

University
of
St Andrews

School of Computer Science

IS 5198 Dissertation in Information Technology

The University of St Andrews Student Experience Dashboard

Magdalena Kovacheva

Student ID: 120024010

23/08/2013

Declaration

I hereby certify that this dissertation, which is approximately 14 800 words in length, has been composed by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a higher degree. This project was conducted by me at The University of St Andrews from June/2013 to August/2013 towards fulfilment of the requirements of the University of St Andrews for the degree of MSc under the supervision of Dr Tom Kelsey Judith Malcolm.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

I, Magdalena Kovacheva, received particular assistance in the writing of this thesis in respect of matters of style, idiom, grammar, syntax or spelling, which was provided by Jane Brooks and Sascha Roschy.

Date: 23/08/2013

Signature:

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr Tom Kelsey and Judith Malcolm, for their insightful feedback, valuable guidance, and constant support.

I also would like to show my greatest appreciation to the University's Business Intelligence team and especially to Simon Caruana for his time, constructive comments, and boundless encouragement.

Special thanks also to the management of the University's Student Services, ASC, IELTS, and IT Service Desk. Without their kind participation this dissertation would not have been possible.

Furthermore, I owe my deepest gratitude to Janie Brooks and Sascha Roschy for their dedicated assistance and warm support. They played an important role in the improvement of my English skills for the writing of this dissertation.

Finally, I want to thank my family and my partner for their patience and constant encouragement throughout my entire master study.

Table of Contents

- Table of Figures I
- Table of Tables I
- Table of Abbreviations II
- Abstract..... III
- 1 Introduction..... 1
 - 1.1 Purpose of the Project 1
 - 1.2 Limitations and Risks..... 1
 - 1.3 Structure 1
- 2 Literature Review..... 2
 - 2.1 Introduction 2
 - 2.2 Business Intelligence (BI) and Analytics 2
 - 2.2.1 Introduction..... 2
 - 2.2.2 Definitions..... 3
 - 2.2.3 Technology 4
 - 2.3 BI and Analytics in Higher Education 12
 - 2.3.1 Motivation..... 12
 - 2.3.2 Benefits 13
 - 2.3.3 Challenges..... 14
 - 2.4 Conclusion..... 15
- 3 The Student Experience Dashboard – Motivators and Potential Benefits 15
 - 3.1 Introduction 15
 - 3.2 Motivation and Potential Benefits..... 15
- 4 Demo Dashboard Requirements 17
 - 4.1 Introduction 17
 - 4.2 Objectives and Potential Benefits 17
 - 4.3 Functional Requirements..... 17
 - 4.4 Non- functional Requirements 18
- 5 Qlikview..... 18
 - 5.1 About Qlikview 18
 - 5.2 Why Qlikview? 19
 - 5.3 Qlikview Syntax..... 20

5.3.1	Introduction.....	20
5.3.2	Load Script.....	21
5.3.3	Set Analysis	26
6	Implementation	27
6.1	Data Modelling.....	27
6.1.1	Data Sources	28
6.1.2	Extract- Transform- Load Process	32
6.2	User Interface	40
6.2.1	Introduction.....	40
6.2.2	‘Help’ Sheet	41
6.2.3	‘Comparison with Last Year’ Sheet.....	42
6.2.4	‘Year by Year Analysis’ Sheet	44
6.2.5	‘Associated Questions’ Sheet	45
6.3	Lessons Learned.....	47
7	Demonstration and Testing	49
7.1	Method	49
7.2	Limitations	50
7.3	Results	50
8	Evaluation	50
8.1	Introduction	50
8.2	Data Storage and Quality	51
8.3	Data Governance and Ownership.....	51
8.4	Culture.....	51
8.5	Processes and Reporting.....	52
8.6	Infrastructure, Tools and Expertise	52
8.7	Conclusion.....	52
9	Recommendations and Future Work	53
10	Conclusion	53
	Bibliography	55
	Ethical Approval Letter.....	59

Table of Figures

Figure 1 The process of turning data into knowledge (Source: (Korczak & Dudycz, 2012)).....	2
Figure 2 Typical BI Platform Architecture (created by the author).....	4
Figure 3: Star Schema (Chaudhuri & Dayal, 1997).....	6
Figure 4: Snowflake Schema (Chaudhuri & Dayal, 1997).....	7
Figure 5: Fact Constellations Schema (Source: datawarehouse4u.info)	7
Figure 6: “Sales” cube	9
Figure 7: An example of hierarchy structure of Product dimension (created by the author)	9
Figure 8 Qlikview Associative Technology (Source: (Qlikview 2012a))	19
Figure 9 Magic Quadrant for BI and Analytics Platforms (Source: (Gartner 2013)).....	20
Figure 10 Exemplifies the need of counter fields	23
Figure 11 Illustrating the difference between JOIN and CONCATENATE (created by the author)...	24
Figure 12 Example of Qlikview Mapping (created by the author).....	25
Figure 13 An Example of Set Analysis (created by the author)	26
Figure 14 Data Model at the end of step 1	35
Figure 15 Data model at the end of Step 2.....	37
Figure 16 Exemplifies a sub set of 'Student Records' table	37
Figure 17 Data model at the end of Step 3.....	38
Figure 18 Data model at the end of Step 4.....	39
Figure 19 Final Data Model.....	40
Figure 20 The ‘Help’ sheet of the demo dashboard.....	41
Figure 21 The 'Comparison with last Year' sheet of the demo dashboard.....	42
Figure 22 The 'Year by Year Analysis' sheet of the demo dashboard	44
Figure 23 The 'Associated Questions' sheet of the demo dashboard	46
Figure 24 Service units' visit load presented in a Radio chart.....	47
Figure 25 Service units' visit load presented in a Line Chart.....	47
Figure 26 Illustrating the benefits of static axes values.....	48
Figure 27 Example of a user-friendly error message from the demo dashboard.....	49

Table of Tables

Table 1 Set Operators (created by the author based on (Qlikview 2012b)).....	27
Table 2 ASC data fields	29
Table 3 UniDesk data fields.....	30
Table 4 ILES data fields	31
Table 5 Student Records data fields.....	31

Table of Abbreviations

AIR..... Association for Institutional Research
ASC..... Advice and Support Centre
BI..... Business Intelligence
DOLAP Database (Desktop) Online Analytical Processing
ECAR EDUCAUSE Centre for Applied Research
ELT Extract-Load-Transform
ETL Extract-Transform-Load
HEI..... Higher Education Institution
HOLAP Hybrid Online Analytical Processing
IELS In-sessional English Language Service
KPI Key Performance Indicator
MOLAP..... Multidimensional Online Analytical Processing
OLAP Online Analytical Processing
OLTP..... Online Transactional Processing
ROLAP Relational Online Analytical Processing
RTOLAP..... Real-time Online Analytical Processing

Abstract

This study proposes and is the first step towards the development of a university-wide business intelligence dashboard related to the student experience – the Student Experience Dashboard. The purpose of the proposed dashboard is to consolidate and display data from different internal and external units related to the student experience at the University of St Andrews. This would allow centralized monitoring and analysis and potentially would lead to better understanding of students' needs and enhanced student experience. The pilot project realized involved the creation of a demo dashboard and a preliminary assessment of the University's readiness for adopting and implementing the Student Experience Dashboard. The findings revealed that a lack of properly collected data, concerns of data misinterpretation, and the University's decentralized structure are the main challenges for the proposed implementation. It is therefore recommended that the dashboard's complete implementation requires clearly communicated benefits, enhanced data collection processes and well established partnerships between the University's departments and between the University and the related external units.

1 Introduction

1.1 Purpose of the Project

The purpose of this dissertation project is to demonstrate the potential of the proposed Student Experience Dashboard and to identify the main challenges for its implementation as well as to provide a preliminary assessment of the University's readiness. This is enabled by the development of a demo dashboard with the business intelligence tool Qlikview. The main tasks of the pilot project involve:

- finding and obtaining access to data from multiple university units;
- developing a demo Student Experience Dashboard;
- demonstrating the demo dashboard to members of the University's Student Services management.

Furthermore the demo dashboard to be developed may serve to further demonstrate, communicate, and explore the perceived benefits of the Student Experience Dashboard.

1.2 Limitations and Risks

- **Lack of senior management support** – without support from the University's senior management the involvement of units related to the student experience might be effort and time-consuming to achieve.
- **Time constraints** – the time for the realization of this project was limited to less than three months. Within this time research about the business intelligence concept, the technology utilized in the project, must be conducted; data related to the student experience must be acquired from multiple units; the requirements for the demo dashboard must be set; the business intelligence tool Qlikview, used for the demo dashboard's development, must be explored; the demo dashboard must be built and demonstrated.
- **Holiday season** – staff absence due to summer holidays might cause a significant delay in the communication with units related to the student experience and thus hamper the process of data acquisition.

1.3 Structure

This dissertation is organised as follows. Chapter 2 provides a comprehensive literature review about the concept and the technology of business intelligence as well as its application in the Higher Education Sector. Chapter 3 introduces the proposed Student Experience Dashboard, the motivation behind the idea and the potential benefits of its implementation at the University of St Andrews. Chapter 4 presents and discusses the requirements for the demo dashboard produced during this dissertation project. Chapter 5 justifies the choice to use the BI tool Qlikview for the dashboard's development and explains its main concepts and syntax. Chapter 6 presents and analyses the

dashboard's implementation process. Chapter 7 discusses the purpose, the method and the results of the dashboard's demonstration session. Chapter 8 evaluates the University's readiness to adopt and implement the proposed Student Experience Dashboard. Chapter 9 provides a set of recommendations. The last chapter concludes the dissertation.

2 Literature Review

2.1 Introduction

This chapter is organized in two main sections. The first part analyses the essence of the business intelligence concept and the technologies hidden behind it. The second part discusses the motivators for applying business intelligence in higher education sector, as well as the benefits and the challenges of its adoption. The content is a result of a comprehensive review of academic journal articles, white papers, and books in both English and German.

2.2 Business Intelligence (BI) and Analytics

2.2.1 Introduction

The large-scale deployment of the database technology and the World Wide Web in the late 90s led to a vast increase of data generation, collection, and storage. Nowadays business enterprises have large amounts of data about different aspects of their operations, such as employees, customers, competitors, and daily transactions. However, in order to take advantage of having this vast amount of data, companies need to turn it into knowledge that can help them make more informed decisions and thus to run their business more effectively (Figure 1). Tools, technologies, and practices that enable the transformation of data into knowledge to ensure enhanced decision-making are generalized under the term business intelligence.

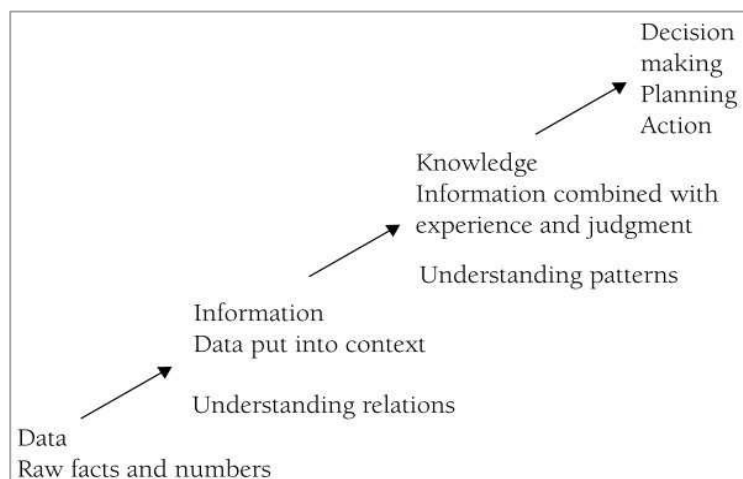


Figure 1 The process of turning data into knowledge (Source: (Korczyk & Dudycz, 2012))

2.2.2 Definitions

Even though the definition of business intelligence is widely discussed in the literature, there is no universally accepted one, as evident from the discussion below. The term was introduced in 1958 as "the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal" (Luhn, 1953). In other words, business intelligence refers to analysing data ("facts") to facilitate decision-making ("guide action"). Examples of "facts" could be data about company's sales, orders, costs etc.

In 1989 Howard Dresner provides a more detailed definition of business intelligence and generalizes it as an umbrella term (Gibson et al. 2004). The comprehensive online research could not reveal the original source of this definition. However, two different versions could be found in business intelligence related articles. According to Chee et al. (2009), Dresner's definition emphasizes on the processes involved in BI and states, that BI is "a broad category of software and solutions for gathering, consolidation, analysing and providing access to data in a way that lets enterprise users make better business decisions". The other definition of BI, connected to Dresner is: "a concepts and methods to improve business decision making by using fact-based support systems" (Power 2007).

One of the well-defined recent interpretations is the one suggested by Nicolson, Junkins, & Evelson (2008), which incorporates the nature of business intelligence ("a set of methodologies, processes, architectures, and technologies), its main task ("that transform raw data into meaningful and useful information") and its purpose ("used to enable more effective strategic, tactical, and operational insights and decision making").

The literature provides a variety of other definitions capturing different aspects of business intelligence. Chee et al. (2009) provide a method for classifying the existing definitions into three different categories depending on the aspect they are focusing on. A definition could be assigned into one or more of the following categories: management (process) aspect, technological aspect, and product aspect.

Since the main purpose of BI is to enable analytics (the process of data analysis that allows companies to measure, improve, and compare their performance (Barneveld et al. 2012)), this document considers both terms as interchangeable.

2.2.3 Technology

2.2.3.1 Overview

A typical BI platform architecture can be divided into five layers (Figure 2). The first layer represents the data sources which can include spreadsheets, operational databases, and/or ERP (Enterprise Resource Planning) systems. The technologies in the second layer are responsible for the transportation of data from the sources and its successful integration to a centralized storage area (data warehouse). These tasks are performed by Extract-Transform-Load (ETL) tools. The third layer is a centralized data repository, which consist of a data warehouse and optionally can include a number of data marts. The fourth layer is the layer, where the data analysis is performed. Typical technologies that enable data analysis are Online Analytical Processing (OLAP) and data mining tools. The last layer is the data presentation layer. Its main goal is to provide data visualization in a way that facilitates the discovery of relationships and triggers effective decisions. Leading vendors of BI platforms are IBM, SAP, SAS, Microsoft, and Oracle (Gartner 2013). The technologies in each layer are discussed further in this section.

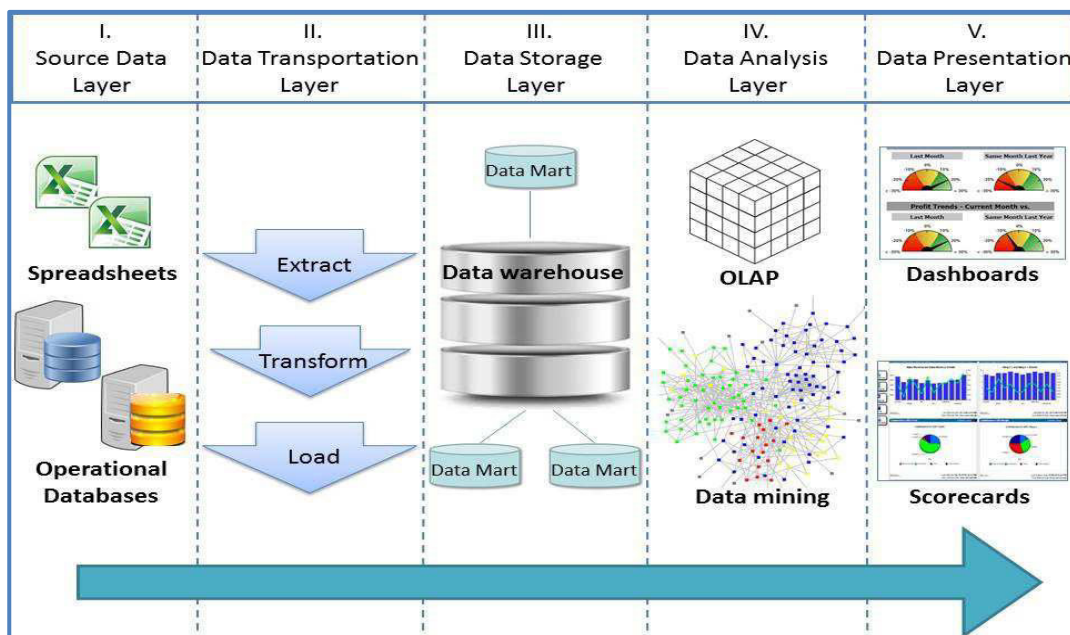


Figure 2 Typical BI Platform Architecture (created by the author)

2.2.3.2 Extract-Transform-Load (ETL)

ETL tools are the main technology used in the Data Transportation Layer. They are responsible for moving the data from a source data stores to a target database, which is typically the company's data warehouse (Section 2.2.3.3). The main goal of ETL tools is to ensure the successful integration of heterogeneous data into a single data store in unified understandable and user-friendly format (van

der Lans 2012) (Sturdy, 2012). To complete this task they run an ETL script, which performs consecutively three main processes or tasks – extract, transform, and load (van der Lans 2012).

- 1. Extract:** The first task forced by the ETL script is extracting data from company's operational data sources (Khan, 2012).
- 2. Transform:** The main purpose of the “transform” phase is to standardize the data extracted from different sources and accommodate it to the data warehouse multidimensional model. (Sturdy, 2012) To complete this task ETL tools apply a set of operations, as well as domain-specific knowledge and business rules. Common operations include reformatting, encoding, joining, filtering, pivoting and adding time elements (Khan 2012) A standard simple operation is renaming attributes to ensure consistency: for example, while in company's HR database employees' details are stored in an “Employees” table and the records are uniquely identified by a key “Employee ID”, in the internal service desk's database employees' data could be stored in a “Users” table with a key “User ID”. Typically, the “transform” process is performed in the so called staging area, which is a particular part of the data warehouse serving as an intermediate storage area for data in a “transform” phase (Sturdy, 2012).
- 3. Load:** The last task of ETL tools is to load the cleaned data into the data warehouse. There the data can be checked for compliance with integrity constraints, summarized, and properly indexed (Sturdy, 2012).

