REYKJAVIK UNIVERSITY DATA WAREHOUSE

Sæmundur Melstað
Master of Science
Computer Science
June 2014
School of Computer Science
Reykjavík University

M.Sc. PROJECT REPORT

ISSN 1670-8539

Reykjavik University Data Warehouse

by

Sæmundur Melstað

Project report submitted to the School of Computer Science
at Reykjavík University in partial fulfillment of
the requirements for the degree of
Master of Science in **Computer Science**

June 2014

Project Report Committee:

Björn Þór Jónsson, Supervisor
Associate professor, Reykjavik University

Jamie Deon Garrett
Sr. Research Scientist, Icelandic Institute for Intelligent Machines

Páll Melsted Ríkharðsson
Associate Professor, Reykjavik University

Heiðar Jón Hannesson
IT Director, Reykjavik University

Copyright
Sæmundur Melstað
June 2014

Reykjavik University Data Warehouse

Sæmundur Melstað

June 2014

Abstract

Access to reliable and accurate information is essential for management of educational institutions. Administrative staff must have access to current and historical information to fulfill their administrative duties.

A data warehouse is a collection of components and tools that retrieve data from disparate systems, transform the data and load into a database designed for analysis and reporting. A data warehouse supports coordinated reporting when companies and/or institutions upgrade or renew their information systems.

Reykjavik University Data Warehouse is designed to allow university administrators to adequately and efficiently deliver their reports. We discuss the design process, architectural design and implementation of the data warehouse solution. Version, one of the data warehouse is presently in operation and will contribute to the reliable reporting of data for the university.

Vöruhús gagna fyrir Háskólann í Reykjavík

Sæmundur Melstað

Júní 2014

Útdráttur

Þörfin fyrir áreiðanlegar og réttar upplýsingar er nauðsynlegt fyrir alla þá sem eru að stjórna menntastofnunum. Stjórnendur þurfa að hafa aðgang að nýjum og sögulegum gögnum til að geta sinnt sýnum stjórnunarlegu skyldum.

Vöruhús gagna er safn aðferða og verkfæra til að sækja gögn í mismunandi gagnasöfn, samræma og hlaða inn í gagnagrunn sem er hannaður fyrir skýrslugerð og gagnagreiningar. Vöruhús gagna styður samræmda skýrslugerð þegar fyrirtæki og stofnanir uppfæra og eða endurnýja upplýsingakerfi sín.

Vöruhús gagna fyrir Háskólann í Reykavík er hannað til þess að stjórnendur skólans geti með viðunandi hætti skilað áreiðanlegum upplýsingum. Við lýsum hönnunarferlinu, hönnun kerfisins og hvernig við útfærðum lausnina.

Fyrsta útgáfa af gagnavöruhúsinu er komin í rekstur og mun styðja samræmda skýrslugerð fyrir háskólann.

To my family who has showed me endless patience and support in this work.

Acknowledgements

I would like to thank Björn Þór Jónsson and Heiðar Jón Hannesson for invaluable input into the project this report represents. I would also like to thank G. Birna Guðmundsdóttir for her cooperation in this mutual project of ours.

The input and feedback of Rósa Gunnarsdóttir and her fellow colleagues in the university administration have also made this project and report doable.

Contents

Chapter 1

Introduction

Educational institutions need a system to keep track of their vital statistics, such as applicants, registered students, what courses they take and what grades they receive from exams and assignments. Educational institutions have a responsibility to store data for a long time because former students have the right to ask for the information about their academic achievement. Management also needs reliable and accurate reports for effective decision making.

Reykjavik University has used the MySchool system for many years for most of its operation. Students use the system in many aspects of their studies, e.g., to see what courses they are registered for, what assignments are in each course, deliver assignments solutions, view their grades, and many other actions that support the students' communication with university faculty staff.

The MySchool system offers multiple reports for the management to retrieve information about students. Unfortunately, however, some reports in the system do not deliver consistent results. Some of this can be attributed to incorrect data input within the MySchool system, some can be attributed to a different logic in code for various reports and some to processes where data is multiplied within the MySchool system.

The administration team of the university recognized that they needed to improve reporting and that further development of the MySchool system would not resolve the issues mentioned above. The mission of this project was therefore to build a first version of a data warehouse for the university, the Reykjavik University Data Warehouse (RUDW), that would store reliable information for reporting and support decision making in administration and operation of the university.

A data warehouse is a collection of components and tools that retrieve data from disparate systems, transform the data and load into a database designed for analysis and reporting. A data warehouse stores current and historical data from a variety of systems. By design, the data in the data warehouse can still be retrievable though its source systems are disconnected from the process and, therefore, a data warehouse is an excellent tool to bridge the gap when older systems are upgraded, or new systems installed.

The project delivery was in two parts; the first part was to design and implement a data warehouse and the second part was to incorporate data quality processes into the data warehouse loading process. The writer took the responsibility for the first part and G. Birna Guðmundsdóttir for the second part. This report covers the design and implementation of the data warehouse.

We met with university staff to gather information about the reporting requirements and to decide what part of the data to focus on. We had several meetings with select part of the university staff to find out what reports they were using in the MySchool system, and that gave us a starting point for the information the data warehouse should deliver. There is little or no documentation available for the MySchool system data structure, so significant work was required in analyzing the data structure of the MySchool system to find out which tables were relevant and which were not. After we had gathered all the necessary requirements, we could start building a data warehouse.

We were aware from the start that there would be many data quality issues. Guðmundsdóttir took the responsibility for flagging all these errors to the university staff so that they could be corrected. The data quality process is run each time the data warehouse is updated, and error output is sent to the owners of the data. The university staff has the responsibility to correct the incorrect data in the MySchool system, so the next time the data warehouse is updated, the correct data will be loaded into the data warehouse.

My part of the project was to create an architecture for the data warehouse, building the data warehouse, the data loading process, and the database views for applications and users to access the data in the data warehouse. My part was also to create an analytical cube, which the university staff would be able to use for reporting and testing the quality of the data that had been loaded into the data warehouse. The user interface we offered to access the analytical cube was Microsoft Excel, but the pivot functions in Microsoft Excel are widely used to access such analytical cubes. Data can also be retrieved with SQL query tools if users so desire.

Version, one of RUDW has presently been put into production within the university IT environment and is already used for reporting by the administration. However, further development and expansion of the data warehouse is required as part of the future work.

The remainder of this report is organized as follows. In Chapter 2 we present a brief overview of data warehousing background. In Chapter 3 we provide an overview of the MySchool system, focusing on those parts that are relevant for the project. In Chapter 4 we describe the implementation of the RUDW and in Chapter 5 we present a preliminary evaluation of the RUDW, and finally in Chapter 6 we conclude and discuss possible extensions for RUDW.

Chapter 2

Data Warehouse

A data warehouse is a collection of components and tools that retrieve data from disparate systems and load into a database designed for analysis and reporting. A data warehouse stores historical data which gives the user the ability to do advanced reporting and statistical comparisons.

Data comes from different sources, e.g., Learning Management System, Student Management System and accounting systems. It could also come from both old and new systems, and when systems are being upgraded. Instead of trying to import data from the old system to the new system when systems are being renewed, with considerable cost, data is loaded from both systems into the data warehouse. When the old system is turned off, the data resides in the data warehouse and can be used for further reporting.

