

Anna Lopatenko

# QlikView as a Data Management Solution

---

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Media Engineering

Thesis

13 November 2015

Author(s) Title	Anna Lopatenko QlikView as a Data Management Solution
Number of Pages Date	35 pages 13 November 2015
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Specialisation option	Audiovisual Technology, Programming
Instructor(s)	Aarne Klemetti, Senior Lecturer
<p>The purpose of this thesis project was to conduct a study of data warehousing concepts and compare the technology to the data processing concepts used in QlikView application development. The project aimed to simulate an environment where data warehouse building stages could be utilized and applied to a QlikView environment.</p> <p>The project consisted of a theoretical part which reviewed the main stages of data warehouse building including the information gathering and consolidation, data transformation and querying. It also provided an overview of business intelligence technologies and tools and an overview of the QlikView platform.</p> <p>The practical part was aimed to support the theory and included an implementation of the final application. The aim was to go through each stage of data report creation in QlikView starting from the raw data extraction and ending with building a visual dashboard. The project aimed to compare both data warehousing and QlikView technologies and estimate the best possible solution for designing a centralized data system and creating further data analyses.</p> <p>On the completion of the project it was concluded that the optimal solution for today's business needs would be using a centralized data warehouse as a base or as one of the source data for future QlikView projects.</p>	
Keywords	QlikView, data warehouse, OLAP, data visualization

## Contents

### List of Terms and Abbreviations

1	Introduction	3
2	Business Intelligence and Data Warehousing	5
2.1	History of Visualization	5
2.2	Data Warehousing Concepts	6
2.2.1	Data Warehouse Design	9
2.2.2	Data Warehousing Characteristics and OLAP	11
2.3	Business intelligence and QlikView	12
2.3.1	QlikView Organization and Data Architecture	15
2.3.2	QlikView Data Modeling	18
3	Project Lifecycle	20
3.1	System Background and Overview	20
3.2	System Analysis and Requirements Specifications	20
3.3	System Design and Implementation	21
3.3.1	Database Design and Data Modeling	21
3.3.2	QlikView Queries and OLAP	25
3.3.3	User Interface and Visualization	28
4	Results and Discussions	31
5	Conclusions	32
	References	34

## List of Terms and Abbreviations

BI	Business intelligence
DBMS	Database management system
Dimension	A table consisting of attributes that represent the definitions of facts items.
DW	Data warehouse
ER	Entity-relationship data model
ETL	Extraction, transformation, loading
Fact	A data table consisting “fact” and measures data
OLAP	Online analytical processing
RAM	Random-access memory
QlikView	A data discovery platform and a data visualization software.
QVD	QlikView data files
RDBMS	Relational database management system
UI	User interface

## 1 Introduction

Nowadays more and more companies are taking into use different data analysis and data visualization technologies and solutions. This is closely related with an overall growth of data volumes and digitalization of pre-existing information. It also helps to collect the data for further evaluation and sharing it with other branches and interested parties.

Current IT market offers the variety of solutions that helps to solve arising data problems. The choice of the right tool depends on the company's size, its role in the market, strategy and mission, and the goals it aims to reach. Before selecting the tools it is crucial to evaluate the tasks and problems to be resolved, the scope and the volumes of the project, and the budget/money that the company is willing to spend/invest.

In last decades the amount of data being stored has increased dramatically. When a company needs quick access to data and numbers it tends to choose fast and easy to use solutions for quicker results. In many of cases old applications, such as spreadsheets, cannot handle large data sets and provide the desired results in building data reports due to their limitations. Bigger data volumes create bigger needs and new demands.

There currently exist a number of tools for data analysis and visualization. QlikView was chosen as the main tool for carrying out this project. The final outcome is a dashboard used for data analysis and decision making. However, this project puts an accent at a different and very important perspective of data analysis. It is the data management process that includes data gathering and consolidation. The main goal of this study is to examine and evaluate QlikView as a possible data management solution and to decide whether it could be used to replace traditional data governance tools. This paper covers the commonly known data management processes and compares them to the QlikView.

This project consists of theoretical and practical parts. The theory part covers the data visualization history and background and looks at data warehousing and business intelligence tools. The practical part covers system design and implementation stages of

the final QlikView application. The project included implementation of the test application for demonstration purposes; however, testing and deployment of the system into production environment were not included.

Although this thesis studies particular software as a data management tool, the results of this study may be applicable to other similar existing data discovery tools.

## 2 Business Intelligence and Data Warehousing

### 2.1 History of Visualization

Throughout the history visualization has served as a means of communication and data representation. Different kinds of methods of data representation have been used by people for thousands of years [1,8]. Visualization came out as a necessity for information sharing in commerce, traveling and communication. Visualization served as an addition or a substitute for textual and verbal information. It provided a compact, rich and easy-to-understand description of the written texts [2,3].

Today's trends demonstrate the interest in data gathering and consolidating. Data is seen as a valuable asset for companies looking to improve their processes and strengthen their businesses. Information is being collected and merged which allows having a wider overview and provides multiple ways of presenting and analyzing data. Previously implemented/existing methods of data visualization are used in support with technology [2,3].

Modern visualization uses digital media as the source of information and as the output tool. With the growth of technology and internet facilities people nowadays deal with a huge amount of data on a daily basis and many are familiar with graphs, charts and maps [2,19]. The original ways of data visualization have been extended and more widely used. Current solutions use different kinds of techniques and technologies for a visual representation of information. Various computational algorithms and processes, user control and other tools and components are combined in order to produce valuable results of data processing and analysis. In the end application users can define the display by posing the queries and this way interact with the data [2].

Visualization is able to provide a very wide or very narrow view of the data. It allows drilling down to very low level with possibility to see even smallest details. The examples of modern data visualizations are simulations, medical reconstructions, bioinformatics, business reports, and many others [2].

Whenever a new application is to be created the designer starts with an analysis of the data to be displayed and the type of information the user wants to retrieve from the presented display. The data can come from a range of sources and have a simple or

very complex structure. The viewer may want to use presented information for the analysis, sharing or inspection purposes [2].

At the early stages of the visualization process data was static but today's technology enables users to assign variables and functions into data points that allows data to be dynamically changed depending on user's actions. The viewer himself, by using controls and filters, may choose what and how data is to be displayed in the application. Different users may have different perceptions and the output can be customized accordingly. Each change will affect the final output and will ideally provide the user with the desired result allowing the user to utilize or extract the data for further use [2].

Visualization is part of a large process of data exploration and analysis. It supports the goal of building a model that represents the desired data views and calculations and helps to identify the data trends, deviations, and errors and discover new knowledge [2,26].

At the initial stages any system or application development involves a data gathering process, whether it to be the customer or user requirements or company data. The collected information represents a base for future system and database design.

Visualization has become an important tool for integrated data processing. It enables people and businesses to easily communicate their needs and helps in decision making. The information presented in charts and tables can help to quickly analyze the data and make actions based on findings. The way data is presented can also impact the perceptions of the information and future decisions. Right visualization tools and techniques allow bringing viewers' attention to convenient figures or dynamics [2,3].

## 2.2 Data Warehousing Concepts

The role and benefits of using of a data warehouse cannot be underestimated. In the large scale data may contain historical, medical, regional, economical or financial information. The majority of companies use more than one data source such as operation, accounting, management, technical or sales systems. All those systems communicate and serve a common goal and business mission. Behind those systems there



are people involved in the process of building the enterprise's intelligence. All of them have their own duties and their own vision of the facts [4].

