NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Business Intelligence to support NOVA IMS Academic Services BI System

João Sequeira Marques Ribeiro

Project Work presented as partial requirement for obtaining the Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

BUSINESS INTELLIGENCE TO SUPPORT NOVA IMS ACADEMIC SERVICES BI SYSTEM

by

João Sequeira Marques Ribeiro

Project Work presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Specialization in Knowledge Management and Business Intelligence

Advisor: Roberto Henriques, Ph.D.

Co Advisor*:* Jorge Nelson Gouveia de Sousa Neves

August 2016

ABSTRACT

Kimball argues that Business Intelligence is one of the most important assets of any organization, allowing it to store, explore and add value to the organization's data which will ultimately help in the decision making process.

Nowadays, some organizations and, in this specific case, some schools are not yet transforming data into their full potential and business intelligence is one of the most known tools to help schools in this issue, seen as some of them are still using out-dated information systems, and do not yet apply business intelligence techniques to their increasing amounts of data so as to turn it into useful information and knowledge.

In the present report, I intend to analyse the current NOVA IMS academic services data and the rationales behind the need to work with this data, so as to propose a solution that will ultimately help the school board or the academic services to make better-supported decisions. In order to do so, it was developed a Data Warehouse that will clean and transform the source database. Another important step to help the academic services is to present a series of reports to discover information in the decision making process.

KEYWORDS

Business Intelligence; Star Schema; Microsoft Excel Power Pivot

INDEX

1. Introduction 1
   1.1. Contextualization 1
   1.2. Goals 2
   1.3. Project Approach 2
2. Literature review 4
   2.1. Introduction 4
   2.2. Initial Planning 5
   2.3. Data Warehouse Architectures 7
      2.3.1. Introduction 7
      2.3.2. Kimball DW Architecture 8
      2.3.3. Other Architectures 15
   2.4. Metadata 18
   2.5. Optimization and Scalability 19
   2.6. Management Processes 20
   2.7. Infrastructure 23
   2.8. Reporting Tools 25
   2.9. Trends 26
      2.9.1. Cloud BI 26
      2.9.2. Big Data 26
      2.9.3. In Memory Analytics 28
      2.9.4. Predictions 30
   2.10. Development Methodologies 30
      2.10.1. Waterfall 30
      2.10.2. Agile 31
3. Methodology 34
   3.1. Planning 34
   3.2. Analysis 35
      3.2.1. Project Methodology 35
      3.2.2. Software 36
   3.3. Implementation 36
      3.3.1. Source Database 36
      3.3.2. Data Warehouse Design 36
      3.3.3. ETL Processes 50

3.4. Reporting Design 67
3.4.1. General 67
3.4.2. College Enrolment 69
3.4.3. Course Enrolment Final Grade 71
3.4.4. Partial Grades 74
3.4.5. Work Allocation 75
4. Conclusions 77
5. Bibliography 78
6. Annexes 81

LIST OF FIGURES

Figure 1.1 – Sources of data kept and managed by the Information System of the Academic Services of a University 2
Figure 2.1 – Core Elements of the Kimball DW Architecture (Kimball & Ross, 2013) 9
Figure 2.2 – Example of Hierarchy (Moody & Kortink, 2000) 10
Figure 2.3 – Generic Structure of a Star Schema 12
Figure 2.4 – Example of a simplified illustration of the independent data mart architecture (Kimball & Ross, 2013) 16
Figure 2.5 – Example of a simplified illustration of the hub-and-spoke Corporate Information Factory architecture (Kimball & Ross, 2013) 17
Figure 2.6 – Hybrid architecture with 3NF structures and dimensional Kimball presentation area (Kimball & Ross, 2013) 18
Figure 2.7 – How in-memory analytics, interactive visualization and associative search affect businesses 29
Figure 2.8 – Example of Waterfall Methodology 31
Figure 3.1 – Master Thesis project Gantt chart 34
Figure 3.2 – Star schema diagram for Fact College Enrolment 37
Figure 3.3 – Star schema diagram for Fact Course Enrolment Final Grade 37
Figure 3.4 – Star schema diagram for Fact Partial Grades 38
Figure 3.5 – Star schema diagram for Fact Work Allocation 38
Figure 3.6 – Dimension Students details 45
Figure 3.7 – SSIS Main package 51
Figure 3.8 – SSIS Load Dimensions package 51
Figure 3.9 – SSIS Load Facts package 52
Figure 3.10 – ETL process for Dimension Academic Year Semester 53
Figure 3.11 – ETL process for Dimension Class 53
Figure 3.12 – ETL process for Dimension Courses 54
Figure 3.13 – ETL process for Dimension Date 54
Figure 3.14 – ETL process for Dimension Partial Grade Types 55
Figure 3.15 – ETL process for Dimension Professors 56
Figure 3.16 – Hierarchy from Dimension Programs Specializations Plans 56
Figure 3.17 – ETL process for Dimension Program Specialization Plan 57
Figure 3.18 – ETL process for Dimension Status 58
Figure 3.19 – ETL process for Dimension Students – Part 1 59
Figure 3.20 – ETL process for Dimension Students – Part 2 60

Figure 3.21 – ETL process for Dimension Term 61
Figure 3.22 – ETL process for Fact College Enrolment 62
Figure 3.23 – ETL process for Fact Course Enrolment Final Grade 64
Figure 3.24 – ETL process for Fact Partial Grades 65
Figure 3.25 – ETL process for Fact Work Allocation 66
Figure 3.26 – Nova IMS Map of Students Country 68
Figure 3.27 – Total Students by Gender and by Civil Status 69
Figure 3.28 – Courses Approved VS Not Approved by Academic Year and by Study Plan Year 70
Figure 3.29 – Total students enrolled in a Program and Specialisation 71
Figure 3.30 – Total Students enrolled by Class Type and the Average Grade of Students by Year and by Periodicity 72
Figure 3.31 – Total Students and Average Students Grade by Term 73
Figure 3.32 – Number of Course with 130+ Students and Total Students by Course Status 74
Figure 3.33 – Students Partial Grades Table 75
Figure 3.34 – Professors Work Allocation by Course 76

LIST OF TABLES

Table 2.1 – Differences between Operational Database and Data Warehouse (Santos & Ramos, 2009) 7
Table 2.2 – Differences between ROLAP, MOLAP, and HOLAP (Kimball & Ross, 2013) 14
Table 2.3 – Benefits and Risks of using a Business Intelligence cloud-based solution (Tamer, Kiley, Ashrafi, & Kuilboer, 2013) 24
Table 2.4 – Example of two Business Intelligence Software Solutions (Business-software, 2016) 25
Table 2.5 – Characteristics of In-Memory BI Solutions (Muntean, 2014) 30
Table 2.6 – Manifesto for Agile Software Development (Beedle, et al., 2001) 32
Table 2.7 – Twelve Principles behind the Agile Manifesto (Beedle, et al., 2001) 33
Table 3.1 – Dimension Academic Year Semester details 39
Table 3.2 – Dimension Class details 40
Table 3.3 – Dimension Courses details 40
Table 3.4 – Dimension Date details 41
Table 3.5 – Dimension Partial Grade Type details 42
Table 3.6 – Dimension Professors details 42
Table 3.7 – Dimension Program, Specialization and Plan details 43
Table 3.8 – Dimension Status details 44
Table 3.9 – Dimension Term details 45
Table 3.10 – Dimension Time details 46
Table 3.11 – Fact College Enrolment details 48
Table 3.12 – Fact Course Enrolment Final Grade details 49
Table 3.13 – Fact Partial Grades details 49
Table 3.14 – Fact Work Allocation details 50
Table 6.1 – SQL command for select the source data for Dimension Academic Year Semester 81
Table 6.2 – SQL command for select the source data for Dimension Partial Grade Type 82
Table 6.3 – SQL command for select the source data for Dimension Program Specialization Plan 82
Table 6.4 – SQL command for select the source data for Dimension Students 83
Table 6.5 – SQL command to populate Dimension Time 85
Table 6.6 – SQL command for select the source data for Fact College Enrolment 85
Table 6.7 – SQL command for select the source data for Fact Course Enrolment Final Grade 85

Table 6.8 – SQL command for select the source data for Fact Work Allocation 86
Table 6.9 – SQL command for select the source data for Fact Partial Grades 86
Table 6.10 – SQL command for lookup AcademicYearSemesterID in Fact Partial Grades 86
Table 6.11 – SQL command to select only DimProfessors latest data..... 87
Table 6.12 – SQL command to select DimProgSpecPlan in force data 87

LIST OF ABBREVIATIONS AND ACRONYMS

IMS	Information Management School
BI	Business Intelligence
IT	Information Technology
IS	Information Systems
EDM	Enterprise Data Model
DM	Data Mart
OLTP	On Line Transaction Processing databases
DW	Data Warehouse
3NF	Third Normal Form
DMS	Database Management Systems
RDMS	Relational Database Management Systems
ER	Entity Relationship
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
SCM	Supply Chain Management
OLAP	Online Analytical Processing
MOLAP	Multidimensional Online Analytical Processing
HOLAP	Hybrid Online Analytical Processing
ROLAP	Relational Online Analytical Processing
CIF	Corporate Information Factory
EDW	Enterprise Data Warehouse
SLAs	Service Level Agreements
DWA	Data Warehouse Administrator
CPM	Corporate Performance Management
RFID	Radio-Frequency Identification

SQL	Structured Query Language
NoSQL	Not Only SQL
IoT	Internet of Things
YARN	Yet Another Resource Negotiator
SDLC	System Development Lifecycle
SCDs	Slowly Changing Dimensions

1. INTRODUCTION

The aim of this Master thesis project is the development of a business intelligence solution to analyse the data from the Academic Services of Nova Information Management School. Consequently, the objective is to present and review conceptual and practical approaches. Therefore, during this project the chosen business intelligence solution will be implemented, which will be followed by the delivery of dashboards for data analysis, allowing the academic services to have a clearer understanding about the data that is producing.

1.1. CONTEXTUALIZATION

Nowadays, organisations are creating and capturing more data than ever before. It is by itself a whole new challenge for companies, which struggle to manage increasingly great amounts of data growing at a rapid rate. In order to deal with this new paradigm, Enterprises turn to Business Intelligence theories and technologies to extract the maximum amount of information from this data so as to make data driven business decisions. In fact, Business Intelligence systems are now quite advanced in some organisations. Therefore, the present project aims at suggesting a Business Intelligence method of developing a Data Warehouse following Kimball's approach that could be used by Higher Education Institutions to extract and analyse data, and ultimately make better-informed decisions (PwC, 2012).

The Information System (IS) of the Academic Services of a University is the source system where the school data is kept. Some examples of such data are summarized in the Figure 1.1. Firstly, the IS are responsible for holding on to the data regarding the enrolment of new students in one of the programs offered by the school, including Bachelor or Masters Degrees, Post Graduations or PhD's. Secondly, it must also contain information on the allocation of professors to the respective courses, which is also known as "Workout Allocation" and can be particularly challenging, in the sense that it involves matching classes to professors' personal schedules. Thirdly, the IS must ensure that every student is given both his partial and final grades from each course he completed. All this tasks and decisions require going through large amounts of data, which is why this report intends to develop a Business Intelligence approach that translates the said data into valuable information, leading to a faster, more efficient decision-making process.

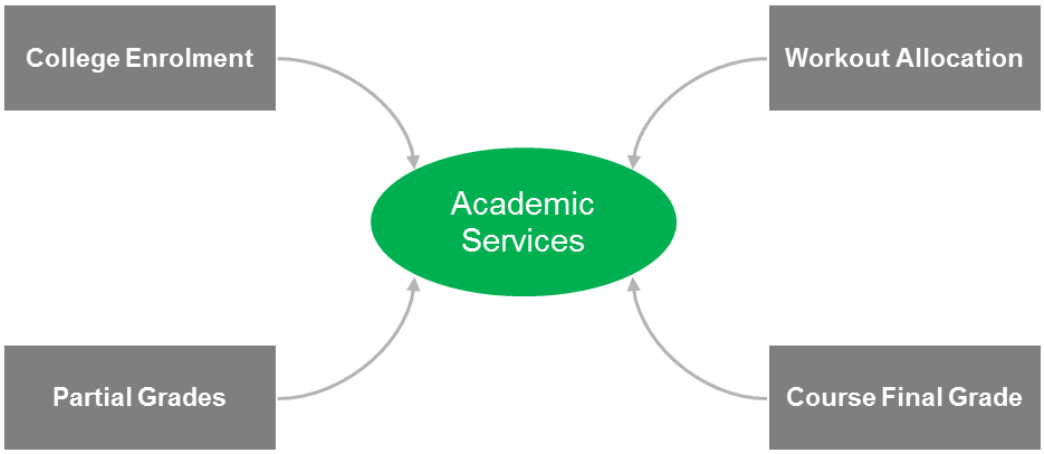

Figure 1.1 – Sources of data kept and managed by the Information System of the Academic Services of a University

1.2. GOALS

The project goal is to provide dashboard(s) to help analyse the extracted data from the NOVA IMS Academic Services Information System using a Business Intelligence approach.

Furthermore, the aim of this project discloses in the following objectives:

1. Analyse the data provided by the Academic Services of Nova IMS
2. Build a Data Warehouse oriented to the main data sources of the Information System of the Academic Services of Nova IMS (College Enrolment, Workout Allocation, Partial Grades and Course Final Grades)
3. Build Dashboards for each source of data of the Information System of the Academic Services of Nova IMS

1.3. PROJECT APPROACH

Business Intelligence is a technique to transform data and to guide decisions for smart business operations. With that in mind, this project will analyse different Business Intelligence approaches so as to find the most appropriate one to help the academic services in their decision making process.

Therefore, we will present several Business Intelligence methods and techniques, find the most suitable for the academic services of a University, apply it to the data it currently manages and deliver a dashboard with the chosen reporting tool.

Other topics will also be addressed, such as how to handle metadata, how to provide an optimized and scalable solution, the management processes that should be covered and the needed infrastructure to create the appropriate Business Intelligence solution.

The project will be composed of three main chapters. The first chapter, the Literature Review, will present and review conceptual and practical approaches for the business intelligence solution as well as the development methodology that this project will follow. The second chapter of this project will be the Methodology where all analysis, implementation and reporting results will take place. Finally, the conclusions chapter where the project accomplishes are described along the future work and limitations.

2. LITERATURE REVIEW

2.1. INTRODUCTION

The term Business Intelligence was firstly defined by the IBM researcher Hans Peter Luhn (1958), as the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal (Luhn, 1958).

Later in 1989, Howard Dresner suggested that Business Intelligence was an umbrella term to describe concepts and methods to improve businesses' decision making by using fact-based support systems. In late 1990s this definition was widespread (Cebotarean, 2011).

Nowadays, data is growing at a rapid rate. Enterprises are turning to Business Intelligence theories and technologies in order to extract the maximum amount of information from this data so as to allow their employees to make better data-driven business decisions (Obeidat, North, Richardson, & Rattanak, 2015). These days, how companies treat their data can determine the success (or failure) of the most important business decisions. The companies that can access the information they need, when they need it – and trust its accuracy – have the upper hand when it matters. Companies today are creating and capturing more information than ever before, which makes their challenge harder, instead of easier (PwC, 2012).

According to Obeidat, North, Richardson, & Rattanak (2015), Business Intelligence applications are periodically used in a popular of search-based applications within a variety of fields, such as Business, Marketing, Law, Education, Visualization, Finance, Engineering, Security Medicine, Health Informatics. Although BI is widely used in private or public organisations for both ordinary business and Internet based business, BI applications are growing in many diverse fields. For instance, in the areas of Mobile Device, Fraud Detection and even in Chronic Disease Management, studies are beginning to designate the benefits of BI applications.

In order to discover different ways on how to handle the subject presented by this project it is important to investigate the current state of art in the Business Intelligence world. Therefore, the investigation will be divided into different sections where different chapters regarding Business Intelligence are presented.

The first section, "Initial Planning", contains a high-level overview of what steps should be completed in order to start the planning of building a Data Warehouse. This section although very important is sometimes missed when a company starts to design the Data Warehouse.

The second section, "Data Warehouse", has the core of this literature review, starting with a brief introduction of the concept of Data Warehouse as well as the goals and benefits of it. Also in this section there are four additional major sub-sections in which different approaches on how to design a Data Warehouse are presented.

The next section is "Metadata" where the importance and need of using metadata on a Business Intelligence system is detailed. It is also contained in the section which documents should contain metadata information.

After metadata, the next section is “Optimization and Scalability” and it explains the scalability major issues and also the challenges that may occur while implementing new solutions.

The next section is “Management Processes” and it focuses on the problem in which this area is usually forgotten in organizations. Another topic is the importance of having a good management, information and data quality. Finally, it will also address the need of having data warehouse policies/administration and consistency.

The next section is “Infrastructure” and it will describe the advantages and impact of data acquisition and data storage have dropped the price in the last years. Another major debate is the strategy of organizations about where to have the data, if locally or in the cloud.

The next section is “Reporting Tools” and it will detail the different Reports/ Dashboard Tools in the market nowadays. This research will show which tool suits better the project.

The next section is “Trends”, aiming primarily to the evolution of the Business Intelligence being more focused on data exploration and visualization. Another trend is the use of web technologies and smartphones, which is not entirely new but is growing more and more. Finally, this section will also include some prediction in the area.

The last section “Development Methodologies” is composed by the different development methodologies that are currently in use in the development market. This master thesis project will be based on Agile methodology.

2.2. INITIAL PLANNING

The initial planning of a Data Warehouse starts with the requirements like business need, outputs, expectations, indication of a scope for the data required and how it should be delivered. It also must be accompanied by defining a subject model[1], beginning to document the understanding of data, helping to communicate the scope of the data warehouse and providing a context for later data models.

The requirements analysis is the succeeding step and they are often stated in a functional form. According to Boateng, Singh, Greeshma & Singh (2012) the following factors affect the architectures selection decision:

- Nature of end-user tasks
- Information interdependence between organizational units
- Social/political factors
- Constraints on possessions
- Professed ability of the in-house IT staff
- Upper management’s information needs

[1] Subject area model is a visual containing a high-level perspective on something of importance to the business - http://www.irmuk.co.uk/articles/Hoberman_What_is_the_subject_area_model.pdf

- Necessity of need for a data warehouse
- Strategic view of the data warehouse former to implementation
- Compatibility with existing system
- Technical Issues

Also according to Boateng, Singh, Greeshma & Singh (2012), the following questions will impact when deciding which architecture should be used:

- What tools will be used to sustain data recovery and analysis?
- Which database management system should be used?
- Will data migration tools be used to load the data warehouse?
- Will parallel processing or partitioning be used?

In agreement with Moody & Kortink (2000), these are essential steps that need to be completed for developing a Data Warehouse. These steps may be found below:

1. Develop Enterprise Data Model[2] (EDM)
2. Design Data Warehouse – based on the enterprise data model, but will be a subset of the model which is relevant for decision support purposes
3. Classify Entities – classify entities in the data warehouse model as either transaction, component or classification entities
4. Identify Hierarchies – identify the hierarchies which exist in the data model
5. Design Data Marts (DM) – develop star cluster schemas for each transaction entity in the data warehouse model. Each star cluster will consist of a fact table and a number of dimension and sub dimension tables. This minimises the number of tables while avoiding overlap between dimensions

There are two major architectures when designing Data Warehouses. The first architecture belongs to Bill Inmon, known as corporate information factory, which follows a top-down design. Starting with a normalized data model it is then defined the dimensional data marts, which contains data required for specific departments that are created from the data warehouse. This architecture is also organised around the applications of the company (Inmon, 2002). The second one presented by Ralph Kimball, known as dimensional data warehouse architecture, considered a bottom-up design in which business users have a simple dimensional structure at first and when combined together it will create a broad Data Warehouse. It is a design to match the fundamental human need for simplicity.

[2] An Enterprise Data Model is an integrated view of the data produced and consumed across an entire organization. It incorporates an appropriate industry perspective - http://tdan.com/the-enterprise-data-model/5205

Simplicity is critical because it ensures that users can easily understand the data, as well as allows software to navigate and deliver results quickly and efficiently (Kimball & Ross, 2013).

A Data Warehouse is a database that is managed independently of an Operational database (OLTP - On Line Transaction Processing databases), according to Santos & Ramos (2009). The Table 2.1 illustrates the main differences between a Data Warehouse and an Operational database.

Operational Database	Data Warehouse
Operational Purposes	Records history
Read/Write access	Read-only access
Pre-defined transactions access	Periodic reports and ad hoc access
Access to a small amount of records	Access to a huge amount of records
Refresh of data near real time	Scheduled data loads
Optimized structure for updates	Optimized structure for processing issues

Table 2.1 – Differences between Operational Database and Data Warehouse (Santos & Ramos, 2009)

Table 2.1 illustrates that both types of databases serve different purposes. The operational database aims to handle near real time transactions (i.e. credit cards transactions) while the data warehouse is more dedicated for analysing a bugger volume of data. All these differences must be taken into account in the process of selecting a new database (Santos & Ramos, 2009).

2.3. DATA WAREHOUSE ARCHITECTURES

2.3.1. Introduction

As stated by Kimball & Ross (2013), information is one of the most important assets of any organization and it is practically always used for two purposes: operational record keeping and analytical decision making. Simply speaking, the operational systems are where you put the data in, and the DW/BI system is where you get the data out.

Data Warehousing is a technique of bringing collectively all of a company's data from different computer systems, together with those connecting to customers, employees, vendors, product, inventory, and financial. The data warehouse connects different database together in order to offer a more inclusive data set for making decision (Boateng, Singh, Greeshma, & Singh, 2012).

There are numerous goals of Data Warehousing and Business Intelligence systems that should be transformed into requirements for the DW system, as reported by Kimball & Ross (2013) it must:

- Make information straightforwardly accessible

- Present information consistently
- Adapt to regular change
- Present information in a timely way
- Secure support that protects the information resources
- Serve as the confident and trustworthy foundation for improved decision making
- The business community must accept the DW system to deem it successful

Despite all of the above requirements being important, the last two, "the data warehouse to serve as the authoritative and trustworthy foundation for improved decision making" and "that the business community should accept a data warehouse in order for it to be successful" are the most overlooked ones. The success of a data warehouse depends not only on the best technicians but also on the best business users, which only by combining both skills the data warehouse will be successfully delivered (Kimball & Ross, 2013).

It is very important to understand that the key to long-term success in query performance, better maintainability, and robust recovery options is to have the right data warehouse foundation and data warehouse design (Petrenko, Rada, Fitzsimons, McCallig, & Zuzarte, 2012).

2.3.2. Kimball DW Architecture

2.3.2.1. Dimensional Modeling

Dimensional modelling is broadly acknowledged as the favourite technique for presenting analytic data because it addresses two simultaneous requirements. Firstly, it delivers data that's logical to the business users and in second it delivers fast query performance. The dimensional modeling is a technique known for being simple, in consonance with Kimball & Ross (2013).

In line with Singh, Singh, & Sriveni (2010), a data warehouse with dimensional modeling is denormalized by nature it is used to transform business rules into useful information as it is the collection of archived operational data which is useful in tactical and strategic business decisions.

Although dimensional models are often instantiated in relational database management systems (RDBMS), they are quite different from third normal form (3NF) models[3] (Kimball & Ross, 2013).

[3] Third Normal form applies that every non-prime attribute of table must be dependent on primary key - http://www.studytonight.com/dbms/database-normalization.php

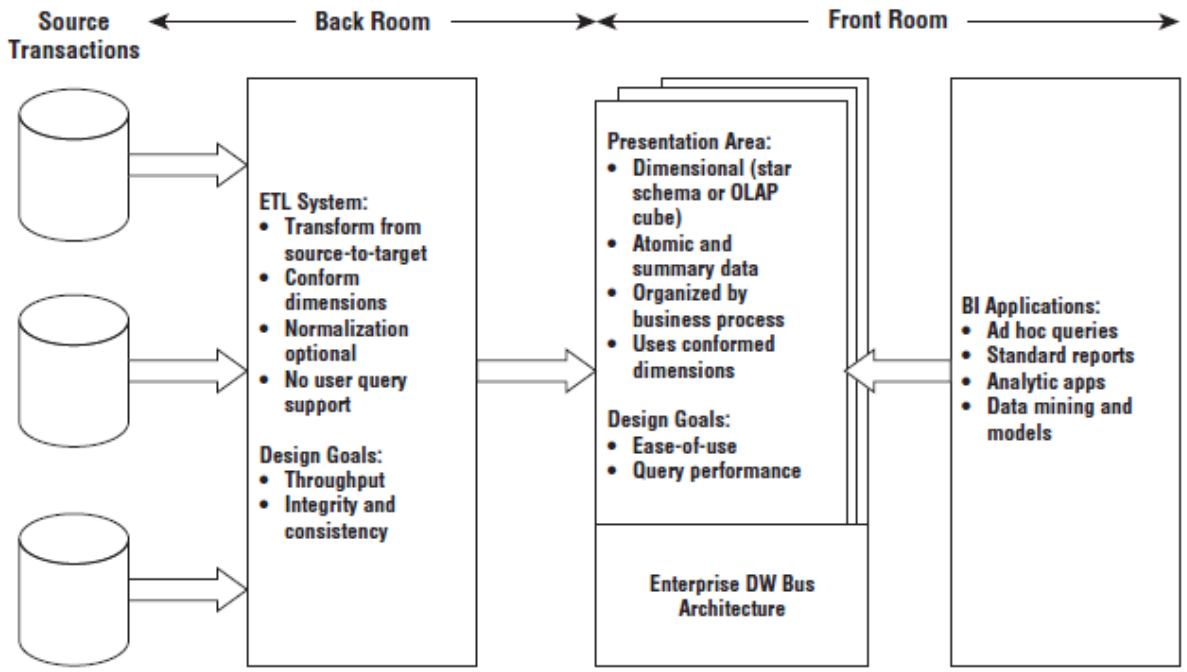

Figure 2.1 – Core Elements of the Kimball DW Architecture (Kimball & Ross, 2013)

Figure 2.1 displays Kimball's data warehouse architecture where it is divided in four main stages. The first is the source transactions (detailed on section 2.3.2.3) which consists in source systems that capture business's transactions. After, the Extract, Transform and Load (ETL) which is made of (more detailed on section 2.3.2.4) a work area, data structures and a set of processes. The ETL is followed by the Presentation Area (more detailed on section 2.3.2.5) where the data is organised and stored and where the business may start to access. Finally, the last price to completely connect the business with the data presentation layer, the business intelligence applications (more detailed on section 2.3.2.6) that enables the business users to use the presentation area for analytic decision making where a range of capabilities are provided.

2.3.2.2. Hierarchies

As reported by Moody & Kortink (2000), hierarchies are an extremely important concept in dimensional modelling, and form the primary basis for deriving dimensional models from Entity Relationship (ER) models[4]. Most dimension tables contain embedded hierarchies. A hierarchy in an Entity Relationship model is any sequence of entities joined together by one-to-many relationships, all aligned in the same direction. Figure 2.2 shows a hierarchy extracted from the example data model, with State at the top and Sale Item at the bottom.

