TÉCNICO LISBOA

Degree Coordination Decision Support System

Gonçalo Nuno Paulos Lopes

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Helena Isabel De Jesus Galhardas
Prof. Nuno João Neves Mamede

Examination Committee

Chairperson: Prof. Maria Luísa Torres Ribeiro Marques da Silva Coheur
Supervisor: Prof. Helena Isabel De Jesus Galhardas
Member of the Committee: Prof. Cláudia Martins Antunes

January 2021

Acknowledgments

First and foremost, to my supervisors, Professors Helena Galhardas and Nuno Mamede, for their continuous support and guidance. For providing me feedback every week during our meetings, for correcting so many Excel files with me, for reviewing this document several times and much more.

To my friends, Catarina Conceição, Ricardo Sousa, João Guerreiro and Bruno Sousa, that were working on their thesis at the same time as me. We supported each other by discussing our problems and contributing with ideas and opinions in a sort of symbiosis.

To everyone who participated in the usability test sessions, for the time they spent on such an indispensable part of this project.

My sincere thanks to all of you!

Resumo

O coordenador de um curso superior deve tomar decisões estratégicas para garantir um excelente desempenho do curso. A tomada de decisão do coordenador de curso deve ser sustentada por dados relacionados com as principais áreas que compõem o curso, dados que, em muitos casos, são obtidos e processados manualmente. Estas tarefas exigem tempo e esforço por parte do coordenador e estão sujeitas a erros humanos, que podem ser atenuados com a introdução de uma aplicação de software para automatizar as tarefas.

Neste documento, propomos um sistema de apoio à decisão para coordenação de cursos, baseado numa data warehouse, que realiza automaticamente as tarefas de obtenção de dados e o seu armazenamento num formato adequado para análise. Os dados são usados para gerar dashboards, que indicam o desempenho das diversas áreas do curso, e que fornecem informações úteis que permitem uma tomada de decisão mais informada por parte do coordenador. A solução proposta visa dar resposta a um conjunto de requisitos definidos, e é guiada por uma análise da literatura e trabalhos relevantes sobre sistemas de apoio à decisão no ensino superior, bem como tecnologias de apoio à implementação do projeto.

Adicionalmente, o sistema proposto foi avaliado, tendo sido determinada a sua capacidade de obter todos os principais indicadores de desempenho. A comparação dos dados armazenados pelo sistema com os dados processados pelo coordenador de curso, resultou numa correspondência de *100%* dos registos comparados. Através de testes de usabilidade, foi obtida uma pontuação média de *85.18* em 100, na escala de usabilidade de sistema.

Palavras-chave: Sistema de Apoio à Decisão, Data Warehouse, Ensino Superior, Coordenação de Curso

Abstract

The coordinator of a higher education degree has to make strategic decisions for ensuring the excellence of the degree's performance. The degree coordinator's decision making needs to be supported by data regarding the degree's main areas, which, in many cases, is manually gathered and processed. These tasks require time and effort from the coordinator and are prone to human error, which could be mitigated by introducing a software application to automate the tasks.

In this document, we propose a decision support system for degree coordination, based on a data warehouse, that automatically performs the tasks of gathering data and storing it in a format suitable for analysis. The data is used to generate dashboards, that indicate the performance of the various areas of the degree, ultimately providing useful insights that will enhance the coordinator's decision making. The proposed solution addresses a set of defined business requirements, and was guided by an analysis of literature and relevant works regarding decision support systems and higher education, as well as technologies for supporting the implementation of the project.

Furthermore, the system proposed was evaluated, having been determined to be capable of obtaining all the required key performance indicators. The data stored by our system had *100%* of matches when compared with data processed by the degree coordinator. Usability tests determined an average System Usability Scale score of *85.18* out of 100.

Keywords: Decision Support System, Data Warehouse, Higher Education, Degree Coordination

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the increasing advances in the areas of science and technology, the importance of training students with rigour and excellence is crucial. Having excellent students can be achieved by having a degree with an excellent organization, which must be ensured by the degree coordinator.

The degree coordinator's decision-making has a strong impact on all parties involved in the degree, mainly on students and instructors. Therefore, it is fundamental for the coordinator to have access to the correct data on how the degree works, to understand which areas need to be improved.

1.1 Problem

Currently, at *Instituto Superior Técnico*[1] (IST), collecting relevant data about students and their performances is mostly a manual, laborious and time consuming process, that the degree coordinators have to perform at the end of each semester. This data is used to generate semiannual reports, describing the overall performance of courses, namely in what concerns the grades obtained by students.

Particularly, the coordinator of the Degree in Computer Science and Engineering - Taguspark (*Licenciatura em Engenharia Informática e de Computadores - Taguspark*, LEIC-T[2] in portuguese) organizes the data collected in Microsoft Excel files, which enable obtaining relevant information by manipulating the data through the application of various formulas. As there is a need for monitoring the evolution of students through time, all the data gathered throughout the semesters must be kept. This results in several files of historical data, where each file contains, in its name, an indication of the academic year and/or semester it refers to.

Given the repetitive nature of gathering and organizing data and its reliance on manual human labor, these tasks are prone to error, as even the slightest unintentional data inconsistency compromises the veracity of the data. As such, there is a need to automate this process, by introducing a software application to automatically perform the tasks of collecting data, storing it according to a specific format and producing dashboards. Automating this process would mitigate the error rate and increase the

[1] https://tecnico.ulisboa.pt
[2] https://fenix.tecnico.ulisboa.pt/cursos/leic-t

overall efficiency. Additionally, it would drastically reduce the time the coordinator spends with this process, while providing accurate data for decision-making.

1.2 Objectives

As outlined in Section 1.1, we propose a a software application, known as *IST Degree Coordination Decision Support System* (SAD-CCIST in portuguese), to automatically gather, structure and store data for analytical purposes and decision-making.

The proposed *decision support system* is based on a *data warehouse*, a repository that stores and unifies data from several operational data sources of an organization. Its development consists of designing the data warehouse, creating processes to load data from a set of input data sources into the data warehouse, and automatically generating data visualizations to be presented in dashboards.

As mentioned in Section 1.1, the LEIC-T Coordinator has gathered several sets of data related to student and course performance. The coordinator extracted this data from the Fénix[3] system, an integrated academic information system designed at IST for managing and supporting the academic tasks. Ideally, our decision support system would extract data directly from the Fénix system, but in the context of this thesis, we will be using the data gathered by the LEIC-T Coordinator.