In this chapter, we are discussing some of the major building blocks of the data warehouse. This report is only a shallow description of this subject; so we refer the interested reader to (Kimball & Ross, 2013).

In Section 2.1 we take a brief look at data warehouse history. In Section 2.2 we discuss the business value of a data warehouse is for the business owners. In Section 2.3 we discuss different data models and the major building blocks in a data warehouse. In Section 2.4 we discuss different operations required to implement a data warehouse. In Section 2.5 we discuss how we can use the data warehouse for reporting, and we summarize in Section 2.6.

2.1 Data Warehouse History

In 1988, IBM researchers Barry Devlim and Paul Murphy (Devlim & Murphy, 1988) coined the term information warehouse and subsequently IT companies began building experimental data warehouses. In 1991, W.H. Inmon made data warehouses practical when he published a how-to guide, *Building the Data Warehouse* (Inmon, 1992). Ralph Kimball published, in 1996, the *The Data Warehouse Toolkit* (Kimball, 1996), which had many practical examples. These two, Inmon and Kimball, are considered the fathers of modern data warehouse concepts. Inmon is known for his top-down centralized view of warehousing but Kimball is known for his bottom-up star-schema approach (Williams, 2014). Our implementation in this project is based on Kimball's approach.

2.2 Business Value

The decision to build a data warehouse should be made from a business perspective but not from a technical perspective. If the data warehouse is only built from a technical perspective, it is highly likely that the business will not utilize the system because it was not built to the needs of the business. Therefore, must the team that builds the data warehouse meets with the business owners and addresses the definitions and scope of the data warehouse. Business requirements determine what data must be available in the data warehouse, how it is organized, how often it is updated, and how the data is retrieved.

2.3 Data Model

The relational model (Codd, 1970) has been widely used in database design for On Line Transaction Systems (OLTP). Relational models are collections of entities and relationships between them. These entities are designed to eliminate redundancy of data and to make transaction processing simple and fast. Modern business systems typically have thousands of entities that are mapped into database tables. To use that model directly for analytical work or reporting, however, is ineffective as end users can not navigate or understand the model quickly (Kimball, 1998). Figure 2.1 shows an example of a relational model that is quite hard for the novice user to understand and utilize for reporting.

The dimensional model is a design technique that presents the data in a standard framework for data warehousing. A dimensional model is structured of one fact table, and a set

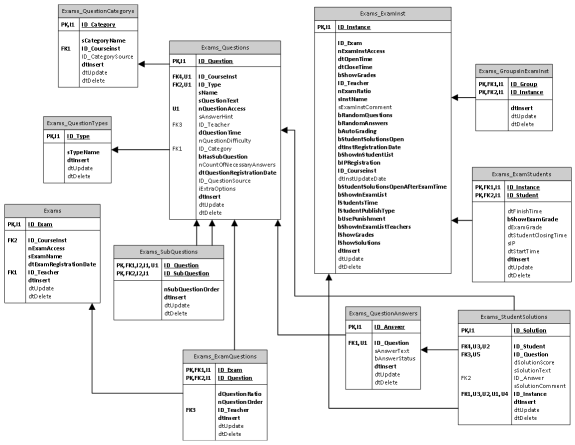


Figure 2.1: Example of relational model

of dimensions tables. Fact tables are the numerical part of the model, and the dimensions tables are the descriptive part. This characteristic star-like structure is often called Star schema or Star-Join schema. A data warehouse will consist of many such models where fact tables share dimensions and users can join fact tables through these dimensions. Figure 2.2 shows an example of a Star schema with one fact table and seven dimensions. This design is much more understandable for the end user for reporting usage.

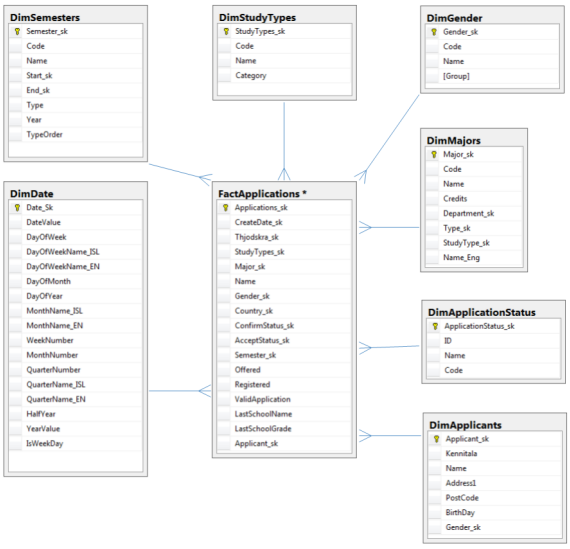
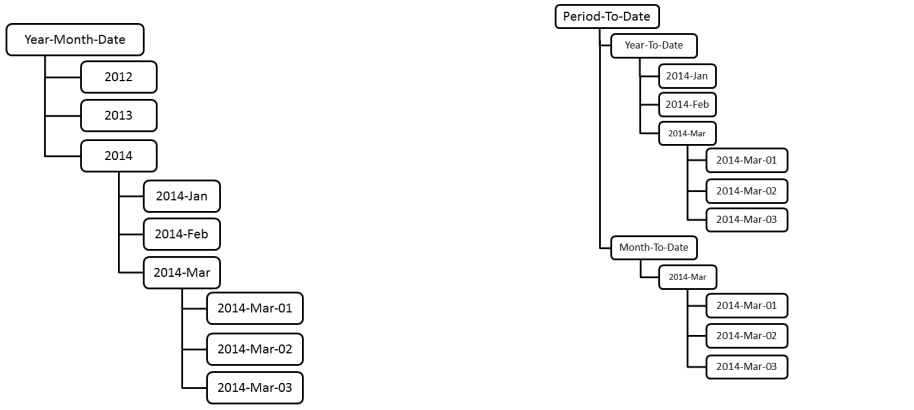


Figure 2.2: Example of dimensional model

2.3.1 Dimensions

A *dimension* is a collection of a text-like attributes that are highly correlated with each other. In an educational data warehouse, e.g., there are student dimension, subject dimension, semester dimension, course dimension, department dimension and time dimension.



(a) Year, Month and Date (b) Period-to-date

Figure 2.3: Example of two different hierarchies

The student dimension, e.g., could have attributes such as ID, name, address, country and birthday.

Attributes are most often text-like fields (such as name of a student), dates, enumerated fields (such as status of student's registration). Attributes are not quantitative, such as number of courses or number of graduate students. Those can be retrieved later from the data warehouse by aggregating facts (see below).

Attributes of a dimension will typically change over time, e.g., the description of a course changes over the years. The administration of the university makes the requirement that these changes are stored so that reports, e.g., diplomas, can be printed out with the right description at the time when the student attended the university. *Slowly Changing Dimension* (SCD) defines this situation because these changes can happen over a long time. Attributes in the same dimension can have different SCD-type; Type 0 is where the value of the attribute is constant and will not change over time, Type 1 is where the value of the attribute is overwritten each time the source changes, and Type 2 is where the history of values is kept, and a new record is stored in the dimension for the new value. Other SCD types have been defined by the industry, but they will not be discussed in this report.