Analyzing the roots of raw data and the ways it is being created might expose the complexity of the data structure and future problems it may result in. Without proper harmonization of the existing information data processing and updating becomes very difficult and this results in growing costs and resources spends [4].

Thus, it seems that there is a need for a right system for handling and storing the data and providing the tools for the data analysis. Data warehousing sounds like a possible solution.

Data warehousing is a concept of collecting information to be used for reporting and data analysis. It aims to help businesses in making faster actions and improving their business processes [3]. The task is to ensure data integrity and to provide a centralized data storage to be used for business needs. The information stored in the warehouse can contain any kind of financial, marketing, operational and other business data [3,5].

Along with the relational database, the data warehouse includes ETL tools, an OLAP engine, client analysis tools and other solutions for gathering data and delivering it to the business users [3,5]. A good data warehouse is:

- subject oriented – concentrates on a particular subject such as sales, finance or inventory;
- integrated – represents a consistency of data retrieved from various sources;
- time variant – contains historical data;
- nonvolatile – consistent; historical data should not be modified;
- containing both detailed and summarized data

[3,5-6].

Data warehouse architecture includes tools for data extracting from various data sources, formatting and linking this data, and loading the normalized data into a warehouse [3,6].

The data warehouse development process can be summarized into several stages. The following stages represent the main concepts of data warehousing and are based on ETL processes which include extraction, transformation and loading of data into a data repository.

First stage is to choose a suitable DBMS – it has to be decided which DBMS to use and why.

**Extract.** Second, the following is the extraction, a phase which includes the retrieval of information from different sources such as files (spreadsheets, text files and others), databases, and other system and services. At this stage the collected raw data is to be extracted into a new database. This includes creating of new tables consisting of raw data that is going to be used later in the analysis (all company data from all sorts of sources). Each of these will become a separate table in this database.

**Transform.** After collecting all necessary data the next stage is to transform the data. During this step the data is converted from raw format and put into another properly normalized database.

Normalization has to be well thought out before starting the data transformations. During this process the data in the relational database is being organized and separated into multiple tables in order to minimize or remove data redundancy [3,65-66].

The transformation step also includes data processing such as numbers and currency formatting, null handling; fixing wrong and generating missing data.

**Load.** Finally, the transformed data can be loaded into the data warehouse. At this stage a new database is created that is going to reflect the questions to be answered. This is going to be a future data warehouse which can later be implemented into business specific views.

All three steps of extraction, transformation and loading can be sorted in separate databases. This helps to follow process and check every stage.

### 2.2.1 Data Warehouse Design

The result of the data consolidation process is a centralized database, a data warehouse (DW), that keeps the data meant to be used for further analysis by decision makers. This database represents a collection of all information about the enterprise and has several features:

- The data presented in this database is a result of transformation and quality optimization of the raw data.
- It aims to support complex queries and requests set by users.
- It connects users to data through the user-friendly end tools allowing user-to-data interactions without a need of a technical support [3,72].

A centralized data warehouse consists of all gathered and normalized corporate data. It represents a collection of all structured data or datasets defined and designed according to the enterprise model. These may include sales, financial, marketing, production or other databases. One of the common techniques used in the data warehouse design is dimensional modeling [3].

The main ingredients of dimensional modeling are facts, measures and dimensions.

A fact table is a larger master table representing a collection of related items (facts), consisting of measures and foreign keys to dimensions tables. A measure is a numeric attribute of a fact, representing the quantity, that can be later used in calculations and analyzed. Examples are prices, volumes, quantities, budget and sales [3,73-74].

A dimension table is a smaller fact-supporting table consisting of attributes that represent the definitions of facts items. The content of dimensions is usually textual and descriptive. The outcomes of the dimensional modeling will be used for building the requirements of future OLAP cubes. Dimensions are the parameters over which the OLAP is performed [3,73-74].

And how is the data held inside the data warehouse? Each database contains a corresponding data model that is used to define the relationships between the tables. The most common types are star and snowflakes models. A schema consists of database objects such as tables, views, measures, indexes, and synonyms [3,81].

The star schema represents a model where all dimensions are directly connected to the main fact table (see Figure 1). In this model the fact table is surrounded by various descriptive dimensions. The more complex star models can contain several fact tables [3,81].

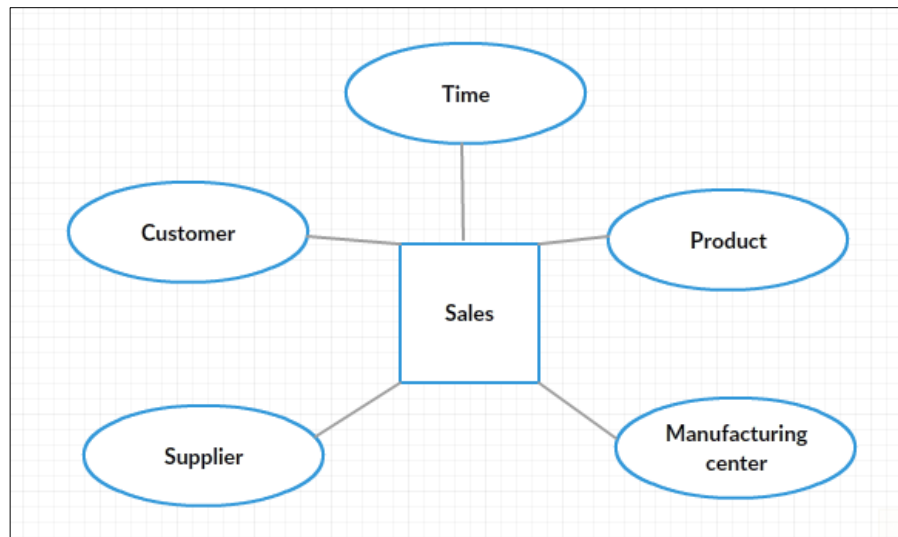


Figure 1. Star model.

The snowflake schema consists of indirect linking of attributes through other dimensional tables. In this mode fact dimensions can be split into separate tables [3,81]. In OLAP the snowflake model requires more joins between the tables and results in a reduced performance of the search outcomes [3]. Figure 2 below shows an example of snowflake model with a single fact table and several dimensions.

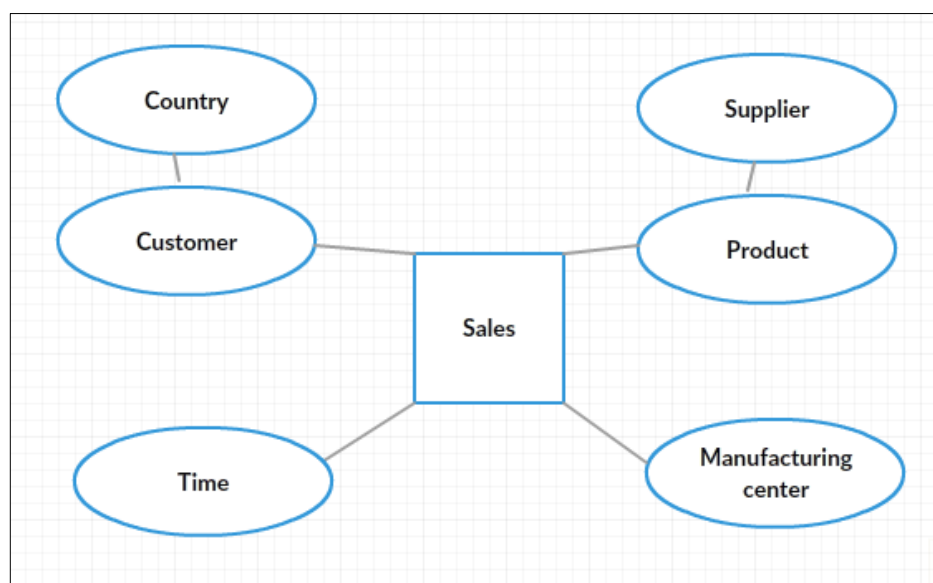


Figure 2. Snowflake data model.