1.3 Main Contributions

The main contributions of this work are:

- Design and implementation of a data warehouse, guided by Kimball's design methodologies, to store data related to the performance of the LEIC degree.
- Design and implementation of extract-transform-load processes, that automate the current manual process of obtaining and processing data into a suitable format.
- Design and implementation of three interactive dashboards, for student activity, student graduation and course related metrics, used for enhancing the LEIC-T Coordinator's decision-making.

Furthermore, a comprehensive evaluation was presented, on three aspects. The dimensional model of the data warehouse was evaluated using validity matrices, which determined the model to be fully capable of obtaining all the required key performance indicators. In terms of the integrity of the data stored in the data warehouse, the data was compared with data processed by the LEIC-T Coordinator and resulted in a match of *100%* of the records compared. Finally, the usability of the dashboards was assessed, by conducting usability test sessions where the test users performed several tasks and filled in a System Usability Scale questionnaire, which determined an average score of *85.18* out of 100, considered by the methodology as *excellent*.

[3] https://ciist.ist.utl.pt/projectos/fenix.php

1.4 Document Outline

This document is organized as following:

- *Chapter 2* provides a detailed background on concepts related to data warehouses and their components, as well as concepts related to the higher education domain.
- *Chapter 3* provides an overview of data warehouse design frameworks and implementation, as well as an overview of several third-party tools that can be used for implementing our solution.
- *Chapter 4* describes the business requirements: the available input files and the *key performance indicators* that should be obtained.
- *Chapter 5* describes the architecture and implementation of our decision support system;
- *Chapter 6* demonstrates how our solution was evaluated and shows the results yielded from the evaluation.
- *Chapter 7* presents the conclusion to this document and future work.

Chapter 2

Basic Concepts

In this chapter, we describe the main concepts regarding data warehousing (Section 2.1), as well as the main concepts about the higher education domain (Section 2.2).

2.1 Data Warehouse

A *data warehouse* (DW) is a central repository that gathers data from various heterogeneous data sources, storing them according to a unified schema [7]. It is typically the central component of a *decision support system* (DSS), and is responsible for enhancing the decision-making of the knowledge workers (e.g., executive, manager, analyst) [8].

The typical DW architecture is presented in Figure 2.1. Initially, there is a set of heterogeneous operational *data sources* from which relevant data is extracted. The data extracted may be transformed and loaded directly to the DW. Optionally, the data can be stored in a *data staging area* (DSA), an intermediate data store in which the data is transformed, validated and finally loaded into the DW. Once in the DW, the data is used by an *OLAP Server*, which is a layer responsible for creating multidimensional data structures, known as OLAP cubes, suited for fast analytical querying. A presentation layer aims at using the data to find useful information and presenting it to the end-user through ad-hoc querying, data mining, reporting or dashboards.

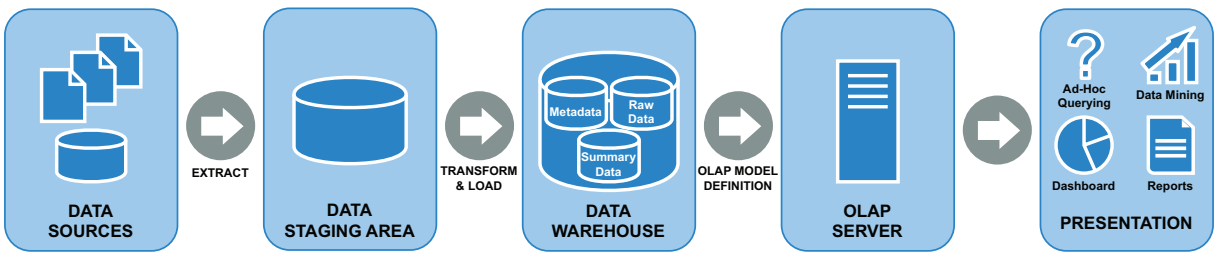

Figure 2.1: Architecture of a data warehouse

2.1.1 Dimensional Model

A *dimensional model* structures data in a way that addresses business understandability and fast query performance [2]. It introduces a view of data as consisting of *facts* linked to *dimensions*, where facts represent the focus of analysis in terms of measurements, usually numeric, whereas dimensions typically contain descriptive data that provides context to the facts.

There are several ways of defining the logical structure of the DW, through the use of different *schemata*. The schema defines the way facts and dimensions are linked. In this sense, a schema may be categorized as:

- *Star Schema*, consisting of a central fact table and a set of associated *denormalized* dimension tables. As dimensions are not normalized, the data from one dimension is stored in a single table, which results in fast queries, despite introducing redundancy.
- *Snowflake Schema*, an extension of the star schema, in which the dimension tables are *normalized*. One dimension is split into several tables. Normalization aims at a reduction in data redundancy, but may harm query performance as more join operations are required.
- *Fact Constellation*, a schema suited for complex DWs, containing multiple fact tables that share common dimensions.

2.1.2 Extract-Transform-Load Process

The process of moving data from multiple sources into a DW, known as *Extract-Transform-Load Process* (ETL) [9], consists of three distinct steps: extracting data from the operational data sources, transforming the data by cleaning and giving it an appropriate format, and loading it into the target DW.

The *extraction* step is the first step of this process and consists in retrieving a subset of data deemed relevant, following a set of established requirements. Since large volumes of data from multiple sources are involved, there is a high probability of errors and anomalies in the data [8], which means they will have to undergo further data processing. For this reason, the extracted data is often moved to an intermediate data store, the DSA, in which the transforming step is applied.

Transforming the data, the second step, imposes a standard format upon the data through the use of a series of customized techniques, to ensure its quality and integrity. These techniques handle inconsistencies and missing values, remove duplicate or redundant records, apply business constraints, among others. The transformations applied to the data using the aforementioned techniques enhance the value that the data has to the organization [2]. Different data presents different challenges, so the techniques applied also differ from one case to another. Therefore, it is fundamental to assess the data beforehand and understand which techniques to apply.

The final step is to *load* the processed data into the target DW. There are two distinct approaches that can be taken when loading the data:

- *Full loading*, the process of loading the entire source data, completely destroying any data already stored.

- *Incremental loading*, the process of loading portions of data in scheduled intervals. The incoming data is compared with the data already stored, and only the records in which changes are introduced will be loaded.

