as court_ordered
    from bi.F_Adv_Authorization
    where authorization_status='Exported'
) as hand_over on  dc.id=hand_over.FK_Case
left join bi.Dim_Airline da on dc.FK_Airline=da.id
group by da.name
) as Airline,
( select count(id) as TotalNCases from bi.Dim_Case) as TotalCases,
( select sum(initial_claim_amount) as TotalClaim from bi.Dim_Case) as TotalClaim
order by cases_share desc
```

FIGURE 4: Top Airlines. OLAP schema SQL-query listing.

Bibliography

[1] Advanced Analytics and Business Intelligence. HaloBI presentation. URL https://www.youtube.com/watch?v=oNNk9-tmsZY#t=215.

[2] Aberdeen Group. Business Intelligence for the Small to Medium Sized Business. Aberdeen Group, Oct 2008. URL http://www.slideshare.net/findwhitepapers/business-intelligence-for-the-small-to-medium-sized-business.

[3] W. W. Eckerson. The keys to enterprise business intelligence: Critical success factors. TDWI. URL http://tdwi.org/research/2005/05/mr-the-keys-to-enterprise-business-intelligence-critical-success-factors.aspx.

[4] Gartner. BI and Analytics Market Trends - 2020 vision. Gartner Business Intelligence & Analytics Summit, Grapevine, TX, USA, Mar 2013.

[5] Fact and figures about the EU's Small and Medium Enterprise (SME). European Commission's Directorate General for Enterprise and Industry website. URL http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/.

[6] SAP BI tools for SME. SAP official website. URL http://www.sap.com/solution/sme/software/analytics/compare.html.

[7] Smeebi. Smeebi official website. URL http://www.smeebi.com/product.

[8] ElegantJBI. ElegantJBI official website. URL http://www.elegantjbi.com/.

[9] Oksana Grabova, Jerome Darmont, Jean-Hugues Chauchat, and Iryna Zolotaryova. Business intelligence for small and middle-sized entreprises. *SIGMOD Rec.*, 39(2):39–50, December 2010. ISSN 0163-5808. doi: 10.1145/1893173.1893180. URL http://doi.acm.org/10.1145/1893173.1893180.

[10] D. Scau R. Bell E. Kim A. Valnic C. Ballard, D. Herreman. *Data Modelling Techniques for Data Warehousing.* IBM Red Book, 1998.

[11] Matteo Golfarelli, Dario Maio, and Stefano Rizzi. Conceptual design of data warehouses from e/r schema. In *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences-Volume 7 - Volume 7*, HICSS '98, pages 334–, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 0-8186-8251-5. doi: 10.1109/HICSS.1998.649228. URL `http://dx.doi.org/10.1109/HICSS.1998.649228`.

[12] C. J. Date. *An introduction to Database systems*. Pearson Education, Inc, 8th edition, 2004.

[13] H. Dresner. Small and Mid-Sized Enterprise BI Market Study. Dresner Advisory Services, Nov 2013.

[14] T. Guarda, M. Santos, F. Pinto, M. Augusto, and C. Silva. Business Intelligence as a competitive advantage for SMEs. In *International Journal of Trade, Economics and Finance, Vol. 4, No. 4*, Aug 2013.

[15] M. Canes. Business Intelligence for the SME. Blue Link Associates Ltd. URL `http://www.bluelinkerp.com/newsroom/whitepapers/BusinessIntelligenceforSME_whitepaper.asp`.

[16] O. Suleymanov. Business Intelligence Strategy for SME's. Grenoble Graduate School of Business. URL `http://www.slideshare.net/olimjons/business-intelligence-strategy-for-smes`.

[17] P. Scholz, C. Schieder, C. Kurze, P. Gluchowski, and M. Boehringer. Benefits and challenges of business intelligence adoption in small and medium-sized enterprises. In *18th European Conference on Information Systems*, 2010.

[18] A. E. and C. Baransel. Architecturing business intelligence for SMEs. In *IEEE 36th International Conference on Computer Software and Applications*, 2012.

[19] Criteria for selecting a business intelligence, reporting and analysis solution. Visual Intelligence. URL `http://www.visualintelligence.co.nz/Content.php?article_uid=34d8da82-ac08-102c-b170-0022191d1e2d`.

[20] M. Lock. Embedded analytics: Enhacing decisions with in-process intelligence. Aberdeen Group, Mar 2013. URL `http://v1.aberdeen.com/launch/report/benchmark/8344-RA-embedded-business-analytics.asp?lan=US`.

[21] C. Imhoff. Business intelligence project pitfalls. BeyeNetwork, . URL `http://www.b-eye-network.com/view/1519`.

[22] L. Wise. Five steps to business intelligence project success. Technology Evaluation Centers, Sep 2009. URL `http://www.technologyevaluation.com/research/article/Five-Steps-to-Business-Intelligence-Project-Success.html`.