### 2.2.2 Data Warehousing Characteristics and OLAP

Data warehouse provides a multidimensional view of data that allows returning the results based on the queries created by the analysts/users and decision makers. OLAP is designed to store data in the “multidimensional”/multifaceted preprocessed cubes that represent data in different angles in order to provide the optimal results for user queries [3,70].

On-line Analytical Processing (OLAP) is a technology that allows users to get insight to the data through fast and interactive summarized data views. It provides a multidimensional analysis of [consolidated) data and allows users to navigate through calculations, follow the trends over the time periods and drill-down to deeper levels along data hierarchy [3,68;5].

The technology itself is not dealing with huge volumes of data but due to consistency and historical perseverance of data OLAP is able to provide a great performance summarizing data in various ways. Regardless the amount of data stored in the database the answers to queries posed by users can be provided within short time [seconds) [3,70]. The key feature that allows OLAP to provide the desired results so quickly is that it contains the summarized multidimensional data [5].

The dimensional models of a data warehouse are used as basis in building of OLAP cubes which process the data and provide summarized information in multidimensional views (see Figure 3). This allows ensuring high performance and fast response to the queries sent by users [3,70].

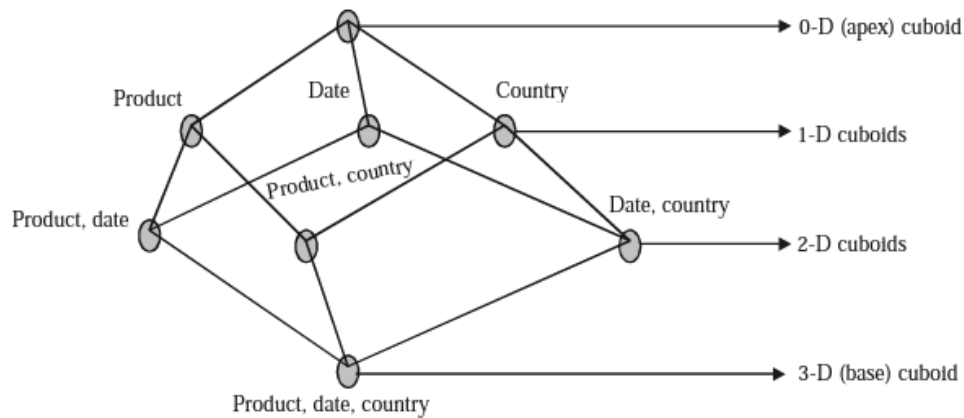


Figure 3. OLAP cubes. Reprinted from Prabhu (2007) [3,73].

The goal of data warehouse architecture is to meet the customer needs by:

- Delivering a good user experience;
- Providing strong analytical tools;
- Presenting a centralized storage of data;
- Handling complex queries and providing fast results

[3,77].

The key for building an efficient data warehouse is to ensure data integrity, keeping it simple for the user and splitting the bigger problems and business processes into smaller tasks, models and databases.

### 2.3 Business intelligence and QlikView

Business Intelligence is a set of tools and technologies used for presenting data to the end customer in a desired format for the purpose of decision making. It allows collecting the data from different resources and quick building of the reports with the possibility to display, drill-down and analyze the company data.

Common BI solutions:

Operational reporting – reporting that aims to reflect current activities.

Ad-Hoc reporting – a process aiming to solve a single, specific business problem on demand.

OLAP analysis – pre-calculated summarized data presented for faster data exploration.

Data visualization – a report with in-memory stored data and procedures.

Dashboards – visualized data reports and detailed data.

Business intelligence is just a part of the data processing and data visualization processes. The biggest work in data processing happens in data warehouses. So it is crucial for data to be well structured. Badly organized data may result in errors and problems when building and analyzing the BI reports.

The amount of data stored in the system and databases nowadays is constantly growing. With a growth of information being stored and passed around current users may need special tools that would allow them easily and quickly access and analyze the data.

There are currently several data discovery tools in the market which provide many different techniques for data visualization. Each of them has strengths and opportunities and the decision on choosing one of them can be endlessly discussed. QlikView is one the modern business intelligence tools and business discovery platforms that is meant for building visual reports for data analyses. The popularity of this tool seems to have been increasing in the last decade and it is being used more often by companies of any size [6,10].

QlikView is a multitask tool that provides the ability to extract and transform raw data, design a data model inside the application and build visual and interactive reports. Several noticeable features that characterize this platform and make it a possible good choice for building the data visualizations are:

- In-memory stored data allows fast processing of selections and filtering possible offline work.

- Use of pre-joined data tables or so called *associative model*. This type of data linking allows all fields, despite of their location in the model, be filtered based on selection.
- Flexible expressions and a good set of data charts and other visualization objects

[6,10;7].

While a data warehouse is intended for creating a well-structured system/database, BI is just using the raw data for building the reports. Thus, the reports built with known data visualization (BI) tools can lack data integrity and result in misleading calculation and data representations. To some extent data warehousing can be done in QlikView with a help of QVD files. As another solution a complete data warehouse system can be used as a source system for QlikView reports.

QlikView can collect the data from the variety of sources and has good visualization capabilities but it is not that effective when it comes to managing big volumes of data, reusing, or harmonizing the metadata across those sources. When choosing QlikView as a possible tool for data visualization, businesses should take into considerations also known drawbacks and limitations of the platform.

QlikView is costly software, as most of the professional software and tools available on the market. Several things are to be taken into consideration such as cost of professional software, licenses, storage, additional tools and features [e.g. for data connectors to other systems), and outsourcing. All of these add extra costs.

The data stored in QlikView is not open-source data. QVD data files are designed to be supported by QlikView and are intended to be used only by this software.

On a big scale it is impossible to survive without a proper data warehouse and it may also be beneficial to use some additional ETL tools. The current setup of the software may satisfy the needs of rather small or very small businesses.

In-memory stored data and procedures may be a useful feature for getting fast filtering results or offline work, but it may also affect the report performance and its ability to handle and process data.



### 2.3.1 QlikView Organization and Data Architecture

The architecture of QlikView development process consists of several stages that represent the structure of different layers that exist between the source tables and final product. A good architecture helps in organizing the whole process of data reports building and organizing the information in the most efficient way. It also allows the application to be more easily scaled and expanded for future needs when the application should be further improved or developed. There are several architecture models most commonly used in the QlikView development environment [7,404].

The diagram below (Figure 4) demonstrates the two-stage data architecture model.