2.1.3 Online Analytical Processing

Online Analytical Processing (OLAP) is a paradigm that aims at facilitating data analysis, providing a connection between the DW and presentation layers, as shown in Figure 2.1.

The OLAP paradigm introduces the concept of *OLAP cubes*, which are data structures that aggregate data according to several dimensions (being called *hypercubes* when exceeding three dimensions). The data in a cube undergoes precalculations, indexing strategies, and other optimizations which enhance the query performance [2]. It is possible to perform a series of *OLAP operations* on cubes, which involve aggregating, summarizing and selecting data according to the dimensions. The most used OLAP operations [10] are:

- *Roll-up*: consists of a navigation from more detailed to more generic data, performing a data cube aggregation, either by rolling up the hierarchy or by dimension reduction.
- *Drill-down*: consists of a navigation from more generic to more detailed data, the reverse of the roll-up operation, performing a data cube summarization, either by stepping down the hierarchy or by introducing a new dimension.
- *Slice*: consists of obtaining a subset of the data cube, performing a selection along one dimension.
- *Dice*: consists of obtaining a subset of the data cube, performing a selection along two or more dimensions.

2.1.4 Presentation

The *presentation layer*, the final component of the DW architecture shown in Figure 2.1, aims at using the data from the DW to provide valuable information to the end-users. Different approaches can be used to obtain this information, such as ad-hoc querying, data mining, reporting or dashboards. The ultimate goal of the presentation layer is the analysis and/or visualization of information, that leads to logical conclusions that are fundamental to a well-informed decision-making.

Ad-hoc querying provides the user with direct access to the data model [2]. When predefined queries are insufficient for the user to obtain the necessary information, ad-hoc queries allow for a broader access to the data, in the sense that users are able to formulate queries to get answers that fulfill their specific needs.

Data mining is a process of data exploration with the intent of finding patterns or relationships that can be made useful to the organization [2]. Using preprocessed data from a DW and applying data mining techniques to the data, such as classification or regression, generates mathematical models capable of identifying relationships that help understanding or predicting behaviors within the data.

Reporting addresses an organization's periodic need for a core set of information about what is going on in a particular area of the business [2]. It is a process that organizes and summarizes data with the purpose of generating paper or web-based reports, providing the end-users with visual context, allowing them to have immediate access to the information in the DW [2]. The generated reports are fundamental for decision-making, as they help end-users extract meaningful insights that lead them to a better understanding of what the business needs to improve. As reports are typically delivered on a regular basis, they tend to follow the same structure and deliver information about the same aspects of the business, albeit related to different time frames. As such, the reports are built through a series of predefined queries that can be reused every time a new report is generated.

Dashboards are graphical user interfaces designed to analyze and keep track of important metrics regarding certain areas of a business, otherwise known as *Key Performance Indicators* (KPIs). Through analyzing the KPIs, the users are offered the possibility of quickly identifying a problem anywhere in the business and drilling down into the detail to identify its causes [2]. It is possible to define thresholds for each KPI represented, specifying the accepted range of values. When a KPI does not conform with this range, the users are notified via *alerts*.

2.1.5 Bus Architecture

The *bus architecture* is a technology and database independent model with the purpose of decomposing the DW planning task into manageable parts [1] in a business-process-aligned manner. It introduces the concept of *conformed dimensions*, which are dimensions with attributes that provide the same contextual meaning to all the facts they are related to. With the dimensions having a uniform interpretation across the enterprise [2], the DW can be developed incrementally, as the dimensions can be reused across the fact tables.

To design the bus architecture, a tabular structure called *bus matrix* is used, establishing a correspondence between the organization's core business processes with the conformed dimensions, which indicates their involvement in processes. Figure 2.2 represents an example of a bus matrix for the retailer domain. The main business processes and dimensions are identified, and placed respectively on the matrix's rows and columns. The matrix's cells reflect a logical relationship between the business processes and dimensions, represented by an "X".

When designing a DW based on a bus matrix, each business process should be implemented individually. Each row of the bus matrix will result in one or more fact tables, that make use of the common dimensions that are involved in the business process represented. Considering the first row of Figure 2.2, the *Issue Purchase Orders* business process can be represented by a fact table sharing three dimension tables, *Date*, *Product* and *Warehouse*, as shown in Figure 2.3.

	COMMON DIMENSIONS						
BUSINESS PROCESSES	**Date**	**Product**	**Warehouse**	**Store**	**Promotion**	**Customer**	**Employee**
Issue Purchase Orders	X	X	X				
Recieve Warehouse Deliveries	X	X	X				X
Warehouse Inventory	X	X	X				
Receive Store Deliveries	X	X	X	X			X
Store Inventory	X	X		X			
Retail Sales	X	X		X	X	X	X
Retail Sales Forecast	X	X		X			
Retail Promotion Tracking	X	X		X	X		
Customer Returns	X	X		X	X	X	X
Returns to Vendor	X	X		X			X
Frequent Shopper Sign-Ups	X			X		X	X

Figure 2.2: Example of a retail enterprise bus matrix - Adapted from [1]

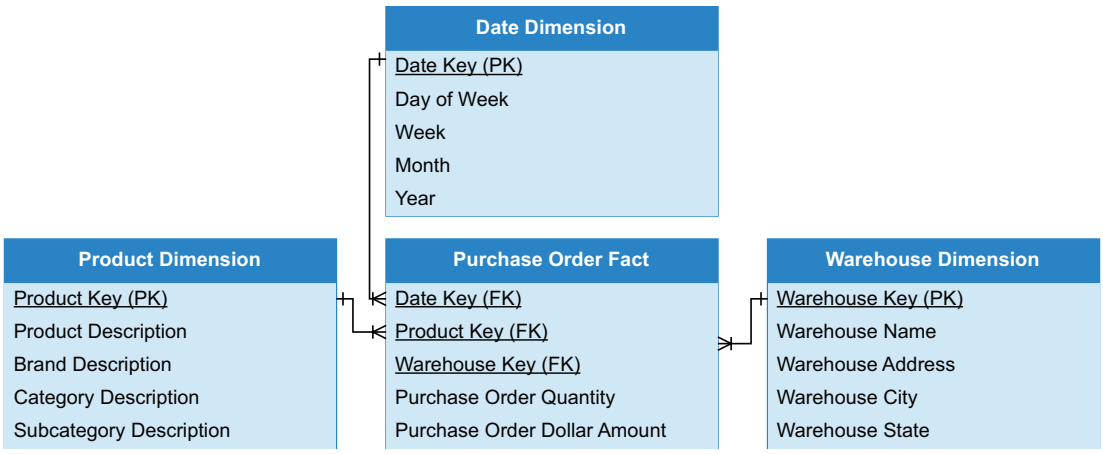

Figure 2.3: Issue Purchase Orders fact and dimension tables - Adapted from [1]

2.1.6 Kimball Business Dimensional Lifecycle

The *Kimball Business Dimensional Lifecycle* approach to data warehousing is a methodology concieved by the Kimball Group[1], that provides the overall framework for the implementation of a DW [2].