[23] Celina M. Olszak and Ewam Ziemba. Critical success factors for implementing business intelligence systems in small and medium enterprises on the example of upper silesia, poland. In *Interdisciplinary Journal of Information, Knowledge, and Management - Volume 7*, 2012.

[24] S. J. Moore E. Fink. 5 best practices for telling great stories with data. Tableau white paper, 2012. URL `http://www.tableausoftware.com/sites/default/files/whitepapers/whitepaper_best-practices_telling_great_stories.pdf`.

[25] S. Jackson. *Cult of Analytics*. Butterworth - Heinemann, 1st edition, 2009.

[26] Google Analytics. Google Analytics official website. URL `http://www.google.com/analytics/`.

[27] Fostering a data-driven culture. Economist Intelligence Unit, 2013. URL `http://www.tableausoftware.com/sites/default/files/whitepapers/tableau_dataculture_130219.pdf`.

[28] Five levels of embedded BI for your saas application. Jaspersoft Corporation, 2013. URL `http://www.jaspersoft.com/five-levels-of-embedded-bi-from-static-to-analytic-applications`.

[29] R. Kimball and M. Ross. *The Data Warehouse Toolkit*. Wiley Computer Publishing, 2nd edition, 2002.

[30] Oyku Isik, Mary C. Jones, and Anna Sidorova. Business intelligence success: The roles of BI capabilities and decision environments. *Information & Management*, 50(1):13–23.

[31] E. M. Rogers. *Diffusion of innovations*. Free press New York, 4th edition, 1995.

[32] L. Estrin, J. T. Foreman, and S. Garcia. Overcoming barriers to technology adoption in small manu-facturing enterprises (SMEs). technical report. In *Pittsburgh: Software Engineering Institute*, 2003.

[33] I. Becerra-Fernandez R. Sabherwal. *Business Intelligence: Practices, Technologies, and Management*. John Wiley & Sons, Inc., 1st edition, 2010.

[34] M. Z. Elbashir, P. A. Collier, and M. J. Davern. Measuring the effects of business intelligence systems: The relationship between business process and organizational

performance. *International Journal of Accounting Information Systems*, (4): 135–153.

[35] T. Ariyachandra and H. Watson. Which data warehouse architecture is most successful? In *Business Intelligence Journal.*, 11(1), 2006.

[36] S. and N. Williams. Critical success factors for establishing and managing a BI program. In *The Profit Impact of Business Intelligence*, Elsevier, Inc., 2007.

[37] W. Yeoh and A. Koronios. Critical success factors for business intelligence systems. In *Journal of Computer Information Systems.*, 50(3), pages 23–32.

[38] C. Imhoff. Business intelligence – five factors for success. BeyeNetwork, . URL `http://www.b-eye-network.co.uk/view/252`.

[39] Salesforce. Salesforce official website. URL `http://www.salesforce.com/`.

[40] OAG. OAG official website. URL `http://www.oag.com/`.

[41] TopLink. TopLink official website. URL `http://www.toplink.de/`.

[42] IAB. IAB official website. URL `http://www.internet-access.de/`.

[43] Weather Underground. WU official website. URL `http://deutsch.wunderground.com/`.

[44] Inter.net. Inter.net official website. URL `http://www.inter.net/`.

[45] Domain Factory. Domain Factory official website. URL `http://www.df.eu/int/`.

[46] Klipfolio. Klipfolio official website. URL `http://www.klipfolio.com/`.

[47] Gartner. Magic Quadrant for Business Intelligence and Analytics Platforms. © 2014 Gartner, Inc. and/or its Affiliates, Feb 2014. URL `http://www.gartner.com/technology/reprints.do?id=1-1QLGACN&ct=140210&st=sb`.

[48] D. Huefner. 5 ingenious KPI dashboards for startups and companies. © 2014 yeebase media GmbH, Jan 2014. URL `http://t3n.de/news/kpi-dashboards-startups-525365/`.

[49] Ducksboard. Ducksboard official website. URL `https://ducksboard.com/`.

[50] Cyfe. Cyfe official website. URL `http://www.cyfe.com/`.

[51] Sumall. Sumall official website. URL `http://www.sumall.com/`.

[52] Qlik. Qlikview official website. URL `http://www.qlik.com/`.

[53] Microstrategy. Microstrategy official website. URL http://www.microstrategy.com/.

[54] Jaspersoft. Jaspersoft official website. URL http://www.jaspersoft.com/.

[55] SpagoBI. Spagobi official website. URL http://www.Spagobi.org/.

[56] Tableau. Tableau official website. URL http://www.tableausoftware.com/.

[57] Statsmix. Statsmix official website. URL http://www.statsmix.com/.

[58] HaloBI. HaloBI official website. URL http://www.halobi.com/.

[59] J. Cutroni. Merging google Analytics with your Data Warehouse. DIGITAL ANALYTICS FOR BUSINESS. URL http://cutroni.com/blog/2011/05/05/merging-google-analytics-with-your-data-warehouse/.