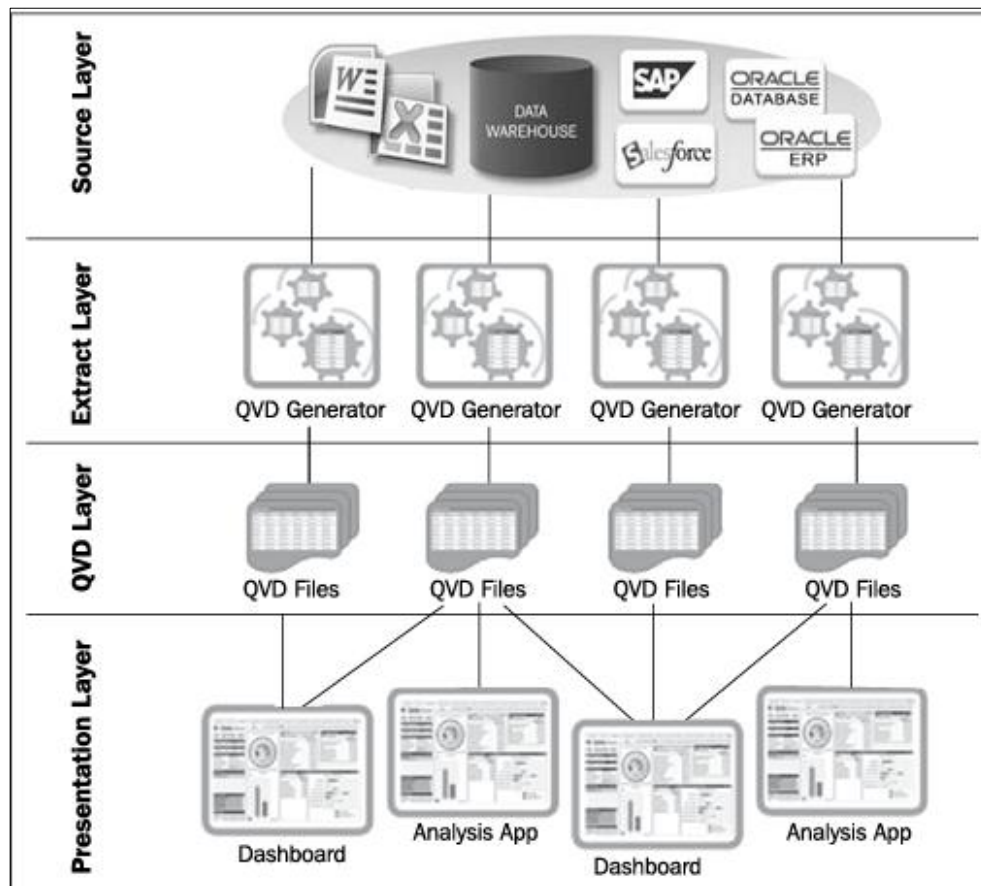


Figure 4. Two-stage QlikView data architecture [7,404].

This architectural model consists of following layers:

- The source layer represents the source data files, systems and data-bases.

- The extract layer consists of QlikView script documents used for generating the QVD data files. The source data is extracted into QVD files in a raw or normalized format.
- The QVD layer consists of QVD files that are used as source data for the final application.
- The presentation layer represents the actual final QlikView reports and dashboards. The generated QVD files in this layer are used as source data and can be further transformed if necessary [7,405].

Creation of a separate QVD layer is beneficial as it allows having a fixed data source that can be later reused in multiple QlikView documents.

The two-stage architecture is suitable for smaller applications where big data transformations are not required. But as the system grows and there is more need in data normalization there could be a need in enhancing the current data architecture in order to allow tracking and controlling the transformation process [7,406].

For these needs there might be a need in implementing more advanced three-stage data architecture (see Figure 5).

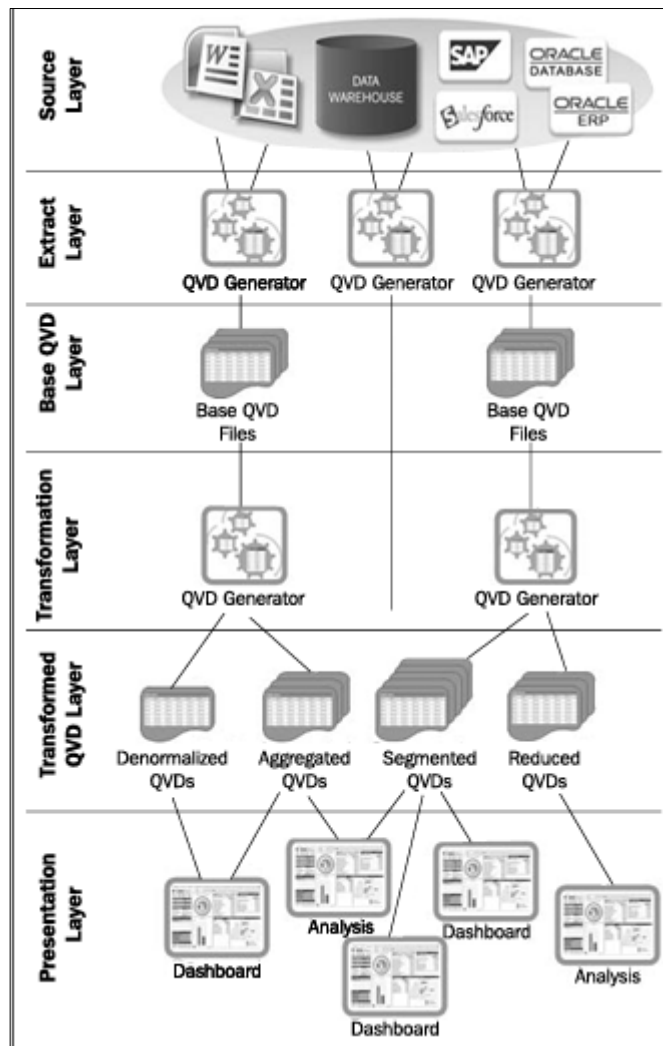


Figure 5. Three-stage QlikView data architecture [7,406].

The architecture is optimized by adding two additional layers – Transformation Layer and Transformed QVD Layer. The purpose of having these layers is to store the information about raw data transformation and normalization. This also allows preserving the Base QVD Layer holding the raw untransformed data. The Transformation Layer will combine all necessary raw base QVD files and processes them in a desired way. The transformed DVQ files will represent the Transformed QVD layer which, along with the Base DVQ Layer, serves as a base for the final Presentation layer [7,406-407].

Any QVD file, which contains raw or normalized data, can always be later reutilized by different QlikView documents. This allows avoiding duplicating unnecessary data and keeping the structure clean and well organized [7].

### 2.3.2 QlikView Data Modeling

As in many other applications and relational database systems, the most common modeling schemas used in QlikView are star and snowflake. Use of dimensional schemas improves the response time and affects the overall performance of the final application. The dimensional model can be optimized by adding new fact and dimension tables into the system [7,8].

Because of the nature of how the QlikView software works the data modeling happens during the data loading into the application. The data for the QlikView application can come from a variety of sources: databases, spreadsheets, the web and other systems. The connection of this data is performed through the associative data model. Any data model in QlikView becomes an associative data model as it contains the different source tables and automatically connects them based on the field/column names [7,94].