[4] ER modeling is an important step in information system design and software engineering - http://www.csc.lsu.edu/~chen/pdf/Chen_Pioneers.pdf

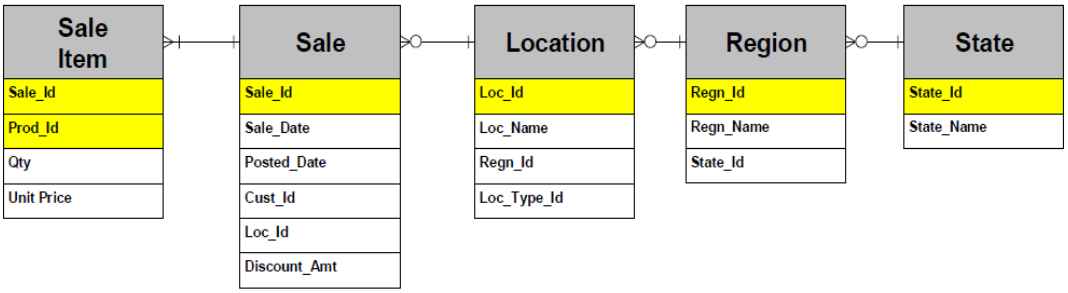

Figure 2.2 – Example of Hierarchy (Moody & Kortink, 2000)

In hierarchical terminology, it is stated by Moody & Kortink (2000) as:

- State is the parent of Region
- Region is the child of State
- Sale Item, Sale, Location and Region are all “descendants” of State
- Sale, Location, Region and State are all “ancestors” of Sale Item

As demonstrated in Figure 2.1, there are four isolated and individual components to consider in the DW Kimball approach, each one of these components will also be covered in the next four sub-sections:

- Data Source
- ETL
- Data Presentation Area
- Business Intelligence Applications

2.3.2.3. Data Source Systems

One of the main goals of a Data Warehouse system is to combine different data resources into information about processes in the company and provide this information in appropriate way and timely to company management. The main source systems can serve enterprise information, systems like Enterprise Resource Planning (ERP), Customer Relationship Management (CRL), or Supply Chain Management (SCM). Also complementary data from external private or public resources can be used. Finally, an integration of technologies with data analysis and visualization tools into BI principles, applicable in small enterprise, is suggested here. This approach presents one possibility how to arrange and process data from different resources and make them available timely for managerial decisions, according to Horakova & Skalska (2013).

These source systems are also known as data sources which can be described, in line with Ranjan (2009), in a simpler way as being operational databases, historical data, external data for example, from market research companies or from the Internet, or information from the already existing data warehouse environment. The data sources can be relational databases or any other data structure that supports the line of business applications. They also can reside on many different platforms and

can contain structured information, such as tables or spreadsheets, or unstructured information, such as plaintext files or pictures and other multimedia information.

2.3.2.4. ETL

In line with Boateng, Singh, Greeshma, & Singh (2012), ETL can be defined as the data warehousing process that consists of extraction, transformation and load which is called putting data to the data warehouse.

According to Kimball & Ross (2013) the data warehouse process of ETL starts out with the extraction of data from the source systems into the ETL system. The extraction process is meant to understand the source systems data and after analysing it should be copied into the data warehouse. After the data is extracted to the data warehouse, there are several possible transformations, such as cleansing the data (correcting misspellings, resolving domain conflicts, dealing with missing elements, or parsing into standard formats), combining data from multiple sources, and de-duplicating data. The ETL organization adds data value with these cleansing and conforming tasks by changing the data and enhancing it.

A major part of the transformation process involves data history and in a real world, dimensions are not static eternally. Some of the dimension attributes may change over time and to handle this, it is necessary to track these changes. Kimball introduced a way of tracking these changes, naming it Slowly Changing Dimensions (SCDs) and it defines that each dimension table attribute should have a specific strategy to handle change.

The final step of the ETL process is the physical structuring and loading of data into the presentation area's target dimensional models. The primary reason of an ETL system is to hand off the dimension and fact tables to the delivery step.

As reported by Simitsis & Vassiliadis (2003), the ETL processes can be simplified with five main task as may be seen below:

- Identification of relevant information at the source side
- Extraction of this information
- Customization and integration of the information coming from multiple sources into a common format
- Cleaning of the resulting data set, on the basis of database and business rules
- Propagation of the data to the data warehouse and/or data marts.

2.3.2.5. Data Presentation Area

As stated by Kimball & Ross (2013), the presentation data area should be organised around business process measurement events. Data in the queryable presentation area of the Data Warehouse system must be dimensional, atomic, business process-centric, and adhere to the enterprise data

warehouse bus architecture. The data must not be structured according to individual departments' interpretation of the data.

There are two approaches to choose from in this stage. Starting out with star schema which according with Moody & Kortink (2000) is the basic building block used in dimensional modelling. A star schema consists of one large central table called the fact table, and a number of smaller tables called dimension tables which radiate out from the central table as may be seen in Figure 2.3.

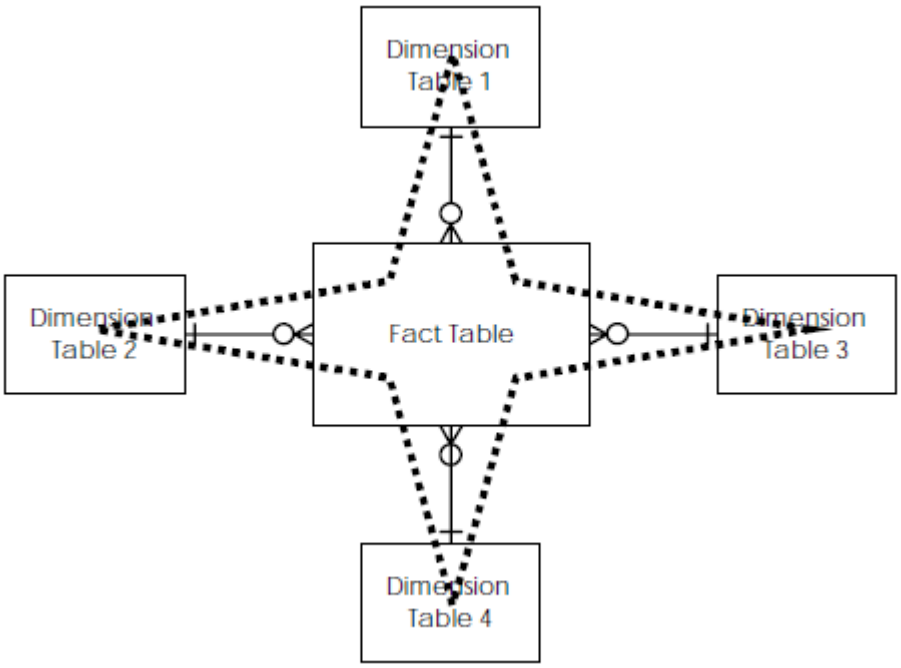

Figure 2.3 – Generic Structure of a Star Schema

Also in line with Moody & Kortink (2000) there some things to consider when handling star schemas:

- The fact table contains measurements (e.g. price of products sold, quantity of products sold) which may be aggregated in various ways
- The dimension tables provide the basis for aggregating the measurements in the fact table
- The fact table is linked to all the dimension tables by one-to-many relationships
- The primary key of the fact table is the concatenation of the primary keys of all the dimension table

The second approach is referred to as online analytical processing (OLAP) cubes. If the Data Warehouse includes either star schemas or OLAP cubes, it leverages dimensional concepts. Both stars and cubes have a common logical design with identifiable dimensions; however, the physical implementation varies. When data is loaded into an OLAP cube, it is deposited and indexed using

formats and methods that are designed for dimensional data. Performance combinations or pre-calculated summary tables are often created and managed by the OLAP cube engine. Consequently, cubes deliver superior query performance because of the pre-calculations, indexing strategies, and other optimizations. Business users can drill down or up by adding or removing attributes from their analyses with excellent performance without issuing new queries. OLAP cubes also provide more analytically robust functions that exceed those available with SQL. The downside is that it pays a load performance price for these capabilities, especially with large data sets.

OLAP is also referred as the way in which business users can slice and dice their way through data using sophisticated tools that allow for the navigation of dimensions such as time or hierarchies. It provides multidimensional, summarized views of business data and is used for reporting, analysis, modeling and planning for optimizing the business. OLAP techniques and tools can be used to work with data warehouses or data marts designed for sophisticated enterprise intelligence systems, as reported by Ranjan (2009).

The OLAP cubes in a data warehouse can be stored in three different modes. The Table 2.2 presents each mode characteristics.

Multidimensional Online Analytical Processing (MOLAP)	**Hybrid Online Analytical Processing (HOLAP)**	**Relational Online Analytical Processing (ROLAP)**
The storage mode causes the aggregations of the partition and a copy of its source data to be stored in a multidimensional structure when the partition is processed	The storage mode combines attributes of both MOLAP and ROLAP. Like MOLAP, HOLAP causes the aggregations of the partition to be stored in a multidimensional structure	The storage mode causes the aggregations of the partition to be stored in indexed views in the relational database that was specified in the partition's data source
Structure highly optimized to maximize query performance	Does not cause a copy of the source data to be stored	Enables users to view data in real time and save storage space when working with large datasets that are infrequently queried, such as purely historical data
Query response times can be decreased substantially by using aggregations	For queries that access only summary data in the aggregations of a partition, HOLAP is the equivalent of MOLAP. Nevertheless, queries that access source data must retrieve data from the relational database and will not be as fast as they would be if the source data were stored in the MOLAP structure	Query response is generally slower with ROLAP storage than with the MOLAP or HOLAP storage modes

The data in the partition's structure is only as current as the most recent processing of the partition	Users will typically experience substantial differences in query times depending upon whether the query can be resolved from cache or aggregations versus from the source data itself	Processing time is also typically slower with ROLAP

Table 2.2 – Differences between ROLAP, MOLAP, and HOLAP (Kimball & Ross, 2013)

According to Kimball & Ross (2013) star schemas and OLAP cubes have a common logical design with recognizable dimensions. However, the physical implementation differs. When data is loaded into an OLAP cube, it is stored and indexed using formats and techniques that are designed for dimensional data. Performance aggregations or pre-calculated summary tables are often created and managed by the OLAP cube engine. Consequently, cubes deliver superior query performance because of the Data Warehousing, Business Intelligence, and Dimensional Modelling pre-calculations, indexing strategies, and other optimizations. Business users can drill down or up by adding or removing attributes from their analyses with excellent performance without issuing new queries. OLAP cubes also provide more analytically robust functions that exceed those available with SQL. The downside is that you pay a load performance price for these capabilities, especially with large data sets.

Kimball & Ross (2013) claims that use of star schemas to design data warehouses results in 80% of queries being single table browses. Star schemas may either be implemented in specialist OLAP tools, or using traditional relational DBMS. The advantage of using star schemas to represent data is that it reduces the number of tables in the database and the number of relationships between them and therefore the number of joins required in user queries.

2.3.2.6. Business Intelligence Applications

Kimball & Ross (2013) refers to the term BI application as the range of capabilities provided to business users to leverage the presentation area for analytic decision making. A BI application can be:

- Ad hoc queries – as simple as an ad hoc query tool or as complex as a sophisticated data mining or modelling application
- Standard reports – Most corporate users will probable access the data through prebuilt parameter-driven applications and templates that do not require users to construct queries directly
- Analytic apps – Ad hoc query tools may be understood and used efficiently by only a minor percentage of the potential data warehouse business users

- Data mining & models – Some of the sophisticated applications, such as modeling tools, might upload results back into the operational source systems, ETL or presentation area

Obeidat, North, Richardson & Rattanak (2015) adds that the Business Intelligence applications are infrequently used in a popular of search-based applications within a diversity of fields, such as Business, Security, Finance, Marketing, Law, Education, Visualization, Science, Engineering, Medicine, Bioinformatics, Health Informatics, Humanities, Retailing, and Telecommunications.

2.3.3. Other Architectures

2.3.3.1. Independent Data Mart Architecture

In line with Moody & Kortink (2000), a data mart represents the lower level of the data warehouse, where data is accessed directly by the department final users. Data is extracted from the data warehouse into data marts (smaller data warehouses) to back analysis requirements. The most important requirement at this level is that data is structured to be easy for users to understand and use. Therefore, dimensional modelling techniques are most appropriate at this level. This ensures that data structures are as simple as possible in order to simplify user queries.

Kimball & Ross (2013) also wrote that a single department classifies requirements for data from an operational source system. The department works with IT staff or external consultants to build a database that satisfies their departmental requirements, reproducing their business rules. In the meantime, another department may be interested in the same source data. It's enormously common for multiple departments to be interested in the same performance metrics resulting from an organization's core business process events. Then again, because this department doesn't have access to the data mart initially constructed by the other department, it proceeds down a similar track on its own, obtaining resources and building a departmental solution that contains similar, but somewhat different data. Once business users from these two departments discuss organizational performance based on reports from their respective repositories, not surprisingly, the numbers won't match because of the differences in business rules.

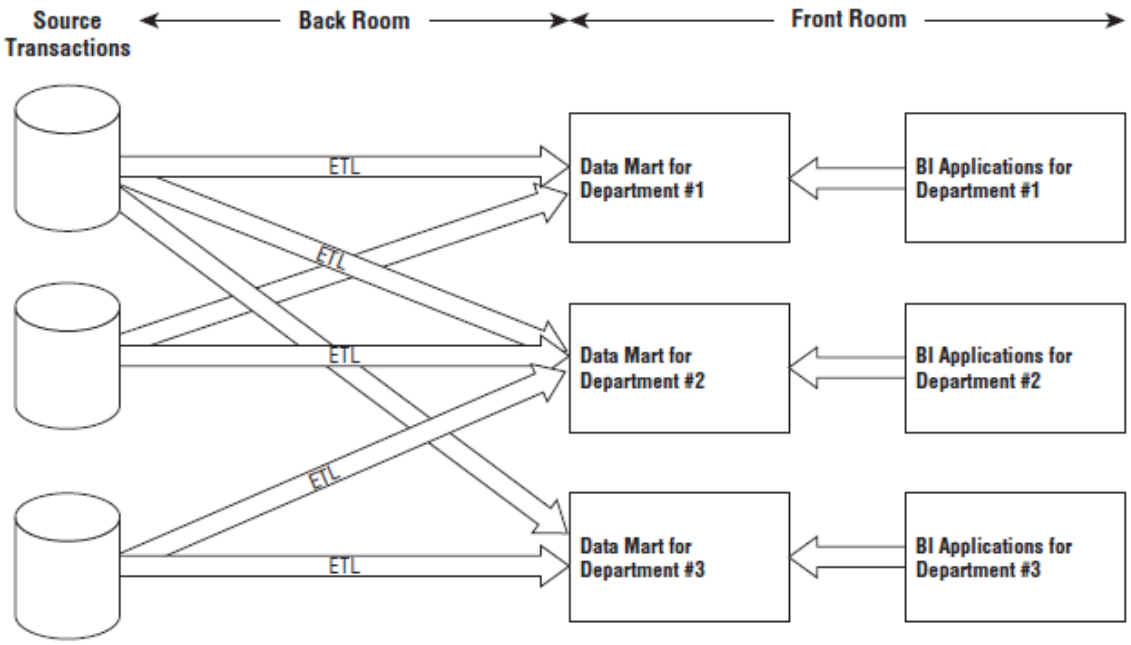

Figure 2.4 – Example of a simplified illustration of the independent data mart architecture (Kimball & Ross, 2013)

These standalone analytic silos represent a DW design that's fundamentally un-architected. Even though no industry leaders promote these independent data marts, this approach is predominant, especially in large organizations. It mirrors the way many organizations fund IT projects, plus it doesn't require cross-organizational data governance and coordination. It's the path of least resistance for fast development at relatively low cost, at least in the short run. Undeniably, multiple uncoordinated extracts from the equivalent operational sources and redundant storage of analytic data are inefficient and wasteful in the long run. Deprived of any enterprise perspective, this independent approach results in myriad standalone point solutions that perpetuate incompatible views of the organization's performance, resulting in unnecessary organizational argument. It's strongly discouraged the independent data mart approach. Nonetheless, often these independent data marts have embraced dimensional modeling because they're interested in delivering data that's easy for the business to understand and highly responsive to queries. Therefore, the theories of dimensional modeling are often applied in this architecture, in spite of the complete disregard for some of the core doctrines, such as focusing on atomic details, building by business process instead of department, and leveraging conformed dimensions for enterprise consistency and integration (Kimball & Ross, 2013).

2.3.3.2. Hub-and-Spoke Corporate Information Factory Inmon Architecture

Kimball & Ross (2013) introduces the hub-and-spoke Corporate Information Factory[5] (CIF) approach as being supported by Bill Inmon and others in the industry. This methodology starts by the data

[5] Overview of a Corporate Information Factory - http://www.inmoncif.com/library/cif/

being extracted from the operational source systems and processed through an ETL system occasionally mentioned to as data acquisition. The atomic data that outcomes from this processing lands in a 3NF database; this normalized, atomic repository is raised to as the Enterprise Data Warehouse (EDW) within the CIF architecture. Even though the Kimball architecture allows optional normalization to support ETL processing, the normalized EDW is a mandatory build in the CIF. As the Kimball approach, the CIF promotes enterprise data coordination and integration. The CIF states the normalized EDW is suited for this role, whereas the Kimball architecture strains the importance of an enterprise bus with fit in dimensions.

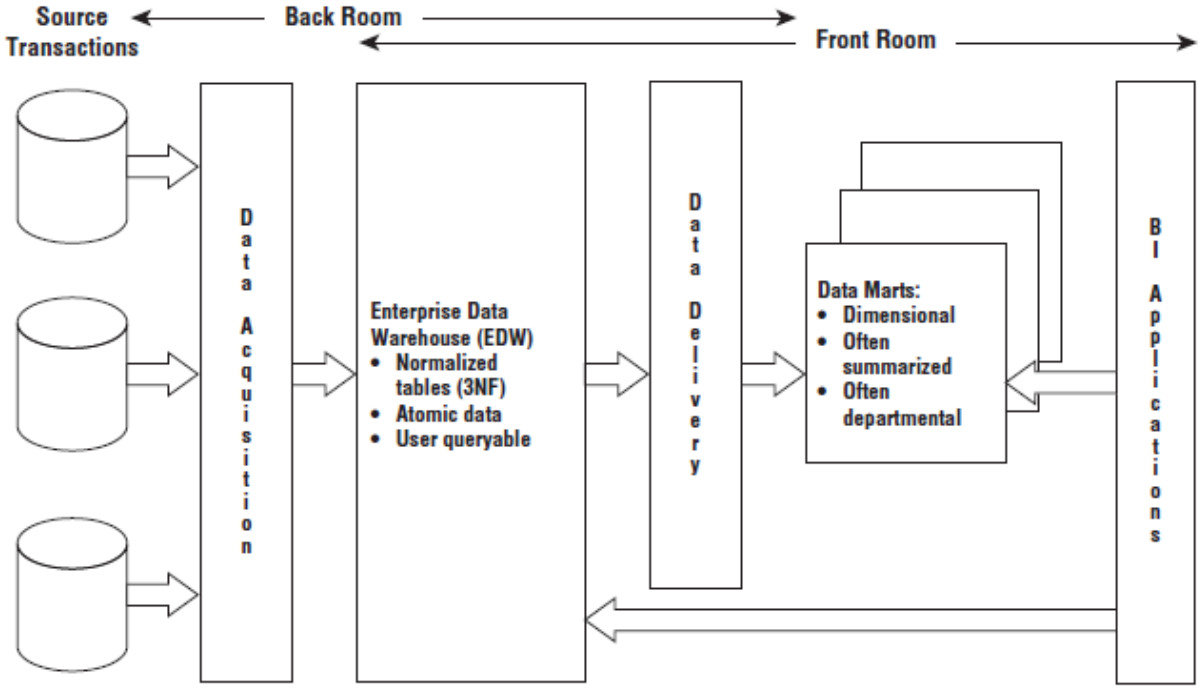

Figure 2.5 – Example of a simplified illustration of the hub-and-spoke Corporate Information Factory architecture (Kimball & Ross, 2013)

Kimball & Ross (2013) also studied that organisations who have adopted the CIF approach often have business users accessing the EDW repository due to its level of detail or data availability timeliness. On the other hand, succeeding ETL data delivery processes also populate downstream reporting and analytic environments to support business users. Even though often dimensionally structured, the resultant analytic databases naturally differ from structures in the Kimball architecture's presentation area in that they are frequently departmentally-centric and populated with aggregated data. If the data delivery ETL processes apply business rules outside basic summarization, such as departmental changing the name of columns or alternative calculations, it could be possibly difficult to tie these analytic databases to the EDW's atomic repository.

2.3.3.3. Hybrid Hub-and-Spoke and Kimball Architecture

The last architecture justifying discussion is the marriage of the Kimball and Inmon CIF architectures. As illustrated in Figure 2.6, this architecture populates a CIF-centric EDW that is entirely off-limits to business users for analysis and reporting. It's purely the source to populate a Kimball-esque presentation area in which the data is dimensional, atomic (complemented by aggregates) and conforms to the enterprise data warehouse bus architecture. Some proponents of this blended approach claim it's the best of both worlds. The situation may perhaps leverage a pre-existing investment in an integrated repository, at the same time addressing the performance and usability matters associated with the 3NF EDW by offloading queries to the dimensional presentation area. In addition, because the final deliverable to the business users and BI applications is constructed based on Kimball tenets, it is quite hard to argue with the approach. Especially if a large investment in the creation of a 3NF EDW was made and it's not delivering on the users' expectations of fast and flexible reporting and analysis. The hybrid approach might be applicable for an organization which is starting with a blank sheet of paper. The hybrid approach will likely cost more time and money, both during development and operation, given the multiple movements of data and redundant storage of atomic details (Kimball & Ross, 2013).

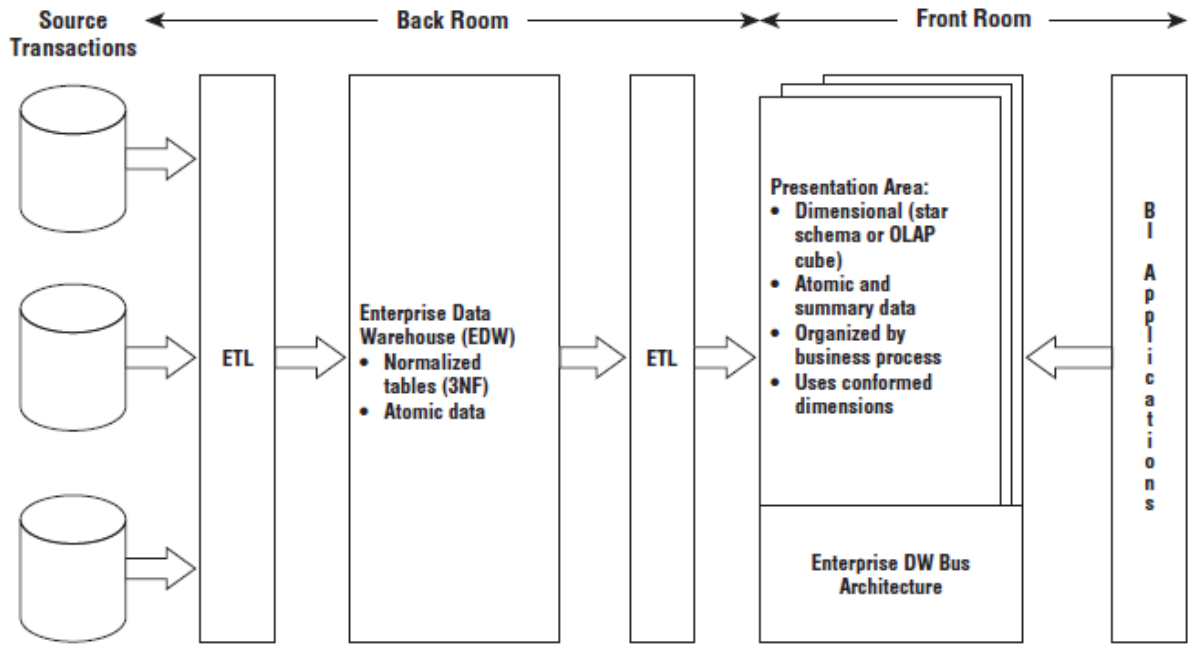

Figure 2.6 – Hybrid architecture with 3NF structures and dimensional Kimball presentation area (Kimball & Ross, 2013)

2.4. METADATA

Metadata is fundamental to solve the problem with the search ability and assessment of the data and this phenomenon is achieved by adding context and identifying this data. A lot of systems already captures some metadata like filenames, authors or size, but metadata about content can me

much more useful. For example, summaries, topics, people or companies mentioned. Two technologies designed for generating metadata about content is to extract information and to automatic categorize (Cebotarean, 2011).

In order to answer the question “what should be included as metadata?” Inmon (2002) has provided a list the important information that should be retained:

- Document ID
- Data of entry
- Description
- Source
- Classification(s)
- Index words
- Purge date
- Physical location reference
- Length
- Related references

Data transformation tools (ETL tools) are used for data extraction from source systems and subsequently for transformation and transmission these data into the specialized database. Advanced ETL suites often handle also data quality control mechanisms as well as metadata management, as reported by Horakova & Skalska (2013). Boateng, Singh, Greeshma, & Singh (2012) has also stated that automatic capturing and delivery of metadata is a major goal of ETL.

The Data Warehouse architect must be vigilant about how to populate and maintain the metadata repository. The attribute definition, interpretations and each staged/archived data set should have accompanying metadata describing the origins and processing steps that produced the data origins should be documented in the metadata. The tracking of this lineage is explicitly required by certain compliance requirements but should be part of every archiving situation. In addition, these activities can be architected to create diagnostic metadata, eventually leading to business process reengineering to improve data quality in the source systems over time, according to Kimball & Ross (2013).

2.5. OPTIMIZATION AND SCALABILITY

Presthus & Sæthre (2015) studied the scalability of a BI system in an organisation and it was found that scaling would affect in attracting diferent partners and new solutions due to majority of the organisations that had employed BI tools indicated that their solutions scaled well. The ease of working with fewer vendors is one explanation. On the other hand, it’s believed that using tools from

several vendors can reduce the risk of lock-in[6]. Lock-in refers to the idea that the switching cost of changing vendor would be too expensive or time-consuming. None of the organisation studied raised concern about lock-in, but it was discovered another reason for the many complex BI solutions with multiple vendors: heritage.

Boateng, Singh, Greeshma, & Singh (2012) has studied and addressed the major issues regarding scability:

- The amount of data in the warehouse
- How quickly the warehouse is expected to grow
- The number of concurrent users
- The complexity of user queries
- Good scalability means that queries and other data access functions will grow linearly with the extent of the data warehouse

Dimensional models are extremely scalable. Fact tables often have a large number of rows; fact tables holding two trillion rows have been stated. The database vendors have enthusiastically embraced data warehouse and continue to integrate competences into their products to enhance dimensional models' scalability and performance (Kimball & Ross, 2013).