Some dimensions have so many members that it is unrealistic or unpractical to browse them as one long list. A good example is the time dimension. Hierarchies are a well known structure to organize such lists and make it easier to browse them. Figure 2.3(a) shows a hierarchy for year, month and date in a time dimension and figure 2.3(b) shows a hierarchy for what is called period-to-date. Period-to-date can have members like Year-To-Date, Month-To-Date and Last-Year-To-Date.

2.3.2 Facts

A fact is data that is not known in advance and most often is numerical. If we take a look at a typical transactional system, such as student management system then the student grades and course credits will become facts in the data warehouse. The student results from an exam will become a record in a fact table with references to dimension tables.

When retrieving data from the data warehouse there is often a need to aggregate data by using counts, summaries, minimum values, maximum values, and also group data together. To achieve that outcome, fact tables are joined together through dimensions, records are grouped together, and the appropriate aggregate is then applied.

2.3.3 Database

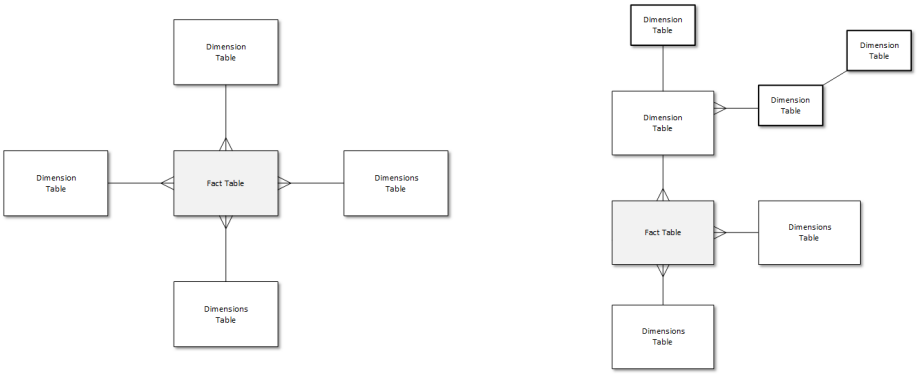
As discussed above the Star schema is a dimensional model composed of a fact table and a set of smaller tables called dimensions tables. A data warehouse will consist of many such models where fact tables share dimensions.

Database designers of a data warehouse are often tempted to save space in the database by breaking up the dimensions into smaller tables. By doing that the data structure is no longer a Star Schema but a so called Snowflake schema. Snowflake Schema is where one or more attributes of a dimension are moved into a separate dimension and linked to the original dimension.

An example of this could be in a customer dimension where demographic information in low cardinality are put into a separate dimension. The demographic dimension has much fewer members than the customer dimension and is loaded into the data warehouse at different times than the customer dimension.

Figure 2.4(a) represents a Star schema model for one fact table and four dimension tables and figure 2.4(b) shows a Snowflake schema where one of the dimension has been divided into sub-dimensions.

It is not recommended in data warehouse design, however as this design often complicates the usage of the data warehouse for the end users.



(a) Star schema

(b) Snowflake schema

Figure 2.4: Examples of Star schema and Snowflake schema

2.4 Data Warehouse Operation

To build a data warehouse, dimensions and fact tables must be created. In a well designed data warehouse where many fact tables reside, they often share dimensions. Few and well formed dimensions make it easier for the end user to retrieve information from the data warehouse. The data warehouse would not be useful if no data were stored in it; so data must be retrieved, cleaned and transformed from other databases or systems, and loaded into the data warehouse.

2.4.1 ETL

The process of building a data warehouse is called Extract, Transform and Load (ETL). The extract phase connects to disparate system and retrieve data from database tables, from single files or even web based systems. The transform phase aligns similar data so it can be loaded into the data warehouse; e.g., gender could be presented with M or F in one system and zero or one in another system. Lastly the load phase stores the cleaned data in the data warehouse.

2.4.2 Data Cleansing

Quality of data is one of the most important factors in a data warehouse. If the quality of the data retrieved from the data warehouse is not trusted, the data warehouse is built in vain. Before the data is loaded into the data warehouse, it is run through several cleansing and mapping rules to find and correct problems, e.g., find missing values, incorrect values or multiple values for the same attribute and map them to one confirmed value. Data

stewards are made aware of incorrect data so they can correct it. Next time the data warehouse is updated, correct data will be loaded. Figure 2.5 shows this process.

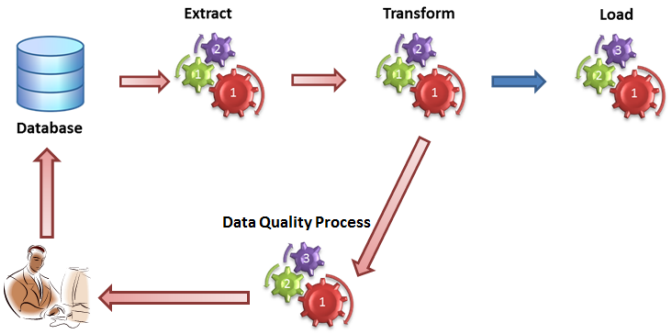


Figure 2.5: Example of data quality process

2.4.3 Meta Data

Meta data is data about data; "*Meta data is all the information that defines and describes the content, structures and operations of the DW system*". (Mundy, Thornthwaite, & Kimball, 2008, p. 524). Meta data describes, e.g.; the data type of each attribute in a dimension, and what dimensions and facts exist in the data warehouse. The metadata repository, figure 2.6, is often stored in a separate database and used by third party software tools to retrieve information about the structure of the data warehouse for data querying purposes and other similar tasks.

2.5 Using the Data Warehouse

When dimensions have been created and fact tables loaded in the data warehouse, all kinds of reporting and analytical work can be done. Fact tables can be joined together through dimensions, summarized, grouped and ordered in many different ways.

Analytical cubes or OLAP cubes can be built from data in the data warehouse, which gives users the possibility to compare different aspects of the data together and use that to either see historical changes or to predict future trends. OLAP is an acronym for OnLine Analytical Processing which is a method for analyzing business data. A cube is a multidimensional dataset that can be viewed in many ways and the major operations that are used on a cube are; slice, dice, drill down, drill up and pivot.

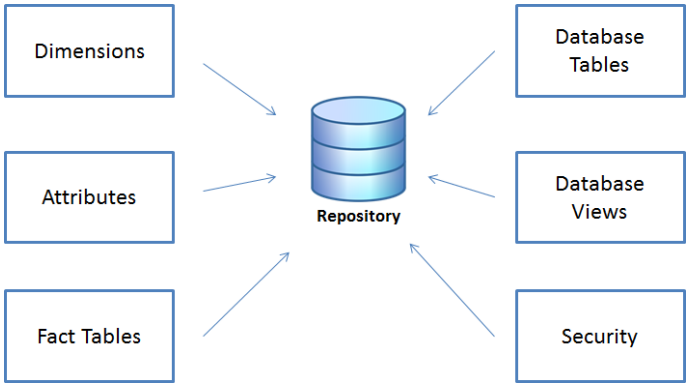


Figure 2.6: Example of information stored in a Metadata Repository

The data warehouse can track historical changes, so users can query the data warehouse for information back in time and retrieve information as they were when the data was loaded into the data warehouse.