Different data modeling approaches can be used depending on the structure of the data. However, each of the data models has its own advantages and disadvantages. The amount and the size of the table stored in the application may affect heavily the end result. The table below compares the data models and explains the behavior of the application when one or another model is used [7,8.]

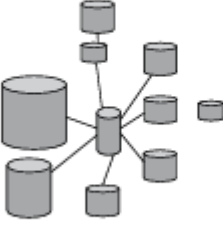
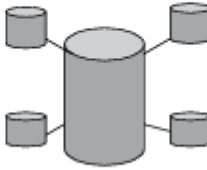

	Option 1 Snowflake	Option 2 Star Schema	Option 3 Single Table
			
Response Time	Satisfactory	Good	Excellent
RAM consumption	Good	Good	Bad
Script run time	Good	Excellent	Bad
Flexibility Model	Poor	Excellent	Excellent
Complexity Script	Poor	Excellent	Excellent

Figure 6. Data model comparison [7,93].

As it can be seen from the comparison table above (Figure 6), it seems that the star schema is an optimal and more balanced solution. But there is no right solution and the final choice of a model depends on the project requirements and data structure.

### 3 Project Lifecycle

#### 3.1 System Background and Overview

The goal of this thesis project is to demonstrate and evaluate data analysis process that could be performed with QlikView. The aim set by the project is to simulate a process of data warehousing building in QlikView and compare the results with traditional data warehousing concepts. The task is to define, design and implement the application for market share and production/material data analysis. The information represented in the application consists of the sample sales and market share data. The practical part aims to support the theoretical part.

#### 3.2 System Analysis and Requirements Specifications

The requirements are created based on the customer's or end user's ideas and specifications. The normal process would include documenting the customer's wishes and visions, translating them into requirements specifications and designing the final application.

As this project was carried out by a student and the application is built without any other parties involved, the student defines the specifications and outcomes of the project. The main requirement was to create a system that simulates a normal data warehousing design process and analyse the results of the study.

### 3.3 System Design and Implementation

#### 3.3.1 Database Design and Data Modeling

The starting point of data design and modeling in QlikView, as in any other database, is converting the ER model (Figure 7) into a dimensional model. Each business process is implemented into a separate data schema [7].

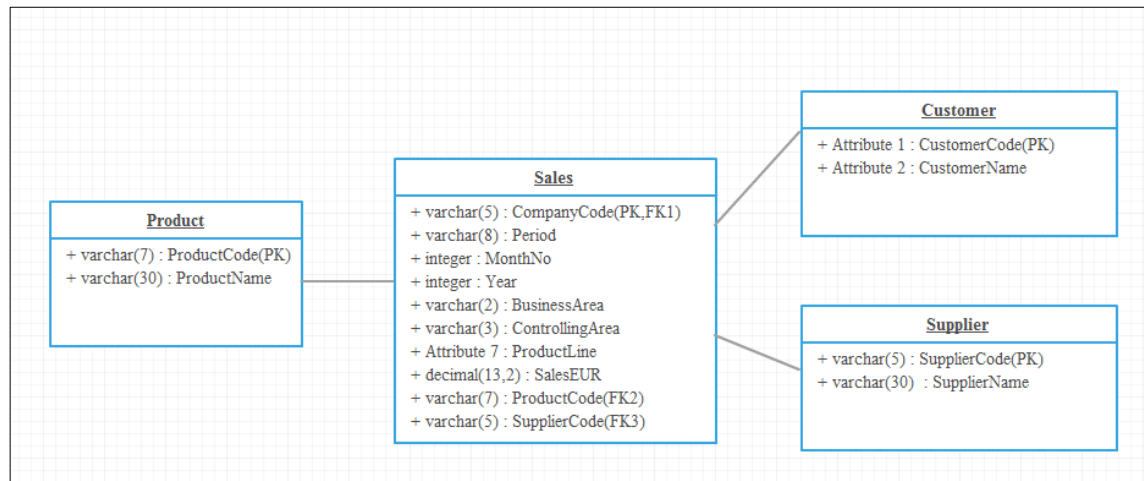


Figure 7. Entity-relational data model. Screenshot.

The following step is to collect all facts and measures into a fact table and flatten the remaining tables into dimensional tables and link them to the master table. The same dimensional tables may be reused in different models. This ensures the data consistency across the databases and applications [7,92].

The data modeling process in QlikView resembles the commonly known ETL process. All extraction, transforming and loading of data happens during the same loading process.

Extraction of the data in QlikView is done through the File Wizard or with a help of SQL based query language. The syntax may vary slightly, depending on the source of data. For accessing data from some other tools and databases, e.g. SAP, it may be required to purchase and install additional tools into the software. Data is loaded from a source file using the LOAD command and from the database with the SELECT clause as demonstrated in the Figure 8 below.

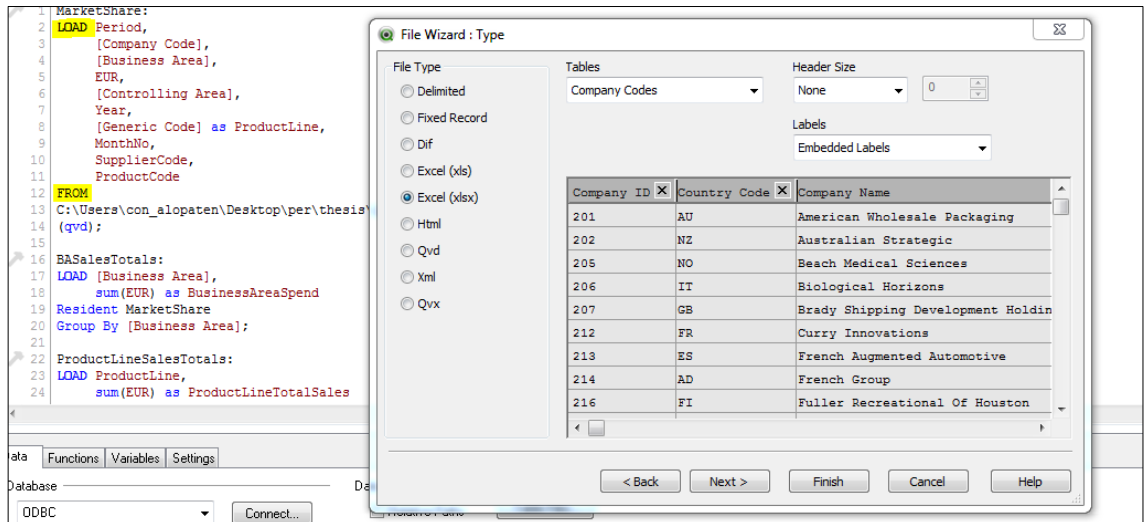


Figure 8. QlikView data load. Screenshot.

The data transformation step includes data cleansing, formatting and fixing. Raw data does not always come in a desired format and may consist of much unnecessary information or be stored in an improper form. The transformation starts with the selecting of data to be loaded and the way it is to be loaded into the report:

- all unnecessary information is sent to the garbage;
- only relevant columns and rows are selected;
- data fields are converted to relevant data types, missing data is generated [7].