A variation of ETL is the ELT process (Extract-Load-Transform). It leads to the same result as ETL, only the process sequence is different. Here the source data is first loaded into the target database and after that transformed (Khan, 2012). The main conceptual difference between an ETL and an ELT tool is that while the former runs transformation operations by itself, the latter delegates this role to the target database. The database server transforms the source data and copies it to another table. The benefits of exploiting the database server for transforming operations are first, the almost instant data loading from the ELT server to the database and second, the accessibility of the original data format from the target data store (van der Lans 2012).

2.2.3.3 Data Warehouse

A data warehouse is a large complex database designed to enable faster and more effective data analysis and decision-making. This is achieved by allowing organizations to store and access consolidated and summarized data, collected from all company's data sources (operational databases, spreadsheets etc.) The term “data warehouse” was introduced by Barry Devlin and Paul Murphy in the 1980s, but the concept was developed and popularized by William H. Inmon in 1991 (Khan, 2012). In his book “Building the data warehouse” he defines data warehouse as “a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decisions”.

(Inmon 1991) The concept of data warehouse was developed to address the main issues of the traditional operational databases. They exploit Online Transactional Processing (OLTP) to support company’s daily operations and to keep detailed up-to-date information about each business transaction (Santos & Bernardino 2009). Since the primary objectives of the OLTP databases are to provide high performance, they use relational data model and normalized tables, eliminating data redundancy. Storing only current raw transaction data, they have limited analytical functionalities (Khan, 2012). In contrast to OLTP databases, data warehouses typically store data in denormalized tables (large tables containing redundant data) and thus they perform less table joins and offer faster query processing. Furthermore, data warehouses provide integrated, summarized, historical data, which allows users to identify trends, forecast customers’ needs, and analyse competitors’ business. To support these functionalities data warehouses typically organize their data using a multidimensional data model.

Multidimensional Schemas

The multidimensional data model can be represented by three schema types: Star, Snowflake and Fact Constellations schema (Hayardisi 2009) (Chaudhuri & Dayal 1997).

Star Schema: Star schemas are the most popular multidimensional schema type. They contain one fact table which is connected to multiple dimension tables through foreign keys (Chaudhuri & Dayal 1997). Each dimension is represented by a single denormalized table, where each record can be uniquely identified by the table’s key (Khan, 2012). Besides the keys for each dimension, the fact table stores numeric measures for each key combination (Hayardisi 2009). As a result of having one large table and multiple small ones, star schemas provide users with quick query processing and high performance (Khan, 2012). A star schema is illustrated in Figure 3.

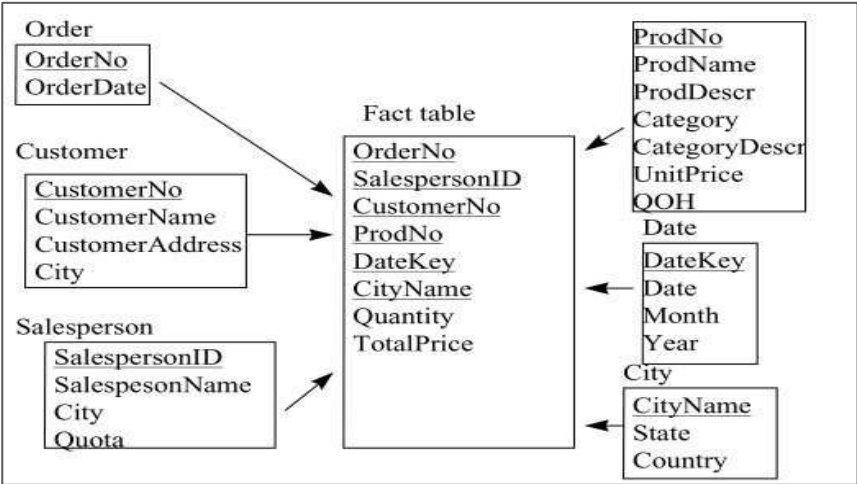


Figure 3: Star Schema (Chaudhuri & Dayal, 1997)

Snowflake Schema: Snowflake schemas are similar to star schemas, but in addition they allow the representation of dimensional hierarchy (Chaudhuri & Dayal 1997). This schema is more complex, because dimension tables are normalized and each hierarchy level is represented by a different dimensional table. Although this solution typically improves query response-time even more, the larger table number can make maintenance more difficult (Khan, 2012). An example of a snowflake schema is presented in Figure 4.

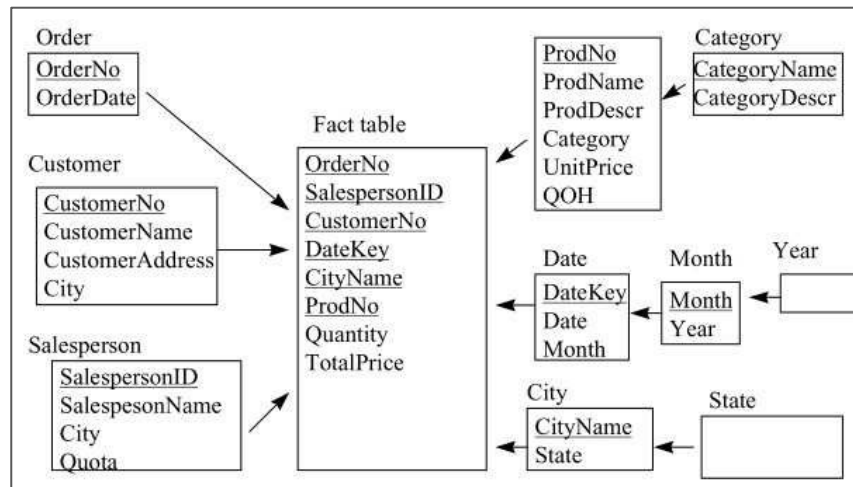


Figure 4: Snowflake Schema (Chaudhuri & Dayal, 1997)

Fact Constellations Schema: The Fact constellations schema is a more complex multidimensional schema, which includes more than one fact tables, which share part of the dimensional tables (Chaudhuri & Dayal 1997) (Hayardisi 2009). An example is presented in Figure 5.

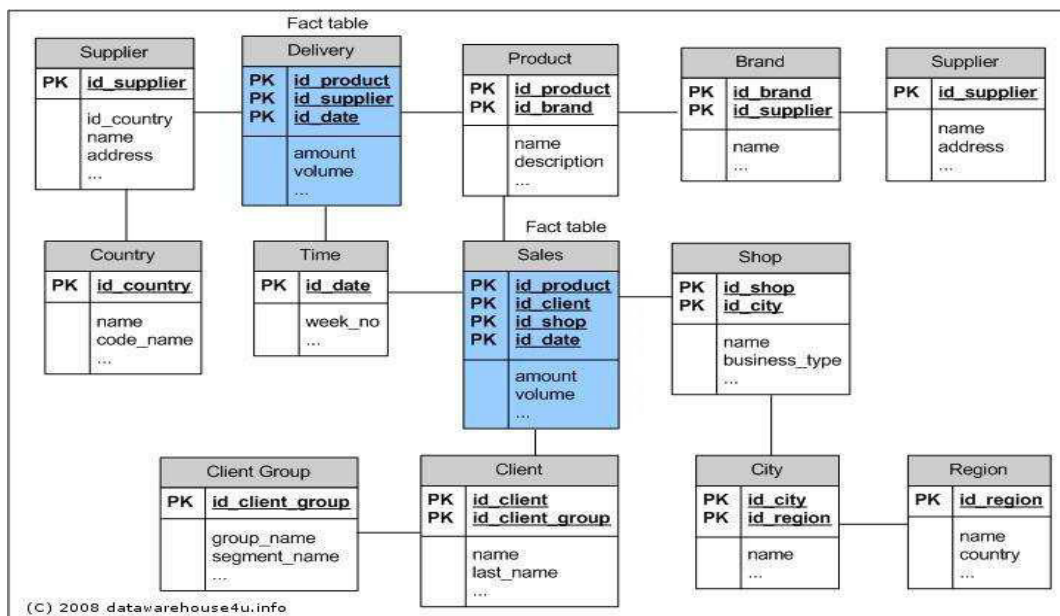


Figure 5: Fact Constellations Schema (Source: datawarehouse4u.info)

Types of data warehouse

According to Khan (2012) there are three data warehouse types: enterprise data warehouse, data mart, and operational data store.

Enterprise data warehouse: An enterprise data warehouse allows companies to store their entire business data in one place. It collects data from all departments within an organization (Chaudhuri & Dayal 1997). Enterprise data warehouses store not only summarized, but also detailed transaction data (Khan, 2012). However, implementing an enterprise data warehouse is a time-consuming, complex, and costly process (Chaudhuri & Dayal 1997).

Data mart: In contrast to an enterprise data warehouse, a data mart does not support a whole organization, but rather a department or a branch of it (Khan, 2012). A data mart stores both transaction and summarized data from different sources (Hajmoosaei et al. 2011) (Khan, 2012). There are two data mart types depending on how they derive data – independent and dependent data marts. While the former obtains data from operational or external sources, the latter retrieves data from a company's central data warehouse. Since the scope of a data mart is typically limited to the needs of a single department, its deployment requires less time, resources, and experience in comparison to an enterprise data warehouse. However, if there is a lack of unified data model within the organization, having separate data marts for each department, can lead to integration issues in a long term period (Chaudhuri & Dayal 1997).

Operational data store: An operational data store is a database obtaining and storing mainly current and recent data from company's operational data sources (Khan, 2012). In contrast to an enterprise data warehouse, an operational data store contains less historical and summarized data and can be constantly updated in real-time. Operational data stores are designed to support the daily operational reporting within an organization or to serve as a staging area preparing data to be integrated into company's central data warehouse (A. A. Mohamadina et al. 2012).

2.2.3.4 Online Analytical Processing (OLAP)

Online analytical processing is a category of database processing (Codd et al. 1993), which could be defined as a software technology that “enables analysts to gain insight into data through fast, consistent and interactive access to the data stored in the data warehouse” (A. a. Mohamadina et al. 2012). The OLAP server acts as an intermediate interface between the data warehouse and the presentation layer (dashboards, reports etc) (Dehne & Zaboli 2012). By utilizing the multidimensional view of the stored data, the OLAP server allows users to perform operations such as aggregation, pivoting, slicing, dicing, rollup, and drill-down on hierarchies (Chaudhuri, Dayal, & Narasayya, 2011). In addition to dynamic analysis, OLAP offers a variety of statistical operations and thus enhances the decision-making process (Dehne & Zaboli 2012). To enable these functionalities, OLAP organizes data using multidimensional data cubes, which allow users to analyse their data

from different perspectives (dimensions). It is important to point out that although a cube has three dimensions, OLAP can support far more dimensions (Kuhl & Stoeber 2006). The measures (values) in the cube are obtained from the fact table records in the data warehouse, while the dimensions are derived from the dimension tables (Khan, 2012). In the example depicted in Figure 6, the “Sales” values can be analysed by dimensions “Customer”, “Product”, and “Store”. In addition, the elements in each dimension could be organized in a hierarchy. This structure enables users to view and summarize data in different levels (Zhao & Huang 2010). The example illustrated in Figure 7 shows a model hierarchy of the elements in the “Product” dimension, which allows users to analyse sales not only by product, but they could also roll up (summarize) the data and view it by sub-subcategory, subcategory, and category. Another common dimension with hierarchy-ordered elements is “time”: year-quarter-month-week-day (Chaudhuri et al., 2011).

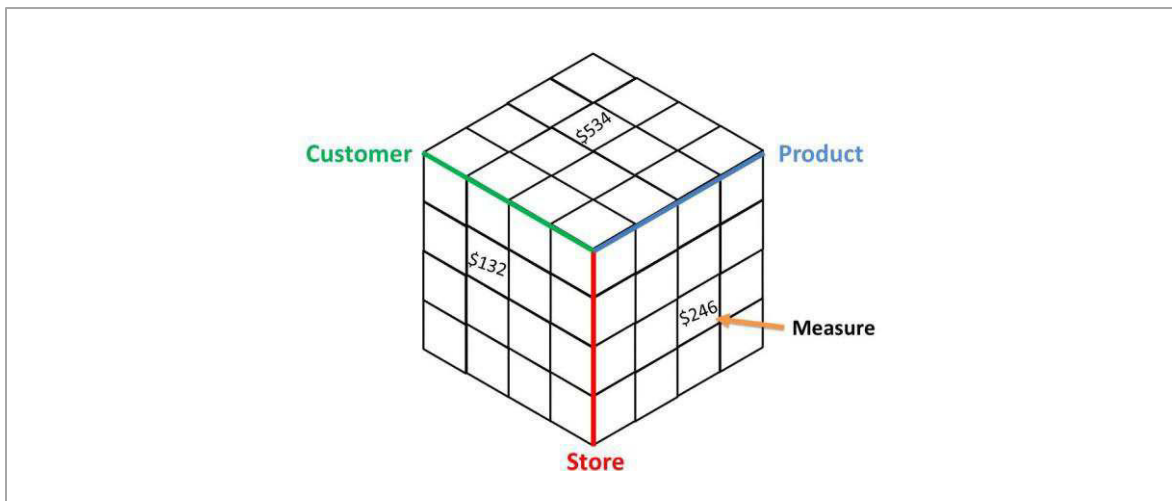


Figure 6: “Sales” cube

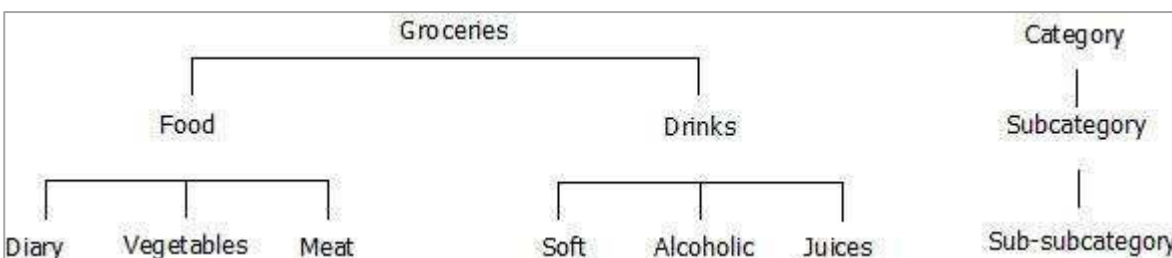


Figure 7: An example of hierarchy structure of Product dimension (created by the author)

There are different OLAP models depending on their implementation method: Multidimensional, Relational, Hybrid, Database (Desktop), Web, and Real-time OLAP.

Multidimensional OLAP (MOLAP): MOLAP is considered to be the classic OLAP model. Here the data derived from the data warehouse is stored on the MOLAP server in proprietary multidimensional format. In order to provide optimized data storage and short response time, MOLAP applies indexing, caching, and data aggregation processes (Khan, 2012).

Relational OLAP (ROLAP): ROLAP systems do not offer a data storage function, but they derive the data directly from the underlying relational database (Kuhl & Stoeber 2006). To enable a multidimensional view over a relational data structure, ROLAP supports not only SQL queries, but also some “aggregate functions such as median, mode, and time window based moving average” (Chaudhuri, Dayal, & Narasayya, 2011). The main disadvantage of this model is that it requires more time to respond to users’ queries (Kuhl & Stoeber 2006).

Hybrid OLAP (HOLAP): HOLAP is a combination of MOLAP and ROLAP architectures. The concept of this model is to use MOLAP for the more frequently analysed data, taking advantage of quick query processing, while the less frequently used data is handled by ROLAP (Kuhl & Stoeber 2006).

Database or Desktop OLAP (DOLAP): In DOLAP architecture, the data cubes are stored on user’s personal machine, since querying locally stored data could result in faster performance. However, this OLAP model is only suitable for small amounts of data and low-end analysis (Khan, 2012).

Web OLAP (WOLAP): The web-based OLAP approach enables users to access and analyse the data warehouse through a web browser (Li et al. 2010).

Real-time OLAP (RTOLAP): In contrast to the traditional static OLAP model, where the system derives data from the data warehouse periodically at a predefined time (e.g. every Monday at 6am), the real-time OLAP gets updated at the moment new data is inserted into the database. Although RTOLAP technology allows users to gain insight into the newest data and thus improves the decision-making process, it still has unsolved issues related to its performance when dealing with large-scale data warehouses (Dehne & Zaboli 2012).

2.2.3.5 Data Mining

In contrast to OLAP servers, providing summarized information, data mining tools offer more complex algorithms for in-depth data analysis (Chaudhuri et al., 2011). They enable users “to discover various patterns, generalizations, regularities and rules in data resources” (Olszak & Ziemia 2007) and to incorporate them in “a representation of reality in the form of a model” (predictive model) (Sturdy 2012). Some of the most common modelling techniques are Neural networks (for classification of large complex data), Cluster analysis (for grouping), Decision tree (for profiling), Regression analysis (for identifying relationship between variables), and Basket analysis (for identifying items which customers buy together) (Ali 2013). By performing tasks such as prediction,

classification, association, and clustering, (Ali 2013) data mining tools transform large amount of data into useful knowledge (Baritchi 2003). However, Rygielski, Wang, & Yen (2002) point out that the knowledge (the models) discovered by data mining tools should not be considered as always right and that verification is an important part of the process.

In a traditional BI architecture, data mining processes are run outside the data warehouse by a specialized statistical software (data mining engine). As a result of the performed statistical analysis on a selected data subset, data mining tools create predictive models, based on the identified patterns and relationships (Sturdy 2012). The final step is the deployment of the constructed predictive models in the operational database (Chaudhuri et al., 2011). Applying these models on the stored data enables users “to find anomalous or unusual data elements” and to create graphs and reports enhancing business decision-making (Sturdy 2012).

However, to avoid data transfer from company’s data warehouse to the data mining engine and to overcome some of the limitations related to the data mining engine itself (e.g. analysing only a subset of the data), more and more BI vendors are moving towards “in-database analytics”. Here data mining tools are integrated in the data warehouse, so that the statistical analysis can be performed inside instead of outside of the database (Chaudhuri et al., 2011).