Additionally, Kimball & Ross (2013) states that simplicity is the key to ensuring that a business used is able to understand the data, allowing a quick and efficient software navigation and delivering results quickly.

2.6. MANAGEMENT PROCESSES

Early studies on critical success factors of data warehouse projects have already emphasised the importance of proper management of BI and holistic concepts of BI maturity also include BI management as a critical dimension. From a theoretical perspective, BI management capabilities can be interpreted as a reflection of resources and learning processes required to combine BI software and organizational strategy into BI solutions, and to ensure the on-going achievement of the objectives associated with the BI process. BI software products, on the other hand, are assets which are readily available in factor markets. Similarly, BI software implementation services can be purchased and the on-going maintenance of BI solutions can also be outsourced. But successful management of BI also requires a close alignment of IT and business throughout the whole BI solution life cycle, in particular matching decisions and requiring information, asking the right questions, gaining and maintaining top management support and championship, and end-user 'buy-in', etc. Providing and maintaining a BI solution in support of "effective problem and opportunity identification, critical decision-making, and strategy formulation, implementation, and evaluation" cannot be fully outsourced, but rather requires internal resources beyond the IT department. Only if

[6] Vendor lock-in is usually the result of proprietary technologies that are incompatible with those of competitors - http://searchcloudapplications.techtarget.com/definition/vendor-lock-in

IT resources and business requirements are aligned through proper management of BI, organisations can realize the potential benefits of BI applications (Wieder & Ossimitz, 2015).

Wieder & Ossimitz (2015) also predict associations between BI management quality, information quality and data quality as may be found below:

- BI management quality is positively related to the quality of managerial decision making
- Information quality is positively related to the quality of managerial decision making
- Data quality is positively related to information quality
- BI management quality is positively related to data quality
- BI management quality is positively related to information quality

Couture (2013) refers to data quality as one of the keys to be truly successful, and writes that there are many dimensions of data quality that can be addressed as part of a data quality assessment program. Data quality itself can be defined as "fitness for use," a very broad definition that entails many aspects of quality at the enterprise level. Also according to Couture (2013) there are four basic dimensions that can be expanded upon over time:

- Completeness - Source-to-target validation; Monitored and reported
- Timeliness – Defined Service Level Agreements[7] (SLAs); Reviewed and approved; Monitored and reported
- Validity – Data profiling[8]; Data cleansing[9]; Inline data quality checks; Monitored and reported
- Consistency – Inline data quality; Trended; Monitored and reported

One of the key deliverables in Business Intelligence is providing consistent, comprehensive, clean, conformed and current information for business people to enable analysis and decision making and to deliver this is not achieved by BI Tools simply accessing unrelated dater, but rather though data integration (Sherman, 2014).

Sherman (2014) states that three quarters of the BI project's time is devoted to data integration when new data sources are added to the data architecture and that it is critical to adopt best practices to design robust, scalable, and cost-effective data integration processes.

[7] A service-level agreement (SLA) is a contract between a service provider and its internal or external customers that documents what services the provider will furnish - http://searchitchannel.techtarget.com/definition/service-level-agreement

[8] Data Profiling is a systematic analysis of the content of a data source - http://datasourceconsulting.com/data-profiling/

[9] Data cleansing is the process of analyzing the quality of data in a data source, manually approving/rejecting the suggestions by the system, and thereby making changes to the data - https://msdn.microsoft.com/en-us/library/gg524800.aspx

Data integration processes should be, according to Sherman (2014):

- Holistic – avoid costly overlaps and inconsistencies
- Incremental – more manageable and practical
- Iterative – discover and learn from each individual project
- Reusable – ensure consistency
- Documented – identify data for reuse, and create leverage for future projects
- Auditable – necessary for government regulations and industry standards.

Kimball & Ross (2013) defined that a foundation of main descriptive followed dimensions involves effort, but after it's agreed upon, following data warehouse efforts can influence the work, both ensuring reliability and reducing the implementation's delivery cycle time. Come to agreement on data classifications, labels and domain values is one of the key functions of data governance. Additional key functions are to create policies and responsibilities for data quality and accurateness, as well as data security and access controls.

Kimball & Ross (2013) also adressed that there was often little effort to ensure consistent common reference data and a strong data governance function is a necessary prerequisite for conforming information regardless of technical approach.

Data Warehouse Administrator[10] (DWA) is responsible for the administration and management of a data warehouse and an effective security in a data warehouse should focus on four main areas, as reported by Boateng, Singh, Greeshma & Singh (2012):

- Founding in effect corporate and security policies and procedures
- Applying logical security measures and techniques to confine access
- Preventive physical access to the data environment
- Instituting an effective inner control evaluation process highlighting on security and privacy

Following are the issues to consider by Boateng, Singh, Greeshma, & Singh (2012) to build a successful data warehouse:

- Delivering data with overlapping and confusing definitions
- Believing promises of performance, capacity, and scalability
- Believing that your problems are over when the data warehouse is up and running
- Focusing on ad hoc data mining and periodic reporting instead of alerts

[10] Data Warehouse Administration involves the overall management of a data warehouse. Administration tasks - http://it.toolbox.com/wiki/index.php/Data_Warehouse_Administration

According to Horakova & Skalska (2013) studies, the BI tools are more and more often focused on Corporate Performance Management[11] (CPM). CPM deals with managing and monitoring of general business efficiency. Performance indicators are usually monitored both at the corporate level and at the level of single department or division. According to this methodology, CPM determines metrics for verification of business efficiency development. Information technology tools, and especially tools from BI area, can support practical realization of CPM.

2.7. INFRASTRUCTURE

Chaudhuri, Dayal, & Narasayya (2011) writes that today is difficult to find a successful enterprise that has not leveraged Business Intelligence technology for their business and one of the reasons for it is that the cost of data acquisition and data storage has declined significantly. This has increased the appetite of businesses to acquire very large volumes in order to extract as much competitive advantage from it as possible. Obeidat, North, Richardson, & Rattanak (2015) confirms that as computer technology advances, larger volume of data are acquired and stored at much lower cost. Any classification of transaction in business, including e-business, radio-frequency identification (RFID) tags, Web sites, emails, blogs, and many more produces new data to be tracked. Cloud virtualization permits virtual servers to be hosted in the cloud, eventually providing much lower cost of hardware and software, while ensuring better utilization of resources. As organisations are moving in the direction of cloud based offerings for increased scalability and flexibility with lower costs, this seems like a great strategy. Nevertheless, with business intelligence applications, businesses sometimes have delicate data that cannot be wholly outsourced to a cloud environment. The scenarios gradually show how more specialized data or the movement of a BI application can trigger events in other systems indicating targeted applications should be moved by following a confined and cloud deployment model rather than an all-or-nothing with cloud infrastructures only.

Advantages	Disadvantages
Increased Elastic Computing Power	Security Risks
Potential Cost Savings	Slow Data Breach Recovery
Easy Deployment	Cloud BI Availability Is Determined By External Factors
Supportive of Nomadic Computing	Potential Compromise of Core BI Capabilities
NA	Costs Are Difficult To Quantify
NA	Changing and Controversial Regulatory Environment

[11] Corporate performance management involves monitoring and managing an organization's performance, according to key performance indicators - http://searchdatamanagement.techtarget.com/definition/corporate-performance-management

Table 2.3 – Benefits and Risks of using a Business Intelligence cloud-based solution (Tamer, Kiley, Ashrafi, & Kuilboer, 2013)

The Table 2.3 displays the numerous advantages and disadvantages of Business Intelligence on the Cloud. Starting with the advantages, the "Increased Elastic Computing Power" in which refer to how fast a machine or software can perform an operation. Project sizes vary greatly and the flexibility in computing power is appealing for companies with fluctuating and growing data sources. Another advantage is the "Potential Cost Savings" as the user on the cloud only has to pay for whatever computing power is needed. Computing needs could vary considerably due to seasonal changes in demand or during high-growth phases. "Easy Deployment" is also an advantage in which the cloud makes it easier for a company to adopt a Business Intelligence Solution and quickly experience the value. Finally, the last advantage presented is the "Supportive of Nomadic Computing" that refers to the information systems support that provides computing and communication capabilities and services to users, as they move from place to place allowing employees and BI users to travel without losing access to the tools. Despite the numerous benefits of adopting cloud-based business intelligence, there are many risks. The following risk emphasis on security of a cloud-based solution. Firstly, the "Security Risks" that come cloud computing, data is stored and delivered across the Internet. Since the location of data is unknown and not controlled by the owner of the data, there is a good chance that several competitors' data and applications reside on the same resources environment. When putting data onto an external server and outside of the user's direct control, there's no way avoiding confidentiality risks. Encryption is a viable option, but it is the responsibility of the user to ensure that data is appropriately encrypted on the cloud. Another risk is "Slow Data Breach Recovery" which is caused due to user does not knowing where the data is actually stored and processed making it difficult to respond quickly, remedy the problem, and provide customers. "Cloud BI Availability Is Determined by External Factors" risk is important because using the Business Intelligence tools is relying on the third party's server availably which is a bit of gambling on the control of its data. Another disadvantage is the "Potential Compromise of Core BI Capabilities" in which the traditional BI solutions offer full control and high-touch data integration, a capability crucial to defining a successful and robust BI solution. On the other side, the cloud presents the potential for compromised data, metadata, and application integration. One more risk is the "Costs Are Difficult to Quantify" which even though a cost benefit analyses for business intelligence is already difficult to do, even more so with cloud solutions. The last risk is "Changing and Controversial Regulatory Environment" which by using the cloud to store and compute data complicates regulation, as there is increased likelihood of cross-border data storage and access (Tamer, Kiley, Ashrafi, & Kuilboer, 2013).

Cloud computing promises significant benefits, but today there are security and several other barriers that prevent widespread enterprise adoption of an external cloud. In addition, the cost benefits for large enterprises have not yet been clearly demonstrated. A recent study shows that 71% of the organizations consider Cloud Computing a realistic technological option, 70% believe that it would lead to increased business flexibility, 62% consider that it would speed up response to market conditions, and 65% consider that it would lead to increased focus on the main aspects of business (Tamer, Kiley, Ashrafi, & Kuilboer, 2013).

2.8. REPORTING TOOLS

Nowadays, technology and major vendors are driving data visualization within the Business Intelligence Dashboard (Wakeling, Clough, Wyper, & Balmain, 2015). However, others defend that in order for business Intelligence to be successful it needs to shift the primary focus from technology (machines assisting) to human being capable on knowledge that a business should rely on to succeed (Few, 2007).

Wakeling, Clough, Wyper and Balmain (2015) proposes that at least one delivery process should be dependent on the user throughout the development of a dashboard solution, mainly due to the users' ability to interpret and comprehend which data is relevant to them.

There is a large number of business intelligence reporting software's available and Table 2.4 presents two of these business intelligence software solutions.

BI Software	Qlikview	Tableau
Key Features	Hybrid platform	Hybrid platform
	Data collection	Data collection
	3rd-party data integration	3rd-party data integration
	Customizable dashboards	Customizable dashboards
	Data visualization	Data visualization
	Ad hoc analytics & reports	Ad hoc analytics & reports
	Self-service	Self-service
	Mobile accessibility	Mobile accessibility
Additional Features	Consolidates data from multiple sources into a single application	Flexible data architecture allows users to connect live to their data source, or extract all or a portion of their data into memory
	Data visualizations with state-of-the-art graphics	Scalable for hardware and memory
	Interactive apps, dashboards and analytics	Automatic updates
	Easily create and manage data definitions and transformations	Embed, share, comment on and subscribe to interactive dashboards

Table 2.4 – Example of two Business Intelligence Software Solutions (Business-software, 2016)

A recent player in the business intelligence world, Microsoft PowerPivot, which since Excel 2013 started to be an inherent part of the Excel technology, stopped having a limited number of rows in storage and also started to be able to compress large amounts of information into small workbooks.

This has evolved significantly the era of self-service business intelligence (Ferrari & Russo, 2014). Microsoft PowerPivot data and Excel presentation objects are contained within the same workbook file, meaning that Excel features to aggregate and interact with data like PivotTables, PivotCharts are instantaneously available (Microsoft, 2012).

2.9. TRENDS

2.9.1. Cloud BI

Obeidat, North, Richardson, & Rattanak (2015) states that as many investigation challenges endure in BI, numerous new open research challenges appear on horizon for recent technologies, such as Cloud Computing. The traditional BI standpoint is focused on Extract, Transform, and Load (ETL), and reporting, while the new BI seems to be more focused on data exploration and visualization. As databases are collecting more and more data, the traditional navigating techniques become inefficient and ineffective, while data exploration and visualization techniques may contribute in a much sophisticated understanding of big data. Furthermore, the authors highlight the importance of providing environments to help users in progressively challenging tasks. Universally, data exploration and visualization techniques deliver an important nexus for the several business organisations to explore, understand and achieve valuable insights to operate and compete globally. Obeidat, North, Richardson, & Rattanak (2015) illustrates how to sustain self-service BI. Firstly, with the definition of self-service reporting, by describing the approaches for operational decision makers, analysts, and knowledge workers to access data required to back decisions and actions to promote business accomplishment. BI software merchants and industry experts distinguish self-service as a key component to eliminate problems to timely insight, decision making, as well as lowering the cost of reporting, analysis, and metrics-driven management by placing data in the proper hands.

Horakova & Skalska (2013) adresses that web technologies are still very popular due to broad availability. Most companies have built intranet or extranet for sharing information between employees and business partners. Web 2.0 conceptions include space for easy creation and maintenance of web content. Very simple is also sharing of web content with others. In connection with BI, web user can online maintain a set of analytical reports in intranet, share this content among other interested users, put comments to interesting indicators or generate dynamically ad-hoc queries directly from web application. The main advantage of this technology is ease of use and fast distribution of analytic outputs.

Smart phones enable to run a lot of applications that were available only on personal computers in the past. It is possible to integrate analytics output with mobile access to applications. The existing BI reports and dashboards will be progressively moved to the mobile platform. Then also more vendors of mobile application for specific BI tasks should appear on the market (Horakova & Skalska, 2013).

2.9.2. Big Data

Data is being collected at an unprecedented scale (Jagadish, et al., 2014). In line with Moorthy, et al. (2015), the term Big Data is to a large extent vague and amorphous. Information technology

professionals look at Big Data as large data sets that require supercomputers to collate, process and analyse to draw meaningful conclusions. Big Data refers to the explosion in the quantity (and sometimes, quality) of available and potentially relevant data, largely the result of recent and unprecedented advancements in data recording and storage technology. In a broad view, Big Data is applied in Business Intelligence for handling a massive amount of digital data which is being collected from all sorts of sources, where sometimes may be too large, raw, or unstructured for analysis through conventional relational database techniques (Kim, Trimi, & Chung, 2014).

Creating value from Big Data is a multistep process, according with Jagadish, et al. (2014):

- Data acquisition
- Information extraction and cleaning
- Data integration, aggregation and representation
- Modelling and analysis
- Interpretation

Moorthy, et al. (2015) stated that Big Data became a reality with the development of certain computing technologies and provided simple explanations of these:

- Hadoop is a disruptive Big Data technology originally initiated by Yahoo to build an advanced search engine and process the generated data. Hadoop has evolved into a large-scale data processing environment.
- Google's Bigtables is a very large distributed storage system for structured data. It can manage petabyte of data on more than thousands of servers. It is a distributed, persistent, sparse, persistent, multidimensional sorted map. The map is indexed by a new row key, a column key, a timestamp and each value in the map is treated as un-interpreted array of bytes.
- HBase started with an objective of strong large tables extending to billions of rows and millions of columns. It is an open source non-relational database designed based on Google's Bigtables.
- HIVE is a data warehouse which facilitates querying and manages large volume data sets residing in the distributed storages. It is built on top of Hadoop to provide easy ETL tools, structuring different data formats. Hive uses a simple query language called QL similar to SQL (Structured Query Language).
- CASSANDRA is a scalable database.
- PIG is a high level data flow language. It can be used for expressing data analytical programmes along with capability for evaluating these programmes. It is capable of substantial parallelization for handling very high volumes of data. PIG uses PigLatin a text based language.

- ZOOKEEPER is a centralized software coordination service with a single interface. It documents, tracks configuration information, naming, distributed synchronization and group services.
- YARN (Yet Another Resource Negotiator) is a more distributed and faster Architecture.
- METADATA refers to 'data about data'. Structural metadata contains data about database design, specification of data structure, containers of data. Descriptive metadata contains data about individual instances of data application and control.
- NoSQL (Not Only SQL) is a database environment which is non-relational distributed database system. It provides ease in speedy organization of data analysis with high volume of data, with desperate data types. Sometimes it is also referred to as Cloud database, non-relational database.
- IoT (Internet of Things) is interconnected uniquely identifiable embedded devices and software. Things can be a wide range of devices that are implants into the living organisms, or embedded in small and large machines such as cars, turbines, sensors in buoys, etc. Information generated from such interconnected devices can be used for large-scale efficient automation.

Jagadish, et al. (2014) writes about the research challenges of big data, ranging from heterogeneity of data, inconsistency and incompleteness, timeliness, privacy, visualization, and collaboration, to the tools ecosystem around Big Data. While the potential benefits of Big Data are real and significant, and some initial successes have already been achieved, there remain many technical challenges that must be addressed to fully realize this potential.

2.9.3. In Memory Analytics

The new class of in-memory BI tools turns a BI solution into an agile BI solution. In the last years, emerging technologies such as interactive visualization, in-memory analytics and associative search marginalized IT role in building BI solutions (Muntean, 2014).

In-memory technology has the potential to help BI systems to become more agile, more flexible and more responsive to changing business requirements. The primary goal of the in-memory BI technology is to replace traditional disk-based BI solutions. The important differences between them are: speed, volume, persistence and price (Muntean, 2014).

In-memory database systems allow for high-speed processing and analysis of large data volumes by storing data directly in main memory, avoiding time consuming hard disk operations (Hahn & Packowski, 2015). Also, most of them can save significant development time by eliminating the need for aggregates and designing of cubes and star schemas (Muntean, 2014).

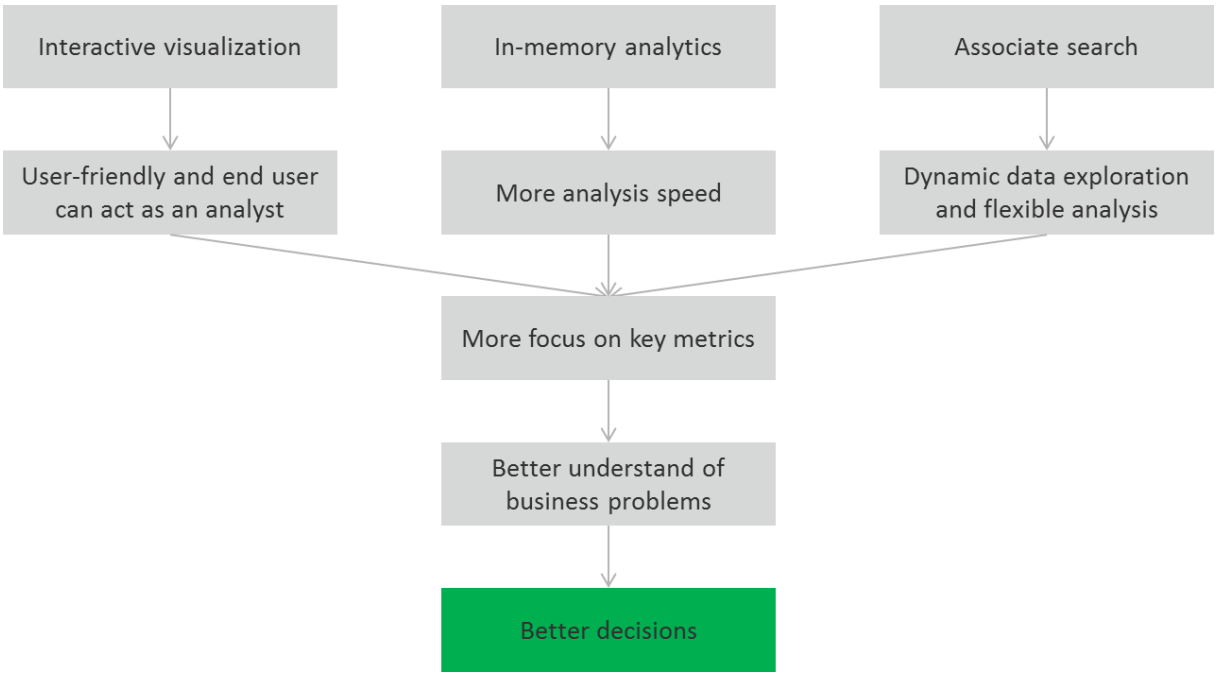

Figure 2.7 – How in-memory analytics, interactive visualization and associative search affect businesses

There are many and different in-memory BI solutions (Muntean, 2014). The Table 2.5 shows the different solutions and its characteristics.

Solution	Characteristics
In-memory OLAP	MOLAP cube and data are all in memory
In-memory ROLAP	only ROLAP metadata loaded in memory although some software can build complete cubes from the subset of data held entirely in memory
in-memory columnar database with data compressions techniques	load and store data in a columnar database
In memory spreadsheet	spreadsheet loaded into memory
In memory “associative” data model Column based storage with compression techniques (with compression ratio near 10:1)	loads and store all data in an “associative” data model that runs in memory; all joins and calculations are made in real time; less modeling required than an OLAP based solution;
Hybrid approach/ dual format approach with data compression techniques	Relational database +columnar database; Both formats are simultaneously active;

Hybrid storage Solution (disk + RAM)	Multidimensional model (traditional OLAP Cube) organizes summary data into multidimensional structures; Aggregations are stored in the multidimensional structure; Tabular model

Table 2.5 – Characteristics of In-Memory BI Solutions (Muntean, 2014)

2.9.4. Predictions

Van der Meulen & Rivera (2015) predicts that almost all data discovery tools are going to have smart data discovery capabilities to be able to increase interactive analysis Influence, by 2017. Nowadays with data discovery capabilities becoming smarter and smarter to streamline pattern detection in data discovery, self-service data preparation capabilities are developing and becoming further capable of semi-automating and augmenting the data preparation activity of data discovery, making it available, for example, to a business analyst. The two developments in combination will produce a next-generation data discovery user understanding that makes advanced types of analysis accessible to a broader range of users.

Another prediction by Van der Meulen & Rivera (2015) is that in 2016 less ten percent of self-service business intelligence initiatives will be governed sufficiently to prevent inconsistencies that adversely affect the business. End-user clamour for access to business data, combined with IT inability to satisfy this need, has manifested in self-service BI initiatives in many organizations. The growing increase in data volume, velocity and, especially, variety has further fuelled this trend. Vendors have responded with mass consumable, broadly deployable, easy-to-use and, often, cloud-based technologies for basic query, analysis and reporting. Often, these solutions are implemented by business units that have circumvented IT and as a result, they are disposed to analytic sprawl — an inconsistent or incomplete use of data, capricious development of metrics and formulae, and either too-restrained or unrestrained sharing of results. To counter these adverse effects, a return to more controlled enterprise BI implementations is expected, or the deployment of self-service BI technologies within a better governed, IT-led project environment. On the technology front, vendors will to continue to play both sides, but more conscientiously — selling simple data discovery technologies broadly throughout their prospects' businesses, while reemphasizing the advantages of controlled, centralized and more-robust enterprise BI technologies.

2.10. DEVELOPMENT METHODOLOGIES

2.10.1. Waterfall

The traditional Waterfall methodology is a downward flowing model that only proceeds to the next stage when the stage before is 100% finished (Mahadevan, Kettinger, & Meservy, 2015). It also has separate linear plan for risk mitigation, resourcing budge and other critical project functions (Grech, 2015).

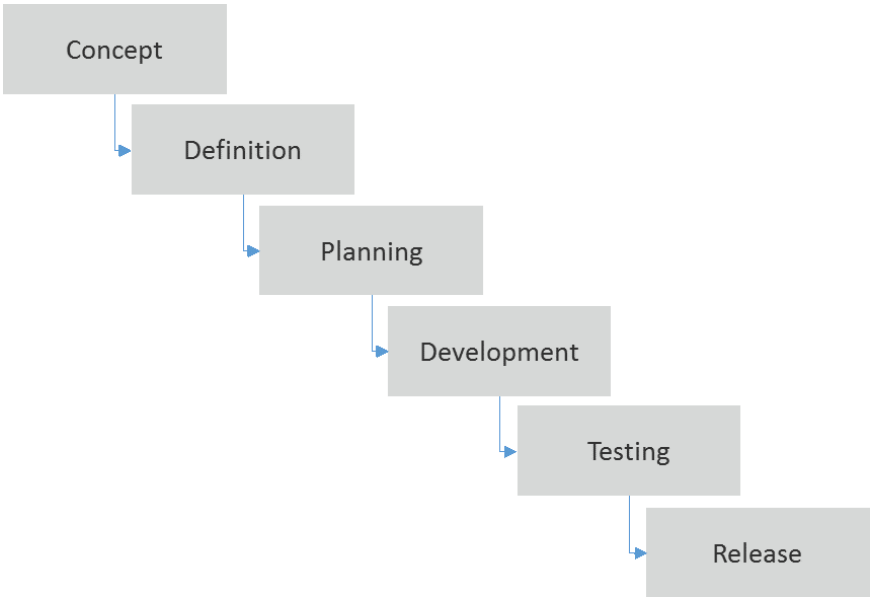

Figure 2.8 – Example of Waterfall Methodology

The traditional waterfall methodology is focused on milestones, meaning that the project scope will be divided into end-to-end features. A topic that must very simple and clear is the control processes for allowing adaptability to the project and in order to be effective it needs to have outlined key approvers who have deep understanding on the project. These people are the decision-makers. Throughout all project using waterfall methodology every team should have a clear idea of the project organizational chart, who has what role and who is accountable for what (Grech, 2015).

2.10.2. Agile

The development market condition of today's worldwide and dynamic environment requires more and more flexible services and additional frequent changing business systems. To better accommodate these markets, need the agile methodology practices and techniques were suggested so that teams can better react and respond to changing dynamics more effectively and seamlessly (Lee & Baby, 2013). The agile methodology promises to deliver increased efficiency, quality and project success percentage over all development projects (Ionel, 2009). It is often referred to as a lightweight approach to project management, this happens because they are in direct contrast to the traditional long-term, plan-driven, document-heavy, bureaucratic approach to managing software development (Milanov & Njegus, 2012).

The idea of agile the traditional methodologies ended up in a retreat, in 2001, where the main innovators and similar approaches authors wrote the Agile Manifest.