The Kimball Business Dimensional Lifecycle depicts a series of sequential high-level tasks required for an effective DW design, development, and deployment, represented in the form of a diagram, on Figure 2.4. It starts with the *Program/Project Planning*, in which the scope of the DW project is defined (i.e., the tasks, duration, resources). This initial step is tied in with the *Business Requirements Specification*, in the sense that the planning process is complemented by the specified business requirements. The Program/Project Planning step also serves as a foundation for the *Program/Project Maintenance*, an ongoing step that monitors the project implementation, ensuring that the Kimball Lifecycle activities are correctly performed.

Following the business requirements specification, the methodology presents three concurrent tracks focusing on *Technology*, *Data*, and *Business Intelligence Applications*, respectively. The Technology track is composed of two sequential steps: the *Technical Architecture Design* step, that defines the overall structure of the DW, and the *Product Selection and Installation*, that defines the technologies that will support the various components of the DW's structure. The Data track is composed of three sequential steps: the *Dimensional Modeling* step, that consists of designing the logical dimensional model, guided by a DW bus matrix, the *Physical Design* step, that turns the dimensional model into a physical data model, supported by a relational database and, if necessary, by OLAP cubes; and the *ETL Design and Development*, that is responsible for populating the tables from the physical data model, extracting data from the various data sources, transforming it and loading it into the DW. The Business Intelligence Applications track is composed of two sequential steps: the *BI Application Design*, that consists of specifying what valuable information must be obtained from the DW and how it will be delivered to the end-users (i.e., ad-hoc querying, data mining, reporting, dashboards), and the *BI Application Development*, that consists of implementing the solutions specified in the BI Application Design step.

The three concurrent tracks converge on the *Deployment* step, and tests are conducted to verify whether each track performs as intended and whether they properly work alongside. When the DW has been deployed, it requires *Maintenance*, more specifically, monitoring the system in terms of performance and address any issues, to keep the system performing optimally. The project's *Growth* is intended to deliver additional value to the business. Expanding the project implies going back to the beginning of the Lifecycle, to the Program/Project Planning, to determine how the additional requirements fit in with the already implemented project.

[1] https://www.kimballgroup.com/

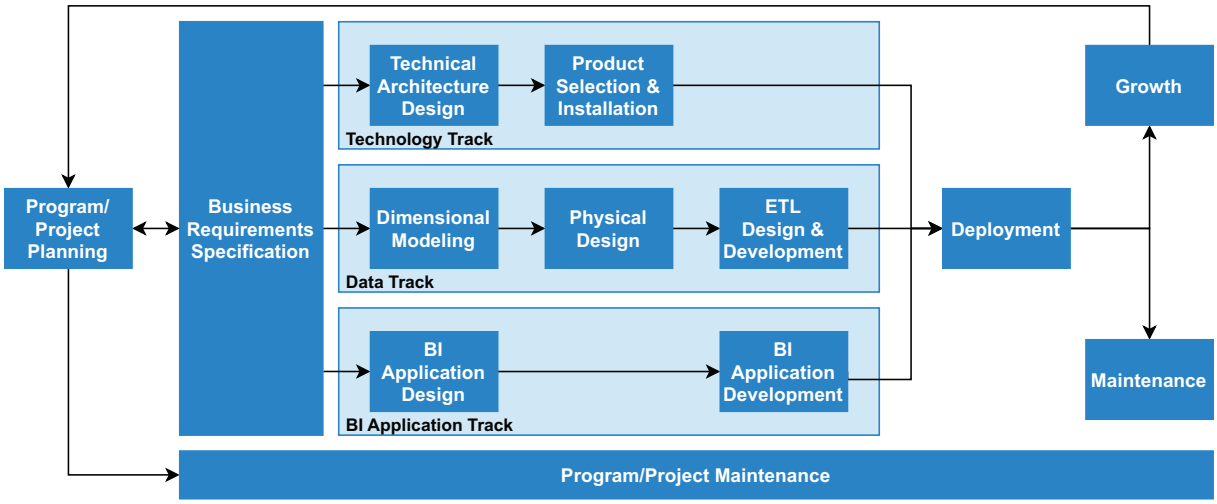

Figure 2.4: Kimball Dimensional Lifecycle diagram - Adapted from [2]

2.2 Domain-Specific Concepts

This section describes concepts related to higher education, applicable to IST, focusing on student activity and performance.

2.2.1 Students

At the beginning of an academic year[2], students apply for a degree and there is an admission process, that can have up to three *admission phases*, in which they are selected to take the degree. A student that is accepted by the university is called an *admitted student*. The set of admitted students of a given degree in a given academic year that enrolled to all the first semester courses are called a *student generation*.

Student Enrollment

An academic degree is composed of a set of courses that students must enroll to and successfully complete, so they can graduate. Each completed course provides the student with an amount of credits. These are part of the *European Credit Transfer and Accumulation System* (ECTS), that defines the courses' workload, estimating the time students must dedicate to them. Each degree year[3] is composed of 60 ECTS credits, divided by a set of courses.
In terms of enrollment in a given course, students may be classified as:

- *Evaluated student*, meaning that the student delivered all mandatory evaluation elements. In terms of evaluation, the student may be further classified into two possible categories:

[2]Academic year is the period in which the academic tasks take place, typically beginning in September of a given calendar year and ending in July of the following calendar year

[3]Degree year is the portion of a degree's curricular plan, that must be undertaken by the students over the course of an academic year. For example, the first degree year of IST LEIC comprises the courses of Linear Algebra, Differential and Integral Calculus I, Foundations of Programming, Introduction to Computer Architecture and Introduction to Information Systems and Computer Engineering in the first semester and Differential and Integral Calculus II, Introduction to Algorithms and Data Structures and Logic for Programming Discrete Mathematics in the second semester.