Specific parts of the data warehouse can be made accessible to different groups of users or certain applications by defining database views on top of the dimensions and fact tables in the data warehouse.

2.6 Summary

We started by looking briefly at the history and business value of data warehouse design. We then described some of the major data warehouse building blocks. We discussed the processes related to loading data into the data warehouse, and finally we looked at how we can use the data warehouse for reporting and analysis.

Chapter 3

MySchool

Reykjavik University has used the MySchool system for many years for most of its operation. MySchool is a Student Management System, Learning Management System and many other systems, bundled into one huge integrated system. It started small and has been growing in functionality over the years. It is a web based system running on Microsoft Windows Server, based on classic ASP and SQL Server 2005. Figure 3.1 shows a screen shot of a student view from the system.

In this chapter, we present a quick overview of the MySchool system and how the administration of the university has been dealing with reporting issues in the MySchool system. In Section 3.1, we look at MySchools database design. In Section 3.2, we discuss what operations are in the MySchool system. In Section 3.3, we discuss reporting in MySchool. In Section 3.4, we discuss data quality in MySchool, and we summarize in Section 3.5.

Figure 3.1: Example of MySchool interface

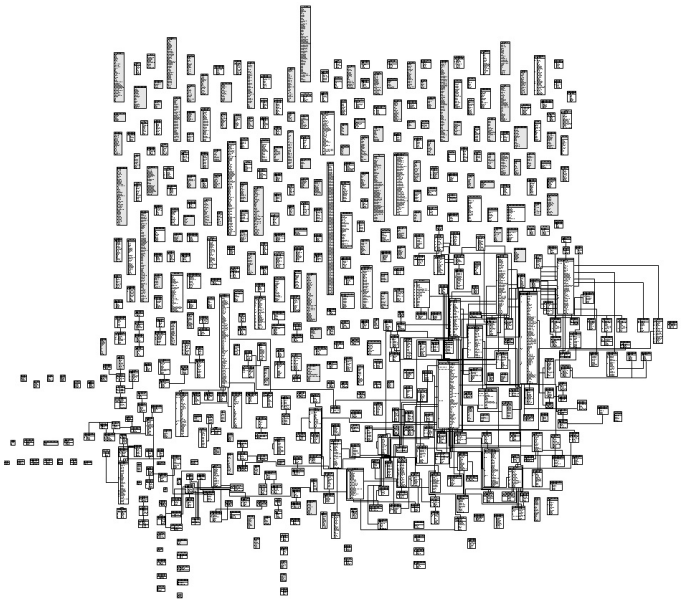


Figure 3.2: Overview of MySchool Database Tables

3.1 Database

Figure 3.2, shows an overview of all the tables in the MySchool system database. It is a typical OLTP design with many small tables and few large one. These tables are very loosely coupled (very few foreign keys) and therefore it is difficult for the end user to understand the data model or use it for the purpose of retrieving data from the system. There is little or no documentation available on the data structure of the system.

Because of this loosely coupled structure, referential integrity is not maintained in the MySchool system database. Data can be inserted with references that do not exist in the database. In the end, information retrieved from the system becomes unreliable.

Figure 3.3 shows a histogram of a number of database tables in the MySchool system by number of records in each table. Most of the tables (334 tables) contain less than one thousand records and only eight of them have one million records or more. None of these big tables is relevant to this project, however, the largest table we encountered and is relevant for this project has around five hundred thousand records.

3.2 Operation

Students use the system for much of their day-to-day activity, e.g., to see what courses they are registered for, what assignments are in each course, to deliver assignments results

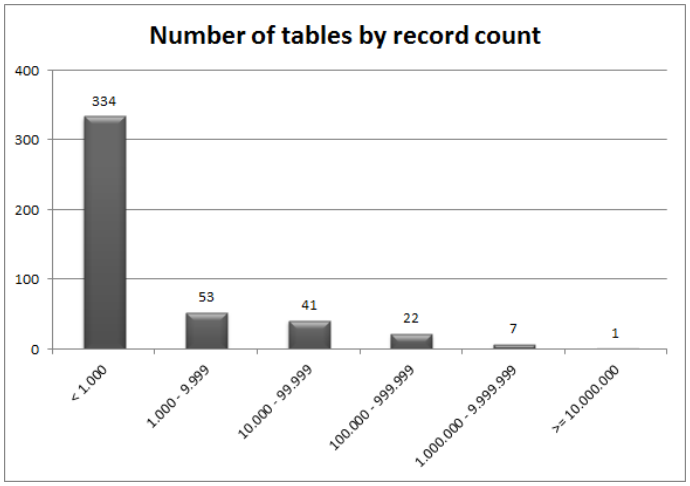


Figure 3.3: Histogram which group tables after number of records

into the system, and to see their grades. Many other functions exist in the system to support the students communication with the university faculty and staff.

The administration creates new courses and teacher's assign projects to students. The system also offers multiple reports for the administration to retrieve information about the students' academic achievements. All in one system, which is a good idea, but in the long run the system has become so huge that maintaining it has become its greatest obstacle. Issues with data quality are frequent and have been dealt with by patching the system, either in the data input part or the reporting part of the system.

3.3 Reporting

Reports are mostly web-page fronts for choosing run time parameters and SQL code which generates HTML- or PDF output. Because of the lack of data integrity in the database, there is a lot of programming logic in these reports to generate the desired output. The data in the MySchool system database would not support the correct information without this programming logic.

Unfortunately, however, there are many reports which give similar outputs, e.g., how many students attended last semester. The total number is not the same from different reports, due to varying program logic, so the management have started to mistrust the reported outcome and have wasted many hours of additional work to verify these numbers.

Some members of the university staff have exported data from the MySchool system into Excel workbooks and done some further data manipulation there. When working on

this data in Excel, some data discrepancies have been revealed and instead of correcting the data in the MySchool system, changes have only been made in the Excel workbook. There is no way back in this situation, as users have now two sources of data instead of one.

In extreme cases, university staff have moved to a manual process of retrieving individual student records from the MySchool system to find the correct information and do the reporting manually. Many hours of unnecessary work have been wasted to get vital information.

3.4 Data Quality

As discussed above data quality has been a major problem within the MySchool system. There has been little or no quality checking on input data, so users of the system have been able to use incorrect data types in input fields (e.g., inserting number into date fields) which has in extreme cases made the MySchool system become inaccessible. So missing input checking has let to many data quality issues in the MySchool system. Some of this can be attributed to incorrect data input within the MySchool system, some can be attributed to a different logic in the code for different reports and some to processes where data is multiplied within the MySchool system.

My partner in this project, G. Birna Guðmundsdóttir has in parallel with my project addressed data quality issues in MySchool and how it would be possible to challenge them in the data warehouse. Her focus was on creating an iterative process where data from the data warehouse is run through quality processes, and incorrect data is reported back to the owners of the data. The owners then correct the data in the MySchool system so that the next time the data warehouse is loaded, the incorrect data is replaced with corrected data.

3.5 Summary

We described the structure of the MySchool system and how data integrity is missing from the system. We discussed data quality and how that has become a major issue in the MySchool system and finally we discussed how the university staff has dealt with getting reliable reports from the system.