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

• **Individuals and interactions** over processes and tools
• **Working software** over comprehensive documentation
• **Customer collaboration** over contract negotiation
• **Responding to change** over following a plan
That is, while there is value in the items on the right, we value the items on the left more
Kent Beck — James Grenning — Robert C. Martin
Mike Beedle — Jim Highsmith — Steve Mellor
Arie van Bennekum — Andrew Hunt — Ken Schwaber
Alistair Cockburn — Ron Jeffries — Jeff Sutherland
Ward Cunningham — Jon Kern — Dave Thomas
Martin Fowler — Brian Marick

Table 2.6 – Manifesto for Agile Software Development (Beedle, et al., 2001)

The agile manifesto has twelve integrated principles on how the agile methodologies should be placed in practice. The Table 2.7 displays those principles as written originally. Below is possible to find the primary focus of the manifesto (Milanov & Njegus, 2012):

- Customer value
- Iterative and incremental delivery
- Intense collaboration
- Small integrated teams
- Self-organization
- Small and continuous improvements

No	Principles
1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
4	Business people and developers must work together daily throughout the project
5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation
7	Working software is the primary measure of progress

8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
9	Continuous attention to technical excellence and good design enhances agility
10	Simplicity--the art of maximizing the amount of work not done--is essential
11	The best architectures, requirements, and designs emerge from self-organizing teams
12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly

Table 2.7 – Twelve Principles behind the Agile Manifesto (Beedle, et al., 2001)

Even though agile development practices have received significant interest from many practitioners (Chong-Leng Goh, Pan, & Zuo, 2013) due to emphasizing simplicity (Ionel, 2009) and authors are outlining the advantages of using the agile methodology, emphasising on individuals and interaction over processes or responsiveness over rigid planning, there are other authors that support the contention of using agile methods may improve the likelihood of project success (Serrador & Pinto, 2015).

There are also suggestions that the two main assumptions between agile and traditional methodologies are the following (Ionel, 2009):

- Traditional methodologies assume that customers' capability to foresee their future requirements is limited, meaning that the developers will need to build in extra functionalities to meet these future needs, frequently leading to overdesigned system.
- Agile methodologies assume that customers and developers together don't have a complete understanding of requirements when the project starts while traditional methodologies assume that customers don't know their requirements, hence they need guidance from the developers. Consequently, agile methodologies customers and developers need to learn together about the system requirements as the development process evolves whereas in traditional software development environments, developers require a detailed specification.

Despite waterfall methodology approach grew to be a dominant software-development methodology in many large companies the Agile software development method argue the Waterfall method is flawed because it is almost impossible for any non-trivial project to finish a system development lifecycle (SDLC) phase completely as pre-specified (Mahadevan, Kettinger, & Meservy, 2015).

3. METHODOLOGY

The implementation methodology for this project is based on four major relevant topics:

- Planning – Draft of the project plan
- Analysis – chosen data warehouse design, project methodology and software
- Implementation – contains the construction of the Data Warehouse, the data loading in the DW selected tool and also the reporting which will show the output of it
- Reporting Design – reporting using PowerPivot, analysis of the reports and discussion of the end results

The data required for this project implementation belongs to the Academic Services of Nova Information Management School. The advisors for this master thesis project have cleaned the data and after that, access was provided to start the project.

3.1. PLANNING

This projects' plan was elaborated with a Gantt chart and all tasks are displayed below.

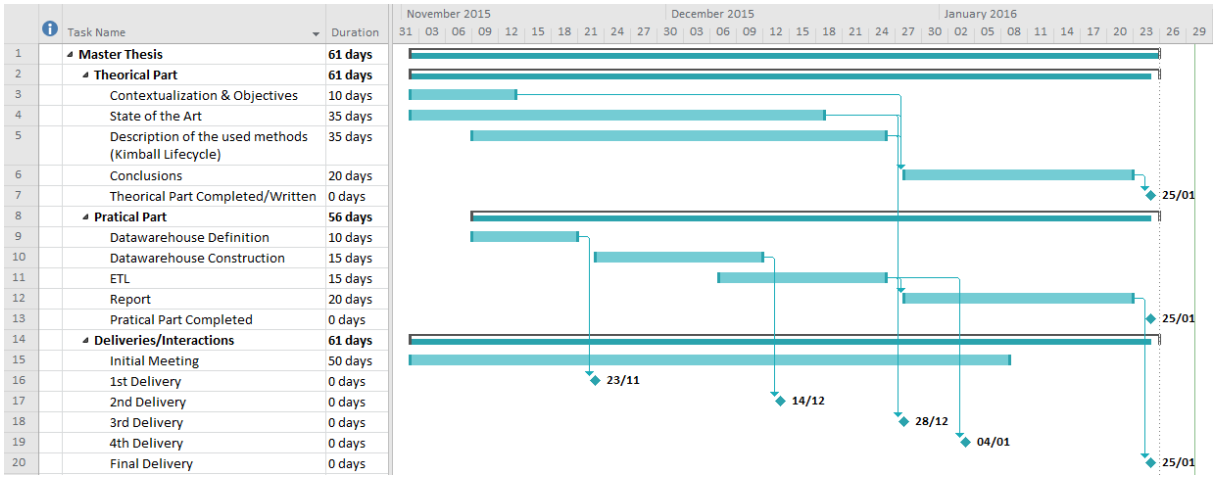

Figure 3.1 – Master Thesis project Gantt chart

Figure 3.1 presents the chronogram for a total period of sixty-one work days, which corresponds to three months.

Even though the official start date for this project was September 2014, it was only started in November 2015. The first version of the project was delivered in the end of January of 2016 and the

second and last version in the end of February 2016. This last month was used to update and adapt according to the master thesis advisors and to end the reporting and conclusion chapter.

3.2. ANALYSIS

The chapter's purpose is to present the chosen concepts and practices for the development of this project.

The main cause to apply a Business Intelligence system to NOVA IMS Academic Services is to explore relevant data that may be hidden and to provide more knowledge on the data that it is currently being extracted. This is something that schools are not taking advantage of and by presenting in the form of a report it may help in the decision making process.

An important factor for the development of this project was the help of the Master Thesis Advisors to provide information and guidance for every step necessary and to be the connection between the student and the academic services.

The project followed Kimball Data Warehouse design, meaning a star schema was the selected schema to build the projects' data warehouse. The reasons to choose the star schema are the following:

- Most common schema
- Lowest maintenance needed as it easier to adapt to change and to add more fields in the future
- Easier to understand and to access (less joins and hierarchical levels), since it is more simple and presents a smaller number of relationships

The data warehouse design, which includes every table, metric and fields is described in detail in section 3.3.2.

3.2.1. Project Methodology

Initially, it was intended for this Master Thesis Project to be developed under a waterfall methodology approach as the high level plan was to gather all relevant information, build the data warehouse and present with reports.

After starting out the project, it was only possible to pursue using an agile methodology. The main reason for this change was due to the initial lack of communication between stages. Also, by following an agile methodology it was possible to bring more value to the final user, an iterative and incremental delivery with a more intense collaboration.

3.2.2. Software

The project can be divided in two major parts, the Data Warehouse creation and the Reporting creation.

The tools for the Data Warehouse creation of this project will be the "SQL Server Management Studio" and the "SQL Server Data Tools", as these Microsoft tools are being used by Nova Information Management School Informatics' Department, Academic Services and Faculty and it wouldn't require any additional licenses.

For the Reporting creation, the tool chosen is Microsoft PowerPivot for Excel 2016, not only due to the lack of need of any additional license by Nova Information Management School but because is a reporting tool that doesn't require much training since it uses Microsoft Excel as a base.

3.3. IMPLEMENTATION

3.3.1. Source Database

This project involves using different database as sources. In order to retrieve all the necessary data SQL Server Database and Microsoft Office Excel documents were used.

The main source to retrieve data for the Data Warehouse designed is a SQL Server Database named "IsegiOnline_Source" that was provided by NOVA IMS Academic Services and the advisors of this project. The database has a total of twenty-one tables and to design the Data Warehouse is being retrieved information from thirteen.

Regarding Microsoft Office Excel documents, a total of six documents were used for different tables. The reasons behind the need to use Microsoft Office Excel documents will be explained more in detail in each sub-section of section 3.3.3. Nevertheless, they were used to add more information that was afterwards provided by the NOVA IMS Academic Services or to create Date/ Time dimensions.

3.3.2. Data Warehouse Design

The adopted Data Warehouse Design for the development of this project is the Star Schema, which consists on large central tables called the fact tables, and a number of smaller tables called dimension table.

The final Data Warehouse has four fact tables and eleven dimension tables and the Star Schema Diagram may be found split between four figures below (Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5), each corresponding to a fact less factual table. Most dimensions are shared between the different fact tables.

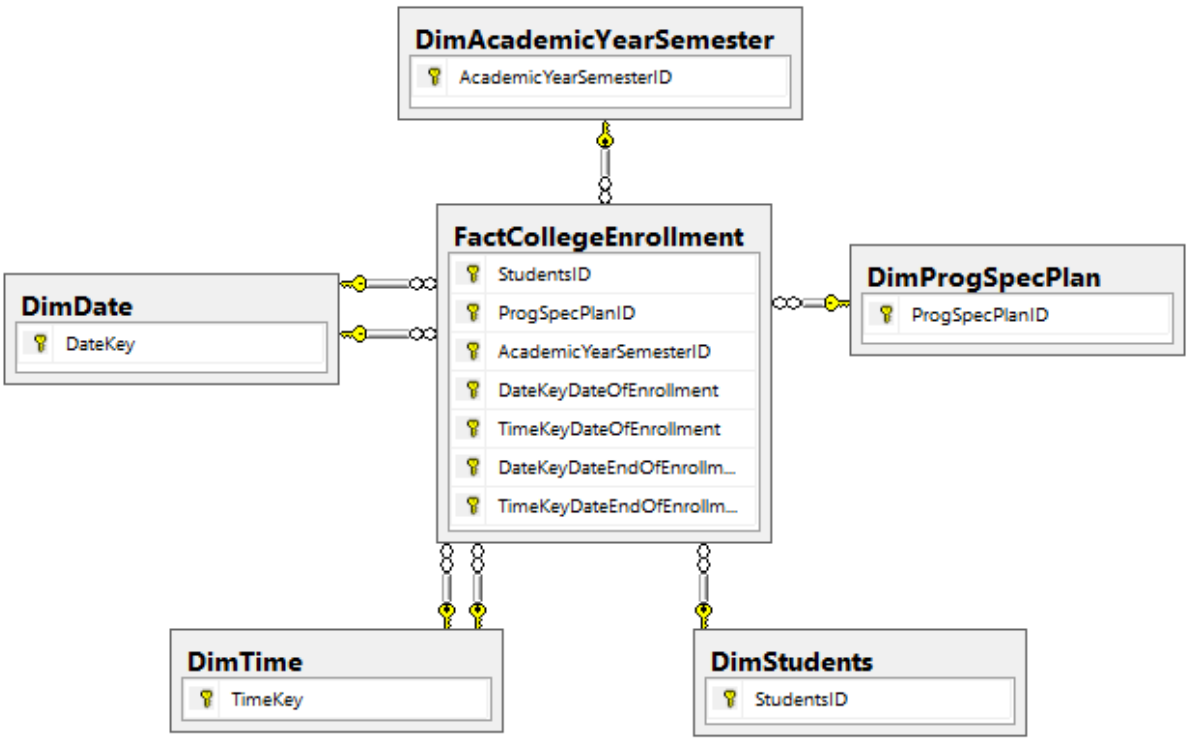

Figure 3.2 – Star schema diagram for Fact College Enrolment

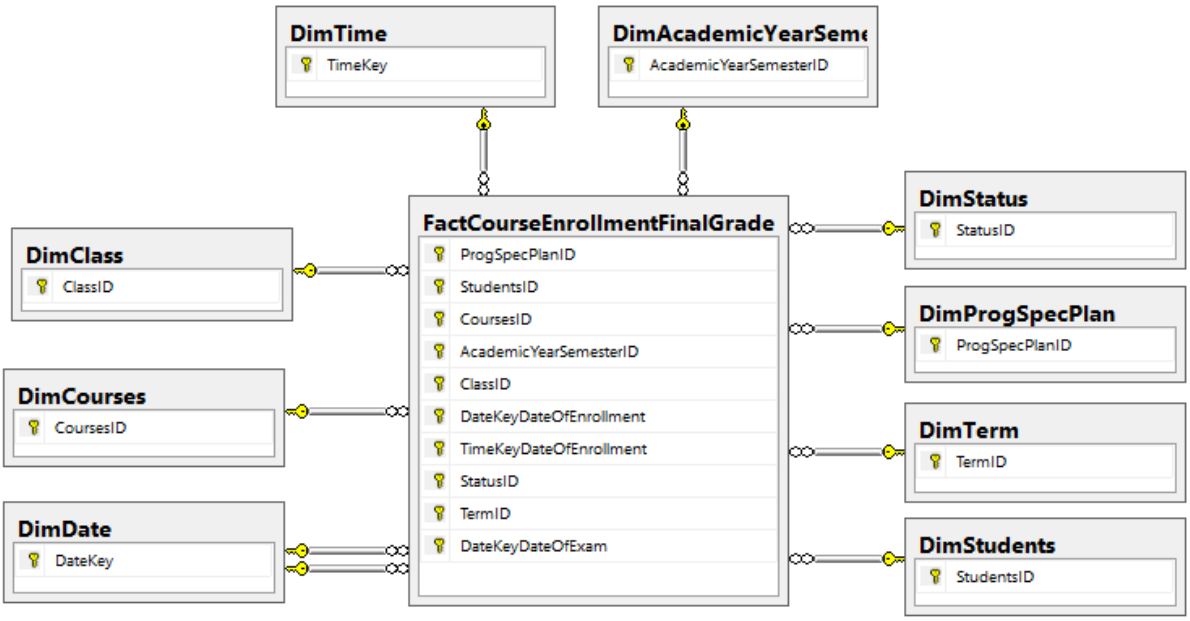

Figure 3.3 – Star schema diagram for Fact Course Enrolment Final Grade

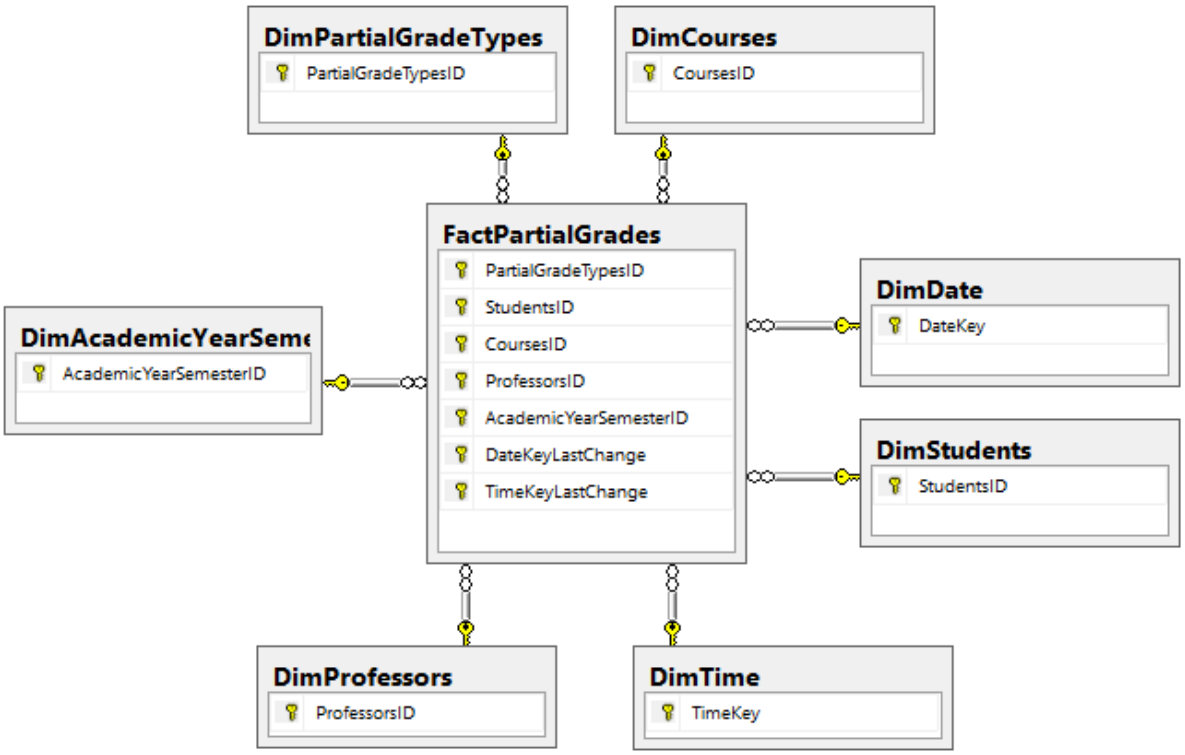

Figure 3.4 – Star schema diagram for Fact Partial Grades

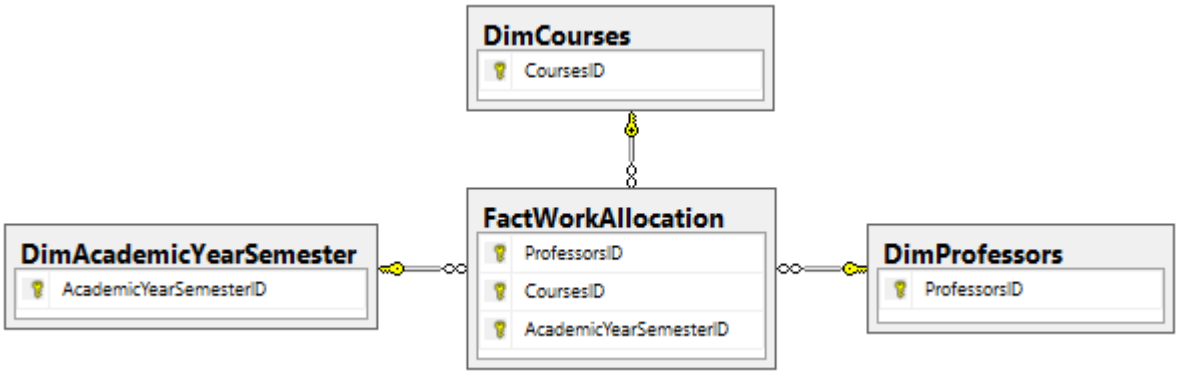

Figure 3.5 – Star schema diagram for Fact Work Allocation

The following sub-sections explain in detail each factual and dimensional table that was created for the Data Warehouse.

3.3.2.1. Dimension Academic Year Semester

The table “DimAcademicYearSemester” contains details of the academic years and the periodicity of the academic year. The academic year in the dimension is formatted as a four digit for the first year

and as a two digit for the second year with both concatenated (i.e. 199899, 200506). The table below displays the fields, the type and the fields' description:

Field	Type	Description
AcademicYearSemesterID	Integer	Primary Key/ Surrogate Key/ Identity Column
AcademicYearPeriodicityChar	Varchar (7)	Natural Key in Character (to connect to two fact tables)
AcademicYearPeriodicityInt	Integer	Natural Key in Integer (to connect to two fact tables)
AcademicYear	Varchar (6)	Academic year formatted as: • Four digit for the first year • Two digit for the second year Both are concatenated (i.e. 199596, 201011)
Periodicity	Varchar (1)	The periodicity is the semester. The following value correspond to each type of periodicity: Null – No periodicity (when only using the year) A or 0 – Annual periodicity 1 – 1st Semester 2 – 2nd Semester 3 – Special Epoch

Table 3.1 – Dimension Academic Year Semester details

3.3.2.2. Dimension Class

The table "DimClass" contains details about each instance of a class type. For example, the dimension includes information such as the type of class description in English and in Portuguese. The table below displays the fields, the type and the fields' description:

Field	Type	Description
ClassID	Integer	Primary Key/ Surrogate Key/ Identity Column
CD_CLASS	Tinyint	Natural Key
Class_EN	Nchar (100)	Class description in English

Class_PT	Nchar (100)	Class description in Portuguese

Table 3.2 – Dimension Class details

3.3.2.3. Dimension Courses

The table "DimCourses" contains details about each course type. For example, the dimension includes information such as the course name. The table below displays the fields, the type and the fields' description:

Field	Type	Description
CoursesID	Integer	Primary Key/ Surrogate Key/ Identity Column
CD_COURSE	Decimal (9, 0)	Natural Key
NM_COURSE	Varchar (200)	Course name

Table 3.3 – Dimension Courses details

3.3.2.4. Dimension Date

The table "DimDate" contains details about the date, for example, it includes information such as the full date, day of week and it is a general dimension that should exist in every Data Warehouse. The table below displays the fields, the type and the fields' description:

Field	Type	Description
DateKey	Integer	Primary Key/ Natural Key
FullDate	Date	Full date (date type) separated by '- '
DateName	Nchar (11)	Full date separated by '/ '
DayOfWeek	Tinyint	Day of the week. Below is the day that corresponds to each day of week: 1 – Sunday 2 – Monday 3 – Tuesday 4 – Wednesday 5 – Thursday 6 – Friday 7 – Saturday

DayNameOfWeek	Nchar (10)	Name of the day in a week (i.e. “Monday”, “Wednesday”)
DayOfMonth	Tinyint	Day of the month. May be a number between 1-31
DayOfYear	Smallint	Day of the month. May be a number between 1-366
WeekdayWeekend	Nchar (10)	Classifies the day as “Weekday” or “Weekend”
WeekOfYear	Tinyint	Number of the week in a year
MonthName	Nchar (10)	Month name
MonthOfYear	Tinyint	Number of the month in a year
IsLastDayOfMonth	Nchar (1)	Classifies the day if it is the last day of the month (“Y”) or not (“N”)
CalendarQuarter	Tinyint	Quarter number in a year
CalendarYear	Smallint	Year number
CalendarYearMonth	Nchar (10)	Year and month number separated by a “-” (i.e. “1990-01”)
CalendarYearQtr	Nchar (10)	Year number and quarter separated by a “-” (i.e. “1990-Q1”)

Table 3.4 – Dimension Date details

3.3.2.5. Dimension Partial Grade Types

The table “DimPartialGradeType” contains details about each partial grade type. For example, the dimension includes information such as the evaluation name and the type of a partial grade description in English and in Portuguese. The table below displays the fields, the type and the fields’ description:

Field	Type	Description
PartialGradeTypesID	Integer	Primary Key/ Surrogate Key/ Identity Column
CD_PARTIAL_GRADE	Integer	Natural Key
NM_NAME_OF_EVALUATION	Varchar (100)	Evaluation name (i.e. exam, essay)
NM_PARTIAL_GRADE_TYPE_PT	Varchar (50)	Partial grade type in Portuguese (i.e. “Teste”, “Trabalho de Grupo”)

NM_PARTIAL_GRADE_TYPE_EN	Varchar (50)	Partial grade type in English (i.e. “Test”, “Work Assignment”)
MAX_GRADE	Integer	Maximum grade for the evaluation type
MIN_GRADE	Integer	Minimum grade for the evaluation type

Table 3.5 – Dimension Partial Grade Type details

3.3.2.6. Dimension Professors

The table “DimProfessors” contains details about the teaching staff. For example, the dimension includes information such as the professor address, zip code and degree. The professors’ names and other personal information were left out on purpose. The table below displays the fields, the type and the fields’ description:

Field	Type	Description
ProfessorsID	Integer	Primary Key/ Surrogate Key/ Identity Column
CD_PROFESSOR	Decimal (9, 0)	Natural Key
ADDRESS_PLACE	Nvarchar (100)	Professors’ address
CD_ADDRESS_ZIPCODE	Integer	Professors’ zip code
DEGREE	Nvarchar (500)	Professors’ degree
IsLatest	Bit	Boolean flag (1 or 0) where 1 is the newest/updated record and 0 the oldest record

Table 3.6 – Dimension Professors details

3.3.2.7. Dimension Program/Specialization/Plan

The table “DimProgSpecPlan” contains details about each program/specialization/plan. The dimension includes information such as the names of the different programs, specializations and plans. The table below displays the fields, the type and the fields’ description:

Field	Type	Description
ProgSpecPlanID	Integer	Primary Key/ Surrogate Key/ Identity Column

CD_PROGRAM	Decimal (4, 0)	Natural Key for the Program
CD_SPECIALIZATION	Decimal (4, 0)	Natural Key for the Specialization
CD_PLAN	Decimal (4, 0)	Natural Key for the Plan
NM_PROGRAM_NAME	Nvarchar (240)	Program name
CD_DEGREE_CODE	Nvarchar (1)	Code for each degree type. Below is possible to find what each code corresponds to: 1 – Bachelor 6 – Postgraduate 4 – Master 5 – Doctorate 3 – Master Erasmus NULL – Bachelor Erasmus
NM_ABREV_PROGRAM_NAME	Nvarchar (40)	Program abbreviated name
NM_SPECIALIZATION	Varchar (280)	Specialization name
NM_PLAN	Varchar (280)	Plan name
ProgSpecPlanEffectiveFrom	Date	Date from when the Program Specialization Plan was effective
ProgSpecPlanEffectiveTo	Date	Date until the Program Specialization Plan was effective. If this field is null, then the Program Specialization Plan is currently effective

Table 3.7 – Dimension Program, Specialization and Plan details

3.3.2.8. Dimension Status

The table "DimStatus" contains details about each instance of a status type. The dimension includes information such as the type of status description in English and in Portuguese. The table below displays the fields, the type and the fields' description:

Field	Type	Description
StatusID	Integer	Primary Key/ Surrogate Key/ Identity Column
CD_STATUS	Tinyint	Natural Key
Status_EN	Nchar (100)	Status description in English

Status_PT	Nchar (100)	Status description in Portuguese

Table 3.8 – Dimension Status details

3.3.2.9. Dimension Students

The table "DimStudents" contains details about the school students. The dimension includes information such as the gender, nationality and civil status. The professors' names and other personal information were left out on purpose. The table below displays the fields, the type and the fields' description:

Field	Type	Description
StudentsID	Integer	Primary Key/ Surrogate Key/ Identity Column
CD_STUDENT	Decimal (9, 0)	Natural Key
CD_GENDER	Nvarchar (1)	Students' gender. It may be Men (represented by the letter "M") or Women (represented by the letter "F")
Title	Nvarchar (100)	Students' Title (i.e. Mr, Mrs or Sir)
CivilStatus	Nvarchar (100)	Students' Civil Status
Nationality	Nvarchar (100)	Students' Nationality
Country	Nvarchar (100)	Students' Country
CountryOfBirth	Nvarchar (100)	Students' Country of Birth
FiscalZone	Nvarchar (100)	Students' Fiscal Zone
AcademicBackground	Nvarchar (100)	Students' Academic Background
BIRTHYEAR	Integer	Students' Birth year
CD_TRANEFERINSTITUTION	Decimal (8, 0)	Institution where student has been prior to enrolment
Employer	Nvarchar (100)	Students' employer
Profession	Nvarchar (100)	Students' profession
EntryType	Nvarchar (100)	Students' type of entry
DT_DATEOFENTRY	Smalldatetime	Date of students' entry

NR_ENTRYGRADE	Decimal (6, 2)	Students' entry grade
DT_DATEOFREENTRY	Smalldatetime	Date of students' re-entry
DT_DATEOFFINALGRADE	Nvarchar (10)	Date of students' final grade
NR_FINALGRADE	Decimal (6, 2)	Students' final grade
DT_DATEOFPARTIALGRADE	Smalldatetime	Date of students' partial grade
NR_PARTIALGRADE	Decimal (6, 2)	Students' partial grade
DT_DATEOFINTERN	Smalldatetime	Date of students' intern
NR_GRADEOFINTERN	Decimal (6, 2)	Students' grade of intern
DT_DATEOFGRAGEIMPROV	Smalldatetime	Date of students' grade improved
NR_GRADEIMPROV	Decimal (6, 2)	Students' grade improved
DT_DATEOFDIPLOMA	Smalldatetime	Date of students' diploma
DT_DATEOFPARTIALDIPLOMA	Smalldatetime	Date of students' partial diploma
DT_DATEDIPLOMAEMITION	Smalldatetime	Date of students' diploma emission
DT_DATEDIPLOMAWITHDRAW	Smalldatetime	Date of students' diploma withdraw