  - *Approved student*, which means that the student obtained a positive final grade (i.e., from 10 to 20), having successfully completed the course.
  - *Failed student*, which means that the student obtained a negative final grade (i.e., from 1 to 9), having failed the course.
- *Non-evaluated student*, meaning that the student failed to deliver at least one of the mandatory evaluation elements.

Students may be evaluated in up to two academic phases: the *regular academic phases* and the *special academic phases*. While the former is open for all students, only specific groups of students have access to the latter, such as students that are high performance athletes, student workers, or students with special education needs.

Student Activity

The students' activity in a semester is determined based on their enrollment and evaluation. According to these criteria, a student may be classified as:

- *Active student*, meaning that the student has been evaluated in at least one course in a given semester.
- *Inactive student*, meaning that the student has either not enrolled to any course, or has not been evaluated in any course, in a given semester.

A modification of a student's activity status from one semester to the next determines two important measures:

- *Comebacks*, that occur when a student previously inactive becomes active in the following semester.
- *Withdrawals*, that occur when a student becomes inactive after being active in the previous semester.

2.2.2 Course Unit Quality

The *Course Unit Quality* (*Qualidade das Unidades Curriculares* - QUC[4] in portuguese) is a system implemented at IST that enables to monitor and evaluate the course units' performance. The evaluation is semiannual and centered on promoting the continuous improvement of the teaching, learning and assessment processes.

At the end of each semester, a *student survey* is conducted online and it is organized in two different sections:

- The first section assesses whether the workload of each course is appropriate to the number of effective credits, by estimating the number of hours of autonomous work each student has spent for each course he/she was enrolled in. If the estimate does not match the actual number of hours of autonomous work, whether because it has more or less hours than the expected ones, students are asked additional questions, to understand the reasons for this mismatch.

[4] http://quc.tecnico.ulisboa.pt/

- The second section assesses the teaching performance, having students to fill in a form in which they classify, on a scale from 1 to 9, a series of aspects from five fundamental topics related the courses: Workload, Organization, Evaluation, Perceived Learning and Teaching Staff.

Not only does this survey provide insight on the way courses are taught, but it also describes the performance of the instructors from the students' point of view. The results of the survey are used to generate web-based reports for each course. The report presents an overview of the courses' workload, organization, evaluation and teaching, as well as statistics detailing the percentage of valid answers to the survey. After the overview,s five different sections are presented:

1. *Course monitoring throughout the semester/Course workload*: details the amount of ECTS predicted beforehand and compares them with the ECTS estimation given by the students' answers, in terms of contact and autonomous work (subdivided into classes and exam preparation). Facing a discrepancy when comparing predicted and estimated ECTS values, the possible reasons for such differences are detailed, also based on the answers to the survey. Additionally, it indicates the percentage of students that indicated, on a scale from 1 to 9, if the knowledge from previous courses is determinant for succeeding in the course, as well as if the importance of the different means of study, such as class attendance, suggested bibliography, notes and documents provided by the instructor and/or other students.

2. *Course organization*: details the percentage of students that indicated, on a scale from 1 to 9, if the course's planned program was taught, how well the course was structured, how appropriate the suggested bibliography was and how appropriate the support materials were.

3. *Course evaluation method*: provides an overview of the enrollments and approval rates of the course, as well as a distribution of the students by their final grade. Additionally, it indicates the percentage of students that indicated, on a scale from 1 to 9, how appropriate the evaluation method was to the course's contents and how fair the evaluation process was.

4. *Course contribution to the acquisition and/or development of competences*: details the percentage of students that indicated, on a scale from 1 to 9, how much the course helped developing the knowledge on its subject, how much the course helped applying the acquired knowledge, how much the course helped developing a critical sense on its subject, how much the course helped improving cooperation and communication, how much the course helped improving autonomous learning and how much the course helped deepen the ability to analyze the implications of its subject in a social and professional context.

5. *Teaching staff*: the instructors that taught the course are listed, indicating what what kind of class they taught (theoretical or practical) and a link to another web-based report detailing the instructor's teaching performance is provided. This report presents an overview of the student class attendance, presential learning benefits, teaching performance and interaction with students.

   (a) *Student class attendance*: details the percentage of students that indicated, on percentage

intervals from [0%;10%[to [90%;100%], what their attendance to the classes was. A justification to low attendance is presented based on the answers to the survey.

(b) *Presential learning benefits*: details the percentage of students that indicated, on a scale from 1 to 9, how the instructor's attendance and punctuality in the classes was and how appropriate the content and pace of the classes were.

(c) *Teaching performance*: details the percentage of students that indicated, on a scale from 1 to 9, if the instructor was committed to the class, if the instructor explained the contents in an attractive way, if the instructor was clear explaining the contents and if the instructor explained the contents with assurance.

(d) *Interaction with students*: details the percentage of students that indicated, on a scale from 1 to 9, if the instructor encouraged participation and discussion and if the instructor was available to answer questions inside and outside the class.

Based on the aforementioned partial sections, each element of the teaching staff is given a final score, indicating their overall performance teaching the course, ranging from 1 to 9.

Chapter 3

Related Work

In this chapter, we discuss relevant works describing decision support systems in the context of higher education institutions (Section 3.1), as well as the most relevant data integration software packages available that are typically used in the implementation of decision support systems (Section 3.2).

3.1 Decision Support Systems in Higher Education

This section analyzes the use of decision support systems in higher education, detailing their design and implementation decisions.

3.1.1 A Decision Support System for IST Academic Information

The Decision Support System for Academic Information [3] (*Sistema de Apoio à Decisão da Informação Académica* - SADIA in Portuguese) was a decision support system proposed in 2003 for managing the IST academic information. It was part of the FENIX project, an integrated academic management information system developed by and for IST, with the purpose of responding to the needs of all participants in the tuition process (i.e., teachers, students and administrative services).

The SADIA system aimed at providing current and historical data organized in terms of Key Performance Indicators, to enhance decision-making. The data would, subsequently, be used to automatically generate tables containing statistics required by the external processes for accreditation and assessment of undergraduate degrees.

To store and organize the data, the authors proposed a DW, designed according to the *Business Dimensional Lifecycle* methodology proposed by Kimball [2]. Using this methodology, the Business Requirements were defined in an interview-oriented way, that led to the identification of the main business processes that the SADIA system would be focusing on: the IST Student Admission process, the Undergraduate Degree Performance Evaluation process, the Course Performance Evaluation process and the Student Performance Evaluation process.