Chapter 4

Reykjavik University Data Warehouse

In this chapter, we present the design (section 4.1) and implementation (section 4.2) of the RUDW. This chapter gives an overview of the process, focusing on a high level view of the data warehouse while a more detailed description will be given in (Melstað, in prep.).

4.1 The Design Process

4.1.1 Business Requirements

The MySchool system offers multiple reports for the management to retrieve information about students. However, some reports in the system do not deliver consistent results. The administration team of the university recognized that they needed to improve reporting and that further development of the MySchool system would not resolve those issues.

The mission of this project was to build the first version of a data warehouse for the university, the Reykjavik University Data Warehouse (RUDW), which would contribute to the reporting part of MySchool. The RUDW would store reliable information for reporting and support decision making in administration and operation of the university. Therefore in this first version of RUDW there is only data from the MySchool system. Future versions will include data from other systems.

We started our work by meeting with the director of education and his staff to gather information about the reporting requirements in order to decide what part of the data to focus on. Our starting point was the reports that the university is required to publish, and the department of education prepares. Most of the numerical data required in these reports are retrieved with reports from the MySchool system.

There is little or no documentation available for the MySchool system data architecture, so significant work was required in analyzing the data structure of the MySchool system to find out which tables were relevant and which were not. After we had gathered all the requirements, we could start building a data warehouse.

4.1.2 MySchool

Our biggest obstacle was to understand the data structure of the MySchool system. Documentation of the system was near to none, so we had to run a trace on the MySchool system database to find out what tables were used for each process in the MySchool system. From these traces, we discovered what database tables would be used for data extraction and how they were related to each other.

The main database tables in the MySchool system that were relevant for this project were related to student applications, student registration, student course registration and student grades. Other related tables were semesters, majors, courses and students.

With this information, we could define the necessary dimensions and fact tables.

4.1.3 Dimension and Fact Tables

The key dimensions in RUDW are:

- Date
- Applicants
- Courses
- Departments
- Majors
- Semesters
- Study Types

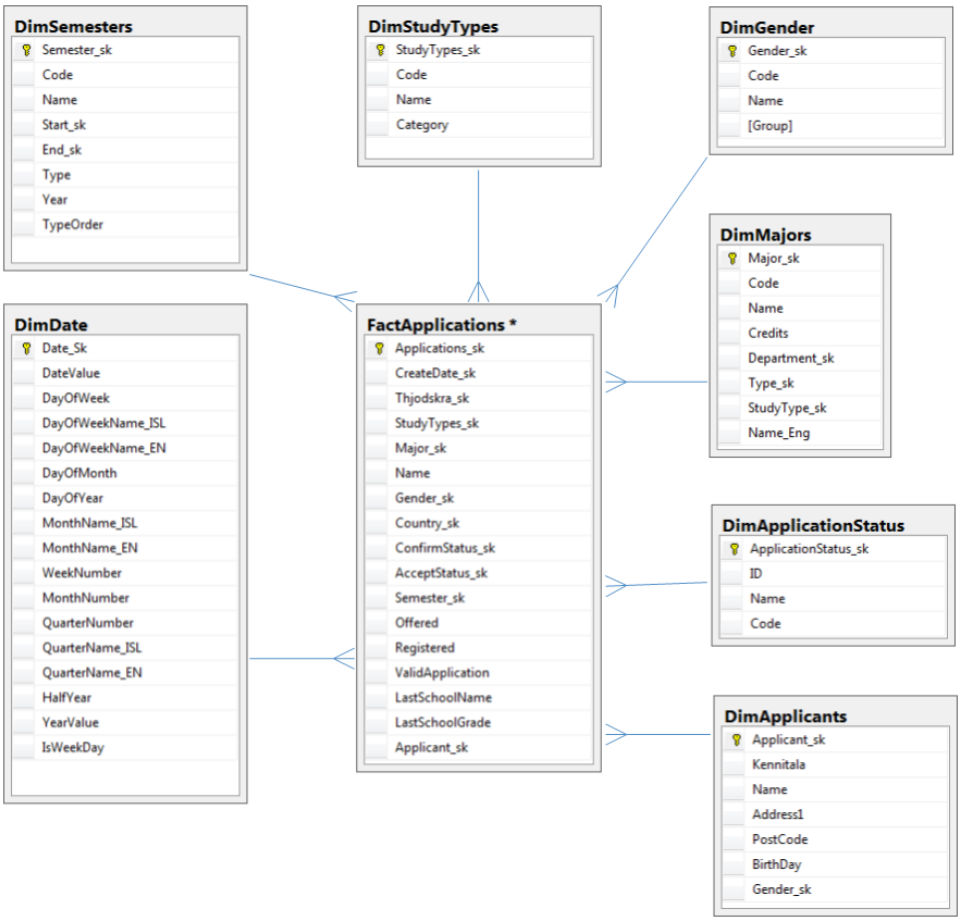


Figure 4.1: Applications fact table and related dimensions

- Students

The major fact tables are:

- Applications
- Student Registrations
- Student Grades

The detail of these tables and views are specified in (Melstað, in prep.). To give an example, however, figure 4.1 shows the details of the Applications fact table and the dimensions that are related to that fact table.

4.1.4 Software and Tools

When we started this project, we had to decide in which software environment the data warehouse would be built. We considered major database vendors such as Oracle and

Microsoft, and also some other database systems which are designed specifically for data warehouse systems. Some of the column store databases, for example, are much more suitable candidates for data warehouse systems than the mainstream transactional databases.

The MySchool system database is based on Microsoft SQL-Server 2005, however, the university has a major investment in Microsoft technology, so it was decided to use the latest version of Microsoft SQL-Server for the project, along with the tools that are included in that suite.

The software used for building the data warehouse therefore was:

- Microsoft Windows Server 2008 R2
- Microsoft SQL SERVER 2012
- Microsoft SQL Server Data Tools (Visual Studio 2010 Professional)
- Microsoft SQL Server 2012 Management Studio (SSMS)
- Microsoft SQL Server 2012 Integration Services (SSIS)

Microsoft SQL Server 2012 Analysis Service (SSAS) was used then to build an analytical cube.

4.2 The Implementation

The RUDW has three major storage components. The first component is the Staging area, where data is extracted from the MySchool system and stored in the RUDW Staging database. The second component is the Data Warehouse area where data from the Staging area is transformed, cleaned and loaded into the RUDW MySchool database. The third component is the Execution area with the RUDW Execution database, where data about the health of the data warehouse is stored. All jobs that load data into the data warehouse write execution logs into the RUDW Execution database. Figure 4.2 shows the overview of this structure and table 4.1 shows the number of database tables and views in each of the database.