Figure 3.6 – Dimension Students details

3.3.2.10. Dimension Term

The table "DimTerm" contains details about each instance of a term type. The dimension includes information such as the type of term description in English and in Portuguese. The table below displays the fields, the type and the fields' description:

Field	Type	Description
TermID	Integer	Primary Key/ Surrogate Key/ Identity Column
CD_TERM	Tinyint	Natural Key
Term_EN	Nchar (100)	Term description in English
Term_PT	Nchar (100)	Term description in Portuguese

Table 3.9 – Dimension Term details

3.3.2.11. Dimension Time

The table "DimTime" contains details about the time, for example, it includes information such as the full time (HH:MM:SS) and the time represented in seconds and along with "DimDate" it is also a general dimension that should exist in every Data Warehouse. The table below displays the fields, the type and the fields' description:

Field	Type	Description
TimeKey	Integer	Primary Key/ Surrogate Key/ Identity Column
TimeAltKey	Integer	Natural Key (Aggregated hour, minute and second)
Time30	Time (7)	Time separated by ":" (i.e. "05:45:30")
Hour30	Tinyint	Hour number
MinuteNumber	Tinyint	Minute number
SecondNumber	Tinyint	Second number
TimeInSecond	Integer	Time represented in seconds
HourlyBucket	Varchar (15)	Hour bucket. One-hour bucket (i.e. 10:00-10:59").
DayTimeBucketGroupKey	Integer	Key representing eight different time buckets
DayTimeBucket	Varchar (100)	Name of the eight different time buckets represented by field 'DayTimeBucketGroupKey', where each number matches a name: 0 – Late Night (00:00 AM To 02:59 AM) 1 – Early Morning (03:00 AM To 6:59 AM) 2 – AM Peak (7:00 AM To 8:59 AM) 3 – Mid Morning (9:00 AM To 11:59 AM) 4 – Lunch (12:00 PM To 13:59 PM) 5 – Mid Afternoon (14:00 PM To 15:59 PM) 6 – PM Peak (16:00 PM To 17:59 PM) 7 – Evening (18:00 PM To 23:59 PM)

Table 3.10 – Dimension Time details

3.3.2.12. Fact College Enrolment

The table "FactCollegeEnrollment" holds information about a student enrolling into a new academic year and semester meaning that this factual table displays the number of courses enrolled, the number of credits and the number of credits ECTS (European Credit Transfer System). This particular

fact table contains a number of foreign key references to the matching dimension table, where the entry corresponds to a primary key in that table for a record. The table below displays the fields, the type, the fields' description and the foreign keys relationship:

Field	Type	Description
StudentsID	Integer	Primary Key/ Foreign Key to Dimension Students
ProgSpecPlanID	Integer	Primary Key/ Foreign Key to Dimension Program/Specialization/Plan
AcademicYearSemesterID	Integer	Primary Key/ Foreign Key to Dimension Academic Year Semester
CD_Y_S	Decimal (2, 0)	Year in which a student is enrolled in. For example, 1, 2, 3
DateKeyDateOfEnrollment	Integer	Primary Key/ Foreign Key to Dimension Date. This field is the date of enrolment of a Student in a Program/Specialization/Plan
TimeKeyDateOfEnrollment	Integer	Primary Key/ Foreign Key to Dimension Time. This field is the time of enrolment of a Student in a Program/Specialization/Plan
DateKeyDateEndOfEnrollment	Integer	Primary Key/ Foreign Key to Dimension Date. This field is the end date of enrolment of a Student in a Program/Specialization/Plan
TimeKeyDateEndOfEnrollment	Integer	Primary Key/ Foreign Key to Dimension Time. This field is the end time of enrolment of a Student in a Program/Specialization/Plan
NR_COURSES	Decimal (3, 0)	Number of Courses Enrolled
NR_COURSES_APPROVED	Decimal (3, 0)	Number of Courses Approved
NR_COURSES_NOTAPPROVED	Decimal (3, 0)	Number of Courses Not Approved
NR_CREDITS	Decimal (12, 2)	Number of Credits
NR_CREDITS_APPROVED	Decimal (12, 2)	Number of Credits Approved
NR_CREDITS_NOTAPPROVED	Decimal (12, 2)	Number of Credits Not Approved

NR_CREDITS_ECTS	Decimal (12, 2)	Number of ECTS
NR_CREDITS_ECTS_APPROVED	Decimal (12, 2)	Number of ECTS Approved
NR_CREDITS_ECTS_NOT_APPROVED	Decimal (12, 2)	Number of ECTS Not Approved

Table 3.11 – Fact College Enrolment details

3.3.2.13. Fact Course Enrolment Final Grade

The table "FactCourseEnrollmentFinalGrade" holds information about a student course enrolment history and grades for each academic year and semester meaning that this factual table displays the courses enrolled by a student, the final grade that it was obtained for that enrolment and the exam grades. This particular fact table contains a number of foreign key references to the matching dimension table, where the entry corresponds to a primary key in that table for a record. The table below displays the fields, the type, the fields' description and the foreign keys relationship:

Field	Type	Description
ProgSpecPlanID	Integer	Primary Key/ Foreign Key to Dimension Program/Specialization/Plan
StudentsID	Integer	Primary Key/ Foreign Key to Dimension Students
CoursesID	Integer	Primary Key/ Foreign Key to Dimension Courses
AcademicYearSemesterID	Integer	Primary Key/ Foreign Key to Dimension Academic Year Semester
ClassID	Integer	Primary Key/ Foreign Key to Dimension Class
CD_Y_S	Decimal (2, 0)	Year in which a student is enrolled in. For example, 1, 2, 3
DateKeyDateOfEnrollment	Integer	Primary Key/ Foreign Key to Dimension Date. This field is the date of enrolment of a Student in a course
TimeKeyDateOfEnrollment	Integer	Primary Key/ Foreign Key to Dimension Time. This field is the time of enrolment of a Student in a course
NR_FINAL_GRADE	Decimal (6, 2)	Students' Final Grade for a Course
CD_STATUS	Decimal (2, 0)	Primary Key/ Foreign Key to Dimension Status

CD_TERM	Tinyint	Primary Key/ Foreign Key to Dimension Term
DateKeyDateOfExam	Integer	Primary Key/ Foreign Key to Dimension Date. This field is the date of exam of a Student
NR_GRADE	Smallint	Exam Grade of a Student

Table 3.12 – Fact Course Enrolment Final Grade details

3.3.2.14. Fact Partial Grades

The table "FactPartialGrades" holds information about the partial grades of a student for each academic year and semester meaning that this factual table only purpose is to display each partial grade of a student. This particular fact table contains a number of foreign key references to the matching dimension table, where the entry corresponds to a primary key in that table for a record. The table below displays the fields, the type, the fields' description and the foreign keys relationship:

Field	Type	Description
PartialGradeTypesID	Integer	Primary Key/ Foreign Key to Dimension Partial Grade Types
StudentsID	Integer	Primary Key/ Foreign Key to Dimension Students
CoursesID	Integer	Primary Key/ Foreign Key to Dimension Courses
ProfessorsID	Integer	Primary Key/ Foreign Key to Dimension Professors
AcademicYearSemesterID	Integer	Primary Key/ Foreign Key to Dimension Academic Year Semester
DateKeyLastChange	Integer	Primary Key/ Foreign Key to Dimension Date. This field is the last change date to a Partial Grade of a Student
TimeKeyLastChange	Integer	Primary Key/ Foreign Key to Dimension Time. This field is the last change time to a Partial Grade of a Student
NR_GRADE	Float	Partial Grade of a Student

Table 3.13 – Fact Partial Grades details

3.3.2.15. Fact Work Allocation

The table "FactWorkAllocation" holds information about the work allocation for the teaching staff to a course for each academic year and semester meaning that this factual table will show that a specific professor has, for example forty hours of teaching for a course in a single semester. This

particular fact table contains a number of foreign key references to the matching dimension table, where the entry corresponds to a primary key in that table for a record. The table below displays the fields, the type, the fields' description and the foreign keys relationship:

Field	Type	Description
ProfessorsID	Integer	Primary Key/ Foreign Key to Dimension Professors
CoursesID	Integer	Primary Key/ Foreign Key to Dimension Courses
AcademicYearSemesterID	Integer	Primary Key/ Foreign Key to Dimension Academic Year Semester
TOTAL_HOURS	Float	Total hours of class given by a professor for a specific course by year/semester

Table 3.14 – Fact Work Allocation details

3.3.3. ETL Processes

The Data Warehouse was designed as an SQL Server database using SQL Server Management Studio due to the easiness, reliability and support that Microsoft. In order for this project to integrate all data, including the extraction, transformation and loading it was chosen the "SQL Server Data Tools".

Inside the "SQL Server Data Tools", the project type used for the master thesis project is the "Business Intelligence Integration Services Project" that was named "NOVAIMSOnlineDW_SSIS".

The integration project will be composed of three SQL Server Integration Services (SSIS) packages and they may be found below:

- Main.dtsx
- LoadDimensions.dtsx
- LoadFacts.dtsx

Figure 3.7 displays the SSIS main package which main function is to control the order of packages that are needed to run. It is essential that when running the SSIS it starts from the Main package allowing every step to be completed with the correct order.

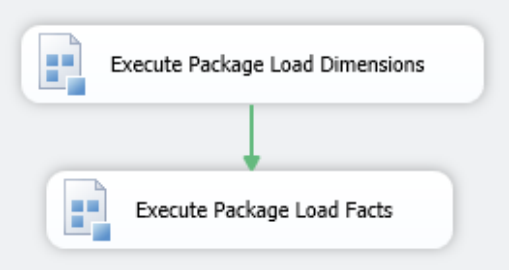

Figure 3.7 – SSIS Main package

Figure 3.8 displays the SSIS load dimensions' package which main function is to populate the dimension of the data warehouse. Starting by cleaning some of the data warehouse tables' records to migrate directly the source latest version (tables without history and always with the latest data) or in some cases to update the table records using Slowly Changing Dimensions (dimensions that will have history). This is the second package that will run.

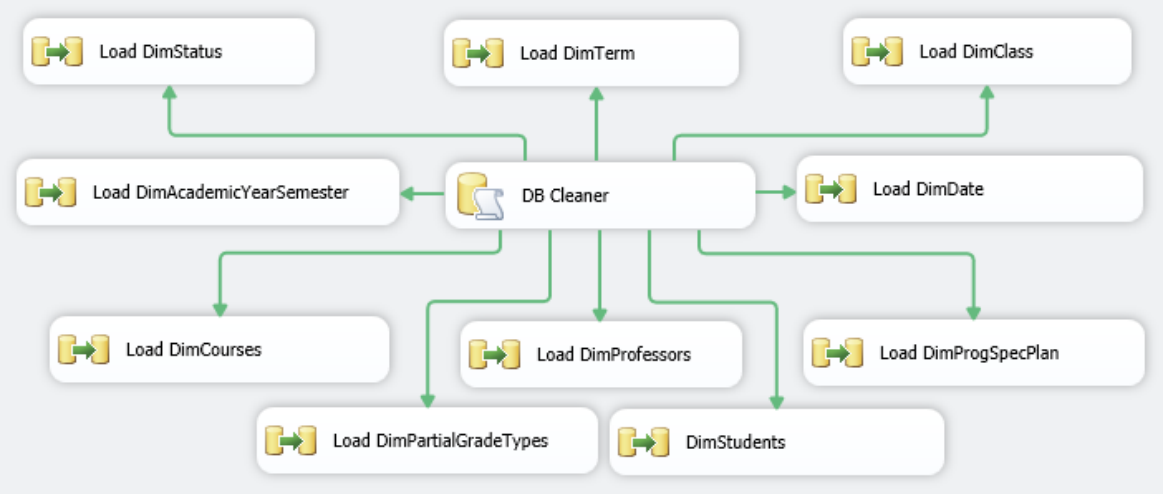

Figure 3.8 – SSIS Load Dimensions package

Figure 3.9 displays the SSIS load facts package which exist to populate the fact tables of the data warehouse. Each fact table queries the sources and then it looks up to match with the selected dimension that were previously loaded. This is the last package that will run.

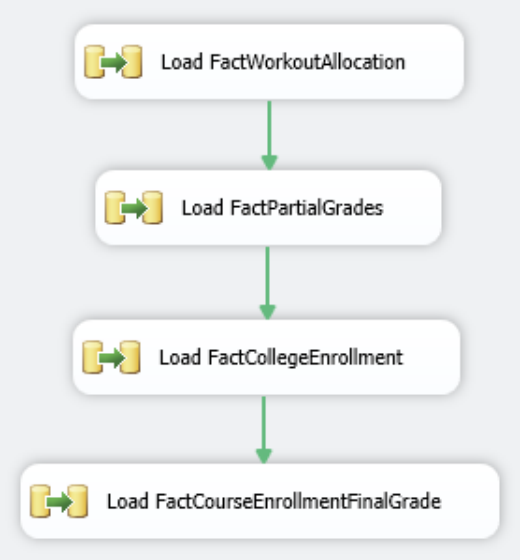

Figure 3.9 – SSIS Load Facts package

The following sub-sections explain in detail the integration process for each dimensional and factual table.

3.3.3.1. Dimension Academic Year Semester

Dimension Academic Year Semester has a single data source that will be "IsegiOnline_Source" database and all data will be migrated from tables PartialGrades, WorkAllocation, Enrollment and CourseEnrollment.

Annexes - Table 6.1 presents the SQL command used to select the source data for Dimension Academic Year Semester and it is possible to check that AcademicYearPeriodicity field needed to be divided into two fields as a varchar and integer type, AcademicYearPeriodicityChar and AcademicYearPeriodicityInt respectively. This division needed to be made due to connections to Fact tables, for example, FactCollegeEnrollment and Fact Course Allocation fields were varchar and Fact Work Allocation field was integer. FactPartialGrades is another table that needed special attention because it could only connect to AcademicYear field since it only has the year and not year plus periodicity. There is a where clause in Table 6.1 SQL command which only allows values "0", "A", "1", "2", "3" or null for the Periodicity field. This step was added to remove any unreliable data that may occur and with the current data it will remove four records spread out between the fact tables.

Figure 3.10 displays the ETL process to migrate Dimension Partial Grade Types data and even though Figure 3.10 only displays the extract and load part of the ETL process, a minor transformation is performed while retrieve data from the source as it is possible to validate on Annexes - Table 6.1.

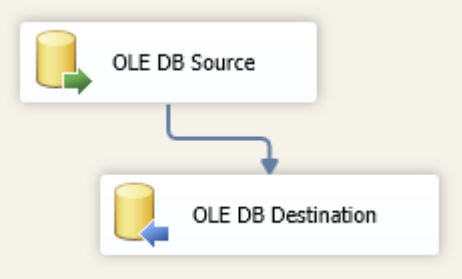

Figure 3.10 – ETL process for Dimension Academic Year Semester

3.3.3.2. Dimension Class

Dimension Class has a single data source named "Dim Class.xls" excel file and the contained data in the file was provided by NOVA IMS.

Figure 3.11 displays the ETL process to migrate Dimension Class data. According to what was mentioned in the beginning of this section the data source is an excel file and the only transformation performed to this dimension is a data conversion where it is necessary to adapt the excel source field types to the SQL Server database field type.

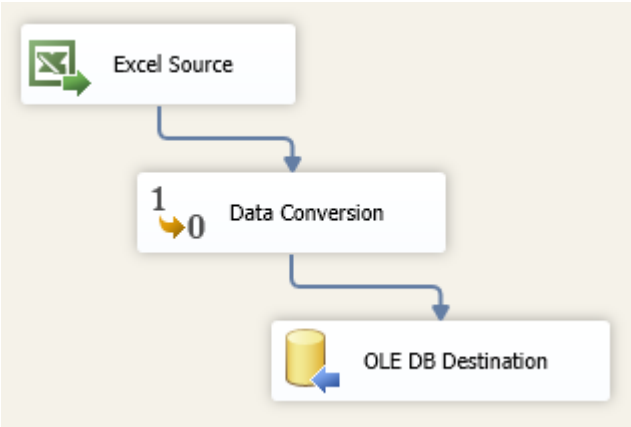

Figure 3.11 – ETL process for Dimension Class

3.3.3.3. Dimension Courses

Dimension Courses has a single data source that will be "IsegiOnline_Source" database and all data will be migrated from table Courses.

Figure 3.12 shows the ETL process to migrate Dimension Courses data and this process only involves extract and load, because there is no transformation needed.

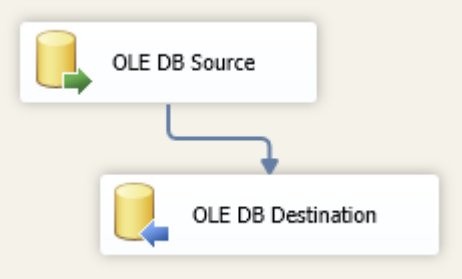

Figure 3.12 – ETL process for Dimension Courses

3.3.3.4. Dimension Date

Dimension Data has a single data source named "Ch07_Date_Dim_2000-2020.xls" excel file and the contained data in the file was created by Kimball Group[12] and adapted to fit the needs of this project. Dimension Date supports dates from 1990 to 2020 and if necessary it is possible to insert more following the guidelines in the Excel file. For the source data from this excel file to be used it was necessary to convert the fields data type in order to fit the Dimension Date.

Figure 3.13 displays the ETL process to migrate Dimension Date data. According to what was mentioned in the beginning of this section the data source is an excel file and the only transformation performed to this dimension is a data conversion where it is necessary to adapt the excel source field types to the SQL Server database field type.

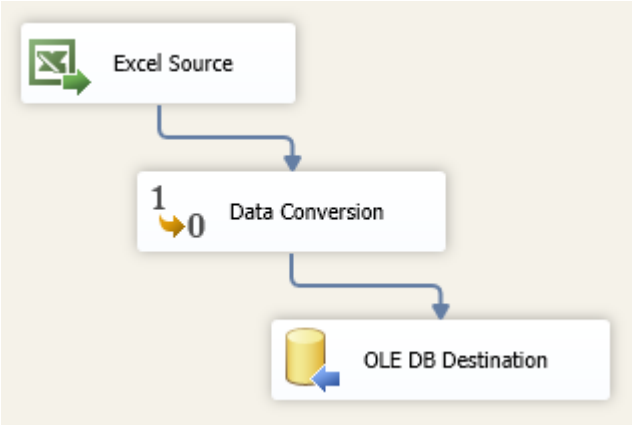

Figure 3.13 – ETL process for Dimension Date

3.3.3.5. Dimension Partial Grade Types

Dimension Partial Grade Types has as data source the "IsegiOnline_Source" database. All data will be migrated from the tables PartialGrades and PartialGradeTypes.

[12] Excel downloaded and adapted from Kimball Group - http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/books/microsoft-data-warehouse-dw-toolkit/

Annexes - Table 6.2 presents the SQL command used to select the source data for Dimension Partial Grade Type and after analysing it is possible to conclude that it is a simple query that only retrieves data from two tables, PartialGrades and PartialGradeTypes. In the case of PartialGrades table only the attributes are being selected as the rest of the information will be used later in a fact table, as it will be possible to check in section 3.3.3.14.

Figure 3.14 displays the ETL process to migrate Dimension Partial Grade Types data and even though Figure 3.14 only displays the extract and load part of the ETL process, a minor transformation is performed while retrieve data from the source as it is possible to validate on Table 6.2.

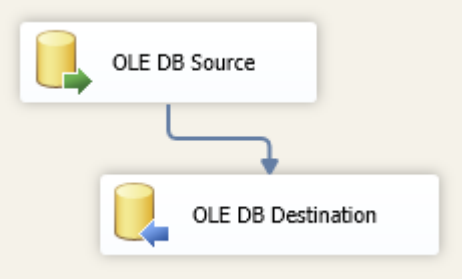

Figure 3.14 – ETL process for Dimension Partial Grade Types

3.3.3.6. Dimension Professors

Dimension Professors has a single data source that will be "IsegiOnline_Source" database and all data will be migrated from table Professors.

Figure 3.15 shows the ETL process to migrate Dimension Professor data and the slowly changing dimension process where in case of an update in the "ADDRESS_PLACE", "CD_ADDRESS_ZIPCODE" or "DEGREE" fields for a professor it will create a new record with the new information. The field "IsLatest" will be "1" for the new record and the old one will change to "0".

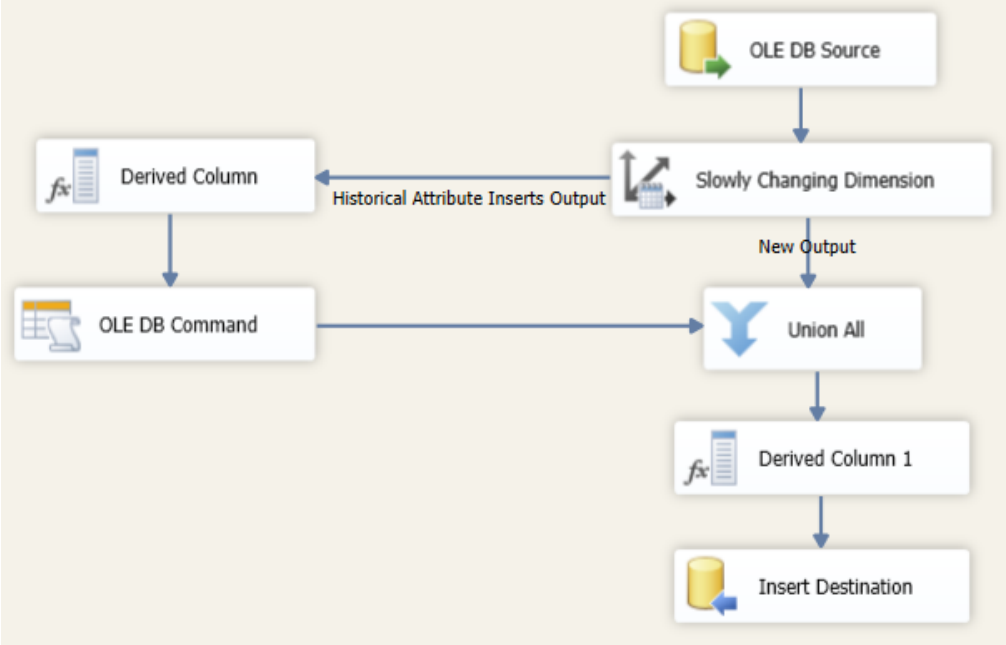

Figure 3.15 – ETL process for Dimension Professors

3.3.3.7. Dimension Program/Specialization/Plan

Dimension Program Specialization Plan has a single data source that will be "IsegiOnline_Source" database and all data will be migrated from tables Programs, Specializations and Plans.

Annexes - Table 6.3 presents the SQL command used to select the source data for Dimension Program Specialization Plan. The query is retrieving data from three tables in total, Programs, Specializations and Plan. Dimension Program Specialization Plan has a hierarchy in which Program table is parent of Specializations table and the same Specialization table is parent of Plans table. Figure 3.16 displays the hierarchy in a user-friendly diagram.

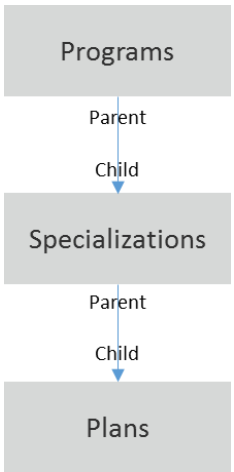

Figure 3.16 – Hierarchy from Dimension Programs Specializations Plans

Figure 3.17 shows the ETL process to migrate Dimension Program Specialization Plan data and the slowly changing dimension process where in case of an update in the “NM_PROGRAM_NAME”, ”NM_ABREV_PROGRAM_NAME”, ”NM_SPECIALIZATION” or “NM_PLAN fields for a program, specialization or plan it will create a new record with the new information. The field “ProgSpecPlanEffectiveFrom” for the new record will have the update date while this field will remain the same the for the old record. However, the field “ProgSpecPlanEffectiveTo” will have the update date for the old record and the null value for the new one.

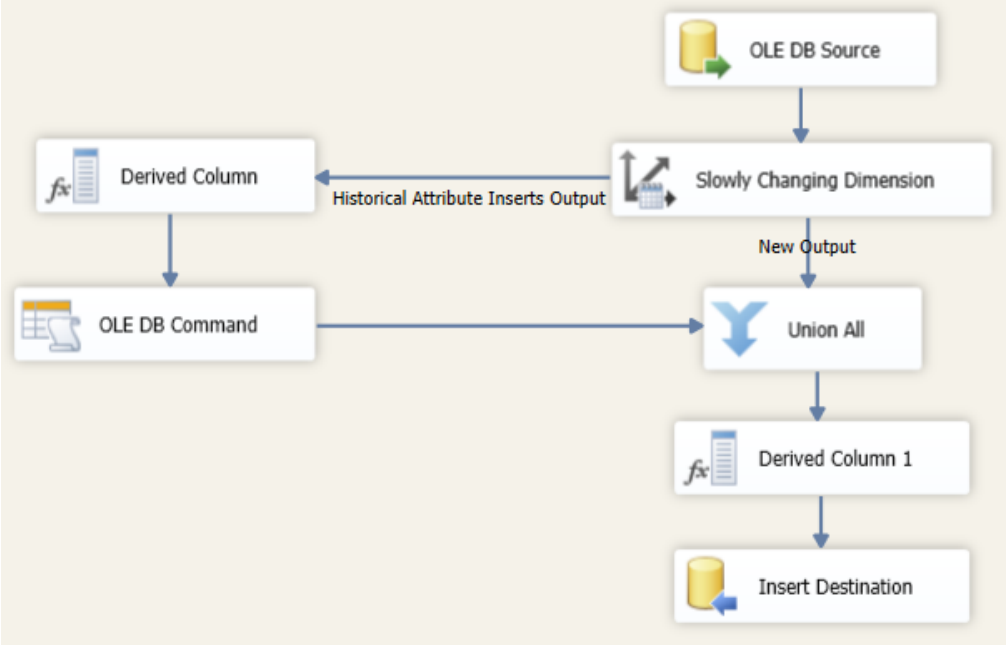

Figure 3.17 – ETL process for Dimension Program Specialization Plan

3.3.3.8. Dimension Status

Dimension Status has a single data source named “Dim Status.xls” excel file and the contained data in the file was provided by NOVA IMS Academic Services and the advisors of this project.

Figure 3.18 displays the ETL process to migrate Dimension Status data. According to what was mentioned in the beginning of this section the data source is an excel file and the only transformation performed to this dimension is a data conversion where it is necessary to adapt the excel source field types to the SQL Server database field type.