Normalization of the data in QlikView happens in the same loading script. Technically, the data is split into appropriate data tables following the data cleansing, formatting and fixing. Figure 4 below shows the formatting of the fields done in the script and converting them into appropriate formats as well as organizing the data table by splitting the data or by combining the data from several tables or fact sheets into one final table. Each transformed table is stored in a separate QVD data file and later uses as a source base for future reports and applications. Combining multiple fact sheets is described later in this chapter. The Figure 9 below shows the process of data transformation.



```

8 SET timestampFormat='D.M.YYYY hh:mm:ss[.fff]';
9 SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
10 SET DayNames='Mo;Tu;We;Th;Fr;Sa;Su';
1
2 Supplier:
3 LOAD SupplierCode,
4     SupplierName
5 FROM
6 C:\Users\con_alopaten\Desktop\per\thesis\QVdocs\Suppliers.xlsx
7 (ooxml, embedded labels, table is Suppliers);
8 STORE Supplier into Supplier.qvd;
9 DROP Table Supplier;
10
11 Company:
12 LOAD [Company ID] as [Company Code],
13     [Country Code],
14     [Company Name],
15     [Country Area],
16     Country
17 FROM
18 C:\Users\con_alopaten\Desktop\per\thesis\QVdocs\COMPANY.xlsx
19 (ooxml, embedded labels, table is [Company Codes]);
20 STORE Company into Company.qvd;
21 DROP Table Company;
22
23
24 //combine multiple fact sheets into one fact sheet
25 MarketShare:
26 LOAD Period,
27     [Company Code],
28     [Business Area],
29     Money(EUR) as EUR,
30     [Controlling Area],
31     Year(Year) as Year,
32     [Generic Code],
33     num(MonthNo) as MonthNo,
34     SupplierCode,
35     ProductCode
36 FROM
37 C:\Users\con_alopaten\Desktop\per\thesis\QVdocs\MARKETSHARE.xlsm
38 (ooxml, embedded labels, table is [2012 DATA1]);
39 LOAD Period,
40     [Company Code],
41     [Business Area],
42     Money(EUR) as EUR,
43     [Controlling Area],
44     Year(Year) as Year,
45     [Generic Code],
46     num(MonthNo) as MonthNo,
47     SupplierCode,
48     ProductCode
49 FROM
50 C:\Users\con_alopaten\Desktop\per\thesis\QVdocs\COMPANY.xlsx
51 (ooxml, embedded labels, table is [Company Codes]);
52 STORE MarketShare into MarketShare.qvd;
53 DROP Table MarketShare;
54

```

Figure 9. QlikView data transformation. Screenshot.

After the data is organized and transformed, the final step is the loading of the data into the application from the data tables. At this step the data structure and data model are optimized and a developer can proceed with building the visualization and dashboards.

Loading of the raw data can happen directly from data sources into the QlikView application. Each data table is usually stored in a separate QVD document, a QlikView supported data file. These files are native data files designed for QlikView which allows reducing the final application data size due to compression and also improving the overall performance of the analysis reports by speeding up the query processing. A collection of the QVD files represents a QlikView database and is later used, along with other sources, as a data base for the report building.

In QlikView any data model becomes an associative model despite the type of source of the data. The tables are automatically connected between themselves based on the similar names fields. Although it may speed up the process of connecting tables inside the model, this feature can also result in wrong links between data tables and undesired data loops. In this type of model each field can have an associative link and act as a dimension in the charts or filtering objects [7,94].

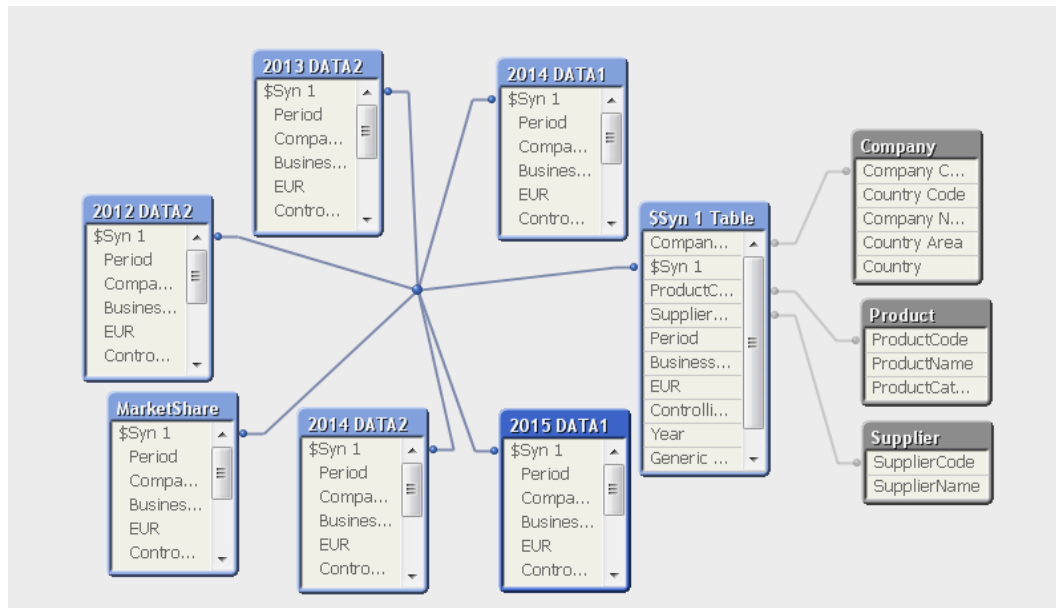


Figure 10. QlikView data model with loops and synthetic keys. Screenshot.

When multiple facts are to be used inside the QlikView the complexity of the data model increases and this may lead to occurrence of the loops and synthetic keys as shown in the Figure 10 above. In order to avoid this problem several fact tables could be concatenated into one fact table [7,92]. This generates a better data model and improves the performance of the final applications. Loops and synthetic keys, ideally, should be avoided in the application. Relevant information, tables or fields, should be combined by concatenating the appropriate tables or by producing composite keys for stronger data connection between the model entities.

The end result is a well-designed data model which consists of fact and dimensional tables directly or indirectly connected to the fact table. Naturally, QlikView itself does not care about fact and dimension tables. It is up to the project developer to decide how to organize the information tables and their structure. Normally, fact tables consist of primary “facts” data and dimension tables would consist of descriptive data and attributes (Figure 11).

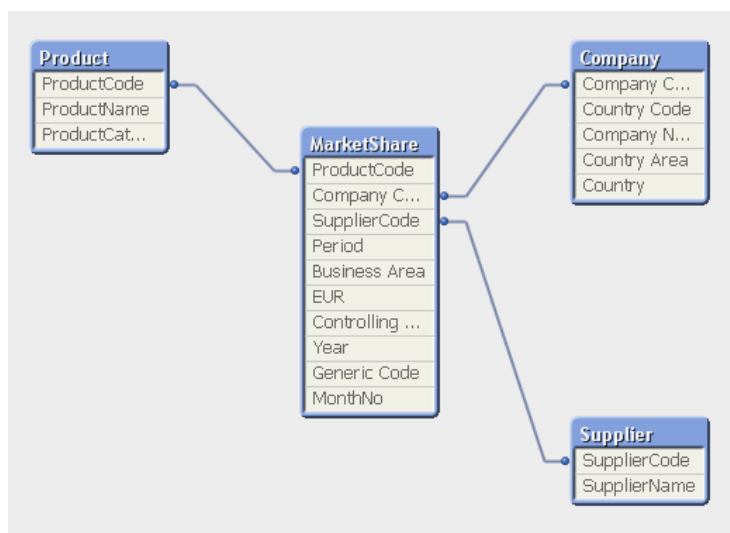


Figure 11. QlikView “clean” data model. Screenshot

QlikView uses many different sources of data and does not require a separate database or data warehouse as such to be created. In fact, as a memory-based software it creates the data structure inside the application and data tables links are created between preloaded QVD files. QVD files are data files supported by QlikView. The process of the creation of QVD files resembles the data warehousing process which includes extracting, formatting and loading of data. However, on a big scale for very big projects using a separate data warehouse or ETL tools might be necessary. This will help to provide a better structure for the data organization and ensure better results of data querying and visualizations.