The common dimensions identified were: Time, Student, Admission, Geography, Course, Degree and Department. The Student Age and Student Sex were modeled as *mini-dimensions*, to enhance

the performance of the user queries, as some impose constraints on the students' age and sex. The Academic Year dimension was modeled separately from the Time dimension, as most user queries intended to analyze data for a particular academic year and semester.

To define the dimensional model, the authors used a DW bus architecture [1], to enable modeling each process individually. The bus matrix details the logical relationships of the aforementioned dimensions with the business processes and is represented in Figure 3.1.

	COMMON DIMENSIONS						
BUSINESS PROCESSES	Time	Student	Admission	Geography	Course	Degree	Department
IST Student Admission	X	X				X	X
Undergraduate Degree Performance Evaluation	X				X	X	X
Course Performance Evaluation	X				X	X	X
Student Performance Evaluation	X	X		X		X	X

Figure 3.1: SADIA Bus Matrix - Adapted from [3]

Using Kimball's enhanced four-step method [11, 12], the IST Student Admission process was chosen and it is the only process whose logical dimensional model definition is covered in the article. This process was modeled as a specific data mart with an accumulating snapshot fact table. The candidate dimensions defined were Time (including the academic year hierarchy), Student, Admission (including admission types and contingents), Geography, Degree and Department. The facts indicate whether the person was applying, admitted or registered and the respective dates of when the application, admission or registration took place. The facts also focus on the grades of the mandatory admission exams (i.e., Chemistry, Biology, Geology and Drawing Geometry), as well as the seriation grade, the high school grade and order of entrance. Figure 3.2 presents the star schema for the IST Student Admission process, containing the key business measures (i.e., elementary facts) identified for this process.

The aggregated and derived facts were identified and, to understand whether they should be included in the aggregated fact table, the business users were questioned about their real usage patterns. Based on the results, a new aggregated fact table was created for the Admission Process for an Undergraduate Degree, which is presented in Figure 3.3.

To ensure that the logical and physical models comply with the business requirements, the authors propose a verification of the ability to respond to each of the user analysis queries. In that sense, the authors created validity matrices, to determine the various dimensions and measures (i.e., elementary, aggregated and derived) that helped responding to each user query. Figure 3.4 represents an excerpt of the validity matrix for the IST Student Admission process. In addition to this validation, the authors consulted the business users to further validate their design decisions. However, for organizational reasons, the SADIA system was never implemented.

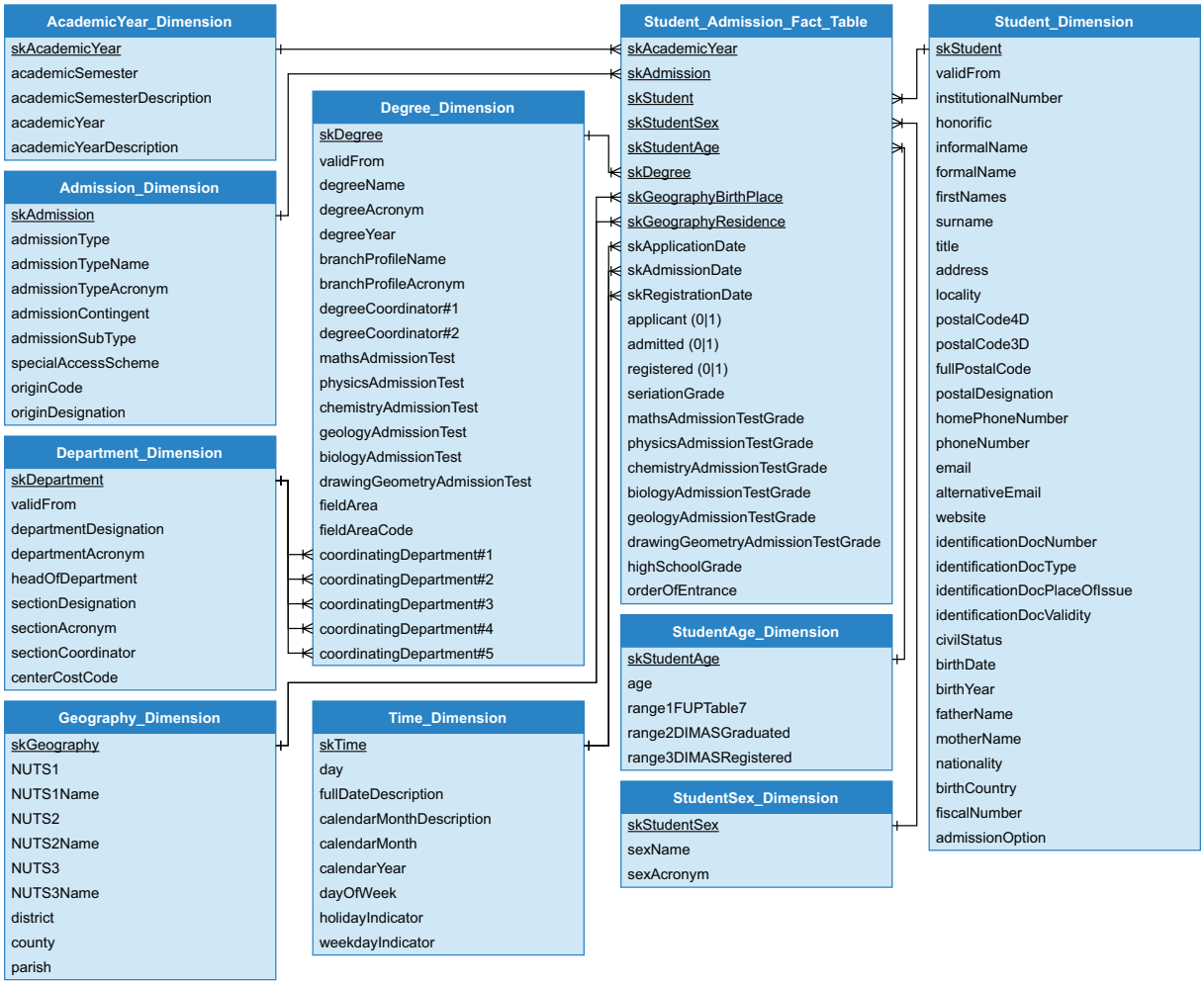

Figure 3.2: IST Student Admission process star schema - Adapted from [4]

Figure 3.3: Aggregated model for the Admission Process for an Undergraduate Degree - Adapted from [4]

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Dimension										
Time										
Academic Year	X	X	X	X	X	X	X	X	X	X
Student										
StudentSex									X	
StudentAge								X		
Geography										
Academic Year	X	X	X	X	X	X	X	X	X	X
Degree	X	X	X	X	X	X	X	X	X	X
Course										
Admission	X	X	X	X	X	X	X	X	X	X
...										
Elementary Metrics										
applicant (0|1)										
admitted (0|1)										
registered (0|1)						X	X	X	X	
numberVacancies	X			X						
...										
Aggregated Metrics										
numberOfApplicants		X								
numberOfAdmitted				X	X		X			
numberOfRegistered					X					
numberFirstOptionApplicants			X							
...										