2.2.3.6 Presentation Layer

The presentation layer is one of the most important components of a BI system, because it is responsible for accessing and displaying the data in a way that facilitates decision making. The data, stored and processed by the backend BI technologies discussed in the previous sections (ELT, data warehouse, OLAP, and data mining,), should be presented in an interactive and intuitive format, allowing users to easily interpret the information provided (Korczak & Dudycz 2012). A good user interface ensures that users will understand the displayed information by helping them to spot trends, anomalies, and relationships. Another characteristic of a good BI user interface is efficient navigation, allowing the users to switch quickly from overview of the information to a specific area of interest (Korczak & Dudycz 2012). Common BI presentation tools are dashboards and scorecards.

Dashboards and Scorecards

A dashboard is a business intelligence software application, which could be described as a data visualization tool that allows users to monitor, analyse, control, and report key performance indicators (KPIs) and critical business metrics on a single screen (Khan 2012). The purpose of a BI dashboard is to present data in a way that facilitates decision making processes. Key dashboard functionalities like real-time graphical presentation of information (charts and graphs), slice-and-dice, and drill-down capabilities (exploration from different perspectives and level of detail) on metrics, and what-if analysis (Bose 2006), enable users not only to spot trends, relationships, and anomalies in the underlying data, but also to investigate reasons for these. BI dashboards could facilitate both exploratory (aims to find unknown relationships and patterns) and confirmatory (aims to confirm

hypothesis) analysis (Jisc 2012). Furthermore, dashboards are often customized for a particular role or department within an organization. There are three main types of dashboards depending on the level of detail and the type of decision which should be facilitated: operational, tactical, and strategic (Eckerson 2011). Typically all three types are used within an organization.

Operational dashboards are commonly used by employees to monitor key processes and to perform their daily tasks (Eckerson 2011). An example of an operational dashboard feature designed for employees taking customers' orders by phone, would be a market basket, showing items which customers tend to buy together. Hypothetically, an instant access to data which suggests that previous customers who purchased wine, also bought cheese and grapes, would enable the employees to make reasonable suggestions to the customers and thus to increase the company's sales.

Tactical dashboards are typically the most widely used in organizations, since they are designed to enable departments and project managers to monitor operations and initiatives (Eckerson 2011). They display summarized as well as detailed information. A tactical dashboard used by the manager of the call centre responsible for taking customers' orders over the phone, would show information such as number of calls (orders) , top employees based on total sales, top products, and cost per order.

Strategic dashboards are designed to meet the needs of executives and senior management and therefore they display summarized data about the whole organization. Strategic dashboards usually contain scorecards (Eckerson 2011), which indicate the progress towards meeting a business goal or objective represented by KPIs.

2.3 BI and Analytics in Higher Education

2.3.1 Motivation

The introduction of tuition fees in 1998 turned the Higher Education sector to a competitive environment (Brookes 2003) (Tapp et al. 2004). Therefore enhancing students' enrolment and retention became main goals for Higher Education Institutions (HEIs). To achieve these, universities started to see students as customers and focused on understanding students' needs and offering a variety of support services and extra-curricular activities (Sturdy 2012). As a result of this shift towards a student-oriented operation and the technological progress in the last decades, today's universities are collecting more data than ever before (Bichsel 2012). The high competition for students motivates universities to take advantage of the collected data by transforming it into knowledge and using it to facilitate strategic decisions, trigger adequate actions, and improve their operation. Instead of using the data to simply meet reporting requirements, HEIs are now realizing the need to adopt business intelligence and analytics practices, allowing them to analyse data and turn it into faster and more effective decisions.

A study conducted in 2012 by the EDUCAUSE Centre for Applied Research (ECAR) in the USA revealed that BI and analytics has increased its importance in higher education in the past three years

and will get even more important in the future (Bichsel 2012). About two-thirds (69%) of the 339 institutions, which responded to the survey, consider BI as a key priority for at least some of their departments or projects. Only 6% do not have any interest towards implementing analytics. In addition to the survey sent to HEIs, which are members of EDUCAUSE or the Association for Institutional Research (AIR), the ECAR study collects information from seven focus groups. Participants in these focus groups were IT and Institutional Research specialists working in higher education.

Besides EDUCAUSE, Jisc is another non-profit organization aiming to enhance higher education by the utilization of information technology.¹ Jisc provides funding and support for Activity Data, Business Intelligence, and Customer Relationship Management projects in UK universities. Together with the ECAR study, Jisc project reports are a valuable source of information about the current state of analytics in higher education outlining key benefits, challenges, and recommendations.

2.3.2 Benefits

The documentation repository of Jisc and EDUCAUSE provides a large amount of examples and case studies from different HEIs that illustrate the benefits of BI in the sector (Jisc 2012) (Kay & van Harmelen 2012) (Bichsel 2012). The advantages discussed in these reports can be mapped out in the following main categories:

Improved Student Performance and Success: the ability to monitor and analyse students' activities (e.g. library visits, support services usage) enables universities to identify factors for student success and to use them to develop early alerts for students at risk of poor performance or dropping out. This allows universities using BI and analytics to act proactively and intervene at an early stage to improve students' success. An example for successful usage of historical data to identify at-risk student behaviour is the Early Warning System implemented in the Roehampton University. The institution reports that in the first year of system's operation, the student progression rates have been improved by 14% (King et al. 2012). A similar project is the Student Engagement Traffic Lighting at the University of Derby (Mutton 2011).

Personalized Services: combining data on students' activities with student records data provides universities with the opportunity to offer personalized services. One of the successful examples illustrating this benefit is the Open University, which consolidated library data with information about students' major. Based on both datasets the institution provided a personalized recommendation service that suggests library materials to students, by showing them sources accessed by other students enrolled in the same program (The Open University 2013).

¹ Jisc, (accessed August 14, 2013); available from <http://www.jisc.ac.uk/about>

Optimized Resource Management: gaining insight into students' needs and activity patterns enables HEIs to distribute and utilize their resources more effectively and efficiently.

Enhanced Strategic Planning: by providing better insight into institutions' operations and students' activities, the adoption of BI and analytics enables data-driven decision-making and leads to more focused and effective strategic planning.

Fostering Communication: the adoption of BI and analytics is reported to improve communication regarding strategic planning. Decisions based on data and facts instead of intuition and personal opinions are more likely to be accepted anonymously.

2.3.3 Challenges

The ECAR study identifies four main challenges for BI and analytics projects in HEIs: affordability, data, culture, and expertise.

Affordability

The main obstacle to the adoption of BI and analytics in HEIs is the size of the required investment (Bichsel 2012). The implementation of BI projects often involves large expenses for infrastructure, tools, experts, and training.

Data

A key reason for most of the issues related to universities' data is that HEIs often operate in decentralized environments. Taken from a variety of different systems and sources the data within a university is often incomplete and inconsistent (e.g. field names are not unified) (Kay & van Harmelen 2012). Furthermore, the heterogeneous environment also causes implications with data ownership, access, and misuse (Bichsel 2012). Common issue is also the unwillingness of departments to share data amongst each other. Furthermore, consolidating and analysing data from different sources, requires new policies regulating data protection, security, and access as well as clearly communicated roles and responsibilities related to them (Kay & van Harmelen 2012).

Culture

Cultural barriers are related to the fact that implementing BI and analytics typically involves changes in working and reporting processes. The fear of new procedures and tasks as well as the mistrust in data often withhold staff to adopt BI and data-driven decision making (Bichsel 2012). Another cultural barrier for university-wide projects is that individual units in HEIs (e.g. schools, service units) often consider themselves to be self-sufficient (Cranfield & Taylor 2008), which can impair cross-departmental communication and data sharing. Furthermore, universities tend to avoid changes in case of successful institutional performance and to react only when a problem is seen. (Cranfield & Taylor 2008)

Expertise

The implementation of BI and analytics is a complex process and requires appointment of BI specialists, expertise in project management, and additional staff training (Bichsel 2012).

2.4 Conclusion

Aiming to increase the number of applicants and to ensure student retention and success, HEIs are starting to implement BI and analytics in their operation. A large number of successful projects have identified that the adoption of BI and analytics leads to improved student experience, optimized resource utilization, enhanced communication and more focused strategic planning. However, the implementation of a BI project often needs to deal with a lack of resources, experts, and properly collected data, as well as a lack of individual units' and staff members' motivation to adopt data-driven decision-making.

3 The Student Experience Dashboard – Motivators and Potential Benefits

3.1 Introduction

The proposed Student Experience Dashboard is a BI dashboard, with a purpose to allow centralized monitoring and analysis of students' behaviour and needs. Its implementation involves the consolidation of data from all units, providing support services and/or engagement opportunities to St Andrews University students, into a single application. This chapter discusses the main motivators behind the proposed Student Experience Dashboard and the potential benefits of its implementation and usage.

3.2 Motivation and Potential Benefits

BI is not a novelty: The development of the proposed Student Experience dashboard is enabled by the fact that the University already has the required BI infrastructure, tools, and experts. Since the implementation of the BI software Qlikview in 2010, a large number of University departments have incorporated dashboards to monitor, measure, and manage their performance.

The Student Experience Importance: Constant enhancement of the student experience is one of the eight main aims outlined in the Strategic Plan 2008-2018 of the University of St Andrews (University of St Andrews 2008a). The document points out that to ensure enhanced student experience the University aims to support students' learning, development and employability by providing services and opportunities to meet students' academic and social needs. The University also strives to engage students by encouraging them to be involved in community and organized activities. As highlighted in the University Student Experience Strategy 2008 (University of St Andrews 2008b), the student experience is one of the main benchmarking metrics in Higher Education. Therefore failure to

understand and meet students' needs may lead to negative consequences for the University's reputation and ranking position.

Following Recommendations: Jisc reports highlight the potential of using data from different service and engagement units across the institution to gain insight into students' activities and needs. They emphasize that the student experience is one of the main aspects that can benefit from the implementation of Bland analytics solutions in HEIs (Jisc 2012).

Supplementing traditional satisfaction-rating and qualitative methods: The traditional satisfaction-rating and qualitative methods, such as satisfaction surveys, questionnaires and student feedback, are not able to provide completely accurate evaluation of the University's performance in terms of the student experience (Baiyun Chen et al. 2012). The major drawback of their methodology is that the responses are often influenced by participants' interpretation and transitory perceptions. The bias could be particularly strong when the questions are related to patterns of usage (Lazar et al. 2010). These limitations could be overcome if the collected qualitative data is cross-referenced with quantitative. Relating data on students' activities and interaction with support services to results of satisfaction surveys may enable the discovery of new factors for satisfaction or dissatisfaction. The importance of the identified factors may be then verified through focus groups with students (Stubbs 2013). For example, the consolidation of data from the University support services with data from units providing engagement opportunities may reveal that students involved in extracurricular activities have fewer interactions with the support units. A hypothesis in this case would be that students involved in societies, sports or music clubs get the needed support and information within the group they are members of. A focus group with members of clubs and societies may help the University to validate this assumption.

Data-driven strategic planning: The better understanding of student needs and behaviour would ideally facilitate strategic decision-making and resource planning processes and help the University to establish a measurable and target-oriented student experience strategy. For instance, if the hypothesis that students involved in different clubs and societies help each other and therefore have less visits at the University's support services, is proven by focus group activity, the University may set as a strategic target to increase students' extracurricular involvement.

Understanding the big picture: Having data from all units related to the student experience in a single application, would enable the University's senior management to quickly access and analyse data from different sources. Instead of looking at different reports they would have the whole information at a glance. The ability to see the big picture could help the University to ask the right questions and to discover unknown patterns and relationships. The findings may potentially identify students at risk and allow the University to perform preventive actions and increase student success and satisfaction.

Customized Services: The ability to identify clusters of students with similar issues may allow the University to offer more customized services. For example the analysis of historical data from

different University services may reveal top questions asked by particular student groups. This knowledge could be used for customizing leaflets (e.g. for different schools) or for providing more personalized and useful information on the University's online portal iSaint.

4 Demo Dashboard Requirements

4.1 Introduction

After discussing the main motivations and the potential benefits of the proposed Student Experience Dashboard, the next chapters of this dissertation discuss the realization of a pilot project whose primary goal is the implementation of a demo Student Experience Dashboard. Going through the process of collecting the needed data, building, and demonstrating the dashboard to members of the University Student Services management, serves as an initial investigation of the University's readiness for the implementation of the university-wide Student Experience Dashboard.

The demo dashboard consolidates data from three University service units: Advice and Support Centre (ASC), IT Service Desk (UniDesk), and In-sessional English Language Service (IELS).

4.2 Objectives and Potential Benefits

The main objective of the demo Student Experience Dashboard is to explore and demonstrate the potential of combining and analysing data from different units related to the student experience.

Potential benefits:

- To help the University management to better understand students' needs.
- To allow the University to monitor students' usage of different University services at a glance.
- To allow the University to identify clusters of students.
- To allow the University to discover unknown narratives, relationships and trends in students' usage of University services.
- To allow the University to confirm hypothetical patterns and relationships in students' usage of the University services.

4.3 Functional Requirements

Since the notion of a university-wide dashboard related to the student experience was a novelty for the University, no requirements for its functionalities were provided. Therefore the requirements presented below are based on the dashboard's objective and the structure of the obtained data sets.

- The dashboard should display the following measures:
 - Number of visits – the number of students' interactions with each of the three services.

- Number of visitors – the number of students who visited each of the three services.
- Visitors percentage in relation to all students- what percentage of all students enrolled in a particular academic year have visited any of the University services.
- The dashboard should allow comparison of the above measures with the previous year.
- The dashboard should allow the possibility for cluster analysis by:
 - Age group
 - Faculty
 - Gender
 - Level of Study
 - Nationality
 - Nationality group
- The dashboard should allow users to identify quickly:
 - Top student groups using University services
 - Top visit subjects for different groups of students
 - Associated questions: students who asked this question, also asked this...in ...(month)
 - Peak months for each of the services.
 - Number of students who visited more than one of the services.

4.4 Non- functional Requirements

- The dashboard should not allow a single student to be personally identified by the data displayed (assuring compliance with data protection regulations).
- The dashboard should provide satisfactory performance and user experience.
- The dashboard should present the data in a clear and useful manner, facilitating users' knowledge discovery process.

5 Qlikview

Having discussed and listed the requirements for the demo dashboard in the previous chapter, the dissertation now introduces the tool that was chosen to be used for the implementation of the demo dashboard and the reasoning behind this choice. Furthermore, it explains the main syntax that was used during the development process.

5.1 About Qlikview

Qlikview is a business intelligence tool, provided by the software company QlikTech, founded in Sweden in 1993. The main components of the Qlikview platform are:

- Qlikview Desktop - the interface for building Qlikview applications (dashboards and reports). After an application is finished, it is then uploaded on the Qlikview Server.

- Qlikview Server – the Qlikview server itself stores and processes the data needed for the applications through its in-memory engine. This means that the server doesn't have to contact the data sources every time when a query is performed.
- Qlikview Portal – allows users to access the Qlikview applications, loaded on the server.

One of the key Qlikview characteristics is that it uses associative technology. This means that the Qlikview in-memory engine makes associative connections between every record (row) in every data field (column) in the schema and thus allows users to instantly see all data associated with a selected record. This is illustrated in Figure 8, provided by Qlikview. The diagram shows that if, for example, a particular product is selected, users can instantly see which sales persons have sold this product and in which states and regions the product has been sold.

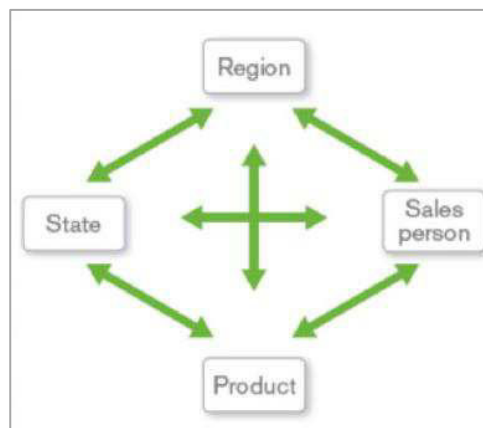


Figure 8 Qlikview Associative Technology (Source: (Qlikview 2012a))

5.2 Why Qlikview?

- 1) Qlikview has been used by the University of St Andrews since 2010, which ensures the following benefits of using it in this project:
 - Access to a license key.
 - Qlikview reading materials and books have been already obtained by the University BI team.
 - Most of the University administrative staff is already familiar with Qlikview dashboards and therefore less or no training time would be required.

2) As can be seen on the diagram below, according to Gartner’s annual report examining the performance of BI vendors, QlikTech is one of the leading BI and analytics vendors on the market (Figure 9). Working with this tool allows the developer to gain knowledge and experience valuable for future BI projects and career development.



Figure 9 Magic Quadrant for Business Intelligence and Analytics Platforms (Source: (Gartner 2013))

- 3) Qlikview and its community provide a variety of excellent support services and reading materials:
- 1) Qlikview Help Menu
 - 2) Qlikview Reference Manual
 - 3) Qlikview Community
 - 4) Qlikview consultants’ blogs and video tutorials

5.3 Qlikview Syntax

The main Qlikview concepts and syntax are explained and exemplified in this section. The aim is to prepare the reader for the following implementation chapter, which discusses the development of the demo dashboard. The content in this section is based on the Qlikview guideline books – (Qlikview 2012b) and (Qlikview 2012a).

5.3.1 Introduction

Developing a Qlikview dashboard consists of two main phases or tasks – data modelling and user interface design. The first one involves creating an optimal data model, in terms of performance and

maintenance. The data model depends on the dashboard's required functionalities and objectives. Data modelling is achieved by using Qlikview Load Script. The second phase is the design phase, which involves the building the dashboard's user interface. It contains mainly charts and tables for displaying the information stored in the data model. The calculations in all charts and tables are defined with expressions and Set Analysis. Although the Qlikview literature distinguishes two roles: developer and designer, both tasks are often performed by the same person, who is referred to as a developer in this section.

5.3.2 Load Script

The Qlikview Load Script functions as an ETL script. It allows Qlikview developers to connect to and extract data from various data sources, to perform data transformation operations and to load (store) this data into a dashboard application. The Load Script is created in the Qlikview Script Editor window.

Before going into detail about Qlikview Load Script, it should be pointed out that Qlikview automatically associates fields from different tables with the same name, therefore extra caution has to be taken when loading tables and naming their fields (columns).