Database	**# of Tables**	**# of Views**
RUDW Staging	41	23
RUDW MySchool	29	48
RUDW Execution	12	2

Table 4.1: Number of database tables and views

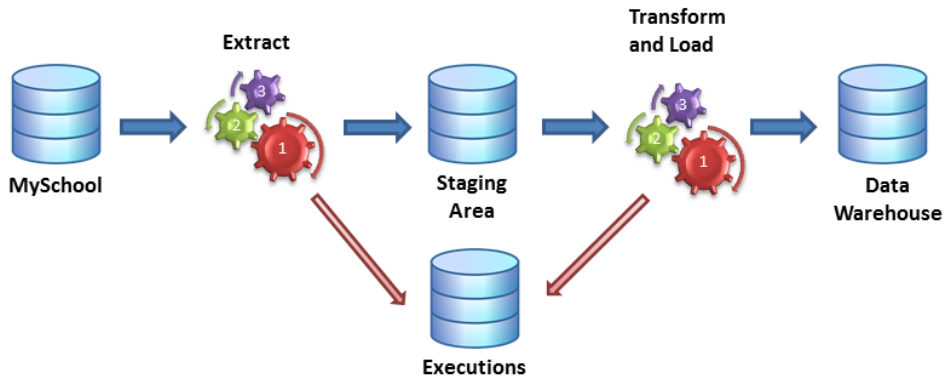


Figure 4.2: RUDW Architecture

The project was then organized into several sub-projects or packages; a package is a data structure that is used by the software tools to keep track of a variety of elements that belong together. There are three Database packages that define the data warehouse databases:

- RUDW Staging database
- RUDW MySchool database
- RUDW Execution database

Then are two Integration Service packages to load data into the databases:

- RUDW LoadStaging
- RUDW LoadDimensionsAndFacts

And finally we have one Analysis Service package to create the analytical cube

- RUDW CubeMySchool

4.2.1 Loading data

An update of the RUDW happens through the two major jobs defined by the LoadStaging and LoadDimAndFacts packages. The former job loads data into the staging database and the latter one loads data into the data warehouse database.

Each data table which is extracted from the MySchool system is loaded into the staging database as is. That is, data fields are copied exactly as they are defined in the MySchool system database and loaded into the staging database.

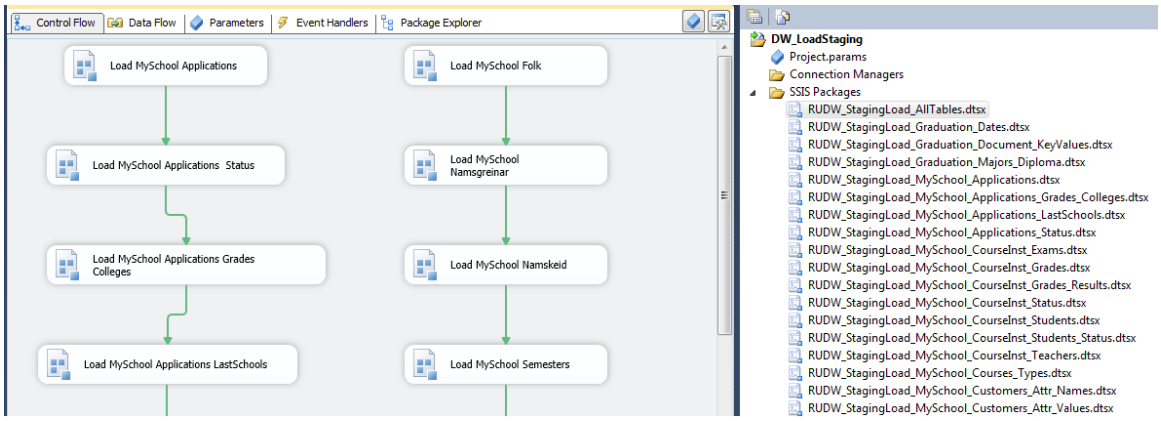


Figure 4.3: RUDW LoadStaging Job

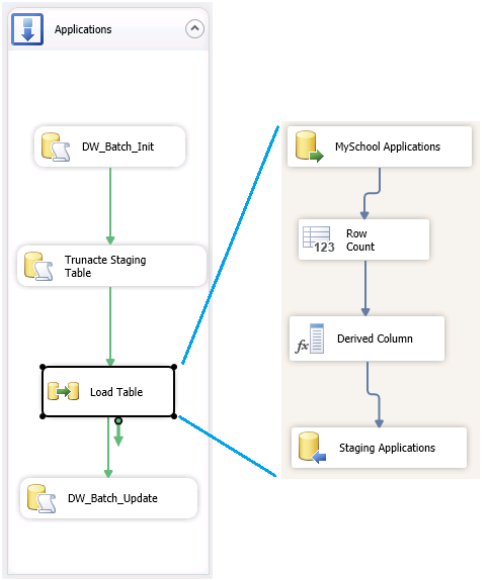


Figure 4.4: Example of Staging flow

Database views are defined on top of the tables in the staging database and used as an input source for loading the data warehouse database. Some of the database views are one to one mapping, but others are database joins where data tables from the staging database are joined together to create input sources for the job that updates either dimension or fact tables in the data warehouse.

The RUDW LoadStaging package, (Figure 4.3) retrieves data from the MySchool system and loads into the RUDW Staging database. It is a collection of many child packages, where each child package loads a single database table. Figure 4.4 shows a detailed data flow for loading the Application's table from the MySchool system into the staging database.

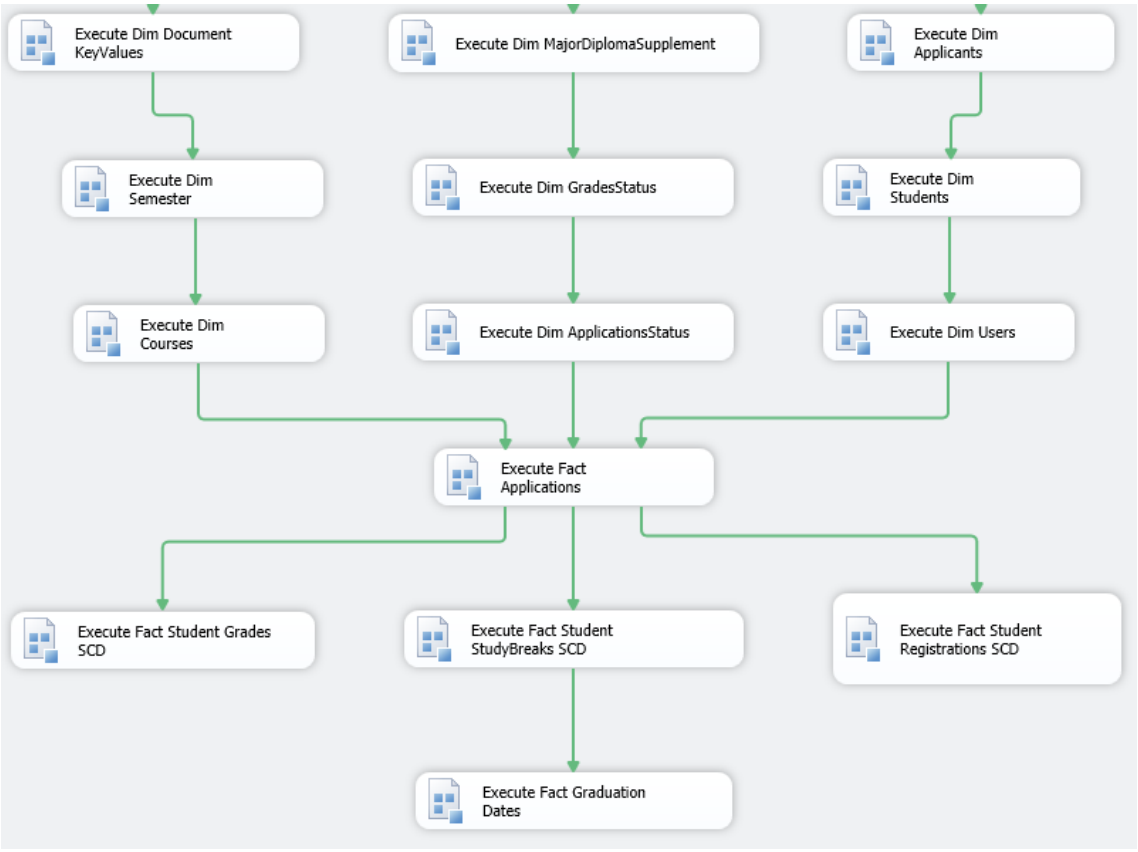


Figure 4.5: RUDW Load Dimensions and Fact Job

The RUDW LoadDimensionsAndFacts package (Figure 4.5) retrieves data from the RUDW Staging database and updates dimensions and fact tables in the RUDW MySchool database.