Figure 3.4: SADIA Validation Matrix - Adapted from [4]

3.1.2 Implementation of Data Warehouse, Data Mining and Dashboard for Higher Education

Conducted at the Bina Nusantara University in Jakarta, the purpose of this research was to develop a system that could integrate data related to the various activities within a higher education institution and be able to perform analysis tasks for a better and more informed decision-making [13]. In that sense, the authors proposed a three step solution: (i) building a DW model and applying data mining techniques, (ii) designing dashboards, and (iii) evaluating the model through interviews.

The authors based the DW's design on the National Higher Education Information System [14], a system composed of ten subsystems that detail the main components of Indonesian higher education institutions (i.e., Academic, Research, Community Service, Personnel, Library, Infrastructure, Financial and Cooperation subsystems). The facts and dimensions were identified according to the composition of this system. In total, 17 dimensional models were defined (e.g., Curriculum Development, Lecturer, Student Intake, Registration and Payment, Teaching Learning, Evaluation, Thesis Guidance) and were implemented using Microsoft SQL Server. We emphasize the Teaching Learning process and the Evaluation process dimensional models, as they are better suited to the scope of our project. The Teaching Learning process star schema provides, as a fact, the total number of student attendances and total number of lecturer attendances, and as dimensions: Semester, Study Program and Course. The Evaluation process star schema provides, as a fact, the total number of approved students, and as dimensions: Course, Department, Major, Grade, Exam Type and Semester. The star schemas were not documented.

From the various star schemas defined, a series of dashboards were created to facilitate monitoring the main areas of the institution. A total of 8 dashboards were developed (i.e., Lecturer, Student Intake

and Payment, Registration, Evaluation, Graduation, Research, Grant and Community Services). Due to the scope of our project, we highlight the Evaluation process dashboard. The focus of this dashboard is to analyze the outcomes of student learning, which enables an overview of the number of students that successfully completed the course, or, contrarily, failed to complete the course. The dashboard also details the distribution of the various courses and their grades.

Despite referencing an interview based evaluation of the proposed model, the authors do not specify whether the model was, in fact, evaluated, which questions would serve as a basis for the interviews, or even who the interviewees would be. However, when detailing the future work, there is a mention to an evaluation of the model, to ensure it meets the standards required by higher education institutions. The authors mention that, in the future, multiple public and private universities in Indonesia may be involved, although it is not specified if the system actually went into a production stage.

3.1.3 Design of a Data Warehouse Model for Decision Support at Higher Education: A Case Study

To address the lack of general consensus and methodologies for designing higher education DWs, the authors propose a methodological framework [5] that offers a set of guidelines to properly design a DW for higher education.

The authors identified, based on [15, 16], two major issues resulting from the application of traditional methods to the design of higher education DWs. The first issue is the *complexity*, more specifically, how elaborate and time consuming designing a DW is. The second issue comes from *uncertain requirements*, which may result in constant changes to the already identified requirements, changes that are not always possible to apply to the DW model. The proposed framework aims at overcoming the aforementioned issues, by introducing two fundamental elements: an *agile development method*, that aims at addressing business requirement changes in a fast and incremental way, and a *method for dealing with uncertain requirements*, that consists in assigning priorities to the requirement changes.

The proposed framework, presented in Figure 3.5, is composed of four procedural steps and, for each procedural step, the most suitable methods are presented, as well as the outcomes they produce.

The first procedural step is *problem observation*, which consists of examining and understanding the business processes, complementing the observations through an interview oriented approach. This procedural step results in a research proposal, a document that presents a formal justification for the DW project, which must be submitted and approved by the organization.

The second procedural step is the *development of the DW model*, which is the outcome of this step. The development step is divided into three different stages: *business requirements specification*, *logical design* and *physical design*. The specification of business requirements is a hybrid method that combines three sequential approaches: beginning with the *user-oriented* approach, a series of interviews are conducted and the business processes are identified and prioritized; afterwards, the *business-process-oriented* approach assesses the value of each business process; finally, the *operational-source-oriented* approach examines the operational source systems and respective data, with the purpose of identify-