5.3.2.1 Loading Data

In general Qlikview can load data from:

- Files (e.g. spreadsheets, XML files), databases, web pages – Load Statement

Load syntax example:

```
[Test Table]:  
LOAD  
ID,  
FirstName & ' ' & LastName as Name,  
Address  
FROM D:\Test\Test.xlsx
```

Table Name

Renaming a field

Combining strings

File path

- previously loaded tables – Resident Load Statement

Resident Load syntax example:

```
LOAD
Name,
Address
RESIDENT [Test Table];
```

Field Names, as from the extant table

The name of the extant table (since the script is executed from top to bottom, the tables' order must be considered)

- data generated in the script – Inline Load Statement

Inline Load syntax example:

```
Colours:
Load * Inline
[ColourID, Colour
0, Blue
1, Green
2, Red];
```

Generated Field Names

Generated Field Values

As can be seen in the examples above, the Qlikview Load Statement is similar to the SQL SELECT statement and also can be followed by WHERE, GROUP BY and/or ORDER BY clauses.

5.3.2.2 Counter Fields

The creation of counter fields (fields used for counting records) is one of the basic concepts in Qlikview data modelling. In contrast to SQL where the table of the field can be specified, in Qlikview the tables cannot be specified and therefore counting keys (which connect two tables) can be problematic.

For example, a data model has two tables 'Orders' and 'Customers', which are connected through the Customer ID field (Figure 10). The 'Customers' table stored records of all customers (5000 customers), where the 'Orders' table stores only the customers which have made an order (3000 active customers). When typing `Count (DISTINCT [Customer ID])`, Qlikview does not know which records to count - the ones from 'Customers' or those from 'Orders'.

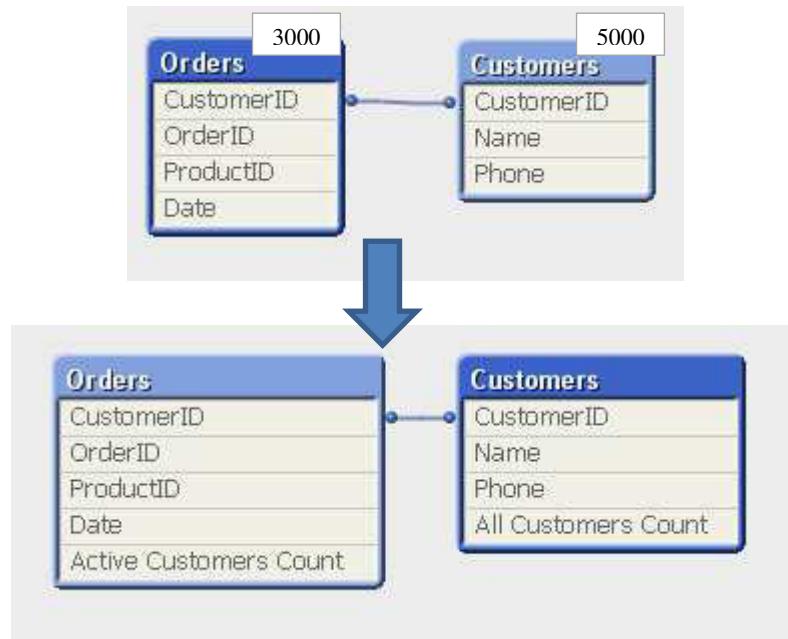


Figure 10 Exemplifies the need of counter fields

To avoid this, counter fields should always be specified. This is implemented through creating new field storing the Customer ID in each table and giving them specific names:

Customers:

```
[Customer ID] as [All Customers Count]
```

Orders:

```
[Customer ID] as [Active Customers Count]
```

Now when typing `Count (DISTINCT [All Customers Count])` Qlikview is sure the records from which table it needs to count and will return 5000. Whereas `Count (DISTINCT [Active Customers Count])` will return 3000.

5.3.2.3 Joins, Concatenation, and Mapping

Joins

Joins in Qlikview are similar to the SQL joins and also can be Inner, Outer, Left or Right. They are used for merging data from multiple tables into a single table. To be joined the tables must have one or multiple matching field names amongst each other. Joins are typically used when developers want to add new fields to already existing table. Joins can be forced by using the JOIN statement with prefix INNER, OUTER, LEFT or RIGHT before the LOAD. When no prefix is specified, Qlikview will perform an inner join.

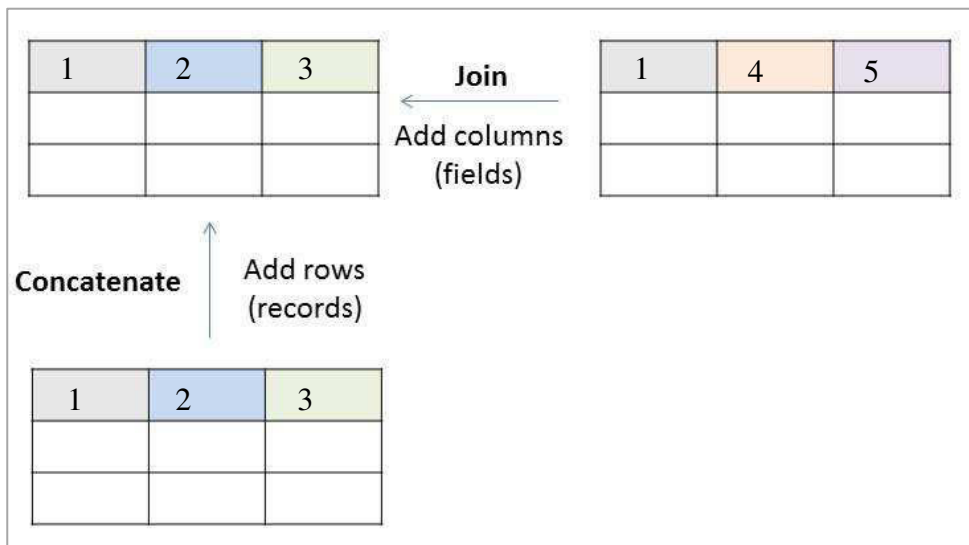


Figure 11 Illustrating the difference between JOIN and CONCATENATE (created by the author)

Concatenation

While joins are used to add new fields, a CONCATENATE statement is used when developers want to add records or in other words when data records from multiple tables needs to be appended into a single one (Figure 11). It has the same function as the SQL Union statement. The concatenation can be automatic in the case that two or more tables have exactly the same fields, or forced when there are some fields in one of the tables that do not exist in the others. When forced concatenation is performed the fields that do not exist in a table will contain NULL values. Concatenation can be forced by using the CONCATENATE statement before the LOAD.

Mapping

Qlikview mapping is used to add data from a single field to an existing table. This is performed by mapping tables and the Applymap() function which is similar to the Excel VLookup functions.

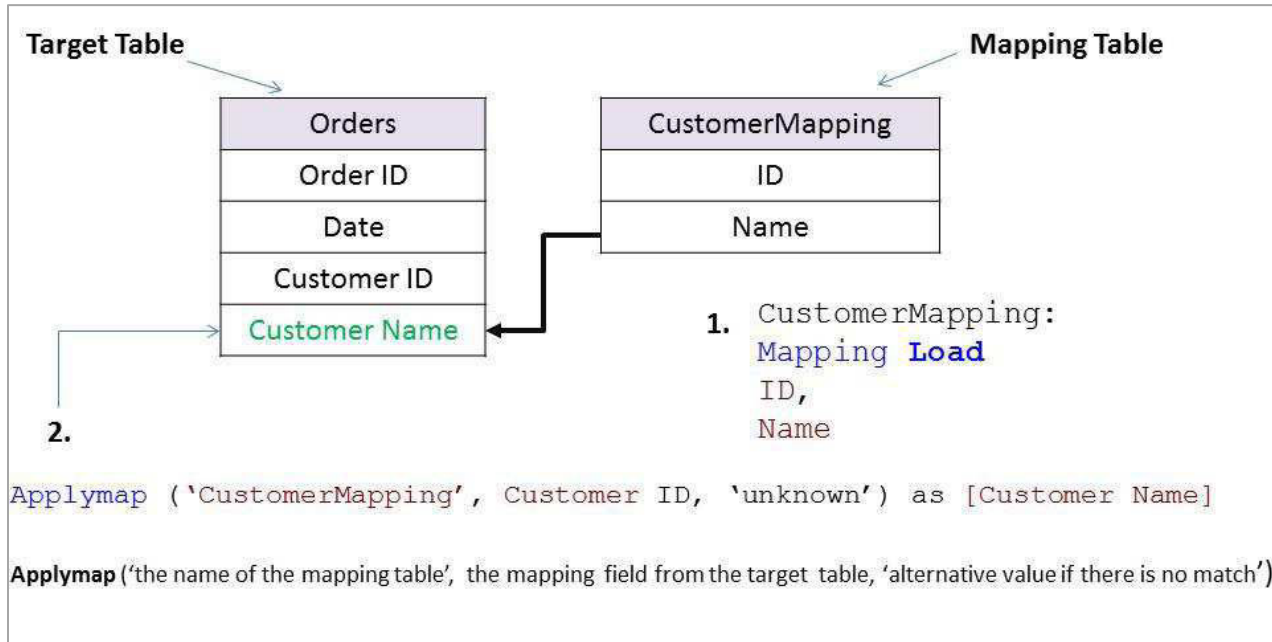


Figure 12 Example of Qlikview Mapping (created by the author)

The Qlikview mapping can be illustrated by the example illustrated in Figure 12: a data model has an 'Orders' table storing Order ID, Date, and Customer ID but the developer wants to add a field with the customers' names. In SQL this is typically resolved by joining tables. However, in Qlikview when developers want to add only a single field, rather than multiple, from one table to another, the Applymap() function is considered a better option. The first step is the creation of a mapping (lookup) table. A mapping table must contain only two fields: the first must be the one that stores values existing in the both tables (ID) and the second must store the values that need to be added (Name). The field names in the mapping table are not important (because no undesired associations can be made), since all mapping tables are automatically dropped at the end of the script execution. The second step involves the insertion of the Applymap() function into the code of the target table (Orders). It is important to notice that the script for the mapping table should be placed before the script of the target table. Furthermore, the Applymap() function cannot handle one-to-many relationships and in such a case it returns only the first value and ignores all others.

5.3.3 Set Analysis

Set analysis is used in the second phase of a dashboard development - the design phase, which involves the creation of charts and tables to display the data from the data model. Set Analysis can be included in aggregate functions to allow developers to modify the data set displayed in each chart or table. Using Set Analysis, developers can add or remove records from a data set based on certain criteria. For example, it allows them to limit the data set to contain only records from a certain year. Set Analysis is enclosed by curly brackets and contains three main components, described further in this section: Identifier, Operator, and Modifier (Figure 13).

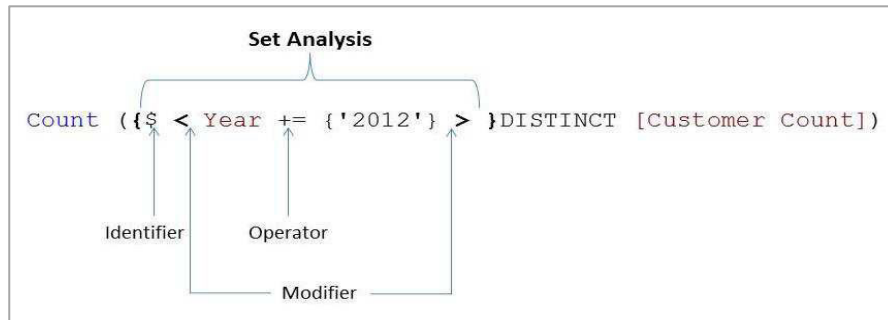


Figure 13 An Example of Set Analysis (created by the author)

Set Identifier

As the name suggests, this component identifies the data set which is needed for the expression. In general the Set Identifier could be either 1, representing all records in the data model, or \$, which is the default state and represents the records of the current selection (the selected data set).

Set Modifier

The Set Modifier is surrounded by `< >` and is used to filter the data set, defined by the Set Identifier. It plays the same role as an SQL WHERE clause and allows developers to add or remove selections in the data set. For example `$ < Month = {'April'} >` represents all records from the currently selected data set (selected by the user) where the month is April. To ignore or remove a certain data field from the current selection data set the following syntax can be used: `< Field name = >`.

Set Operator

The Set Operator is used to specify an interaction between two data sets. There are four types of Set Modifiers: Union (+), Exclusion (-), Intersection (*) and Symmetric Difference (/). A description of each of them is presented in Table 1.

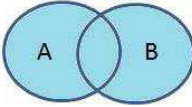
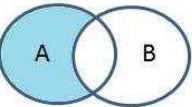
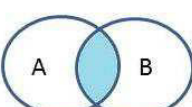
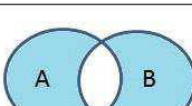
Set Operator Name	Usage in Set Analysis	Illustration	Description
Union	+		Returns a set that contains records from any of the both initial sets A and B.
Exclusion	-		Returns a set that contains records from set A but not from set B.
Intersection	*		Returns a set that contains common records that exist in both sets.
Symmetric difference	/		Returns a set that contains all records from both set except those which are common.

Table 1 Set Operators (created by the author based on (Qlikview 2012b))

Let's use the expression from Figure 12 to summarize. It means: return the number of distinct records in Customer Count field where the year is the currently selected year or 2012.

```
Count ({$ < Year += {'2012'} > }DISTINCT [Customer Count])
```

6 Implementation

6.1 Data Modelling

Having introduced the tool and the main syntax used in the demo dashboard's implementation, the dissertation continues with a chapter that analyses the obtained data sets and discusses the decisions made in the process of building the dashboard's data model. Furthermore, it presents the produced interface and explains its main capabilities. Finally, it analyses the lessons learned during the implementation of the demo dashboard.

6.1.1 Data Sources

The process of finding and obtaining access to data from units related to the student experience was the most time and effort-consuming part of the project. The reasons for this were a lack of senior management support for the project, delayed communication with the different units due to holidays, as well as cultural barriers, which are explained in detail in Chapter 8 of this document. Furthermore, due to the high sensitivity of the data included in the dashboard, the access to it was a matter of ethical considerations. To ensure compliance with data protection regulations the University of St Andrews CIO was involved in the early stages of the project. A short briefing session was held aiming to discuss the data privacy concerns and the required mitigation measures. As a result the project's ethical application has been approved by the University Teaching and Research Ethics Committee.

Finally, data from three service units at the University was obtained and included in the demo dashboard:

- 1) ASC – The Advice and Support Centre is the central front-office where students can seek help for any issues (e.g. academic support, visa issues, accommodation payments, counselling).
- 2) IT Service Desk - The IT Service Desk provides technical support to all students and staff members of St Andrews University. It uses the UniDesk self-service portal for keeping records (tickets) for each incident.
- 3) IELTS – The In-sessional English Language Service helps non-native speaker students to develop their English language skills.

The acquired data sets contain records of students' interaction with each of these services. In order to fulfil one of the main objectives of the dashboard, namely to allow comparison with the previous year, records for both academic years - 2011/2 and 2012/3 were needed. Since the academic year at St Andrews University starts on the 1 August and ends on the 31 July, it was required to have records from the beginning of August 2011 till the end of July 2013. However, the provided data did not cover the whole period and additional records have been randomly generated using the trial version of AbleBits data generator for MS Excel. Furthermore, I was provided with a Student Records list, extracted from the University's data warehouse.

6.1.1.1 ASC data set

The data set from ASC has been provided to me in a QVD file - a file containing tabular data exported from Qlikview. The fields (columns) in the table are described in Table 2, where the ones loaded in the dashboard are marked as bold:

Field (Column) Name	Description
ASC_VISIT	Contains the ID of each visit at ASC as a string. One visit can be related to more than one ASC_VISIT_SUBJECT_GROUP and ASC_VISIT_SUBJECT. Therefore the same Visit ID could exist in more than one record (row).
ASC_VISIT_DATE	Stores the visit date in the following format 'dd/MM/yyyy' (e.g. 31/08/2011).
ASC_STUDENT_ID	Stores the Student ID, if the visitor is a student or 'non student' in case that the visitor is not a student.
ASC_VISIT_SUBJECT_GROUP	Contains the visit category. (e.g. 'Student Finance', 'Registry Processes', 'Student Accommodation').
ASC_VISIT_SUBJECT	Stores the visit subject (subcategory). (e.g. 'Status/Bank Letter', 'Matriculation', 'I need this form stamping').
ASC_STUDENT_FLAG	Contains two values ('Y' and 'N'), showing whether the visitor is a student or not.
ASC_YEAR_OF_STUDY	Keeps the year of study of the student visitors at the moment of the visit. Although this data field is related to the dashboard objectives, it has not been included in it, because none of the other data sets contain students' year of study. Loading the data field would result in null values for the year of study of all students, which have not visited ASC.
ASC_VISIT_TYPE	Shows whether the visit (interaction) was 'In-person', per 'Email' or by 'Phone'.
ASC_LOCATION	Contains the place where the visit took place at: 'ASC', 'Union' or 'Other'.

Table 2 ASC data fields

6.1.1.2 UniDesk data set

The obtained UniDesk data set was stored in a Microsoft Excel spreadsheet extracted from the UniDesk database (Table 3).

Field (Column) Name	Description
TICKET_ID	Contains the ID of each ticket (visit) at UniDesk as a numeric value. One ticket could be related to only one issue (request).
POSTING_DATE	Stores the visit date as a string in the following format '31-AUG-11 00.00.00'
OPENEDBY_PERSON_EMAIL	Contains the email address of the person who requested assistance (in this case called 'visitor'). This is the only unique identifier of the visitors in this data set.
TICKET_CATEGORY	Contains the visit category. (e.g. 'Web & Business Apps', 'Accounts & Authentication').
TYPE_OF_PROBLEM	Stores the visit subject (subcategory). (e.g. 'Eduroam', 'Hardware', 'Email').
SLA_COMPLETED_INTIME_YN	Shows whether the ticket was completed ('Y') or not ('N') within the agreed SLA time.
TICKET_ENTRY_METHOD	Shows whether the service has been requested by 'Email', 'In person', by 'Telephone' or through the system's 'Web interface'.
PRIORITY	Stores ticket's priority.