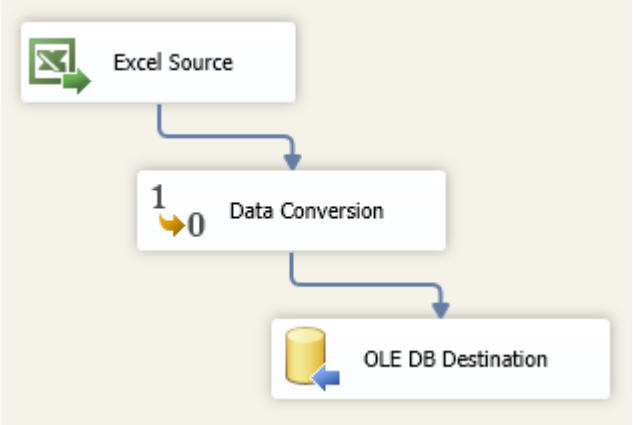

Figure 3.18 – ETL process for Dimension Status

3.3.3.9. Dimension Students

Dimension Students has several data sources, the main one will be "IsegiOnline_Source" database and another data source named "Auxiliary Dim Students Info.xls" excel file, which contains several sheets with metadata that will be needed for students' dimension. The excel file was provided by NOVA IMS Academic Services and the advisors of this project. For the source data from this excel file to be used it was necessary to convert the fields' data type in order to fit the Dimension Status

Figure 3.19 and Figure 3.20 displays the ETL process to migrate Dimension Students data. According to what was mentioned in the beginning of this section this dimension has two data sources.

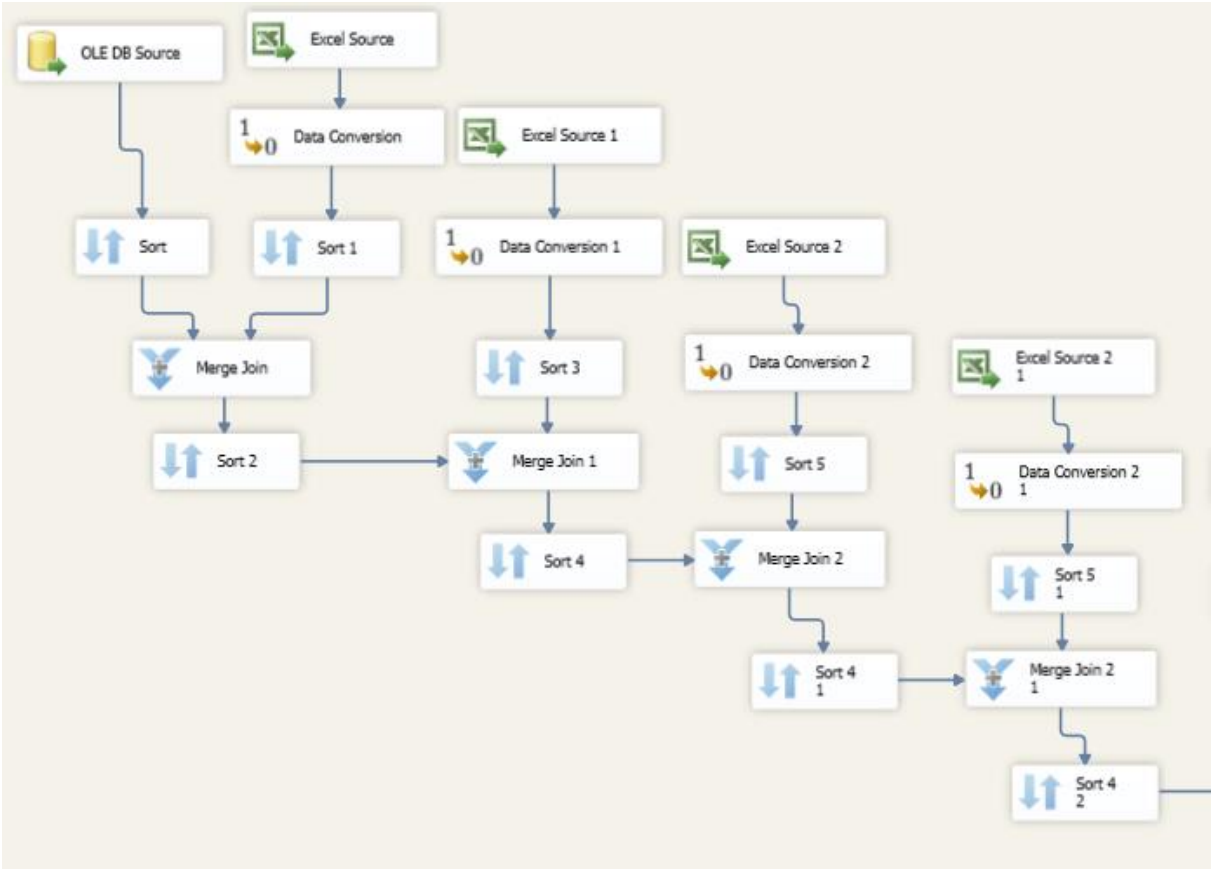

Figure 3.19 – ETL process for Dimension Students – Part 1

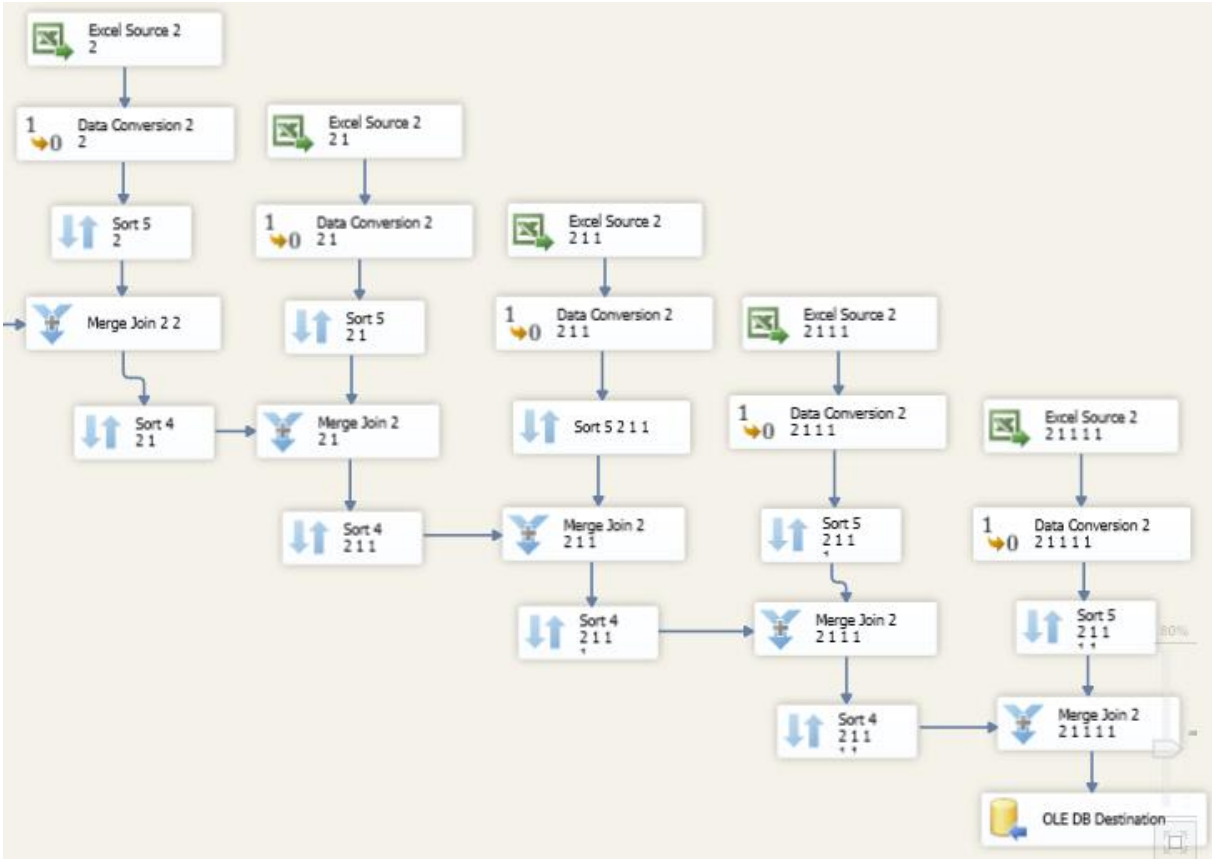

Figure 3.20 – ETL process for Dimension Students – Part 2

First, the "IsegiOnline_Source" database where an SQL command was used to select data, as presented in Annexes - Table 6.4. The query is retrieving data from only one table, Students. Students' source table has two relevant keys, "CD_PROGRAM" and "CD_STUDENT", which allows to store the same students in two different programs. From a business side this is something that should never happen, in case a student enrols in a different program it will have a different student number and for that reason and to ensure data quality due to lack of information on a student being in two different programs it was decided that 46 from a total of 5149 records should be removed.

Secondly, it was necessary to transform a few fields from the "IsegiOnline_Source" database as these were only codes. For that happen, the excel file was adapted to create "Auxiliary Dim Students Info.xls" containing additional attributes for the dimension and Figure 3.19 displays the process of transformation necessary to retrieve a more completed information. The transformation process was performed by joining the "IsegiOnline_Source" database with several tables from the excel file. For example, the transformation process allowed for Dimension Students to be populated with "Civil Codes" like "Solteiro(a)" instead of "1".

3.3.3.10. Dimension Term

Dimension Term has a single data source named "Dim Term.xls" excel file and the contained data in the file was provided by NOVA IMS Academic Services and the advisors of this project. For the source

data from this excel file to be used it was necessary to convert the fields' data type in order to fit the Dimension Term.

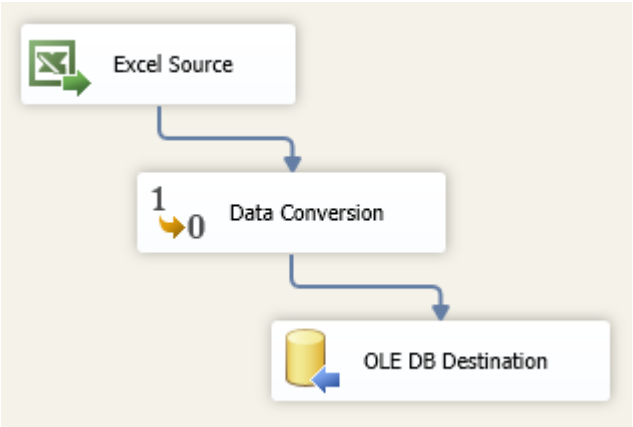

Figure 3.21 – ETL process for Dimension Term

Figure 3.21 displays the ETL process to migrate Dimension Status data. According to what was mentioned in the beginning of this section the data source is an excel file and the only transformation achieved to this dimension is a data conversion where it is necessary to adapt the excel source field types to the SQL Server database field type.

3.3.3.11. Dimension Time

Dimension Time has been populated via SQL command, which may be validated in Annexes - Table 6.5 and it was decided to do it with this technique because it is type of dimension that shouldn't need maintenance unless more fields need to be added. In case something happens to Dimension Time (i.e. corrupted data) it is only necessary to run the stored procedure "FillDimTime". Time30 field was originally of type Varchar (8) but is need to be changed to Time (7) to in order to match with all fact tables.

3.3.3.12. Fact College Enrolment

Fact College Enrolment has a single data source that will be "IsegiOnline_Source" database and all data will be migrated from table Enrollment.

Annexes - Table 6.6 presents the SQL command used to select the source data for Fact College Enrolment. The query is only retrieving data from table Enrollment and the reason why it is necessary to adapt this query is to ensure that data and time types are returning values (for example, 1900-01-01 and 00:00:00, accordingly) that are not null, which could happen in the data source table. It is something mandatory because these fields in table FactCollegeEnrollment will be primary/foreign

keys. The field “CD_Y_S” displayed inconsistency on the semester value and taking into account that data quality is essential when analysing data, it was removed leaving only the course year.

Figure 3.22 – ETL process for Fact College Enrolment

Figure 3.22 displays the ETL process to populate Fact College Enrolment. We start by selecting the source data (query from Table 6.6). Afterwards we will look up for the associated dimensions, which in this case will be DimStudents, DimProgSpecPlan, DimDate, DimTime and DimAcademicYearSemester. Finally, it will populate Fact College Enrolment and in the event of not having a match with the look up dimension the unmatched record will be forwarded to a text file document named “Load FactCollegeEnrollment Error File”. When looking up for DimProgSpecPlan we used the SQL command presented in Annexes - Table 6.12 to select the program, specialization and plan.

3.3.3.13. Fact Course Enrolment Final Grade

Fact Course Enrolment Final Grade has a single data source that will be “IsegiOnline_Source” database and all data will be migrated from tables CourseEnrollment and FinalGrades.

Annexes - Table 6.7 presents the SQL command used to select the source data for Fact Course Enrolment Final Grade. The query is only retrieving data from the two tables mentioned above where all data is being migrated from and also the same changes done on section 3.3.3.12 to date and time field because these fields in table FactCourseEnrollmentFinalGrade will be primary/foreign keys. The field “CD_Y_S” displayed inconsistency on the semester value and taking into account that data quality is essential when analysing data, it was removed leaving only the course year.

As part of the ETL process, the CD_FINALGRADE field was not used because it was only a surrogate key in FinalGrade table. In this specific case only the NR_GRADE field brings value while CD_FINALGRADE is an ordinary code. Another field that didn’t had any relevant data is CD_DURATION which only after analysing data it was possible to understood that it would only give the semester and that can be already obtained from another field (AcademicYearSemester).

Figure 3.23 – ETL process for Fact Course Enrolment Final Grade

Figure 3.23 displays the ETL process to populate Fact Course Enrolment Final Grade. Starting by selecting the source data (query from Annexes - Table 6.7) it will afterwards look up for the associated dimensions, which in this case will be DimCourses, DimStudents, DimProgSpecPlan, DimTerm, DimClass, DimStatus, DimDate, DimTime and DimAcademicYearSemester. Finally, it will populate Fact Course Enrolment Final Grade and in the event of not having a match with the look up dimension the unmatched record will be forwarded to a text file document named “Load FactCourseEnrollmentFinalGrade Error File”. It was necessary, when looking up for DimProfessors and DimProgSpecPlan to use a SQL command presented in Annexes - Table 6.11 and Annexes - Table 6.12, accordingly, in order to select the latest professor information due to DimProfessors having history and also to the select the program, specialization and plan in force.

3.3.3.14. Fact Partial Grades

Fact Partial Grades has a single data source that will be “IsegiOnline_Source” database and all data will be migrated from tables PartialGrades and StudentPartialGrades.

Annexes - Table 6.9 presents the SQL command used to select the source data for Fact Partial Grade Type and after analysing it is possible to conclude that is a simple query that only retrieves data from two tables, PartialGrades and StudentPartialGrades. In the case of PartialGrades table only the metrics are being selected as the rest of the information was used for Dimension Partial Grade Type and it is possible to validate in section 3.3.3.5. For this fact table the same changes to date and time in section 3.3.3.12 were performed due to the date and time fields being primary/foreign keys in Fact Partial Grades.

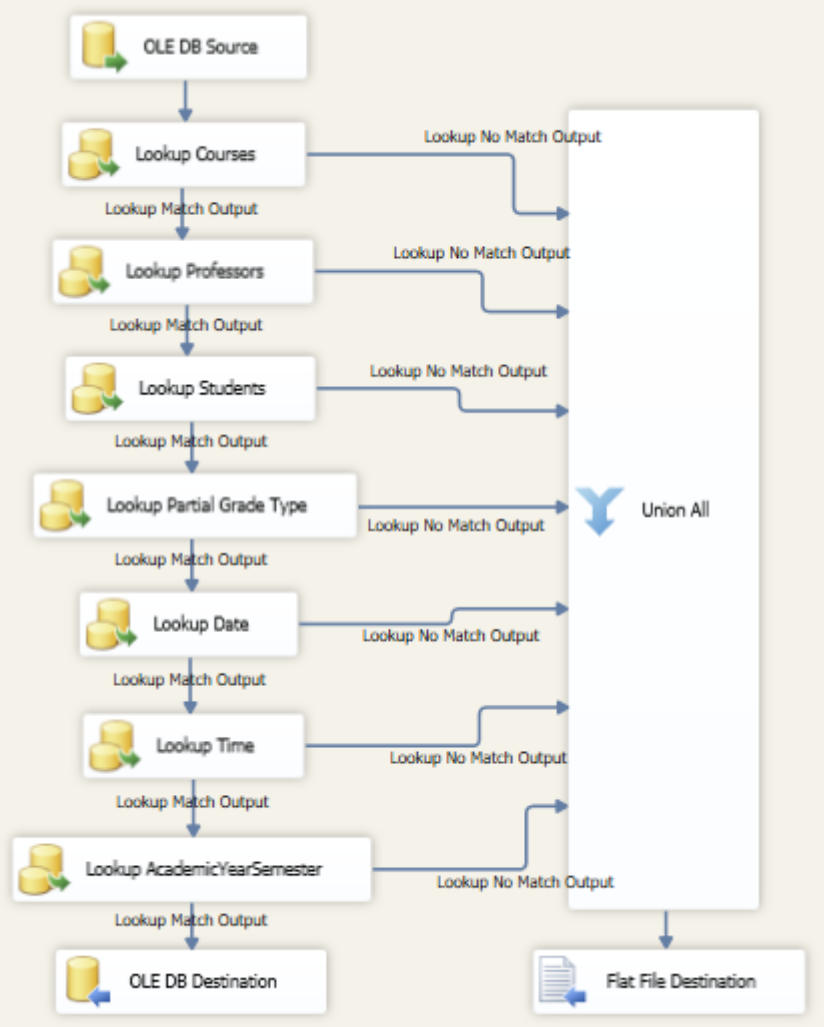

Figure 3.24 – ETL process for Fact Partial Grades

Figure 3.24 displays the ETL process to populate Fact Partial Grades. Starting by selecting the source data (query from Annexes - Table 6.9) it will afterwards look up for the associated dimensions, which

in this case will be DimCourses, DimProfessors, DimStudents, DimPartialGradeType, DimDate, DimTime and DimAcademicYearSemester. Finally, it will populate Fact Partial Grades and in the event of not having a match with the look up dimension the unmatched record will be forwarded to a text file document named “Load FactPartialGrades Error File”. It was necessary, when looking up for DimProfessors to use a SQL command presented in Annexes - Table 6.11 in order to select the latest professor information due to DimProfessors having history.

The process of populating Fact Partial Grades was fairly simple, despite having a data quality issue due to AcademicYearPeriodicityInt field, which was the surrogate key for Dimension, having a different number of characters (dimension had seven and fact only six). In order to overcome this data quality problem, the AcademicYear (same as AcademicYearPeriodicity without the Periodicity) was used as surrogate key when the Periodicity was null. The SQL command used to lookup for this values may be confirmed in Annexes - Table 6.10.

3.3.3.15. Fact Work Allocation

Fact Work Allocation has a single data source that will be “IsegiOnline_Source” database and all data will be migrated from a single table named WorkAllocation.

Annexes - Table 6.8 presents the SQL command used to select the source data for Fact Work Allocation. The query is retrieving data from the single source data needed. However, there was a business decision being this query that was referring to the need or not of the field CD_PROGRAM. Finally, it was decided that PROGRAM was not going to be needed as there was no additional information added by it. Also, it would make no sense since a professor teaches a course in a semester, it didn’t matter the program itself. There were only two records in total that were removed due to this decision.

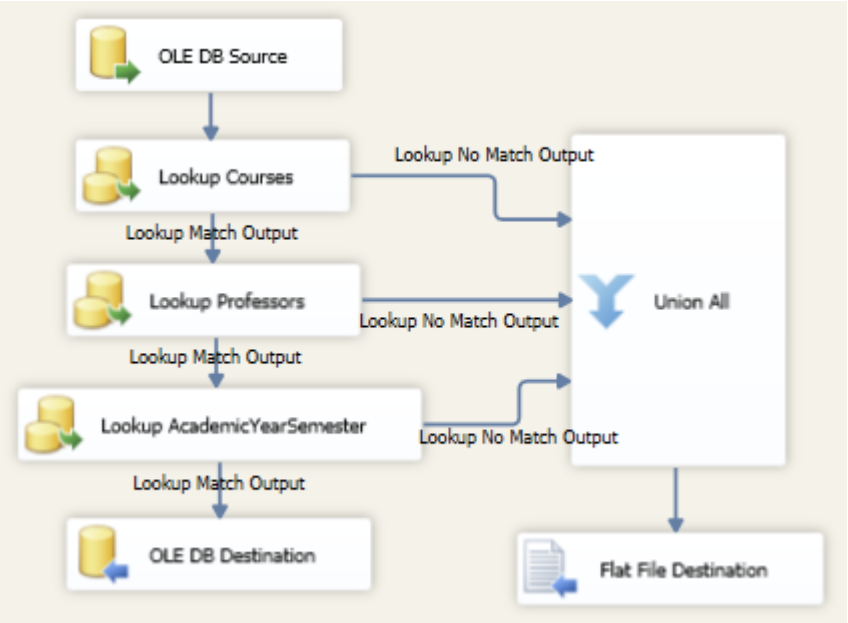

Figure 3.25 – ETL process for Fact Work Allocation

Figure 3.25 displays the ETL process to populate Fact Work Allocation. Starting by selecting the source data (query from Annexes - Table 6.8) it will afterwards look up for the associated dimensions, which in this case will be DimCourses, DimProfessors and DimAcademicYearSemester. Finally, it will populate Fact Work Allocation and in the event of not having a match with the look up dimension the unmatched record will be forwarded to a text file document named "Load FactWorkoutAllocation Error File".

3.4. REPORTING DESIGN

The reporting design chapter aims to present trends and results from the NOVA IMS Academic Information System. To best demonstrate these trends and facts a number of charts using Microsoft Excel Power Pivot were created. The reports only contain data until the Academic Year of 2014/2015.

In following sub-sections, we will explain in detail each developed chart.

3.4.1. General

In the General chapter, we will present and explain the graphics containing the overal information about the students of Nova IMS.

Starting with Figure 3.26, a map showing students' nationalities, it is possible to verify that Nova IMS attracts students from fifty-one different countries, making it a quite international school. Moreover, the majority of these students come from Portugal, Europe (namely Spain, due to geographical proximity and Slovenia, possibly because of similar higher education costs) and Africa (especially from the PALOP's countries). This map is especially important to take into account seen as it contains vital clues that can help the school improve their internationalization strategy, in terms of what countries should they be focusing on when establishing priority markets and designing policies, which would be become more relevant and suitable for students from different nationalities.

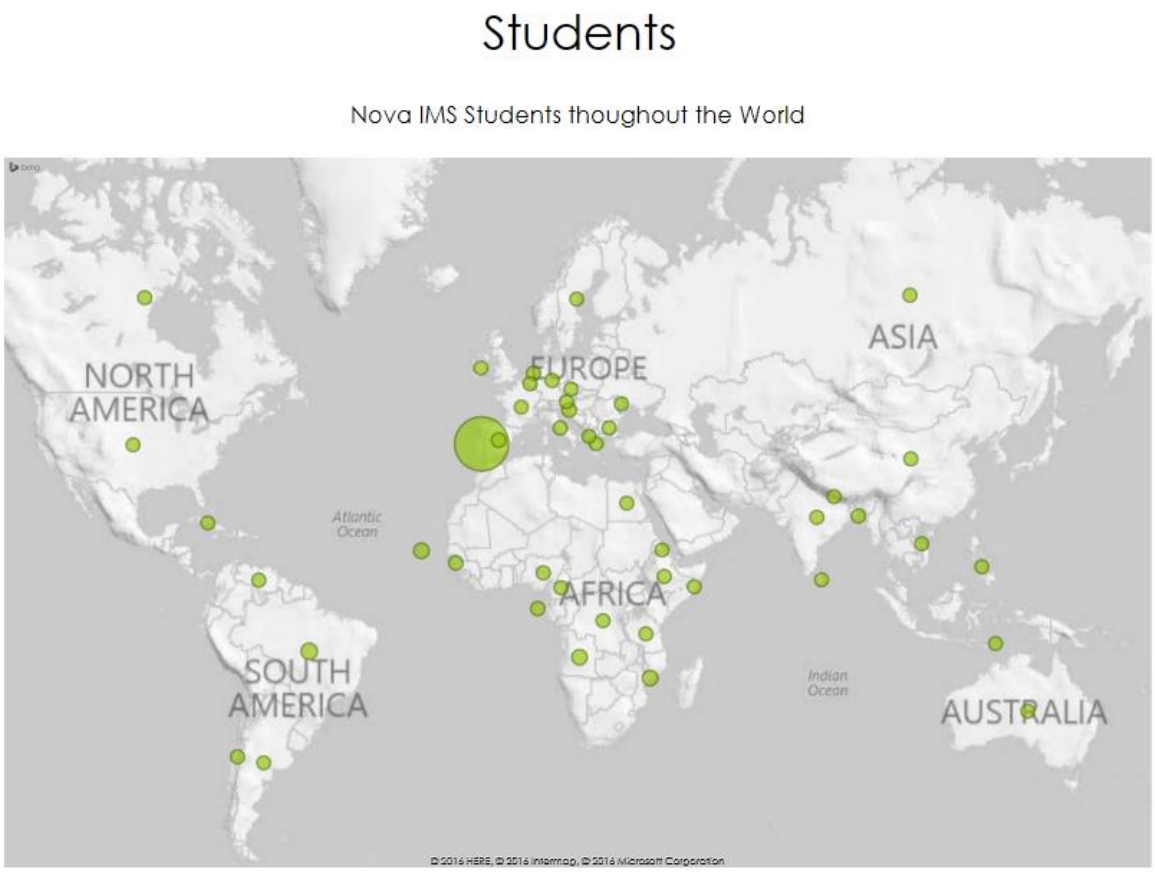

Figure 3.26 – Nova IMS Map of Students Country

Continuing on with Figure 3.27, it illustrates two pie charts, the first one referring to Nova IMS students' Gender and the second one referring to Civil Status. Regarding Gender, 59% of the students are men and 41% are women, showing that the school has a ratio of men and women of almost 1:1, which can be especially relevant for technology schools to prove they are gender equality schools. Considering the Civil Status, around 90% of Nova IMS students are single, which is an indicator that most students pursue a higher education degree straight out of high school, which in turn can help the school define better marketing and strategies.