Packages are loaded into Integration Services Catalog in the SQL Server and run on a daily basis from a job in SQL Server Agent.

4.2.2 Database Views

Access to data in the data warehouse is always given through database views. Users or applications are newer allowed direct access to database tables. This setup gives the administrator of the data warehouse the ability to control access to the data warehouse and present the data in different shape and format for different purposes.

Figure 4.6 shows how database views are used both in the Staging Area and in the Data Warehouse area.

We created database views for an application that required access to specific data. In these views, dimensions and fact tables are joined together, filtered, grouped, ordered and aggregated to output the data in the right format for the application.

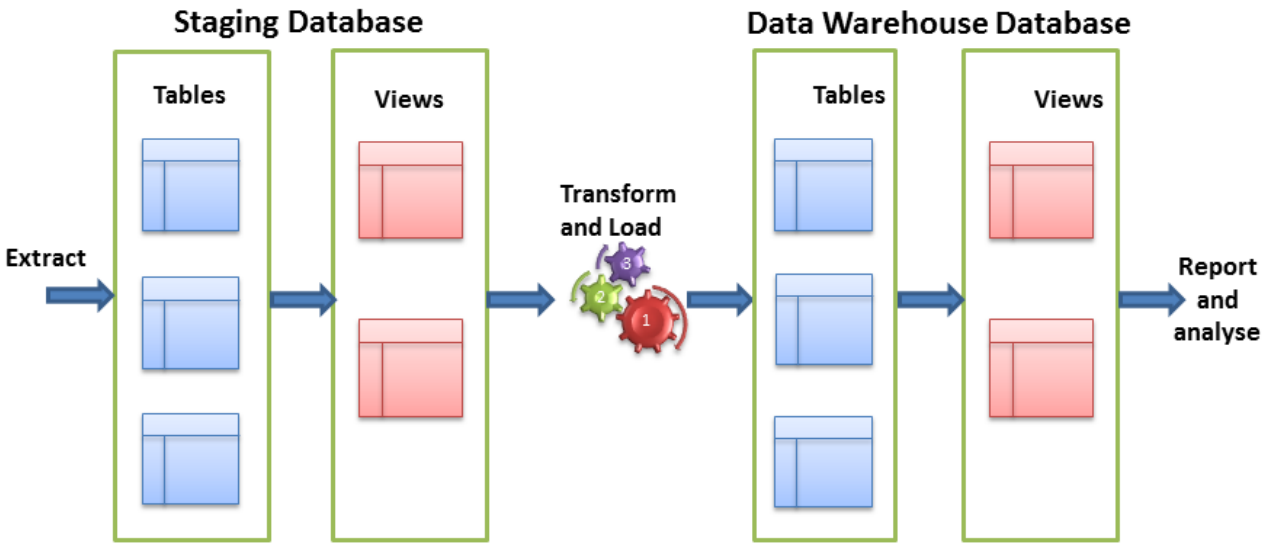


Figure 4.6: Example of usage of database views

```
CREATE VIEW [Applications].[vLINApplicants]
AS
SELECT
      MIN(Grades.Grades_sk) AS ID
      ,Students.[Kennitala]
      ,Fact.[Registration_ID]
      ,Majors.Name AS Major
      ,MajorTypes.Name AS MajorType
      ,Semesters.Code AS Semester
      ,SUM(CASE WHEN Grades.[Status] IN ('Lokið','Staðið')
            THEN Grades.Credits ELSE 0 END) AS ECTSCompleted
      ,SUM(CASE WHEN Grades.[Status] = 'Metið'
            THEN Grades.Credits ELSE 0 END) AS ECTSEvaluated
  FROM [dbo].[vDimStudents] Students
  INNER JOIN [dbo].[vFactStudentRegistrations] Fact ON Students.Student_sk = Fact.Student_sk
  INNER JOIN [dbo].[vFactStudentGrades] Grades On Fact.Registration_ID = Grades.Registration_ID
  INNER JOIN [dbo].[vDimMajors] Majors ON Fact.Major_sk = Majors.Major_sk
  INNER JOIN [dbo].[vDimMajorTypes] MajorTypes On Majors.Type_sk = MajorTypes.MajorTypes_sk
  INNER JOIN [dbo].[vDimSemesters] Semesters ON Grades.Semester_sk = Semesters.Semester_sk
  WHERE Students.LINApplicant = '1'
    AND Grades.[Status] IN ('Lokið','Metið','Staðið')
  GROUP BY
         Students.Kennitala
        ,Fact.Registration_ID
        ,Majors.Name
        ,MajorTypes.Name
        ,Semesters.Code
```

Figure 4.7: Example of definition of a database view

End users have access to the data warehouse through database views where they can retrieve data from the data warehouse. They need not be knowledgeable about SQL because the complexity of the data warehouse is hidden inside these views. Dimensions and fact tables have already been joined together, and data from many tables is presented as one table with large records.

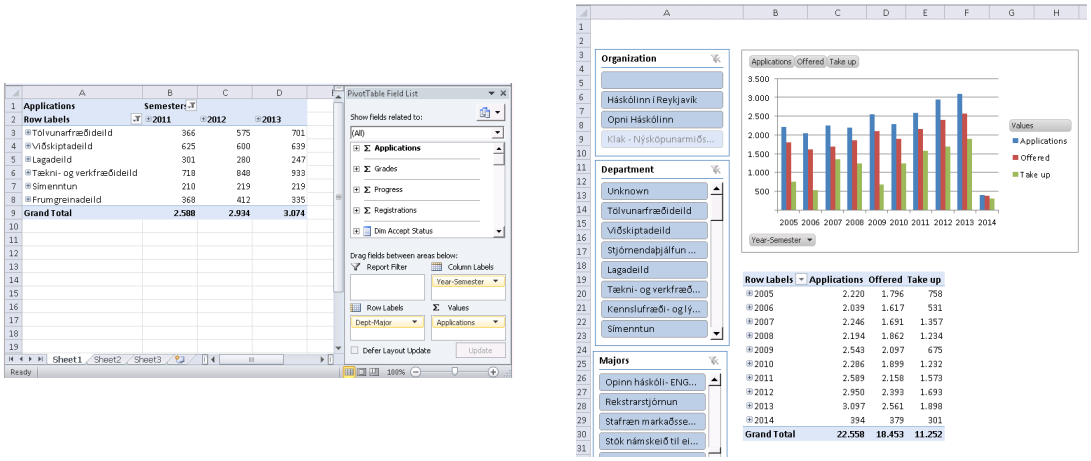


Figure 4.8: Two examples of usage of Microsoft Excel to access analytical cube

Figure 4.7 shows the definition of a view where two fact tables and four dimensions are joined together to create a database view for an application to retrieve some specific data about student applications from the data warehouse.