### 3.3.2 QlikView Queries and OLAP

Querying in QlikView happens inside the load script as well as through the application layout and its objects. Visual objects with filtering possibilities on the dashboard allow the user to get a wide or narrow overview of the data. In UI users can have various hypercubes in forms of Charts or Pivot tables, which are calculated on demand. Each of these objects represents aggregated data views and may be queried by a simple click on the filter or the chart or table area [7].

In the script side of the QlikView application it is also possible to build the querying views similarly to OLAP cubes. Sometimes there might be a need to read the same

table more than once inside one document. In this case the data is loaded and reprocessed in a way that is more relevant for the application needs. Once the table is loaded into the memory of the application it is possible to access the same table by command Resident without the need of re-reading the original data source [7,401].

The resident load allows reprocessing the data that has been already pulled into the RAM memory and applying new queries on the same data. The output consists of aggregated data over a selected dimension [7].

As the first step the data is loaded from the original source and is stored in a RAM model (see Figure 6). Adding a title to the table inside the script allows later setting the references to it in the script whenever necessary. In this case I am planning to use *MarketShare* table for calculating the needed information.

```
MarketShare:
LOAD Period,
      [Company Code],
      [Business Area],
      EUR,
      [Controlling Area],
      Year,
      [Generic Code],
      MonthNo,
      SupplierCode,
      ProductCode
FROM
C:\Users\con_alopatzen\Desktop\per\thesis\QVdocs\MarketShare.qvd
(qvd);
```

Figure 7. Fact table loading. Screenshot.

The *Resident* command in the same loading script, accessing the same table, allows aggregating data using a *Group By* command. When referencing the tables preloaded into RAM, it is important to refer to the name of the fields as they are defined in the script which may differ from the names that are stored in the original raw data. In other words, if the field name was renamed, the new name of the field would be used a reference in the resident load [7,411]. An example of Resident load can be seen in the following Figure 12.

```

5 BASalesTotals:
7 LOAD [Business Area],
8     sum(EUR) as BusinessAreaSpend
9 Resident MarketShare
10 Group By [Business Area];
11
12 ProductLineSalesTotals:
13 LOAD ProductLine,
14     sum(EUR) as ProductLineTotalSales
15 Resident MarketShare
16 Group By ProductLine;
17

```

Figure 12. Resident table loading. Screenshot.

As a result the latest data model consists of information on a more detailed level and also summarized data for each business area and product chain [7]. It is now possible to access the summarized information directly and faster through the layout of the application (see Figure 13).

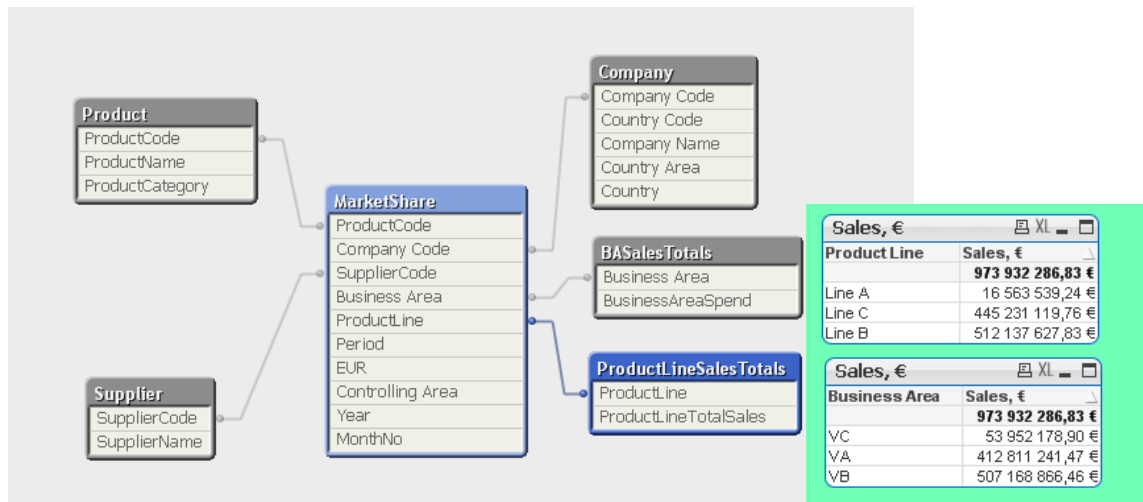


Figure 13. QlikView data model with summarized data tables. Screenshot.

After performing the final load of normalized data in the memory of the application a developer may proceed to building the report's user interface and visualize the processed data in many different views according to the end user needs.

### 3.3.3 User Interface and Visualization

A QlikView document, in general, is used to visualize the data. The various types of objects are used to display facts and dimensions in a way that would be useful for end analysis [6,7].

The purpose of the software is to visualize data in a way that would satisfy the end users. Thus, data visualization is the end result of the project. That includes displaying the existing data in form of text, numbers, calculations, tables and charts. This part combines technical and visual perspectives and requires a good eye and ability to build easy-to-understand and visually appealing layouts.

The final application consists of several objects that can be created using the user interface of QlikView. Table 1 below lists the main controls displayed in the dashboard interface.

Table 1. QlikView UI objects and controls.

Object	Object description
Button	An object with an assigned action or a combination of actions (for example, filters selecting and clearing or printing).
Chart	An object that displays the selected or calculated information in the form of a graph (different chart types), or a combination of text and trend lines.
Current Selection	A box for displaying the names of fields currently being selected and the names of the specific values being selected in the specified fields.
List Box	An object consisting of a single field/data column as a filter with

	the possibility to add any extra dimensions.
Search	A box that allows the searching of text through predefined fields. This object allows the user to enter multiple search entries. Search functions and outputs are predefined by the report developer.

One of the benefits of QlikView is that it provides the user with the ability to interact with the displayed data. All filters and selections affect the layout of the application and allow data representation to be changed dynamically. These features allow saving screen space, creating separate tabs, defining access rights for different user groups and dynamic filtering with possibility to drill-down. All fields in the QlikView application can act as dimensions. Thus data can be filtered in many different ways. Many screen objects, such as filtering boxes, charts, or pivot tables allow data filtering [7,6].

The final application layout represents an outcome of data processing and visualization (Figure 14). It consists of multiple document objects that allow users to navigate through data. Users can filter, drill-down and measure the information in many possible ways in order to get the desired outcomes. Final data analysis allows identifying possible data trends, errors and deviations and seeing the overall performance of a particular process that is being visualized. All these results help in making the appropriate decisions, taking the corrective actions and redefining the business strategies when and if necessary.