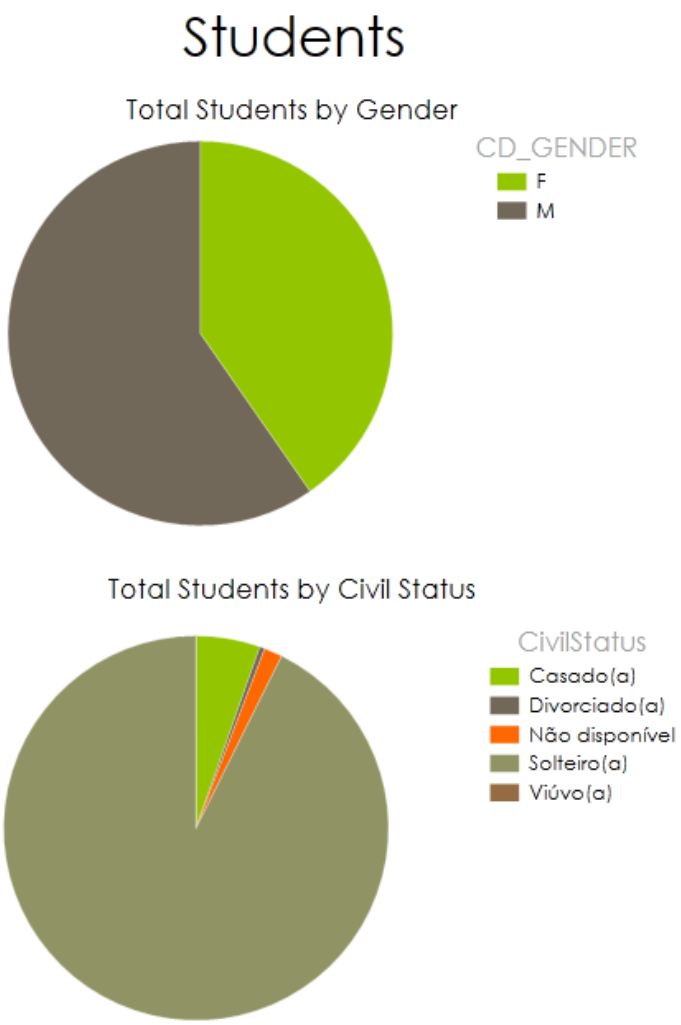

Figure 3.27 – Total Students by Gender and by Civil Status

3.4.2. College Enrolment

Moving on now to the College Enrolment chapter, we will focus on information regarding the total number of students and courses that they took in Nova IMS.

Figure 3.28 compares the courses approved and not approved for a student at time of enrolment. Starting with the graph on the top, which is divided by academic years, it is possible to conclude that in the last 5 years' students have more approved courses than in any previous year. Furthermore, enrolments in courser have also been steadily increasing, which may b explained be the increase in the total number of students attending Nova IMS. Moreover, still regarding the top graph, the school years of 1990/91, 1991/92 and 1992/93 were left out due to the lack of data for approved/not approved courses.

Now considering the bottom graph, it displays the approved and not approved courses divided by study plan year, meaning that a specific student is enrolled in the 1st year of school, 2nd year, etc. An interesting fact shown in this second graph is the fact that the total courses approved and not approved are larger in the first year than in the remaining ones. This may be explained by the fact

that Masters'/ Post-Graduation courses only have 1 year of academic courses and that after the 1st year of the Bachelor Degree, some students end up dropping out of college.

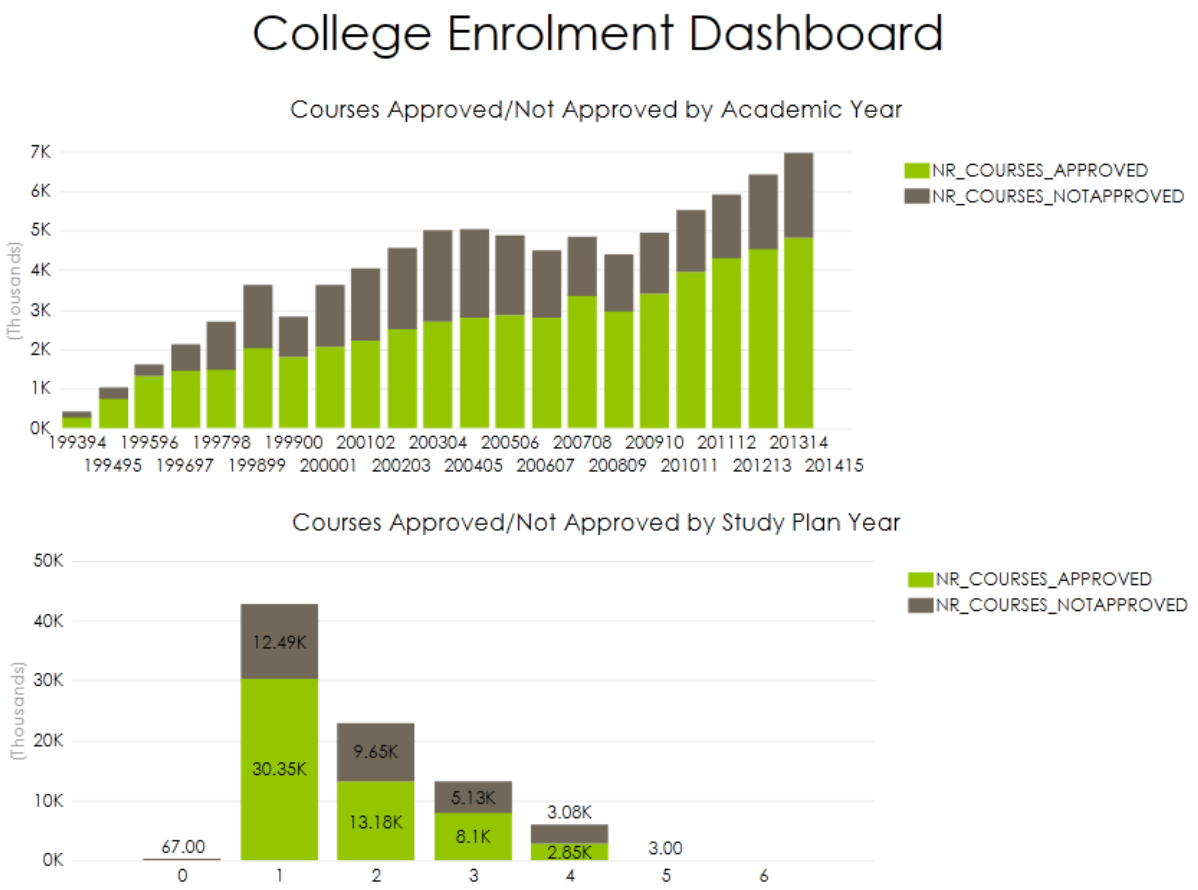

Figure 3.28 – Courses Approved VS Not Approved by Academic Year and by Study Plan Year

Considering Figure 3.29, it illustrates the total number of students enrolled in Nova IMS Programs & Specialisations. With that being said, it enables the College Board to understand which programs are the most popular when searching for the enrolled students. Hence, and taking a look at the number of students of each program, it is easily concluded that the Bachelor Degree in Information Management and the Masters Degree in Statistics and Information Management are the ones attracting more students. Another popular specialisation is Business Intelligence and Knowledge Management again for the Masters in Information Management.

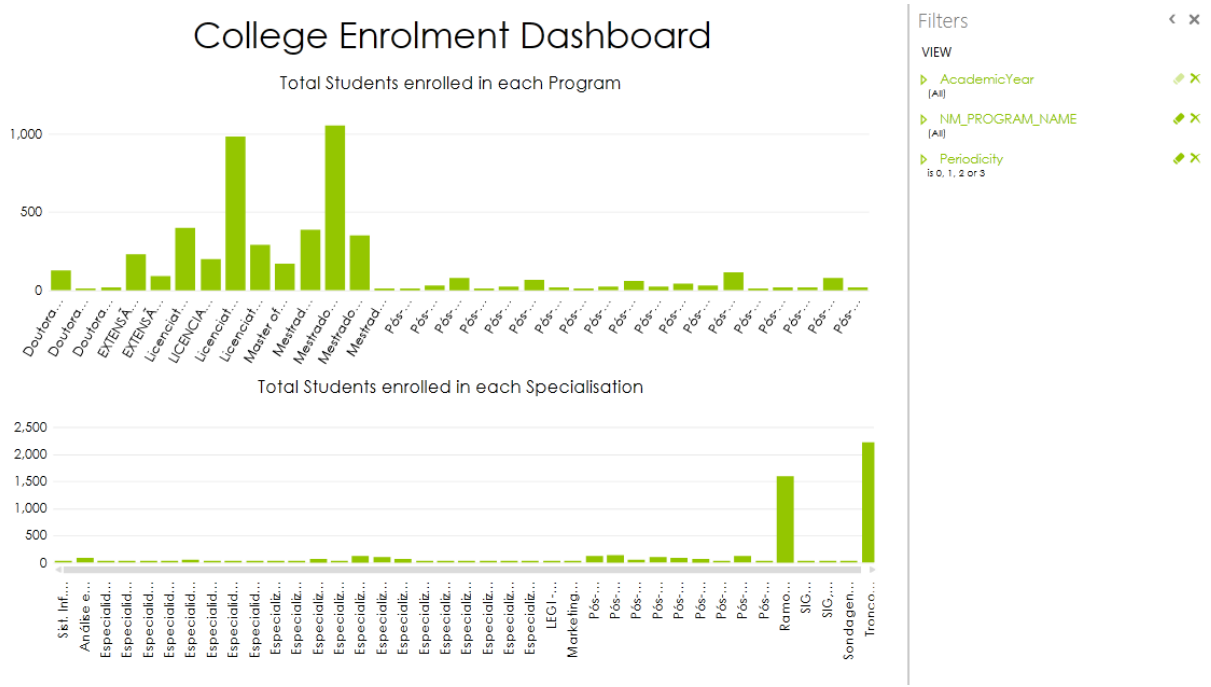

Figure 3.29 – Total students enrolled in a Program and Specialisation

3.4.3. Course Enrolment Final Grade

As for the Course Enrolment Final Grade chapter, it is probably one of the most relevant as it contains informantion on every course enrolled by students and their correspondent grades (a student might have no grade or multiple grades in a course).

Figure 3.30 top graph displays the total students enrolled in Nova IMS per academic year and fragmented by class type. This figure is especially important in helping out to discover what are the types of classes that Nova IMS should focus on more. The main conclusions that are possible to draw from this graph is that almost all classed are given in a Theoretical Practice format and that the total number of students have been increasing over the years as was also presented in Figure 3.29.

Regarding the bottom graph of Figure 3.30, it represents the average grades of the students, making it possible to conclude that there aren't many discrepancies between the semesters. Nevertheless, this graph is relevant in the sense that it might help Professors assess the general difficulty of a subject, by identifying years with especially higher or lower grades.

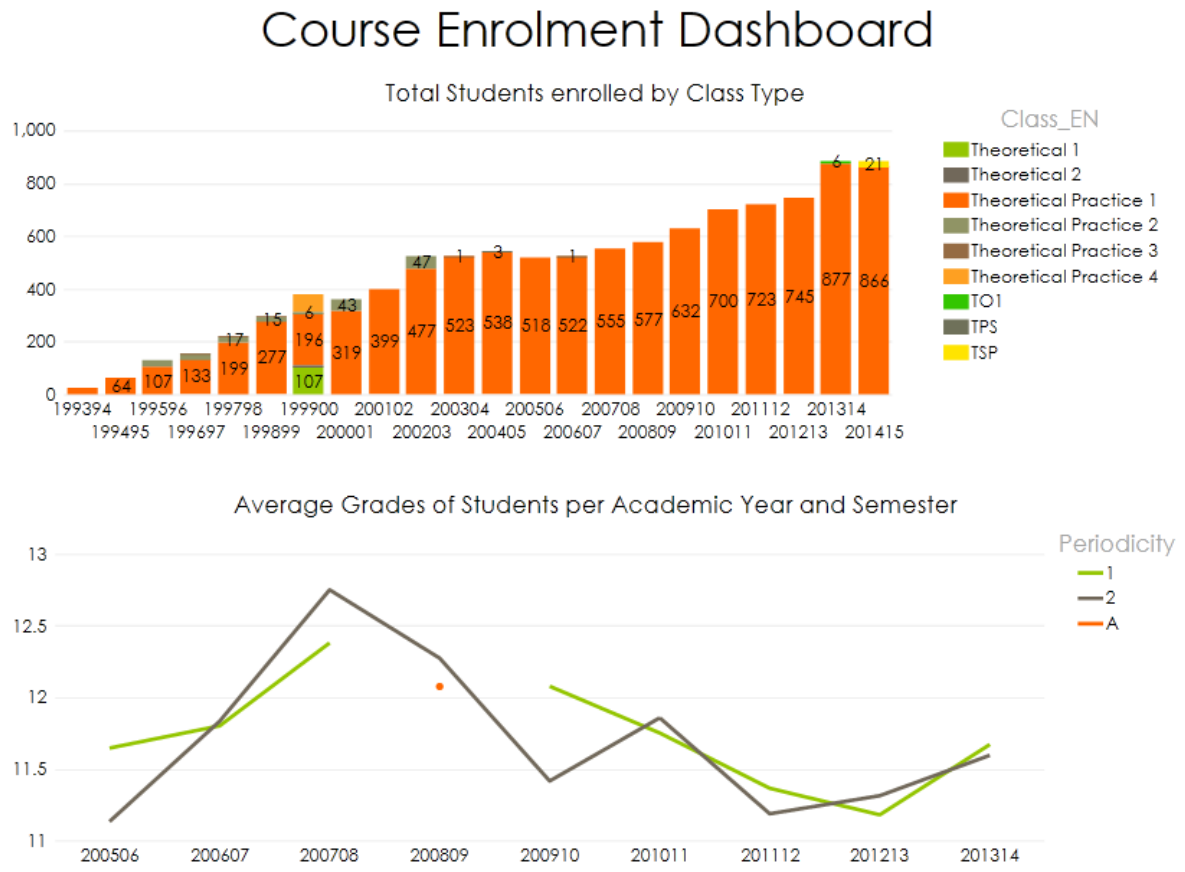

Figure 3.30 – Total Students enrolled by Class Type and the Average Grade of Students by Year and by Periodicity

Figure 3.31 presents, in turn, information on the total number of students at Nova IMS as well as on their final grades by term across the different academic years. Starting with the top graph, representing the total number of students, it is possible to notice that the number of students enrolled in the 2nd epoch is almost the same as the number in the 1st epoch, which may be due to the high number of not approved courses presented in Figure 3.28 or students preferring to go to the 2nd epoch, in an attempt to improve their grades. As for the bottom graph containing students' average grades by term, one can verify, again, that there aren't many discrepancies between the academic years (average of 12 to 13) in terms of average grade. Another fact is that the 1st epoch grades are always better than the 2nd epoch. Both graphs have been filtered for the null values of Term, not considering students who have not enrolled in the 1st nor 2nd epoch. Additionally, one can also filter this graph by a specific student (using their ID numbers only as theirs names were not provided), allowing professors and faculty staff members to view a specific student's enrolment and grades.

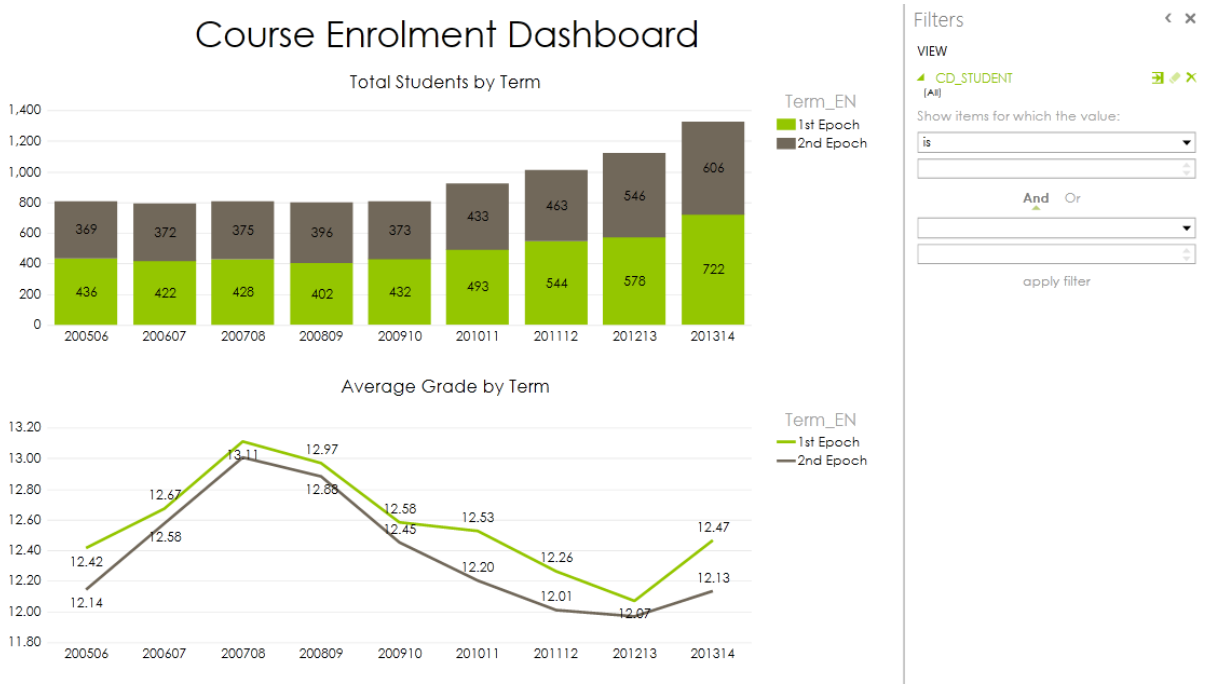

Figure 3.31 – Total Students and Average Students Grade by Term

Finally considering Figure 3.32, the first graph presents the Nova IMS' courses with the higher students' enrolment, filtering for the ones with more than 130 students. The objective behind this graph is helping the school by demonstrating which courses have more students and then proceed to understand what are the reasons behind it. Therefore, the courses with most students enrolled are "Computação I", "Computação II", "Matematica I" and "Matematica II". As for the bottom graph in Figure 3.32, it's purpose it's to display the status of Nova IMS's students, whether they fall into the category of being approved or disapproved. The graph demonstrates that despite the fact that more students are enrolling into Nova IMS, there percentage of approved courses keeps increasing. The gaps in the graph information may be due to bad quality information and/or because the courses didn´t have more than 130 students enrolled in all presented years.

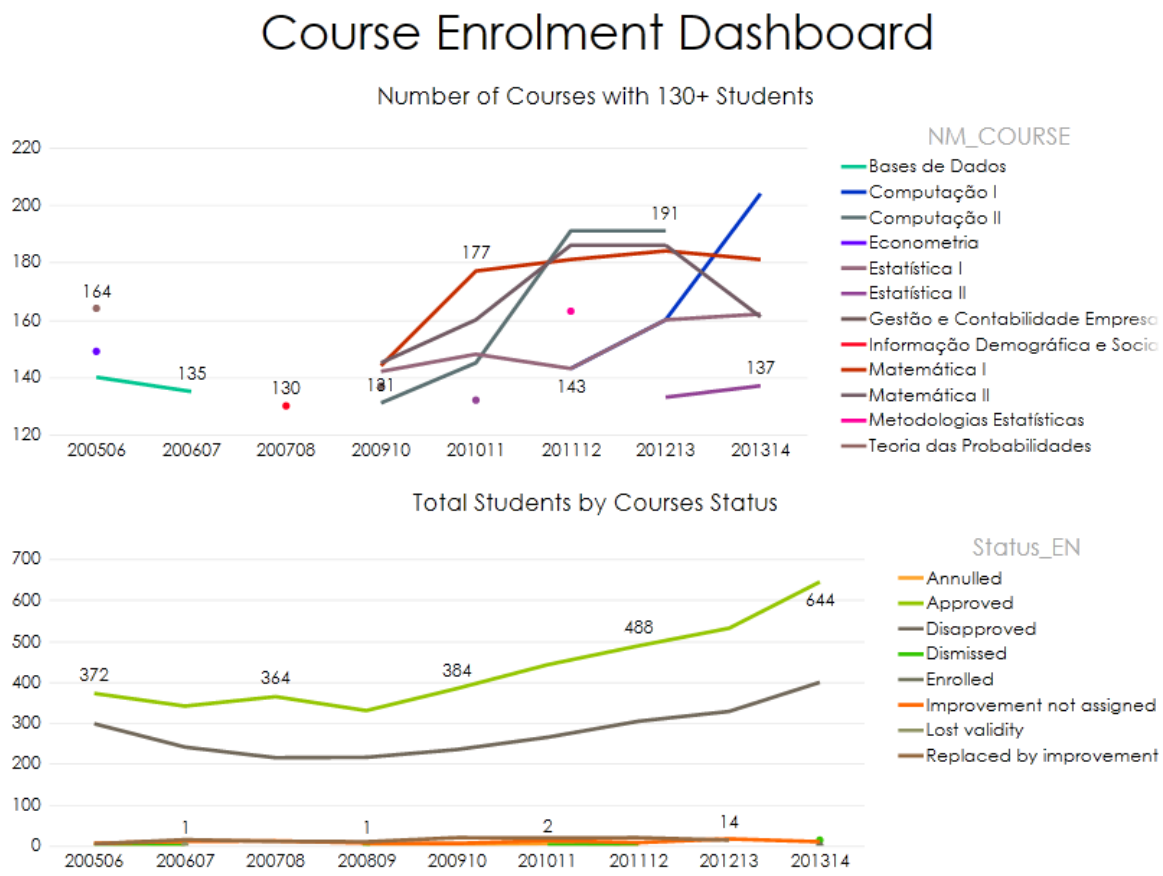

Figure 3.32 – Number of Course with 130+ Students and Total Students by Course Status

3.4.4. Partial Grades

Continuing on to the Partial Grades factual table, it presents information on Nova IMS' students' partial grades in a specific course. Therefore, this chapter will focus on offering some insights about a student's partial grades.

Figure 3.33 then refers to students' partial grades for each course, which is relevant in the sense that it aids the school providing Nova IMS with an exploratory base where it can search for any partial grade. In this specific figure, the information was translated into a table, as a user should be able to explore the data by himself. Moreover, on the right-hand side of the table it is possible to perceive that there are a large number of options available for the user to filter the data as needed. Options like the academic year, students' number or the course will help the user to analyse the sought information.

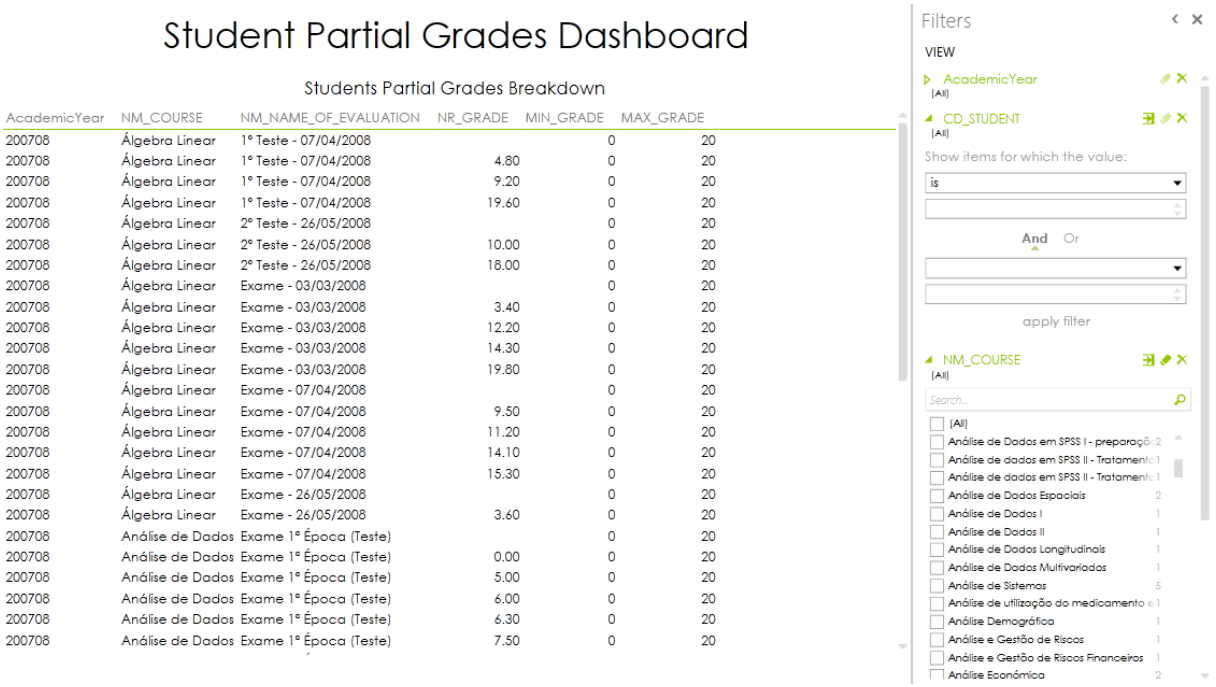

Figure 3.33 – Students Partial Grades Table

3.4.5. Work Allocation

Finally, the Work Allocation chapter intends to focus on the total number of hours professors teaching at Nova IMS spend in each course.

Hence, Figure 3.34 displays the total working hours of all professors at Nova IMS. This dashboard was made so the user interactively chooses what information he wishes to display, therefore helping the school better distributing professors per classes. In the X-axis it is possible to find the years and in the legend, the Periodicity (first semester or second semester). The filtering options allow the user to select one or more professors and one or more courses. Particularly regarding Figure 3.34, it does not contain any specific course or professor, which allows Nova IMS to see the total hours of every single professor for every course. With such graph, it is possible to conclude that the hours have been increasing, which is also aligned with the fact that Nova IMS is receiving more students, making it necessary to have more classes.

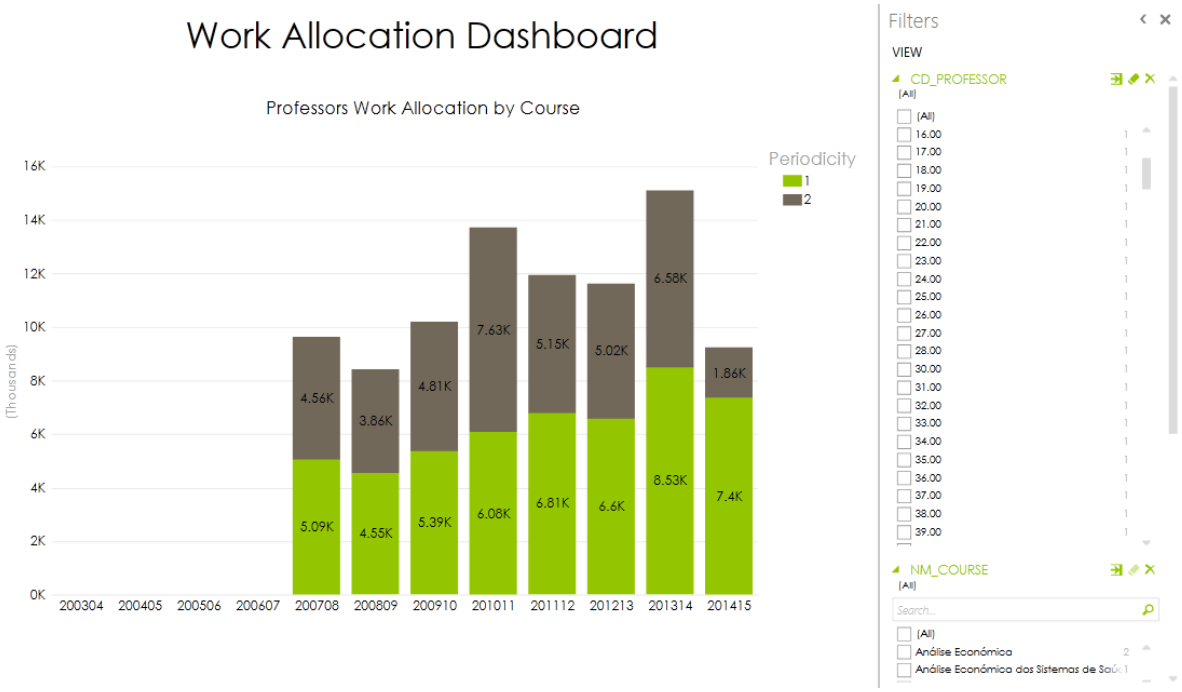

Figure 3.34 – Professors Work Allocation by Course

4. CONCLUSIONS

This project aimed to develop a business intelligence solution to improve the understanding of data being collected by the Academic Information System of Nova Information Management School. With that in mind, and after a thorough analysis of the data provided by NOVA IMS, we chose to build a Data Warehouse following Kimball's approach. Furthermore, a set of dashboards was also designed for further data analysis.