Table 3 UniDesk data fields

6.1.1.3 IELS data set

The data set provided by IELS was also stored in a Microsoft Excel spreadsheet (Table 4).

Field (Column) Name	Description
User Name	Contains the username of each person (visitor) who attended an IELTS tutorial.
Date of Tutorial	Stores the visit date in the following format 'dd-MM-yyyy' (e.g. 31-08-2011).
Faculty/School	Stores the faculty or the school, which the visitor is enrolled in.
Status	Shows visitor's level of study.

Table 4 ILES data fields

6.1.1.4 Student Records data set

The Student Records data set has been extracted from the University's data warehouse and has been provided to me in an Excel spreadsheet (Table 5). It contains information about students for each academic year from 2010/1 till 2012/3.

Field (Column) Name	Description
STUDENT_ID	Contains a student's Student ID.
AYR_CODE	Stores the academic year in format '2012/3'
USER_ID	Stores a student's Username.
AGE	Contains a student's age.
GENDER	Stores a student's gender ('M' or 'F')
NATIONALITY_NAME	Keeps a student's nationality.
PROGRAMME_NAME	Stores the name of the programme that a student is enrolled in.
PROGRAMME_CODE	Contains the code of the programme that a student is enrolled in during the corresponding academic year.
EMAIL	Contains a student's email address.

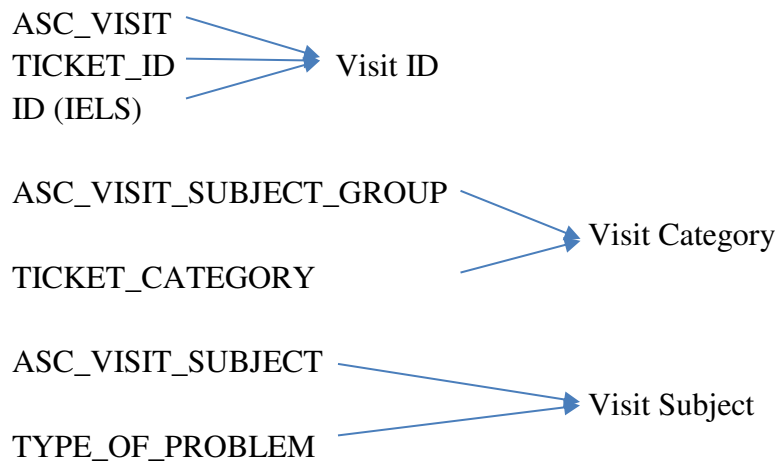
Table 5 Student Records data fields

6.1.2 Extract- Transform- Load Process

6.1.2.1 Step 1: Loading the source data

The first step of the implementation process was to extract and load the needed source data into the dashboard application. The following decisions and data transformations were made at this step:

1. Only the data fields which contribute to fulfilling the dashboard’s objectives were loaded into the application. This avoids confusion caused by having too many fields and improves the dashboard’s performance, since it will reduce the amount of data stored in Qlikview’s in memory engine.
2. In order to ensure consistency, all fields storing student information (e.g. level of study, age, faculty) were extracted from the Student Record data set and not from the Service units’ ones.
3. In order to create unified field names, an interaction with any of the services was called a “Visit”. Hence the following data fields were renamed.



Since Qlikview automatically associates fields from different tables with the same name, extra caution were taken when renaming fields. To avoid undesired associations, the source of the data has been added to each field name (e.g. Visit Category ASC).

4. The ASC data set:

Since the dashboard aims to display students’ usage of the University services, all records flagged as ‘non students’ (having ‘N’ in the ASC_STUDENT_FLAG field) were excluded from the loaded data set. This has been implemented through a WHERE clause after the LOAD Statement:

```
WHERE (ASC_STUDENT_FLAG = 'Y');
```


5. The IELTS data set:

As showed in Table 3 the IELTS data set does not contain a unique identifier for each tutorial. Therefore a new column was added to the source Excel file called 'ID', where a unique numeric value, starting from 1, was assigned to each tutorial record. Another possible solution could have been the creation of a composite key, combining both fields Username and Date of Tutorial. However, in some cases a student can have more than one tutorial per day and therefore this solution was rejected.

6. The UniDesk data set:

The only information which this data set contains about the service users (visitors) is their email addresses. Since it stores neither student ID nor Username, the only possible solution was to derive the Username from the email address. This is possible only for email addresses which are in the University domain (username@st-andrews.ac.uk) therefore all records having a non-university email were excluded from the data set through a WHERE ...LIKE clause:

```
WHERE (OPENEDBY_PERSON_EMAIL like '*@st-andrews.ac.uk');
```

The extraction of the Username from the University email address has been achieved through a Subfield () string function. This function has the following syntax: subfield (string, 'delimiter', index), where index is an optional parameter showing which substring should be returned: 1- the string before the delimiter, and 2- the string after the delimiter. In this case the string is the email address (username@st-andrews.ac.uk) stored in the OPENEDBY_PERSON_EMAIL field for each record; the delimiter is @; and the substring, which should be returned (username), is the one before the delimiter '@':

```
Subfield (OPENEDBY_PERSON_EMAIL, '@',1) as [Username IT]
```

Another data field in the UniDesk data set that required to be transformed was the POSTING_DATE, whose format had to be changed from text to a numeric value.

7. The Student Records data set:

All records which do not have assigned PROGRAMME_CODE were considered as non-students and therefore not loaded into the dashboard application. The records for academic year 2010/1 were also excluded, since the service units' data sets span only 2011/2 and 2012/3 academic years. These decisions were implemented through a WHERE NOT clause and an IsNull () function, which returns all records having a NULL value in the defined field:

```
WHERE NOT IsNull (PROGRAMME_CODE) AND NOT (AYR_CODE = '2010/1');
```

Important transformation to the Student Records data set is the assignment of AGE and NATIONALITY_NAME field values into groups. The aim is to allow better cluster analysis and to avoid personal identification of students from the data displayed on the dashboard.

- Creating the Age Group field: the values from the AGE field were assigned into three groups through a nested if statement:

```
if(AGE<= 21, 'under 21', if(AGE> 21 and AGE <=30, '22-30', if(AGE>30, 'over 30')
) )as [Age Group]
```

- Creating the Nationality Group field: the field was created by assigning the values from the NATIONALITY_NAME field into three groups – ‘Home’, ‘EU’ and ‘Non EU’, using a nested IF Statement. It’s logical meaning is that that if the value in the NATIONALITY_NAME field is part of the United Kingdom, it should return the value ‘Home’, in the case that the value is not part of UK it should check if the value is part of the European Union and return ‘EU’ in the case of yes and ‘Non EU’ in the case of no.

```
if (NATIONALITY_NAME = 'Welsh' or NATIONALITY_NAME = 'Northern Irish' or
NATIONALITY_NAME = 'Scottish' or NATIONALITY_NAME = 'English' or NATIONALITY_NAME like
'Brit*', 'Home',
if(NATIONALITY_NAME = 'Austrian' or NATIONALITY_NAME = 'Bulgarian' or
NATIONALITY_NAME = 'Cypriot (European Union)' or NATIONALITY_NAME = 'Czech' or
NATIONALITY_NAME = 'Danish' or NATIONALITY_NAME = 'Dutch' or NATIONALITY_NAME
= 'Estonian' or NATIONALITY_NAME = 'Finnish' or NATIONALITY_NAME = 'French'
or NATIONALITY_NAME = 'German' or NATIONALITY_NAME = 'Greek' or
NATIONALITY_NAME = 'Hungarian' or NATIONALITY_NAME = 'Irish' or
NATIONALITY_NAME = 'Italian' or NATIONALITY_NAME = 'Latvian' or
NATIONALITY_NAME = 'Lithuanian' or NATIONALITY_NAME = 'Luxemburger' or
NATIONALITY_NAME = 'Polish' or NATIONALITY_NAME = 'Portuguese' or
NATIONALITY_NAME = 'Romanian' or NATIONALITY_NAME = 'Slovak' or
NATIONALITY_NAME = 'Slovenian' or NATIONALITY_NAME = 'Spanish' or
NATIONALITY_NAME = 'Swedish' or NATIONALITY_NAME = 'Croatian' or
NATIONALITY_NAME = 'Maltese', 'EU', 'non EU')) as [Nationality Group]
```

Another key data transformation to the Student Records data set is the creation of Faculty and Level of Study fields. Since the variety of values in the PROGRAMME_NAME field was too large for clustering and analysis, it was decided to transform it into two new fields providing more summarized and meaningful information.

- Creating the Faculty field: St Andrews University has 4 faculties - Science faculty, Arts faculty, Medicine faculty and Divinity faculty. Since not all values in the PROGRAMME_NAME field contain information about the faculty, some of the records have ‘unknown’ Faculty. This was achieved through another nested IF statement:

```
if (PROGRAMME_NAME like '*Science*', 'Science', if (PROGRAMME_NAME like '*Art*',
'Arts', if (PROGRAMME_NAME like '*Divin*', 'Divinity',if (PROGRAMME_NAME like
'*Medicin*', 'Medicine', 'unknown')))) as Faculty,
```

- Creating the Level of Study field: Using a similar nested IF statement, the following values were derived: 'Diploma/ Certificate', 'International Foundation Programme', 'Master', 'Non-degree/ non-graduating', 'PhD', 'Research Visiting', 'Sabbatical year', 'Summer School', 'Undergraduate'.

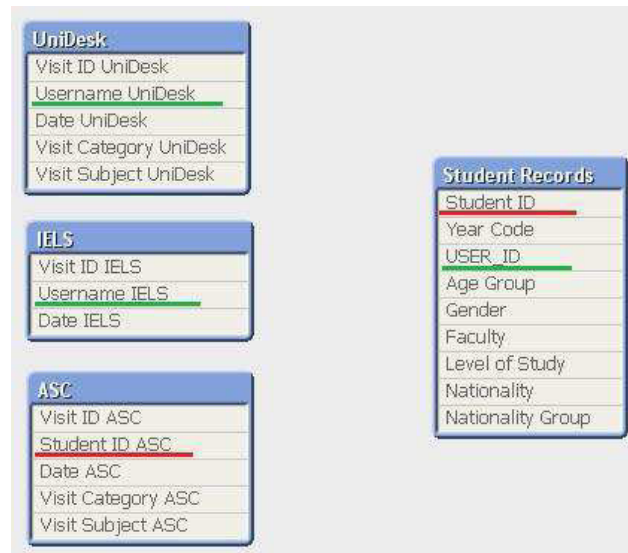


Figure 14 Data Model at the end of step 1

6.1.2.2 Step 2: Creating the 'Visits' Fact Table

As can be seen in Figure 14 at the end of step 1, the needed fields from the four source data sets were loaded, renamed to user friendly and unified field names, and transformed. The next step involves combining the service units' tables into a single fact table named 'Visits' and creating the measures identified in Section 4.3 of this document: number of visits, number of visitors and number of all students (required for calculating the visitors percentage in relation to all students). Important considerations and tasks at this stage were:

1. One of the first issues that can be spotted in the data model displayed above is that while in the 'IELS' and the 'UniDesk' tables students are identified only by their username, in the 'ASC' table the username is missing and the students are identified only by their student ID. To resolve this issue the student ID needs to be matched with the corresponding username. This was achieved by using the 'Student Records' table containing both fields. Since a single field was needed to be appended from one table to another, the best option was to use the Applymap() function. By applying the syntax explained in Section 5.3.2.3 two separate mapping tables were created. The first one allows the Applymap() function added to the 'IELS' and the 'UniDesk' tables to use the username and look up the student ID. The second mapping table allows the Applymap() function inserted in the 'ASC' table to use the student ID and look up the username. An important lesson that needs be pointed out here is that the

look up field and the field in the mapping table must have the same format. Knowing this may save a valuable time when searching for the reason behind a not working Applymap() function.

2. The next task was to create the measures - Number of Visitors, Number of Visits, and Number of All Students, which was enabled by adding counter fields in the script. The Visitors Count field was created from the Student ID field, the Visits Count field was created from the Visit ID, and the All Students Count field from the STUDENT_ID.
3. Since 'ASC', 'IELS' and 'UniDesk' tables contain the same field names, it was decided that they should be concatenated into a single fact table named 'Visits'. Two tasks were performed to prepare the tables for the concatenation. First, in each of the initials tables a Service Type field containing the name of the corresponding service was manually added.

e.g. 'ASC' as [Service Type ASC]

Second, the Visit ID fields in the 'ASC' and the 'IELS' tables were numerical values and therefore there was a risk of having duplicate numbers, which could have been problematic after the concatenation. To resolve this issue a corresponding letter was added to the ID numbers.

e.g. 'A' & ASC_VISIT as [Visit ID ASC]

After the completion of these two steps, the fields were renamed (the name of the service units were removed) and a forced concatenation was implemented. Since the IELS table does contain two of the fields – Visit Category and Visit Subject, the value in these fields for the IELS records is NULL.

4. For easier maintenance and code readability all heavy data transformations were performed in the 'ASC', 'IELS' and 'UniDesk' tables, where the source data was initially loaded. After the creation of the concatenated 'Visits' table, all three tables with the initial data transformations were dropped.

The image shows two database tables side-by-side. The left table is titled 'Visits' and the right table is titled 'Student Records'. Both tables have a blue header and a yellow body with a blue border.

Visits
Visit ID
Student ID
Username
Service Type
Date
Visit Category
Visit Subject
Visitors Count
Visit Count

Student Records
STUDENT_ID
Year Code
USER_ID
Age Group
Gender
Faculty
Level of Study
Nationality
Nationality Group
All Students Count

Figure 15 Data model at the end of Step 2

6.1.2.3 Step 3: Connecting the 'Visits' and the 'Student Records' Fact Tables

After creating the measures and consolidating the initial 'ASC', 'IELS' and 'UniDesk' tables into a single fact table 'Visits' (Figure 15), the next step involves connecting both fact tables 'Visits' and 'Student Records'.

1. The main problem with the 'Student Records' table was that the STUDENT_ID field is not a unique identifier of the records in the table. As can be seen in Figure 16, the table contains information about each student in each year. The example in the figure shows that although the value in the STUDENT_ID field is the same, the values in all other fields are different, since the student was studying an Undergraduate program in the Science Faculty in 2011/2, but a Master program in the Arts Faculty in 2012/3. This issue was solved by creating a composite key of STUDENT_ID and Year Code fields:

```
STUDENT_ID & ' ' & 'in' & ' ' & AYR_CODE as [Student in Year]
```

However, this solution cannot deal with rare situations when one student is enrolled in two programs at the same year.

STUDENT_ID	Year Code	Level of Study	Faculty
A	2011/2	Undergraduate	Science
A	2012/3	Master	Arts

Figure 16 Exemplifies a sub set of 'Student Records' table

2. To enable an accurate analysis of, for example, how many students from the Arts Faculty have visited any of the University service units, both tables 'Visits' and 'Student Records' were connected on the created Student in Year field. The interesting part here was the creation of this field in the 'Visits' table, since it required to extract the academic year from the Date field and to turn it in the proper format ('2012/3') for each of the concatenated tables. This was implemented through a complicate if statement:

```
if(num(month(ASC_VISIT_DATE))<=7,Year(ASC_VISIT_DATE)-1&'/'&right
(Year(ASC_VISIT_DATE),1),Year(ASC_VISIT_DATE)&'/'&right(year(ASC_VISIT_DATE)+1,1))
as [AYear ASC]
```

The created field was after that used in the 'Visits' table composite key:

```
[Student ID ASC]& ' ' & 'in' & ' ' & [AYear ASC] as [Student in Year]
```

The data model at the end of this step can be seen in Figure 17.

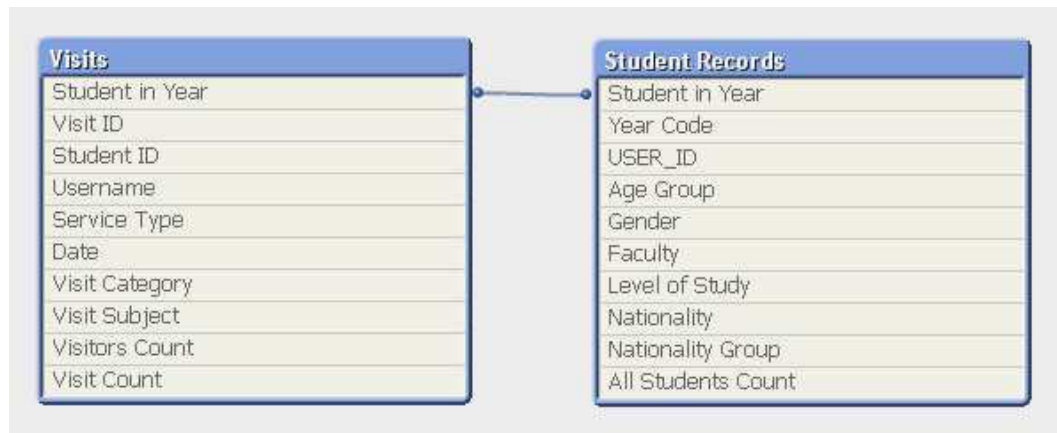


Figure 17 Data model at the end of Step 3

6.1.2.4 Step 4: Creating the 'Master Calendar' Table

During the previous step the both fact tables 'Visits' and 'Student Records' were connected on the derived Student in Year field. The next task is the creation of the 'Master Calendar' table. This is a common table for Qlikview data models, since it is considered that storing data fields outside the fact tables is a good practice. The process and the script for creating a Master Calendar table are explained and demonstrated in (Qlikview 2012b). In summary it involves the creation of a temporary table containing all dates between the minimum and the maximum date in the fact table, in this case the 'Visits' table. The records from this temporary table were then used for deriving different date fields, such as Day, Month, Year, Month-Year, and WeekDay. A special field that was derived in this case was Academic Year by using the same syntax as in Step 3 for turning the Date into '2011/2', '2012/3' format. Finally, the 'Master Calendar' table was connected to the 'Visits' table on the Date field. The data model resulted is depicted in Figure 18.