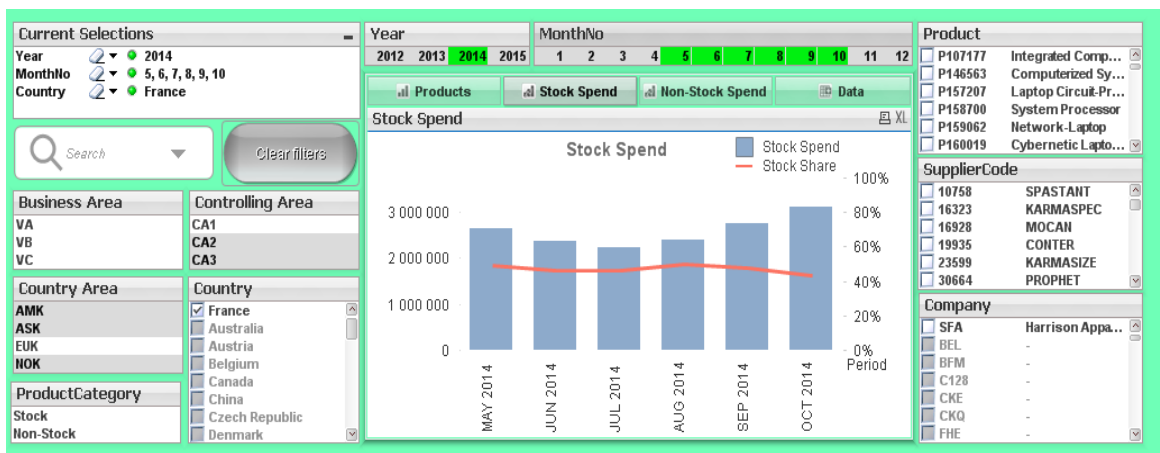


Figure 14. Final UI and user selections. Screenshot.





## 4 Results and Discussions

Although QlikView is known as a data visualization tool, this project was aimed to demonstrate the abilities of the QlikView software to handle large data sets. The fact that QlikView can load and store millions, and even billions of entries, and later process and display them in the report makes this product far superior than many known traditional applications such as, for example, Excel. However, the way this software manages the data demonstrates the obvious limitations that may affect the user's choice of the development tool.

The overall project flow was not that smooth due to some obstacles. Limited time and rather little experience with the software resulted in quite a small final application. Also, it was quite challenging to create a complex enough system with limited resources. The project's scale was not big enough and did not allow performing reasonable testing of the software and its abilities. Some knowledge and software limitations findings are partly technical and some are pretty theoretical. Data visualization, one of the most important features of the QlikView software, does require better visualization skills. Proper presentation of the data through different views and angles allows deeper analysis of the data.

The main challenge of this project was the scale which was very small. Building a proper big data system and data model with many tables and items for QlikView requires enough planning time and storage. Some data accesses require appropriate tools and privileges for direct data accessing. The current configuration allows access to the spreadsheets, text files and some local databases. An appropriately big database would serve well for demonstrating how the data is actually loaded and structured inside the application. That would create a good case appropriate for comparing it to the data warehouse systems. The current project is a small scale project and serves as a small demonstration of the software abilities though perhaps a big and well scaled project would just require a data warehouse.

The project base, on the other hand, helped to gain a general understanding of data warehousing concepts, OLAP and data visualization processes. The results of the projects also provided good ideas on improving future reports and dashboards built with QlikView. Particularly important was to go through all data modeling principles and techniques.

## 5 Conclusions

As was stated in the beginning of this paper the goal of the project was to examine the QlikView software and compare it to the commonly used data warehousing concepts and technologies.

Although the main task of QlikView is data visualization, this project also tried to look at another important perspective by targeting the data structuring and analysis which is the key assignment in system design and application development process. A proper data modeling is the key in building the application that ensures future data integrity and improves application performance. It helps to achieve a better user experience and provide the desired results required for decision making.

The result of this final year project is a small-scale QlikView application which is perhaps too small for testing all the possibilities of this platform. One way to improve the outcomes would be reserving enough time for studying the subject and coming up with a better project idea and gathering more data. After completing this project I can say that I gained a better understanding of the overall system design flow and it gave me a good about how my other current and future projects could be improved. It also demonstrated that there is still so much to learn about data modeling and visualization.

Based on the studies and my previous experience, I can state that, although QlikView can handle multiple sources and millions and even billions of rows it can hardly replace a well-designed data warehouse. The limitations are cost, application size growth with the data expansion, QVD files being tied to specific software only and a complex data modeling process. The data warehousing solution provides a better data structure and delivers information in a format that is more universal in use and can be utilized by many different applications which makes it far superior to QlikView.

What are the possible conclusions one can come up with when comparing a traditional data warehouse and QlikView? There is no clear answer. Deciding whether to use a data warehousing solution or QlikView depends on the needs stated by the company. Perhaps, small companies, having smaller data volumes and data problems, can avoid using a middle database layer between data and QlikView.

A better approach is to use an open "data warehouse" as a source for future QlikView reports. That ensures a faster access to the centralized and properly formatted data and making easier the further data analysis process. It allows users (managers, sales people or customer support team) a faster and more flexible access to the data when there is a need for a fast solution and there is no time for learning new (BI) tools.

The topic of data warehousing and visualization itself is quite big and interesting and may initiate many questions, discussions and projects. This makes one think more carefully about data storage and management subject in general and provides new ideas for future projects and challenges.

## References

1. Kirk, Andy. Data Visualization: a successful design process. Olton, Birmingham, GBR: Packt Publishing, 2012.  
<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10642563>. Accessed 10 April 2015.
2. Matthew Ward, Georges Grinstein, Daniel Keim. Interactive Data Visualization: Foundations, Techniques, and Applications. Natick, MA, USA: A K Peters/CRC Press, 2010. Accessed 10 April 2015.
3. Prabhu, S. Data Mining and Warehousing. Daryaganj, Delhi, IND: New Age International, 2007.  
<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10323316>. Accessed 15 April 2015.
4. Floyd, Matt. QlikView Scripting. Olton, Birmingham, GBR: Packt Publishing Ltd, 2013.  
<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10813407>. Accessed 10 November 2015.
5. Nagabhushana, S. Data Warehousing OLAP and Data Mining. Daryaganj, Delhi, IND: New Age International, 2006.  
<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10318716>. Accessed 7 November 2015.
6. Pover, Karl. Learning Qlikview Data Visualization. Birmingham, GBR: Packt Publishing Ltd, 2013.  
<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10772079>. Accessed 7 November 2015.
7. Harmsen, Barry, and Garcia, Miguel. QlikView 11 for Developers. Olton, Birmingham, GBR: Packt Publishing, 2012.  
<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10632721>. Accessed 10 November 2015.
8. Redmond, Stephen. QlikView for Developers Cookbook. Olton, Birmingham, GBR: Packt Publishing Ltd, 2013.  
<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10728416>. Accessed 10 November 2015.
9. QlikTech International AB, 1993-2015. Lund, Sweden. <http://www.qlik.com/>. Accessed 5 April 2015.
10. Williams, Steve, and Williams, Nancy. Profit Impact of Business Intelligence. Burlington, MA, USA: Morgan Kaufmann, 2006.

<http://site.ebrary.com.ezproxy.metropolia.fi/lib/metropolia/detail.action?docID=10158423>. Accessed 10 November 2015.