4.2.3 Analytical Cube

We created an analytical cube, RUDW CubeMySchool, on top of the data warehouse, for reporting and testing the quality of the data that has been loaded into the data warehouse. The user interface we offer is Microsoft Excel, and figure 4.8 shows two examples of Excel worksheet with data from the analytical cube.

4.3 Summary

We described what business requirements were taken into account and how we designed and implemented the RUDW. We discussed how data is loaded into the data warehouse and what software and tools were used. Then we discussed how we implemented the data loading process with integration packages and finally how we use database views to give access to data in the data warehouse.

Chapter 5

Evaluation

In this chapter, we briefly evaluate the usefulness of the data warehouse from a user perspective. The system was put into production at the beginning of 2014 and a group of university staff had been given access to the system. Members of this group included administrative directors and program administrators of the university departments (schools), dean of all schools and administrative director of the department for Teaching Affairs and Registry.

Two sessions were held where the system was introduced to this group and a link to an Excel document was sent in email to the group members. This Excel document had a connection to an analytical cube which was build on top of the data warehouse. Group members were asked to pilot test the system and report back if anything was broken or malfunctioning in the system. Group members also had access to the IT department (UTS) staff for assistance. My project partner G. Birna Guðmundsdóttir became an employee of the university IT department and had much interaction with the group members.

5.1 Questionnaire

After four months period of usage, we decided to evaluate how the system had been used. Six questions were created and sent to all four of the administrative directors of the university departments and to the administrative director of Teaching Affairs and Registry. They were asked to gather opinions from other members in the working area. This group was chosen because of their involvement in the testing group and to narrow the number of interviewers. Interviews were scheduled later with the same people to get their opinions.

The questions were presented in Icelandic but we show them here in English and they were:

1. Has anyone in your unit used the system?
2. What part of the system have you used?
   (a) Excel with connection to cube
   (b) Excel with connection to a view in the data warehouse
   (c) A query on the data warehouse with other tools than Excel
   (d) None
3. How did it perform with respect to speed?
4. Was the definition of dimensions and measures as you expected?
5. Is there anything that is missing from the system, e.g., different interface?
6. Is there any data that you need that is not in the system?

5.2 Results

We were able to interview three of the four administrative directors and the administrative director of Teaching Affairs and Registry.

Some key observations from the interviews, for each of the questions, are the following:

1. None of them was using the system directly. Some of them had tried to use the Excel-document but stopped after a short time. Either they did not know how to use the interface to the cube or, as one of them was using Microsoft Office for Mac, the connection did not work.
2. Some had tried option (a) but no one option (b) and (c)
3. Of those who had used the system the performance was acceptable. The only comparison they had, however, was the MySchool system.
4. Because of little usage of the system the interviewees had no opinion on this question.
5. Most of the interviewees are using Excel on a regular basis but found the interface to the cube to complex to use. They would like to have a simpler interface to the

system and some of them asked for a web like interface. They are used to running pre-cooked reports from the MySchool system and asked for a similar setup but with more flexible interface which would provide them with additional options such as adding columns to the reports.

6. Most of the interviewees had been sending requests to the university staff of the IT department (UTS) to extract data from the data warehouse and send the result back. All of them were satisfied with the outcome, the correctness of the data and the response time of the IT department.

The interviews also revealed that the following items were found missing in the data warehouse:

- Which students were exchange students (mostly foreign students).
- Which students had been exchange students at other universities. Reykjavik University offers their students to take one semester abroad.
- Ranking final grade of students' with their following graduate students.
- Distribution of grades for the same course over semesters. (Used to analyze changes in grades between semesters, maybe because of changes in teaching staff.)
- Teaching evaluations. Teaching evaluation is performed on a regular basis in the MySchool system, and the outcome of that is used to provide the administrators of the university assessment of the teaching staff and also provide the teaching staff feedback on how their teaching was perceived from the students' point of view.
- A list of students that have taken the same course twice and have not fulfilled the minimum grade.

5.3 Summary

Results from the questionnaire and the writers twenty years of experience in the IT industry, reveal that there are certain things that have to be taken in account before the data warehouse is developed further.

Users require an interface with standard reports that they can run in a simple way. This interface could be a dashboard with the standard reports that users will preferably be using. The interface would give the users additional features, such as adding columns, or modify the filters of the reports. Currently, there are many solutions that offer such features, and it would be practical to use one of them instead of building the solution from

scratch. Additional effort involved in defining requirements and setting up the system will always be necessary.

A group of key users should be formed with the function of addressing the development of the data warehouse. The group should be involved in decisions on what user interface to use, the data domains added to the warehouse and what additional analytical cubes were constructed. Members of the group could also promote the usage of the data warehouse inside the university and become a super user.

For the data warehouse to become the pivotal solution the university expects, the necessary manpower and effort has to be allocated for further development. This project was only the first step in building a data warehouse for Reykjavik University. One full time employee, with a reasonable knowledge of data warehouse and business intelligence theory should be adequate in the beginning.

To estimate the business value of the data warehouse for the university, it could be immense. The availability of standard reports for administrative directors that would give them almost instantaneous outcome with reliable information would be of a great value, instead of waiting for days or weeks for reports prepared by them self or by another member of the university staff.

Chapter 6

Conclusions

In this report, we presented the design and implementation of a data warehouse for Reykjavik University. It is the first step of a long journey towards a comprehensive data warehouse solution. A data warehouse is not a system that is designed once and installed; it needs to be maintained and developed to the needs within the organization. It is a continuous project. Future work will focus on expanding the data warehouse into other data domains and extending the functionality of the data warehouse.

Bibliography

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, *13*(6), 377–387.

Devlim, B., & Murphy, P. (1988). An architecture for a business and information system. *IBM System Journal*, *27*(1), 60-81.

Inmon, W. H. (1992). *Building the Data Warehouse*. New York, NY, USA: John Wiley & Sons, Inc.

Kimball, R. (1996). *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. John Wiley & Sons.

Kimball, R. (1998). *The Data Warehouse Lifecycle Toolkit: Expert methods for designing, developing, and deploying data warehouses*. John Wiley & Sons.

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons.

Melstað, S. (in prep.). *RUDW System Manual.*

Mundy, J., Thornthwaite, W., & Kimball, R. (2008). *The Microsoft Data Warehouse Toolkit, Second Edition*. Indianapolis, Indiana: Wiley Publishing, Inc.

Williams, P. (2014). *A Short History of Data Warehousing.* Retrieved 2014-05-03, from `http://www.dataversity.net/a-short-history-of-data-warehousing/`

School of Computer Science
Reykjavík University
Menntavegi 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.reykjavikuniversity.is
ISSN 1670-8539