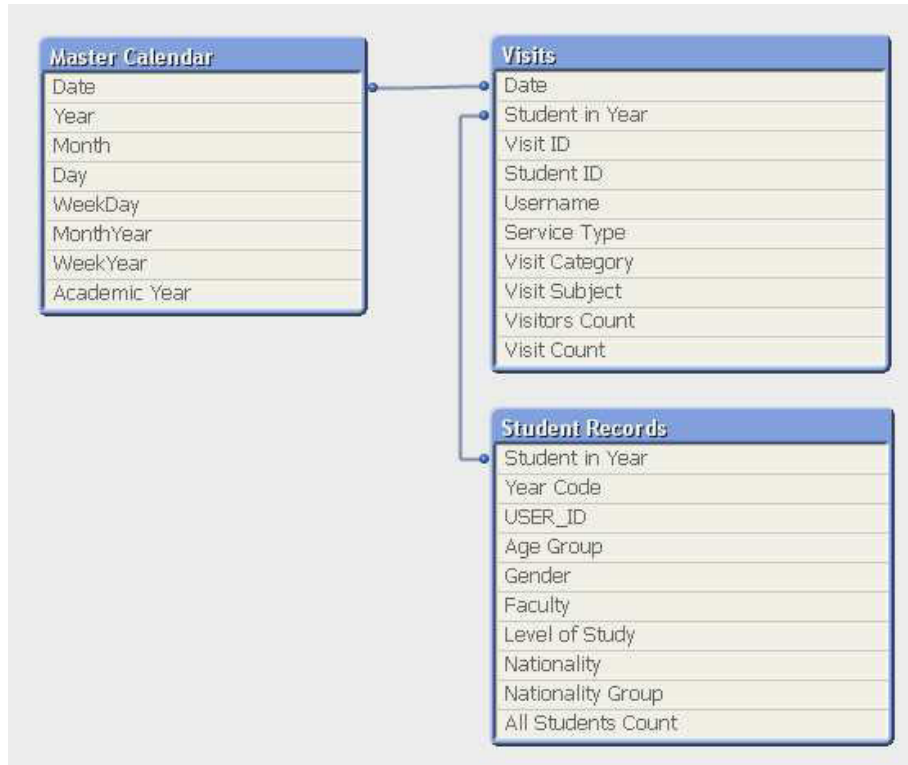


Figure 18 Data model at the end of Step 4

6.1.2.5 Step 5: Creating the ‘Associated Visit Subjects’ Table

The last step involves the creation of the ‘Associative Visit Subject’ and its additional calendar table. The idea for the Associative Visit Subjects was inspired by the Market Basket concept used in retail. Similarly to the Market Basket, which shows ‘the customer who bought this..., also bought ...’, the Associative Visit Subject shows ‘the visitors who asked this... also asked...’. The implementation of Associative Visit Subject is enabled by the associative Qlikview technology and is based on the implementation of a Market Basket, suggested by Lith (2009), who is a Qlikview Consultant. First the Visit Subject, the Service Type, and the Date for each Visit record were duplicated in another table, named ‘Associated Visit Subjects’ Table. The new table was connected to the ‘Visits’ table through the Student ID field. In contrast to the Market Basket, where we want to see the products which were ordered together and therefore the connection filed is the Order ID, here we want to see the questions asked by the same visitors and therefore the connection filed is the Student ID. To be able to see the dates of the associated visits a new calendar table was created and connected to the ‘Associate Visit Subject’ Table.

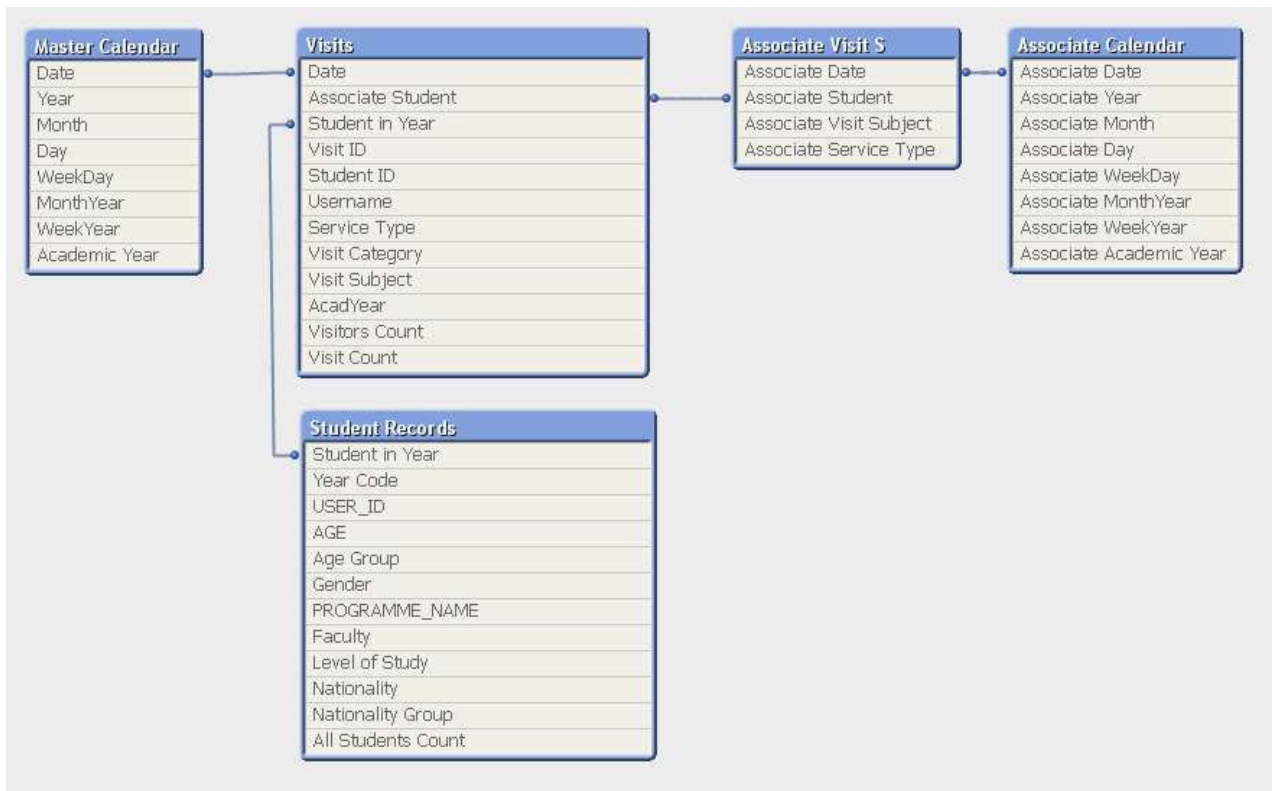


Figure 19 Final Data Model

6.1.2.6 Conclusion

The final data model, presented in Figure 19, is similar to a fact constellation schema, but the dimensional tables are merged with the fact tables. This structure is a result of the associative Qlikview technology, the dashboard's requirements and the nature of the data sources, which were extracts in tabular but in relation or dimensional format.

6.2 User Interface

This section discusses the main considerations which were taken into account when building the user interface of the demo dashboard. Furthermore it explains the main functionalities of the dashboard produced.

6.2.1 Introduction

After the completion of the dashboard's data model, the development of the user interface was started. The most important part of this phase was the writing of Set Analysis and expressions implemented in the charts. The design of the demo dashboard was based on the theme used in the already existing University's dashboards. However, when building the dashboard's interface best practices in dashboard design provided by Few (2006) were considered:

- The dashboard should not exceed the boundaries of a single iPad screen (980-610px).
- The visualized data should be arranged in a way that facilitates users' discovery process.
- The dashboard should provide an easy and intuitive navigation.
- The dashboard should accurately present the information.
- Additionally, the dashboard should have a Help sheet providing short instructions about Qlikview and key information about the data presented in the dashboard.

The rest of this section presents a short overview of each sheet (tab) of the dashboard. A brief description of each chart is provided as well as the Set Analysis and the expressions used in their implementation. Here should be emphasized that a considerable amount of the presented data was randomly generated, therefore it is important to state that no conclusions should be drawn based on this data either for particular clusters of student or for any of the three service units.

6.2.2 'Help' Sheet

The first sheet of the dashboard aims to familiarize the user with basic instructions about working with Qlikview as well as with the key terms used in the dashboard: Academic year, Student, Visit and Visitor (Figure 20).

The screenshot shows a Qlikview interface with a 'Help' tab selected. The main content area is titled 'Working with Qlikview - The Basics' and contains the following sections:

- How to select values?**
 - To select a value click on it or on the checkbox in front of it.
 - To select more than one value from the same category, press the Ctrl key and click on the desired values.
- What do the colours mean?**
 - When a value is selected, it turns green, all values associated with it turn white, and all values which are not associated turn grey.
 - Selected values ('2011/2' and Level of Study 'Diploma/Certificate')
 - Associated values ('Arts' and 'Science' Faculty)
 - Not associated values ('2012/3', 'Divinity', 'Medicine' and 'unknown' Faculty)
- How to clear selections?**
 - To clear a particular selection click on the selected value again or click on the 'rubber' icon on the Current Selections Box.
 - To clear all selections click on the **Clear** button on the Toolbar.

On the right side, there are two interactive elements:

- A filter for 'Faculty' with a dropdown menu showing 'Arts', 'Science', 'Divinity', 'Medicine', and 'unknown'.
- A filter for 'Level of Study' with a dropdown menu showing 'Diploma/ Certificate'.

Below these filters is a 'Current Selections Box' showing:

- Academic Year: 2012/3
- Level of Study: Diploma/ Certificate

Below the 'Current Selections Box' is a section titled 'About the Student Experience Dashboard' with the following content:

- When does the Academic year start?**
 - The Academic Year starts on 01 August and ends on 31 July.
- Who is considered a Student?**
 - Only persons existing in the Student Records List and who having assigned Programme in the corresponding academic year, are considered students
- What is a Visit?**
 - A Visit could be ASC visit, UniDesk Ticket, or IELTS Tutorial.
- Who is called a Visitor?**
 - A Visitor is a Student who used any of the three services: ASC, UniDesk and/or IELTS.

Figure 20 The 'Help' sheet of the demo dashboard

6.2.3 'Comparison with Last Year' Sheet

The second sheet was designed to enable year on year comparison. (Figure 21).

The items in the selection area on each sheet allow users to filter the data presented on the charts:

- by time period: academic year; and month.
- by student category: age group; faculty; gender; level of study; nationality; and nationality group.
- by service unit: ASC; IELTS; and UniDesk.

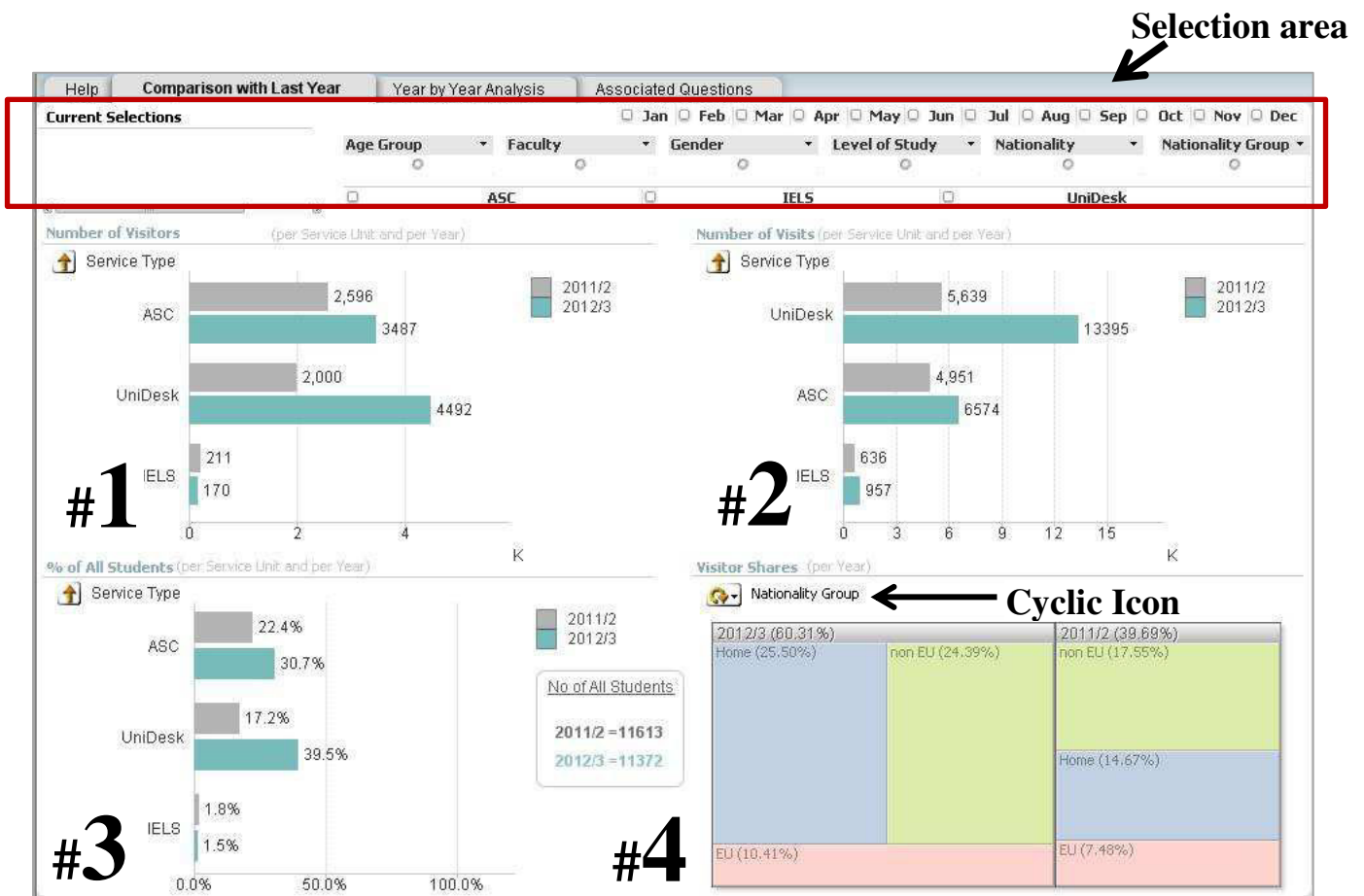


Figure 21 The 'Comparison with last Year' sheet of the demo dashboard

Chart #1

This chart shows the number of Visitors of each of the three service units in both academic years 2011/2 and 2012/3. In the implementation of the chart are used the following two expressions containing Set Analysis:

2011/2: `Count ({$<[Academic Year] = {'2011/2'} >}DISTINCT [Visitors Count])`

2012/3: `Count ({$<[Academic Year] = {'2012/3'} >}DISTINCT [Visitors Count])`

Chart #2

The second chart on this sheet displays the number of Visits to each service units in both academic years 2011/2 and 2012/3.

2011/2: `Count ({$<[Academic Year] = {'2011/2'} >}DISTINCT [Visit Count])`

2012/3: `Count ({$<[Academic Year] = {'2012/3'} >}DISTINCT [Visit Count])`

Chart #3

This chart presents the percentage of students who visited each of the in both academic years 2011/2 and 2012/3.

2011/2: `Count ({$<[Academic Year]={'2011/2'}>}DISTINCT [Visitors Count])/`
`Count (total{$<[Year Code]={'2011/2'},[Service Type] = , [Visit Category] =`
`,[Visit Subject] = , [Academic Year]= , Month = , MonthYear= >}`
`DISTINCT [All Students Count])`

2012/3: `Count ({$<[Academic Year]={'2012/3'}>}DISTINCT [Visitors Count])/`
`Count (total{$<[Year Code]={'2012/3'},[Service Type] = , [Visit Category] =`
`,[Visit Subject] = , [Academic Year]= , Month = , MonthYear= >}`
`DISTINCT [All Students Count])`

Chart #4

This block chart presents the distribution of Visitors by the different student categories. For example the chart on the sheet's screenshot (Figure 21) shows the distribution of Visitors by the category nationality group. It can be seen what percentage of the Visitors were EU, non EU, or Home students. By clicking on the Cyclic Icon users can switch between all six student categories, listed in the beginning of this section.

`Count (DISTINCT [Visitors Count])`

6.2.4 'Year by Year Analysis' Sheet

This sheet is design to allow more detailed analysis for each year (Figure 22). All charts on this sheet can be seen only after an academic year is selected (in the Selection area).