Meanwhile, with the development of this project we were also able to gather some interesting facts such as the fact that Students in Nova IMS come from all around the world and especially from the PALOP's (Portuguese-speaking African countries) and Brazil, also regarding the current students' population from NOVA IMS is 41% women and 59% men. Additionally, it is possible to notice an increase that has become more and more relevant since the academic year for 2009/2010, and finally the fact that the courses "Computação I", "Computação II", "Matematica I" and "Matematica II" are the ones which register most students, possibly due to the fact that these are considered the hardest courses of the Bachelor's degree and where students' fail the most.

As for challenges faced during the elaboration of this report, there were some limitations on the data provided (metadata), which led to a slower pace especially in the beginning of the project. Nevertheless, as we got deeper and deeper into the project, we were able to gather the missing information and, in the end, only a few fields were left without an answer, which had near zero impact in the report. Another challenge was the fact that had we been provided with a document with the explanation of the fields, the Data Warehouse design and the SSIS processes would have been done faster and we could have spent more time in reporting, rather than spending it on what to do with some of the fields. Moreover, having a normalized database would also have helped the construction of the Data Warehouse and facilitate the transformation in the ETL process.

Finally, for the future we recommend that, there is more data quality analysis using SSIS, that surveys are done to the Academic Services or school board so as to better understand their expectations and what they require, that a real Project Implementation is performed with different environments and including testing, and that a generic Schools' Star Schema is built so that is possible to implement it in other schools.

5. BIBLIOGRAPHY

Ballard, C., Herreman, D., Schau, D., Bell, R., Kim, E., & Valencic, A. (1998). *Data Modeling Techniques for Data Warehousing.* San Jose: IBM Corp.

Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., . . . Thomas, D. (2001, 11 13). *Manifesto for Agile Software Development*. Retrieved from agilemanifesto: http://agilemanifesto.org/

Boateng, O., Singh, J., Greeshma, & Singh, P. (2012). Data Warehousing. *Business Intelligence Journal*, 11.

Business-software. (2016, 1 1). *Top 10 Business Intelligence Software*. Retrieved from business-software: http://c3330831.r31.cf0.rackcdn.com/top_10_bi.pdf

Cebotarean, E. (2011). Business Intelligence. *Journal of Knowledge Management, Economics and Information Technology*, 12.

Chaudhuri, S., Dayal, U., & Narasayya, V. (2011, August). An Overview of Business Intelligence Technology. *Communications of the ACM*, p. 11.

Chong-Leng Goh, J., Pan, S. L., & Zuo, M. (2013). Developing the Agile IS Development Practices in Large-Scale IT Projects: The Trust-Mediated Organizational Controls and IT Project Team Capabilities Perspectives. *Journal of the Association for Information Systems*, 36.

Couture, N. (2013). Implementing an Enterprise Data Quality Strategy. *Business Intelligence Journal*, 7.

Ed Price MSFT; Peter Geelen MSFT. (2015, Novmber 3). *Differences Between OLAP, ROLAP, MOLAP, and HOLAP*. Retrieved from Microsoft TechNet Wiki: http://social.technet.microsoft.com/wiki/contents/articles/19898.differences-between-olap-rolap-molap-and-holap.aspx

Ferrari, A., & Russo, M. (2014). *Microsoft Excel 2013: Building Data Models with PowerPivot.* United States of America: Microsoft Press.

Few, S. (2007, 1 10). Data Visualization Past, Present and Future. *Perceptual Edge*, p. 12.

Grech, T. (2015). The intersection of agile and waterfall. *Industrial Engineer*, 4.

Hahn, G., & Packowski, J. (2015). A perspective on applications of in-memory analytics in supply chain management. *Elsevier*, 8.

Horakova, M., & Skalska, H. (2013). Business Intelligence and Implementation in a Small Enterprise. *Journal of Systems Integration*, 12.

Inmon, W. H. (2002). *Building the Data Warehouse, Third Edition.* John Wiley & Sons, Inc.

Ionel, N. (2009). Agile Software Development Methodologies: An Overview of the current State of Research. *University of Economic Studies (ASE) Bucharest Faculty of Management*, 6.

Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J., Ramakrishnan, R., & Shahabi, C. (2014). Big Data and Its Technical Challenges. *COMMUNICATIONS OF THE ACM*, 10.

Kim, G.-H., Trimi, S., & Chung, J.-H. (2014). Big-Data Applications in the Government Sector. *communications of the acm*, 9.

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

Lee, D., & Baby, D. (2013). Managing Dynamic Risks in Global IT Projects: Agile Risk-Management using the Principles of Service-Oriented Architecture. *International Journal of Information Technology & Decision Making*, 31.

Luhn, H. (1958). A Business Intelligence System. *IBM Journal*, 6.

Mahadevan, L., Kettinger, W. J., & Meservy, T. (2015). Running on Hybrid: Control Changes when Introducing an Agile Methodology in a Traditional "Waterfall" System Development Environment. *Communications of the Association for Information Systems*, 28.

Microsoft. (2012, 1 1). *Learn About PowerPivot Capabilities*. Retrieved from Microsoft Developer Network: https://msdn.microsoft.com/en-us/library/gg399131(v=sql.110).aspx

Milanov, G., & Njegus, A. (2012). Analysis of Return on Investment in Different Types of Agile Software Development Project Teams. *Informatica Economică*, 13.

Moody, D., & Kortink, M. (2000). From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design.

Moorthy, J., Lahiri, R., Biswas, N., Sanyal, D., Ranjan, J., Nanath, K., & Ghosh, P. (2015). Big Data: Prospects and Challenges. *VIKALPA The Journal for Decision Makers*, 24.

Muntean, M. (2014). Toward Agile BI By Using In-Memory Analytics. *Informatica Economică*, 16.

Obeidat, M., North, M., Richardson, R., & Rattanak, V. (2015). Business Intelligence Technology, Applications, and Trends. *International Management Review*, 11.

Petrenko, M., Rada, A., Fitzsimons, G., McCallig, E., & Zuzarte, C. (2012). *Best Practices: Physical database design for data warehouse environments.* IBM Corp.

Presthus, W., & Sæthre, S. (2015). The Secret of my Success An exploratory study of Business Intelligence management in the Norwegian Industry. *Elsevier*, 8.

PwC. (2012). *How to drive innovation and business growth.* Retrieved from https://www.pwc.com/us/en/supply-chain-management/assets/pwc-oracle-innovation-white-paper.pdf

Ranjan, J. (2009). BUSINESS INTELLIGENCE: CONCEPTS, COMPONENTS, TECHNIQUES AND BENEFITS. *Journal of Theoretical and Applied Information Technology*, 11.

Santos, M., & Ramos, I. (2009). *Business Intelligence - Tecnologias da Informação na Gestão do Conhecimento.* Lisbon: Editora de Informática.

Serrador, P., & Pinto, J. (2015). Does Agile work? - A quantitative analysis of agile project success. *International Journal of Project Management*, 12.

Sherman, R. (2014). *Business Intelligence Guidebook: From Data Integration to Analytics.* Waltham: Morgan Kaufmann/ Elsevier.

Simitsis, A., & Vassiliadis, P. (2003). A Methodology for the Conceptual Modeling of ETL Processes. *Online Proceedings for Scientific Workshops*, 12.

Singh, J., Singh, B., & Sriveni, Y. (2010). A Convenient Way From Normalized Database To Denormalized Database. *International Journal of Computer Communication and Information System (IJCCIS)*, 4.

Tamer, C., Kiley, M., Ashrafi, N., & Kuilboer, J.-P. (2013). RISKS AND BENEFITS OF BUSINESS INTELLIGENCE IN THE CLOUD. *Northeast Decision Sciences Institute Annual Meeting Proceedings*, 11.

van der Meulen, R., & Rivera, J. (2015, January 27). Gartner Says Power Shift in Business Intelligence and Analytics Will Fuel Disruption. *Gartner Newsroom*, p. 2.

Wakeling, S., Clough, P., Wyper, J., & Balmain, A. (2015). Graph Literacy and Business Intelligence: Investigating User Understanding of Dashboard Data Visualizations. *BUSINESS INTELLIGENCE JOURNAL*, 13.

Wieder, B., & Ossimitz, M.-L. (2015). The impact of Business Intelligence on the quality of decision making – a mediation model. *Elsevier*, 9.

6. ANNEXES

```
SELECT
  AcademicYearPeriodicityChar,
  cast(convert(int,AcademicYear) as varchar) + '' + cast(PeriodicityInt as varchar) as
AcademicYearPeriodicityInt,
  AcademicYear,
  Periodicity
FROM(
SELECT
  AcademicYear + Periodicity as AcademicYearPeriodicityChar,
  cast(CASE
       WHEN Periodicity = 'A'
         THEN '0'
       ELSE Periodicity
         END AS int) as PeriodicityInt,
  AcademicYear,
  Periodicity
FROM(
SELECT DISTINCT
  convert(varchar(6),left([CD_YEARS_SEMESTER],6)) as AcademicYear,
   null as Periodicity
FROM [IsegiOnline_Source].[dbo].[PartialGrades]
UNION
SELECT DISTINCT
  convert(varchar(6), left([CD_YEARS_SEMESTER],6)) as AcademicYear,
  convert(varchar(1), right([CD_YEARS_SEMESTER],1)) as Periodicity
FROM [IsegiOnline_Source].[dbo].[WorkAllocation]
UNION
SELECT DISTINCT
  convert(varchar(6), left([CD_YEARS_SEMESTER],6)) as AcademicYear,
  convert(varchar(1), right([CD_YEARS_SEMESTER],1)) as Periodicity
FROM [IsegiOnline_Source].[dbo].[Enrollment]
UNION
SELECT DISTINCT
  convert(varchar(6), left([CD_YEARS_SEMESTER],6)) as AcademicYear,
  convert(varchar(1), right([CD_YEARS_SEMESTER],1)) as Periodicity
FROM [IsegiOnline_Source].[dbo].[CourseEnrollment]
) as tbl
) as tbl1
WHERE Periodicity = 'A'
OR Periodicity = '0'
OR Periodicity = '1'
OR Periodicity = '2'
OR Periodicity = '3'
OR Periodicity is null
```

Table 6.1 – SQL command for select the source data for Dimension Academic Year Semester

```
SELECT
  PG.[CD_PARTIAL_GRADE],
  PG.[NM_NAME_OF_EVALUATION],
  PGT.[NM_PARTIAL_GRADE_TYPE_PT],
  PGT.[NM_PARTIAL_GRADE_TYPE_EN],
  PG.[MAX_GRADE],
  PG.[MIN_GRADE]
FROM [IsegiOnline_Source].[dbo].[PartialGrades] PG
INNER JOIN [IsegiOnline_Source].[dbo].[PartialGradeTypes] PGT
ON PG.CD_PARTIAL_GRADE_TYPE = PGT.CD_PARTIAL_GRADE_TYPE
```

Table 6.2 – SQL command for select the source data for Dimension Partial Grade Type

```
SELECT
  PR.[CD_PROGRAM],
  PL.[CD_PLAN],
  SP.[CD_SPECIALIZATION],
  PR.[NM_PROGRAM_NAME],
  PR.[CD_DEGREE_CODE],
  PR.[NM_ABREV_PROGRAM_NAME],
  PL.[NM_PLAN],
  SP.[NM_SPECIALIZATION]
FROM [IsegiOnline_Source].[dbo].[Specializations] SP
INNER JOIN [IsegiOnline_Source].[dbo].[Plans] PL
ON SP.CD_PROGRAM = PL.CD_PROGRAM
AND SP.CD_PLAN = PL.CD_PLAN
INNER JOIN [IsegiOnline_Source].[dbo].[Programs] PR
ON SP.CD_PROGRAM = PR.CD_PROGRAM
AND PL.CD_PROGRAM = PR.CD_PROGRAM
```

Table 6.3 – SQL command for select the source data for Dimension Program Specialization Plan

```
SELECT
  ST.[CD_PROGRAM],
  ST.[CD_STUDENT],
  ST.[CD_GENDER],
  ST.[CD_CIVIL_STATUS],
  ST.[CD_NATIONALITY],
  ST.[CD_COUNTRYOFBIRTH],
  ST.[CD_ENTRYCODE],
  ST.[DT_DATEOFENTRY],
  ST.[NR_ENTRYGRADE],
  ST.[DT_DATEOFREENTRY],
  ST.[DT_DATEOFFINALGRADE],
  ST.[NR_FINALGRADE],
  ST.[DT_DATEOFPARTIALGRADE],
  ST.[NR_PARTIALGRADE],
  ST.[DT_DATEOFINTERN],
  ST.[NR_GRADEOFINTERN],
  ST.[DT_DATEOFGRAGEIMPROV],
  ST.[NR_GRADEIMPROV],
  ST.[DT_DATEOFDIPLOMA],
  ST.[CD_FISCALZONE],
  ST.[DT_DATEOFPARTIALDIPLOMA],
  ST.[CD_TRANEFERINSTITUTION],
  ST.[CD_TITLE],
  ST.[CD_EMPLOYER],
  ST.[CD_PROFESSION],
  ST.[DT_DATEDIPLOMAEMITION],
  ST.[DT_DATEDIPLOMAWITHDRAW],
  ST.[CD_STUDENT_ACADEMICBAK],
  ST.[BIRTHYEAR]
FROM [IsegiOnline_Source].[dbo].[Students] ST
INNER JOIN
  (
  SELECT
    COUNT(ST1.cd_student) as StudentCount,
    ST1.[CD_STUDENT]
  FROM [IsegiOnline_Source].[dbo].[Students] ST1
```

```
  GROUP BY [CD_STUDENT]
  ) as ST2
ON ST2.StudentCount = 1
AND ST2.CD_STUDENT = ST.CD_STUDENT
```

Table 6.4 – SQL command for select the source data for Dimension Students

```
/***** Create Stored Procedure in NOVAIMSOnline_DW and Run SP to Fill Time Dimension with
Values****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[FillDimTime]
as
BEGIN
--Specify Total Number of Hours You need to fill in Time Dimension
DECLARE @Size INTEGER
--iF @Size=32 THEN This will Fill values Upto 32:59 hr in Time Dimension
Set @Size=23
DECLARE @hour INTEGER
DECLARE @minute INTEGER
DECLARE @second INTEGER
DECLARE @k INTEGER
DECLARE @TimeAltKey INTEGER
DECLARE @TimeInSeconds INTEGER
DECLARE @Time30 varchar(25)
DECLARE @Hour30 varchar(4)
DECLARE @Minute30 varchar(4)
DECLARE @Second30 varchar(4)
DECLARE @HourBucket varchar(15)
DECLARE @HourBucketGroupKey int
DECLARE @DayTimeBucket varchar(100)
DECLARE @DayTimeBucketGroupKey int
SET @hour = 0
SET @minute = 0
SET @second = 0
SET @k = 0
SET @TimeAltKey = 0
WHILE(@hour<= @Size )
BEGIN
if (@hour <10 )
begin
set @Hour30 = '0' + cast( @hour as varchar(10))
end
else
begin
set @Hour30 = @hour
end
--Create Hour Bucket Value
set @HourBucket= @Hour30+':00' +'-' +@Hour30+':59'
WHILE(@minute <= 59)
BEGIN
WHILE(@second <= 59)
BEGIN
set @TimeAltKey = @hour *10000 +@minute*100 +@second
set @TimeInSeconds =@hour * 3600 + @minute *60 +@second
If @minute <10
begin
set @Minute30 = '0' + cast ( @minute as varchar(10) )
end
```

```
else
begin
set @Minute30 = @minute
end
if @second <10
begin
set @Second30 = '0' + cast ( @second as varchar(10) )
end
else
begin
set @Second30 = @second
end
--Concatenate values for Time30
set @Time30 = @Hour30 +':'+@Minute30 +':'+@Second30
--DayTimeBucketGroupKey can be used in Sorting of DayTime Bucket In proper Order
SELECT @DayTimeBucketGroupKey =
CASE
WHEN (@TimeAltKey >= 00000 AND @TimeAltKey <= 25959) THEN 0
WHEN (@TimeAltKey >= 30000 AND @TimeAltKey <= 65959) THEN 1
WHEN (@TimeAltKey >= 70000 AND @TimeAltKey <= 85959) THEN 2
WHEN (@TimeAltKey >= 90000 AND @TimeAltKey <= 115959) THEN 3
WHEN (@TimeAltKey >= 120000 AND @TimeAltKey <= 135959) THEN 4
WHEN (@TimeAltKey >= 140000 AND @TimeAltKey <= 155959) THEN 5
WHEN (@TimeAltKey >= 50000 AND @TimeAltKey <= 175959) THEN 6
WHEN (@TimeAltKey >= 180000 AND @TimeAltKey <= 235959) THEN 7
WHEN (@TimeAltKey >= 240000) THEN 8
END
--print @DayTimeBucketGroupKey
-- DayTimeBucket Time Divided in Specific Time Zone
-- So Data can Be Grouped as per Bucket for Analyzing as per time of day
SELECT @DayTimeBucket =
CASE
WHEN (@TimeAltKey >= 00000 AND @TimeAltKey <= 25959) THEN 'Late Night (00:00 AM To 02:59 AM)'
WHEN (@TimeAltKey >= 30000 AND @TimeAltKey <= 65959) THEN 'Early Morning (03:00 AM To 6:59 AM)'
WHEN (@TimeAltKey >= 70000 AND @TimeAltKey <= 85959) THEN 'AM Peak (7:00 AM To 8:59 AM)'
WHEN (@TimeAltKey >= 90000 AND @TimeAltKey <= 115959) THEN 'Mid Morning (9:00 AM To 11:59 AM)'
WHEN (@TimeAltKey >= 120000 AND @TimeAltKey <= 135959) THEN 'Lunch (12:00 PM To 13:59 PM)'
WHEN (@TimeAltKey >= 140000 AND @TimeAltKey <= 155959) THEN 'Mid Afternoon (14:00 PM To 15:59
PM)'
WHEN (@TimeAltKey >= 50000 AND @TimeAltKey <= 175959) THEN 'PM Peak (16:00 PM To 17:59 PM)'
WHEN (@TimeAltKey >= 180000 AND @TimeAltKey <= 235959) THEN 'Evening (18:00 PM To 23:59 PM)'
WHEN (@TimeAltKey >= 240000) THEN 'Previous Day Late Night (24:00 PM to '+cast( @Size as
varchar(10)) +':00 PM )'
END
-- print @DayTimeBucket
INSERT into NOVAIMSOnline_DW.dbo.[DimTime] (TimeKey,TimeAltKey,[Time30],[Hour30]
,[MinuteNumber],[SecondNumber],[TimeInSecond],[HourlyBucket],DayTimeBucketGroupKey,DayTimeBucket)
VALUES (@k,@TimeAltKey ,@Time30 ,@hour
,@minute,@Second,@TimeInSeconds,@HourBucket,@DayTimeBucketGroupKey,@DayTimeBucket )
SET @second = @second + 1
SET @k = @k + 1
END
SET @minute = @minute + 1
SET @second = 0
END
SET @hour = @hour + 1
SET @minute =0
END
END
Go
Exec [FillDimTime]
go
```

Table 6.5 – SQL command to populate Dimension Time[13]

```
SELECT
  ENR.[CD_YEARS_SEMESTER],
  ENR.[CD_PROGRAM],
  ENR.[CD_STUDENT],
  left(ENR.[CD_Y_S],1) as CD_Y_S,
  ISNULL(cast(ENR.[DT_DATEOFENROLLMENT] as date), '1900-01-01')as DateOfEnrollment,
  ISNULL(cast(ENR.[DT_DATEOFENROLLMENT] as time), '')as TimeOfEnrollment,
  ISNULL(cast(ENR.[DT_DATEENDOFENROLLMENT] as date), '1900-01-01')as
DateEndOfEnrollment,
  ISNULL(cast(ENR.[DT_DATEENDOFENROLLMENT] as time), '')as TimeEndOfEnrollment,
  ENR.[CD_PLAN],
  ENR.[CD_SPECIALIZATION],
  ENR.[NR_COURSES],
  ENR.[NR_COURSES_APPROVED],
  ENR.[NR_COURSES_NOTAPPROVED],
  ENR.[NR_CREDITS],
  ENR.[NR_CREDITS_APPROVED],
  ENR.[NR_CREDITS_NOTAPPROVED],
  ENR.[NR_CREDITS_ECTS],
  ENR.[NR_CREDITS_ECTS_APPROVED],
  ENR.[NR_CREDITS_ECTS_NOT_APPROVED]
FROM [IsegiOnline_Source].[dbo].[Enrollment] ENR
```

Table 6.6 – SQL command for select the source data for Fact College Enrolment

```
SELECT DISTINCT
  CE.[CD_PROGRAM],
  CE.[CD_PLAN],
  CE.[CD_SPECIALIZATION],
  CE.[CD_STUDENT],
  CE.[CD_COURSE],
  CE.[CD_DURATION],
  CE.[CD_YEARS_SEMESTER],
  CE.[CD_CLASS],
  left(CE.[CD_Y_S],1) as CD_Y_S,
  ISNULL(cast(CE.[DT_DATEOFENROLLMENT] as date), '1900-01-01') as DateOfEnrollment,
  ISNULL(cast(CE.[DT_DATEOFENROLLMENT] as time), '') as TimeEndOfEnrollment,
  CE.[NR_FINAL_GRADE],
  CE.[CD_STATUS],
  ISNULL(FG.[CD_TERM], '-') as Term,
  ISNULL(CONVERT(date, FG.[DT_DATEOFEXAM],103), '1900-01-01') as DateOfExam,
  FG.[NR_GRADE]
FROM [IsegiOnline_Source].[dbo].[CourseEnrollment] CE
LEFT JOIN [IsegiOnline_Source].[dbo].[FinalGrades] FG
ON CE.CD_PROGRAM = FG.CD_PROGRAM
AND CE.CD_COURSE = FG.CD_COURSE
AND CE.CD_STUDENT = FG.CD_STUDENT
AND CE.CD_DURATION = FG.CD_DURATION
AND CE.[CD_YEARS_SEMESTER] = FG.[CD_YEARS_SEMESTER]
```

Table 6.7 – SQL command for select the source data for Fact Course Enrolment Final Grade

[13] This dimension has been retrieved and adapted from http://www.codeproject.com/Tips/642912/Create-Populate-Time-Dimension-with-Hourplus-Va

```
SELECT
  WA.[CD_PROFESSOR],
  WA.[CD_COURSE],
  WA.[CD_YEARS_SEMESTER],
  WA.[HEIGHT_MULTIPLIER],
  WA.[TOTAL_HOURS],
  WA.[LOAD_MULTIPLIER]
FROM [IsegiOnline_Source].[dbo].[WorkAllocation] WA
INNER JOIN(
SELECT
  [CD_PROFESSOR],
  [CD_COURSE],
  [CD_YEARS_SEMESTER],
  count (distinct [CD_PROGRAM]) as ProgramCount
FROM [IsegiOnline_Source].[dbo].[WorkAllocation]
GROUP BY
  [CD_PROFESSOR],
  [CD_COURSE],
  [CD_YEARS_SEMESTER]
  ) as WA1
  ON WA1.CD_PROFESSOR = WA.CD_PROFESSOR
  AND WA1.CD_COURSE = WA.CD_COURSE
  AND WA1.CD_YEARS_SEMESTER = WA.CD_YEARS_SEMESTER
  AND WA1.ProgramCount = 1
```

Table 6.8 – SQL command for select the source data for Fact Work Allocation

```
SELECT
  SPG.[CD_PARTIAL_GRADE],
  SPG.[CD_STUDENT],
  PG.[CD_COURSE],
  PG.[CD_YEARS_SEMESTER],
  PG.[CD_PROFESSOR],
  SPG.[NR_GRADE],
  ISNULL(cast(SPG.[DT_LAST_CHANGE] as date), '1900-01-01') as DateLastChange,
  ISNULL(cast(SPG.[DT_LAST_CHANGE] as time), '') as TimeLastChange
FROM [IsegiOnline_Source].[dbo].[StudentPartialGrades] SPG
INNER JOIN [IsegiOnline_Source].[dbo].[PartialGrades] PG
ON SPG.CD_PARTIAL_GRADE = PG.CD_PARTIAL_GRADE
```

Table 6.9 – SQL command for select the source data for Fact Partial Grades

```
SELECT
  [AcademicYearSemesterID],
  [AcademicYearPeriodicityChar],
  convert(int, left([AcademicYearPeriodicityInt],6)) as AcademicYearPeriodicityInt,
  convert(int, [AcademicYear]) as AcademicYear,
  [Periodicity]
FROM [NOVAIMSOnline_DW].[dbo].[DimAcademicYearSemester]
WHERE [Periodicity] IS NULL
```

Table 6.10 – SQL command for lookup AcademicYearSemesterID in Fact Partial Grades

```
SELECT
  [ProfessorsID],
  [CD_PROFESSOR],
  [ADDRESS_PLACE],
```

```
  [CD_ADDRESS_ZIPCODE],
  [DEGREE],
  [IsLatest]
FROM [NOVAIMSOnline_DW].[dbo].[DimProfessors]
WHERE [IsLatest] = 1
```

Table 6.11 – SQL command to select only DimProfessors latest data

```
SELECT
  [ProgSpecPlanID],
  [CD_PROGRAM],
  [CD_SPECIALIZATION],
  [CD_PLAN],
  [NM_PROGRAM_NAME],
  [CD_DEGREE_CODE],
  [NM_ABREV_PROGRAM_NAME],
  [NM_SPECIALIZATION],
  [NM_PLAN],
  [ProgSpecPlanEffectiveFrom],
  [ProgSpecPlanEffectiveTo]
FROM [NOVAIMSOnline_DW].[dbo].[DimProgSpecPlan]
WHERE [ProgSpecPlanEffectiveTo] is null
```

Table 6.12 – SQL command to select DimProgSpecPlan in force data