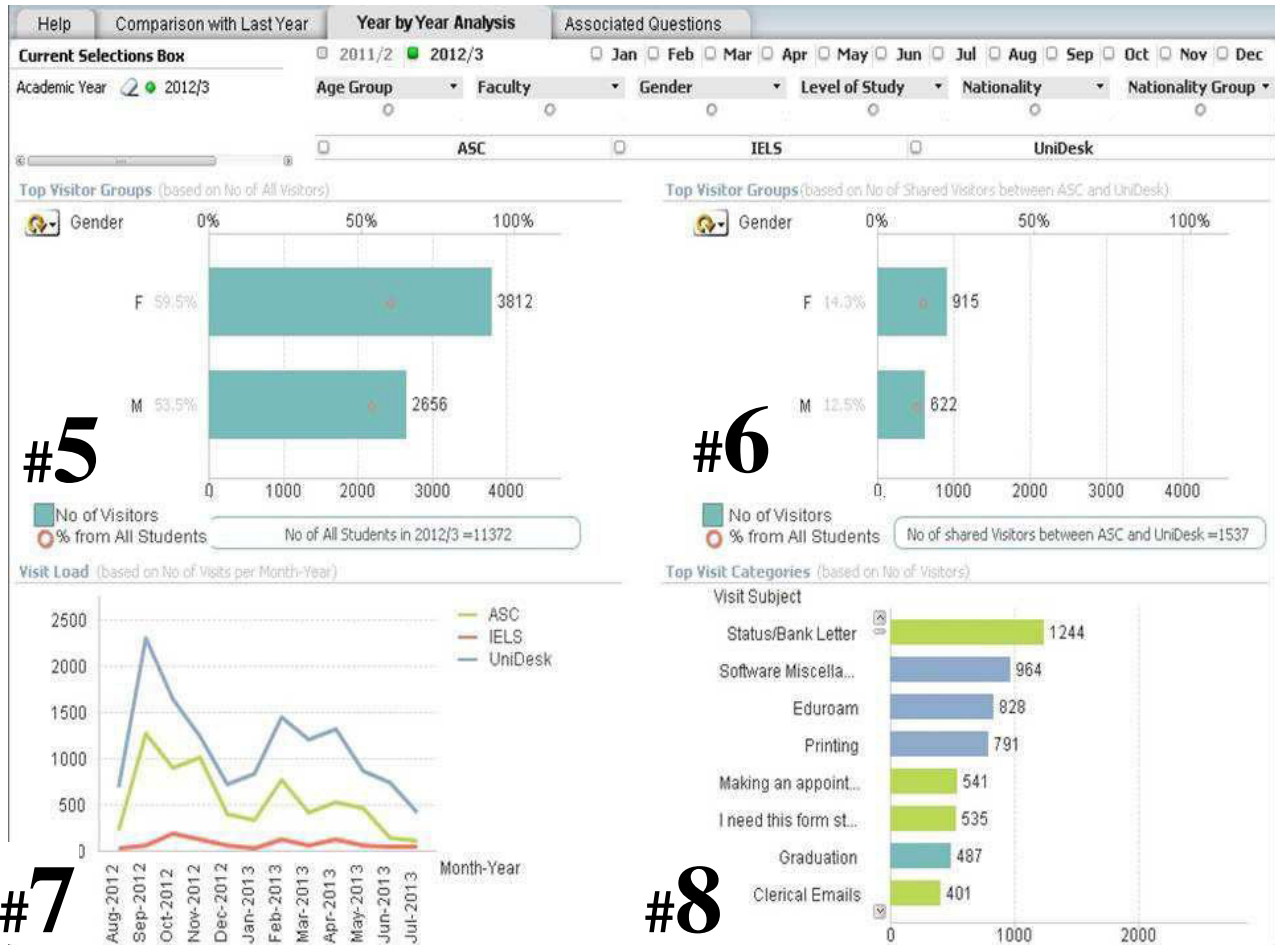


Figure 22 The 'Year by Year Analysis' sheet of the demo dashboard

Chart #5

This combo chart allows users to instantly see top Visitors groups by all six categories. Clicking on the Cycling icon again provides the opportunity to switch the displayed category. The chart also shows the percentage of Visitors in relation to all students from the particular group. For example the chart on the screenshot (Figure 22) displays that the top Visitors by gender are the females – 3812 of all Visitors in 2012/3 were females, while only 2656 were males. It also shows that these 3812 female Visitors were 59.9% of the female student body at the University in this year.

No of Visitors: `Count (DISTINCT [Visitors Count])`

% from All Students: `Count (DISTINCT [Visitors Count]) / Count ({ $ < [Year Code] = { "=" & $ (vTest) " }, [Academic Year] = , [Service Type] = , [Visit Category] = , [Visit Subject] = , Month = , MonthYear = > } DISTINCT ([All Students Count]))`

Chart #6

This chart is the same as the previous one, but it presents only the number of Visitors who have visited both services – ASC and UniDesk. The IELTS visitors were excluded because their Visitors are only international students and this would limit the analysis.

No of Visitors: `Count({<[Visitors Count]=P({<[Service Type]='ASC'})>}[Visitors Count])>*<[Visitors Count]=P({<[Service Type]='UniDesk'})>}[Visitors Count])>} distinct [Visitors Count])`

% from All Students: `Count({<[Visitors Count]=P({<[Service Type]='ASC'})>}[Visitors Count])>*<[Visitors Count]=P({<[Service Type]='UniDesk'})>}[Visitors Count])>} distinct [Visitors Count])/
Count ({$< [Year Code] ={"=$ (vTest) " }, [Academic Year] = , [Service Type] = , [Visit Category] = , [Visit Subject] = , Month= , MonthYear= >}
DISTINCT([All Students Count]))`

Chart #7

This Line chart is designed to present the visit load of each of the service units throughout the selected academic year.

`Count (DISTINCT [Visit Count])`

Chart #8

The last chart on this sheet shows the top Visit Subjects (based on the number of Visitors who asked about them), where the ASC subjects are represented by green bars, the UniDesk subjects – by blue bars, and the subjects which exist in both services are represented by blue-green bars.

`Count (DISTINCT [Visitors Count])`

6.2.5 ‘Associated Questions’ Sheet

The last sheet was designed to allow more advanced analysis. It allows users to see which Visit Subjects are associated with each other. In other words it shows the Visit Subjects (questions) which a single student asks within an academic year.

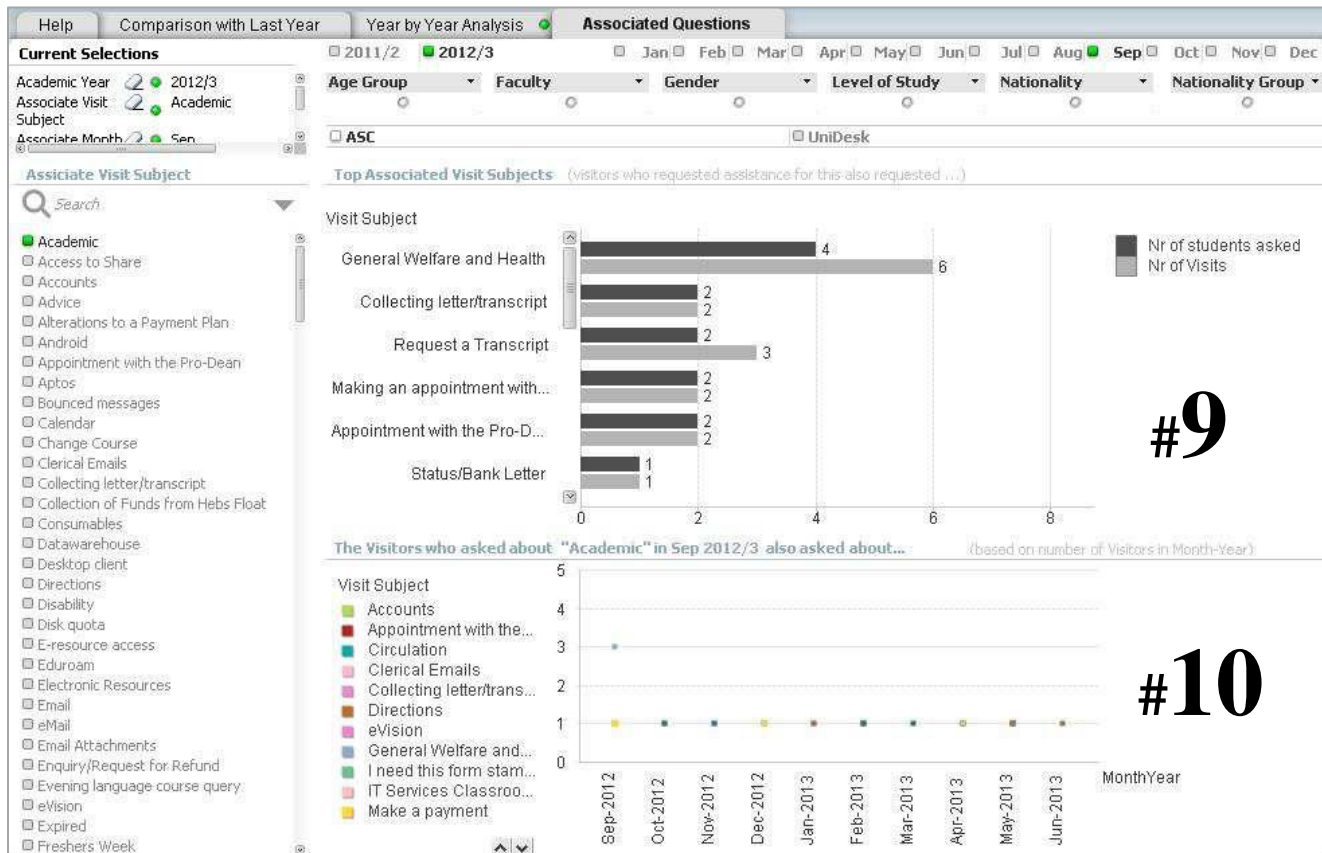


Figure 23 The 'Associated Questions' sheet of the demo dashboard

Chart #9

This chart presents Visit Subjects associated to a subject selected from the list on the left side of the sheet. For example the chart on the screenshot (Figure 23) shows that 4 of the Visitors, who asked about 'Academic' (Visit Subject) in September 2012/3, also made 6 Visits about 'General Welfare and Health'. Since the bar values are presented in descending order this means that most of the Visitors who asked about 'Academic', also asked about 'General Welfare and Health'.

No of Students asked: `count({<[Visit Subject]=[Visit Subject]-[Associate Visit Subject], [Academic Year]= {"$(vMarketTest)"}>}distinct([Associate Student]))`

No of Visits: `count({<[Visit Subject]=[Visit Subject]-[Associate Visit Subject], [Academic Year]= {"$(vMarketTest)"}>} distinct([Visit Count]))`

Chart #10

The second chart on this sheet shows the number of Visitors, who asked about each of the Associated Visit Subjects in each month of the selected academic year.

```
Count ( (<[Visit Subject]=[Visit Subject]-[Associate Visit Subject], [Academic Year]=
{"$(vMarketTest)"}> }DISTINCT [Associate Student])
```

6.3 Lessons Learned

This section presents the lessons learned during the implementation of the demo dashboard in Qlikview.

1. When using the applymap() function, it should be checked whether the field in the target table and the field in the mapping table are in the same format.
2. A good practice is to distribute the code in separate tabs in the Qlikview Scrip Editor, since this makes it more readable and easier to maintain.
3. Another good practice is to save a new version of the application after each main data transformation, so that the last successful application is always available.
4. A useful guideline is also to test the dashboard's presentation on browsers during the development, because it is not always correctly displayed. This allows timely corrections in the dashboard's design, when needed.
5. The chart that seems to be the proper one at the beginning may not be suitable for the particular data set. For instance radio charts are often used to display loads throughout the year. As can be seen in Figure 24, however, the difference between the loads of the service units was so large that the data presented in a radio chart was not legible (the IELTS data cannot be seen). A better solution was to present this data in a line chart, which handles large amplitudes much better, that the radio chart (see the solution in Figure 25).

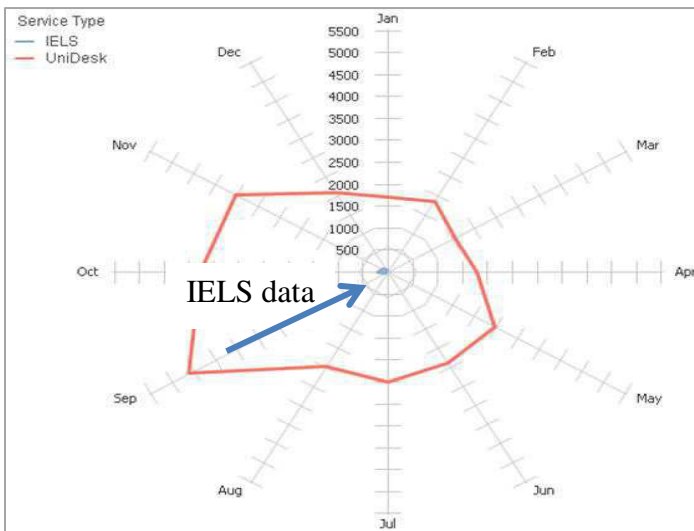


Figure 24 Service units' visit load presented in a Radio chart

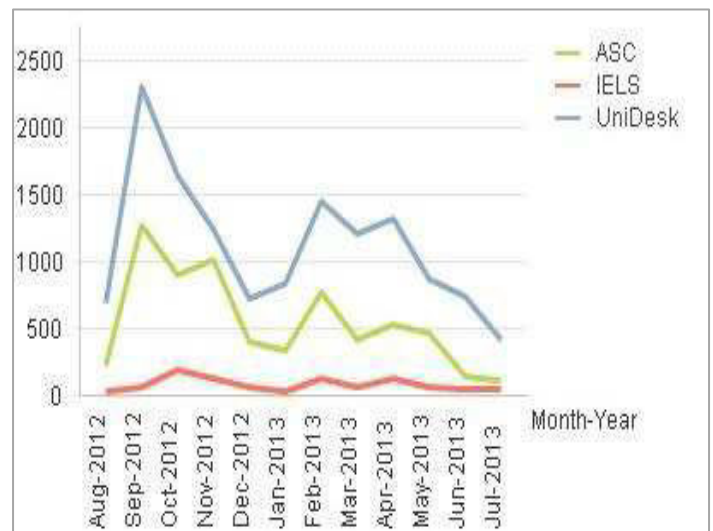


Figure 25 Service units' visit load presented in a Line Chart

- Setting static axes values was also found to be a good practice. The issue of having non static axes values is illustrated in Figure 26: Chart A shows the number of All Visitors. Chart B is the same as A, but only male students were selected. It can be seen that although the numbers are much smaller, the size of the bars doesn't change and thus leads to misinterpretation of the data. Chart C illustrates how by static axis values the change in the number of visitors is also represented by a change in the bars' size.

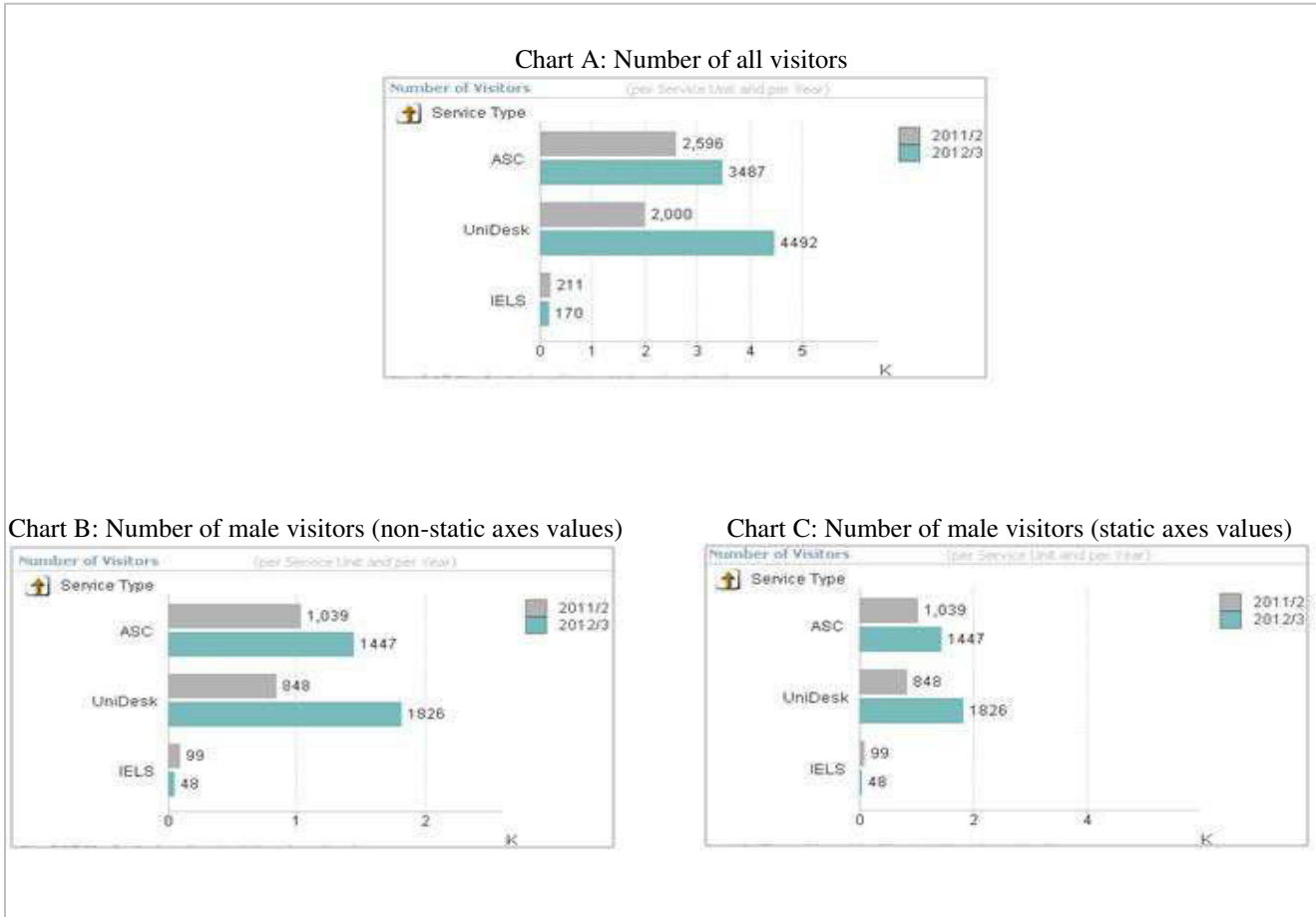


Figure 26 Illustrating the benefits of static axes values

- A useful guideline to be followed is setting informative error messages to improve the dashboard's usability and enhance users' experience. An example of the produced demo dashboard can be seen in Figure 27.

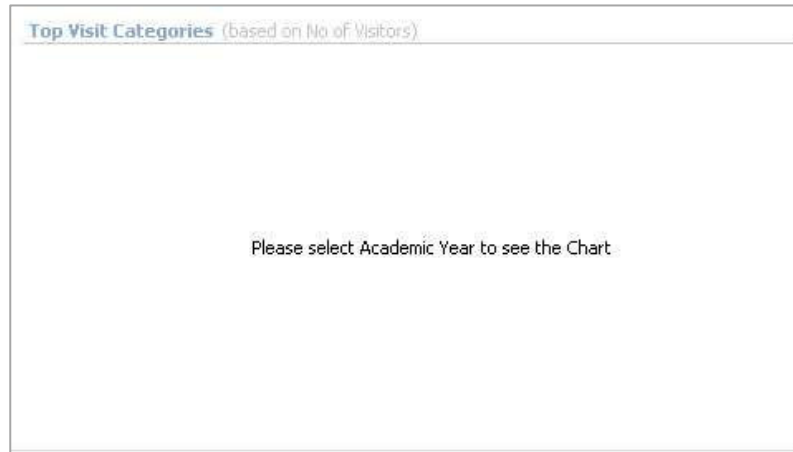


Figure 27 Example of a user-friendly error message from the demo dashboard

7 Demonstration and Testing

After the demo dashboard was built it was demonstrated to two members of the University's Student Services management. This chapter discusses the chosen method and the procedure for the demonstration session as well as the revealed findings.

7.1 Method

Purpose of the session

- To demonstrate the functionalities of the demo dashboard.
- To evaluate the ease of use of the demo dashboard.
- To assess whether the data is presented in a clear and useful manner.
- To investigate the potential and the challenges for the proposed Student Experience Dashboard.

Method

The method chosen for the demonstration/testing session is a semi-structured interview, which is considered to be an effective method for capturing user feedback in application development process. (Lazar et al. 2010) It is often used for collecting users' reactions and evaluation of prototypes and already completed systems. The participants in the demonstration/testing session were two members of the University Student Services management. Both of the participants are familiar and have

experience with Qlikview dashboards, but do not use such in their daily work. As advised by Lazar et al. (2010), the questions for the session were designed to be simple, clear and avoiding bias.

Procedure

First the participants were informed about the purpose of the session and were provided with a short description of the project. After that a short explanation of the data and the terms used in the dashboard was given. In the main part of the session the functionalities of each of the tabs were demonstrated and after that tested by the participants. Finally, a short semi-structured interview was conducted.

7.2 Limitations

- The given time for the session was 30 minutes.
- When the participants know that they are testing a product designed by the interviewer they tend to give more positive feedback. (Lazar et al. 2010)

7.3 Results

The dashboard was found to be easy to use “once you know what you are looking for”. Furthermore, both participants stated that the data is presented in a clear and useful manner. The possibility for year on year comparison was pointed out as the most useful capability of the dashboard. The reason stated was that it allows spotting trends and changes, which might have “implications about resourcing” or that might help the University to identify problematic areas. Being asked about the benefits of consolidating data from different units related to the student experience, the participants commented that the proposed Student Experience Dashboard would be “very good for the high level management at the University”, because instead of receiving reports once a year, they would have “access to the statistics on on-going basis”. Furthermore, it would allow them to spot university-wide trends. However, it was highlighted that the units might feel uncomfortable sharing their data. The fear that the data can be misinterpreted when it is not accompanied by contextual background was identified as a very significant obstacle to the adoption of the proposed dashboard. In this relation it was pointed out that the sharing of data should be voluntary.

8 Evaluation

8.1 Introduction

This section provides an evaluation of the University’s readiness for the implementation of the proposed Student Experience Dashboard. The assessment is based on the factors identified in the EDUCAUSE maturity framework, which is suggested to be used by HEIs to self-assess their maturity when starting a BI project (Kay & van Harmelen 2012). The factors included in the framework are

Investment; Data Storage, Quality, Governance, and Ownership; Culture; Processes and Reporting; and Infrastructure, Tools, and Expertise. (Bichsel 2012) Since it does not fall within the scope of this project the Investment factor is not considered. The evaluation provided is based on data collected through email communication and short semi-structured interviews, as well as on my experience in finding and obtaining access to data for the demo dashboard. The purpose of the investigation conducted was to collect information about current data storage, analysis, and reporting procedures in different units, related to the student experience, as well as their perception of BI and Analytics. The information was requested from managers of different support services and units offering extracurricular engagement opportunities. Due to summer holidays, seven responses were received in the project's time frame. Although the collected information does not cover all units related to the student experience, its analysis outlines important challenges for the Student Experience Dashboard implementation.

8.2 Data Storage and Quality

The initial investigation revealed that most of the University's units collect and analyse data about students' interactions with them. However, this is not always the case in the external units. For example no data is kept about which students are members of societies. Another data issue is that some of the data collected is hard to interpret or is insufficient for a proper analysis. This is the case with the University's Sport Centre, which keeps records about the students who purchase membership packages and about the student cards used for accessing the building. However, as explained by a member of Sports Centre staff, this data is not sufficient to provide an accurate and complete number of students training at the Sport Centre. The reason is that students can train at the Sport Centre without having a membership package and can access the building without their student card.

8.3 Data Governance and Ownership

My experience with obtaining access to the data sources used in the pilot dashboard revealed that roles, responsibilities and procedures related to data governance are not clearly communicated to the University's staff and it is not clear for them who should be contacted and what procedure should be followed in such cases. Furthermore, some of the contacted units refused to provide information about whether and what kind of data they store. The specified reason for this was the concern about violating data protection and privacy regulations. Another main issue relates to data ownership. Some units were unwilling to share their data for analytical purposes, because the benefit for them is not obvious. Furthermore, in some cases it was hard to find first the person who can allow access to the data (data owner), and then the person who can physically grant the access or provide the data.

8.4 Culture

All units expressed a positive attitude towards using business intelligence and making data-driven decisions. All units which do not store data about students' interactions or in which the quality of the collected data is not sufficient for proper analysis, are planning the implementation of new tools and

systems. They expressed their hope that the new systems will allow them to collect and store valuable data. However, the demonstration session revealed that University's units might be cautious about giving access to their data without the opportunity to provide contextual background for it, since this may lead to incorrect conclusions about their performance.

8.5 Processes and Reporting

One of the biggest challenges for the implementation of the Student Experience Dashboard is that key units related to the student experience are actually separate organizations, which are not managed by the University. Collecting data from external sources could be problematic and requires adequate communication practices and policies. Key third party organizations related to the student experience are the Chaplaincy, the Student Association and the Athletic Union. Currently these units do not provide data other than summary reports to the University. Another obstacle to the Student Experience dashboard is that there is no single department or person who manages the student experience as a whole - reporting and data analysis are performed departmental-wide rather than university-wide.

8.6 Infrastructure, Tools and Expertise

The University has the infrastructure, the tools, and the business intelligence experts required for the implementation of the Student Experience Dashboard. It administrates an Oracle data warehouse storing central administrative data such as student records, admissions, and finance data. The University also has a team of business intelligence experts having years of experience in building Qlikview dashboards for different departments at the University.

8.7 Conclusion

The experience and the information gained during the pilot project revealed that the University is not ready for the complete implementation of the proposed Student Experience Dashboard. The main reasons for this are lack of proper data collection procedures in the related third party units and a lack of partnership, communication and data sharing processes between different University departments and between the University and the involved external organizations. Another key obstacle is the units' considerations that the presentation of data without a contextual description may lead to inaccurate conclusions about their performance and trigger incorrect senior management decisions regarding human resources allocation and funding. The identified challenges are not surprising and are consistent with those discussed in the literature. The next chapter recommends measures that could help the University to overcome the identified challenges and to move one step further towards centralized monitoring and analysis of students' behaviour and needs.

9 Recommendations and Future Work

It is recommended that the University consider the following suggestions:

1. **Enhance data protection policies** – the rules for data sharing and transfer across the different units and between the University and the external organizations needs to be clearly specified by the enforcement of new policies. The new regulations should also address issues regarding confidentiality of departmental data as well as confidentiality of student data.
2. **Encourage data collection** - new data collection tools and practices need to be established in the third party units to provide the required data.
3. **Ensure data coherence** – to enable the successful integration of data from different sources standardized field names and unified level of data granularity need to be established.
4. **Communicate benefits** - to ensure the units' willingness to cooperate and share their data, the benefits of the project need to be clearly communicated.
5. **Establish partnerships** – to be able to capture as many aspects as possible of the student experience, the University could consider the establishment of partnerships and communication processes with the external organizations connected to the student experience and to encourage their involvement in analytics initiatives.

The next step could be the demonstration of the produced demo dashboard to the University senior management and key roles involved in the student experience. This would allow the further exploration of the perceived benefits of the adoption and implementation of the Student Experience Dashboard.

10 Conclusion

This study has examined and evaluated the University's readiness for adoption and implementation of the Student Experience Dashboard - a BI dashboard that consolidates data from multiple internal and external units related to the student experience at the University of St Andrews.

This was achieved by running a pilot project going through the process of developing such an interdepartmental dashboard. It involved the following steps: obtaining data from multiple units after preliminary investigation of the potential data sources; consolidating the obtained data and displaying it on a single dashboard; and investigating perceived benefits for the University Student Services management by demonstrating the dashboard.

For the evaluation of the University maturity level for development and adoption of the Student Experience Dashboard an assessment framework has been applied. The revealed main obstacles to the project were the University's decentralised structure, the departments' concerns of data misinterpretation, and the lack of properly collected data in the external units. Based on the experience from the pilot project and the findings, the following recommendations have been made: enhancing data protection policies; encouraging data collection; ensuring data coherence; communicating the benefits of the Student Experience Dashboard; and establishing partnership with the related external units.

The main limitation of this dissertation project were the time constrains. The given time of three months was additionally limited by a lack of University senior management support, a heavy and time-consuming process of ensuring compliance with data protection regulations, and staff absence due to summer holidays.

Despite these challenges, the demo dashboard has been successfully implemented and has received a positive feedback from members of the University Student Services management. The developed demo dashboard may be used for demonstration purposes and for further exploration of the perceived benefits of the proposed Student Experience Dashboard.

Bibliography

- Ali, M.M., 2013. Role of Data Mining in Education Sector. *International Journal of Computer Science and Mobile Computing*, 2(4), pp.374–383.
- Baiyun Chen, Sugar, A. & Bauer, S., 2012. Effective Faculty Development through Strategies for Engagement and Satisfaction. *EDUCAUSE*.
- Baritchi, A., 2003. Business Intelligence in the Digital Economy M. S. Raisinghani, ed., IGI Global. Available at: <http://www.igi-global.com/chapter/data-mining-knowledge-discovery/6064/> [Accessed June 30, 2013].
- Barneveld, A. Van, Arnold, K. & Campbell, J., 2012. Analytics in higher education: Establishing a common language. ELI paper, (January), pp.1–11. Available at: <http://sites.ewu.edu/elearningservices/files/2012/06/Analytics-in-Higher-Education-Establishing-a-Common-Language-ELI3026.pdf> [Accessed August 16, 2013].
- Bichsel, J., 2012. Analytics in Higher education: Benefits, Barriers, Progress and Recommendations. Louisville, CO: EDUCAUSE Center for Applied Research. Available at: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Analytics+in+Higher+Education+on+Benefits+,+Barriers+,+Progress+,#0> [Accessed July 7, 2013].
- Bose, R., 2006. Understanding management data systems for enterprise performance management. *Industrial Management & Data Systems*, 106(1), pp.43–59. Available at: <http://www.emeraldinsight.com/10.1108/02635570610640988> [Accessed July 27, 2013].
- Brookes, M., 2003. Evaluating the “Student Experience”: An Approach to Managing and Enhancing Quality in Higher Education. *The Journal of Hospitality Leisure Sport and Tourism*, 2(1), pp.17–26. Available at: <http://www.heacademy.ac.uk/assets/hlst/documents/johlste/vol2no1/0027.pdf> [Accessed July 6, 2013].
- Chaudhuri, S. & Dayal, U., 1997. An overview of data warehousing and OLAP technology. *ACM Sigmod record*. Available at: <http://dl.acm.org/citation.cfm?id=248616> [Accessed June 25, 2013].
- Chaudhuri, S., Dayal, U. & Narasayya, V., 2011. An overview of business intelligence technology. *Commun. ACM*, 54(8), pp.88–98. Available at: <http://doi.acm.org/10.1145/1978542.1978562>.
- Chee, T. et al., 2009. Business Intelligence Systems : State-of-the-art Review and Contemporary Applications. , pp.96–101.
- Codd, E.F., Codd, S.B. & Salley, C.T., 1993. Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate.

- Cranfield, D. & Taylor, J., 2008. Knowledge management and higher education: a UK case study. *Electronic Journal of Knowledge Management*, 6(2), pp.85–100. Available at: <http://www.ejkm.com/issue/download.html?idArticle=145> [Accessed August 17, 2013].
- Dehne, F. & Zaboli, H., 2012. Parallel Real-Time OLAP on Multi-core Processors. 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), pp.588–594. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6217470> [Accessed June 23, 2013].
- Eckerson, W.W., 2011. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business* 2nd ed., John Wiley & Sons, Inc.
- Few, S., 2006. *Common Pitfalls in Dashboard Design*. , (February).
- Gartner, 2013. *Magic Quadrant for Business Intelligence and Analytics Platforms*. Available at: <http://www.gartner.com/technology/reprints.do?id=1-1DYXHVT&ct=130206&st=sg>.
- Gibson, M. et al., 2004. Evaluating the Intangible Benefits of Business Intelligence: Review & Research Agenda. *Proceedings of the 2004 IFIP International Conference on Decision Support Systems (DSS2004): Decision Support in an Uncertain and Complex World*, pp.295–305.
- Hajmoosaei, A., Kashfi, M. & Kailasam, P., 2011. Comparison plan for data warehouse system architectures. *Data Mining and Intelligent Information Technology Applications (ICMiA), 2011 3rd International Conference on*, pp.290–293.
- Hayardisi, G., 2009. Data warehouse and web-based OLAP for hotspot distribution in Indonesia. *Data Mining and Optimization*, (October), pp.4–7. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5341921 [Accessed June 23, 2013].
- Inmon, W.H., 1991. *Building the Data Warehouse* John Wiley & Sons, Inc. John Wiley & Sons, Inc.
- Jisc, 2012. *Activity Data: Delivering Benefits from the Data Deluge*. , (January), pp.1–16.
- Kay, D. & van Harmelen, M., 2012. *CETIS Analytics Series Vol.1, No.2, Analytics for the Whole Institution, Balancing Strategy and Tactics*.
- Khan, A., 2012. *Business intelligence and data warehousing simplified: 500 questions, answers, and tips*. [Books24x7 version] Available from <http://common.books24x7.com.ezproxy.st-andrews.ac.uk/toc.aspx?bookid=43955>.
- King, J. et al., 2012. *Project fulCRM Case Study*. Jisc Projects. Available at: http://www.roehampton.ac.uk/uploadedFiles/Pages_Assets/PDFs_and_Word_Docs/JISC_Projects/fulCRM/Project_documents/Case-study-final-document.pdf.
- Korczak, J. & Dudycz, H., 2012. *Architecture and functions*. , pp.1003–1007.

- Kuhl, M. & Stoeber, O., 2006. Data Warehousing and Customer Relationship Management als Grundlagen des wertorientierten Kundenmanagements B. Günter & S. Helm, eds. , pp.531–548. Available at: <http://www.springerlink.com/index/10.1007/978-3-8349-9288-8> [Accessed June 12, 2013].
- Van der Lans, R.F., 2012. Data virtualization for business intelligence systems : revolutionizing data integration for data warehouses.
- Lazar, J., Feng, J.H. & Hochheister, H., 2010. Research Methods in Human-Computer Interaction. Wiley Publishing.
- Li, J. et al., 2010. The design and implementation of web-based OLAP drilling analysis system. 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, (Fskd), pp.2570–2573. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5569837>.
- Lith, W. van, 2009. How to: Market Basket Analysis. Quick-Qlear-Qool. Available at: <http://www.quickqlearqool.nl/?p=965>.
- Mohamadina, A. a. et al., 2012. Business Intelligence: Concepts, Issues and Current Systems. 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), pp.234–237. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6516358> [Accessed June 21, 2013].
- Mutton, J., 2011. Project Plan: Student Experience Traffic Lighting (SETL). Jisc Projects. Available at: http://www.jisc.ac.uk/media/documents/programmes/bce/DerbySETLprojectplan_v1.0_FINAL_June_11_Redacted.pdf.
- Nicolson, N., Junkins, J. & Evelson, B., 2008. Topic Overview : Business Intelligence Topic Overview : Business Intelligence.
- Olszak, C. & Ziemia, E., 2007. Approach to building and implementing business intelligence systems. Interdisciplinary Journal of Information, Knowledge, and Management, 2. Available at: <http://www.ijikm.org/Volume2/IJIKMv2p135-148Olszak184.pdf> [Accessed June 22, 2013].
- Power, D.J., 2007. A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web, <http://DSSResources.COM/history/dsshistory.html>, version 4.0, March 10, 2007.
- Qlikview, 2012a. Designer. QlikTech International AB, Sweden.
- Qlikview, 2012b. Developer. QlikTech International AB, Sweden.
- Rygielski, C., Wang, J.-C. & Yen, D.C., 2002. Data mining techniques for customer relationship management. Technology in Society, 24(4), pp.483–502. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0160791X02000386>.

- Santos, R.J. & Bernardino, J., 2009. Optimizing data warehouse loading procedures for enabling useful-time data warehousing. Proceedings of the 2009 International Database Engineering & Applications Symposium on - IDEAS '09, p.292. Available at: <http://portal.acm.org/citation.cfm?doid=1620432.1620464>.
- Stubbs, M., 2013. 5 top tips to enhance your students' experience. Available at: www.jisc.ac.uk/blog/5-top-tips-to-enhance-your-students-experience-11-mar-2013.
- Sturdy, G., 2012. Customer Relationship Management using Business Intelligence Cambridge Scholars Publishing. Cambridge Scholars Publishing.
- Tapp, A., Hicks, K. & Stone, M., 2004. Direct and database marketing and customer relationship management in recruiting students for higher education. International Journal of Nonprofit and Voluntary Sector Marketing, 9(4), pp.335–345. Available at: <http://doi.wiley.com/10.1002/nvsm.258>.
- The Open University, 2013. RISE - Recommendations Improve the Search Experience. The Open University. Available at: <http://www.open.ac.uk/blogs/RISE/>.
- University of St Andrews, 2008a. Strategic Plan: 2008-2018. Available at: [http://www.st-andrews.ac.uk/media/principals-office/planning/strategicplanning/universityoperationalstrategies/University_of_St_Andrews_Strategic_Plan_2008-2018\(Graphics\).pdf](http://www.st-andrews.ac.uk/media/principals-office/planning/strategicplanning/universityoperationalstrategies/University_of_St_Andrews_Strategic_Plan_2008-2018(Graphics).pdf).
- University of St Andrews, 2008b. The Student Experience Strategy. Available at: <http://www.st-andrews.ac.uk/studentexperience/about/what/strategy/>.
- Zhao, X. & Huang, Z., 2010. A quality evaluation approach for OLAP metadata of multidimensional OLAP data. 2010 2nd IEEE International Conference on Information Management and Engineering, pp.357–361. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5477583>.