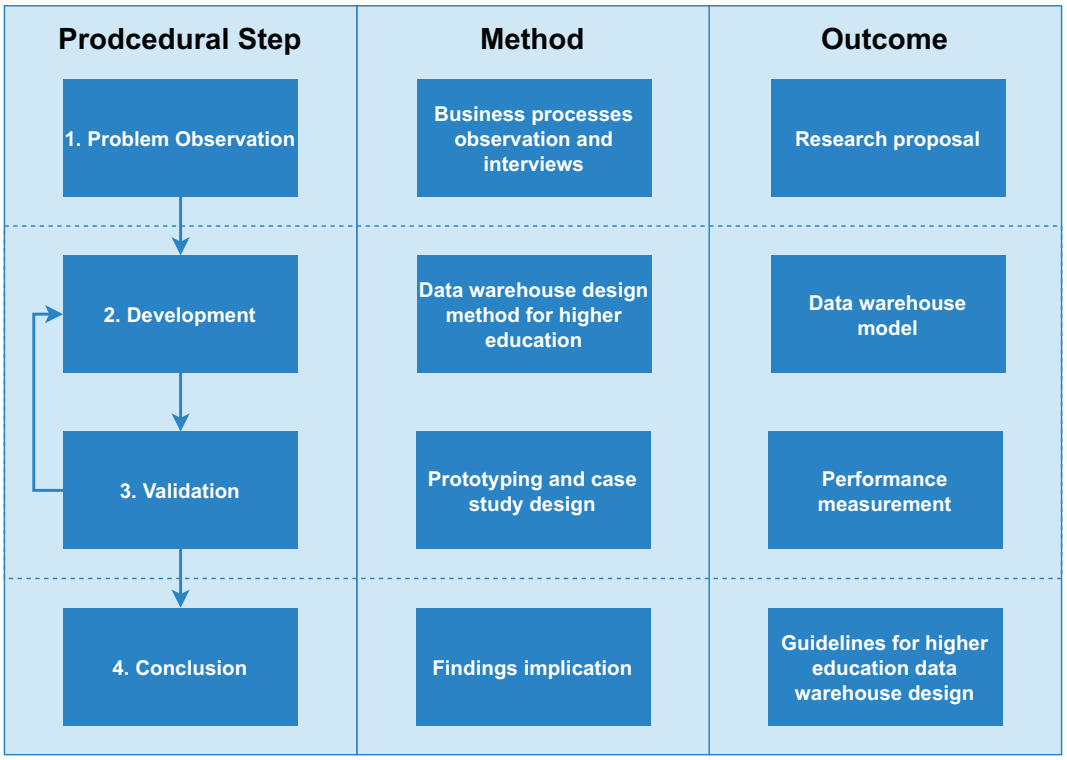

Figure 3.5: Proposed framework for designing a higher education data warehouse - Adapted from [5]

ing the technical requirements for designing the DW model. Upon defining the business requirements, the dimensional models are created in the logical design step, making use of a dimensional modeling technique (e.g., Kimball's four-step dimensional design process [1]). Following the logical design stage comes the physical design stage, which consists of translating the designed dimensional model into physical tables created in a database management system (e.g., SQL Server 2012 Database Engine).

The third procedural step is the *validation* of the DW model, that is, building a prototype and measuring its performance, ensuring the business requirements are fulfilled. Not meeting the business requirements implies a revision and consequent restructuring of the development step, as suggested by Figure 3.5. Only when the requirements are fully met, comes the fourth and final procedural step, the *conclusion* of the project, that consists of documenting the implementation guidelines.

In comparison to Kimball's Business Dimensional Lifecycle methodology, referenced in Section 2.1.6, the framework at issue presents a less detailed development process, that is composed by the business requirement specification and the logical and physical designs. Kimball's methodology differentiates the business requirement specification from the development process, and splits the latter into three concurrent tracks: the Technology, Data and Business Intelligence Application tracks. The proposed framework's development process coincides with the Data track, with both having the purpose of building the dimensional model. One noticeable difference is that the Data track includes the ETL design and development, whereas in the proposed framework, the validation step is where the ETL is developed.

In the framework at issue, validation implies creating a functional prototype, to assess its ability to respond to strategic questions that are instrumental to the organization, as well as its compliance with the defined business requirements. In Kimball's methodology, validation takes place at the dimensional

model level, even before its transition to a physical model. On one hand, the proposed framework's validation step can provide a better understanding of the DW's capabilities and limitations; on the other hand, if the validation is not successful, as it occurs at a prototype level, it forces a full restructuring of the model and, consequently, the creation of a new prototype.

The authors present a case study of the application of the proposed framework on the implementation of a DW for the University of Business and Technology (UBT) in Jeddah, Saudi Arabia. Following the aforementioned procedural steps, the DW design process started with problem observation, more specifically, with monitoring the business processes at UBT. From this step, it was determined that the traditional system in use at UBT at the time was unable of supporting strategic decision-making, thus highlighting the need for a DW system.

After ensuring that a DW system was, in fact, needed, the business requirements were determined through the use of the previously described hybrid method. This method combined interviews (user-oriented approach), business process observation (business-process-oriented approach) and document examination (operational-source-oriented approach).

Kimball's bottom-up approach was used, in the logical design step, meaning that the business processes modeled would be merged to form an enterprise-wide DW for UBT. Using Kimball's four-step dimensional design process, and considering the business requirements identified, two business processes were selected: Course Registration and Academic Performance. The two business processes resulted in two data marts.

The Academic Performance data mart, whose model is represented in Figure 3.6, is the one that better suits the scope of our project and as such constitutes our main focus. This model provides a way of analyzing the students' performance along the various courses and departments, enabling the identification of possible issues in them.

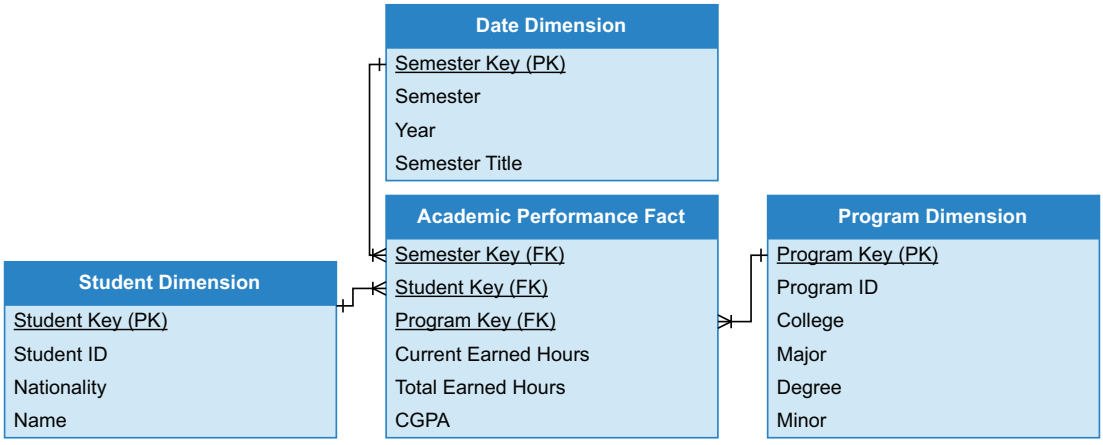

Figure 3.6: Student academic performance dimensional model - Adapted from [5]

The physical design consisted of implementing the models as physical tables using SQL Server 2012 Database Engine. To deal with slowly changing dimensions, surrogate keys were used as primary keys in the dimensional tables, as opposed to business keys.

The validation of the implemented models was performed using reports, to answer a series of strategic questions provided by the executive managers at UBT, regarding the registration of students by

major per year. Using Microsoft SQL Server Reporting Services, queries were constructed to answer the questions in the form of a report, containing a table with the years and majors as rows and columns respectively, as well as the respective number of registrations, as represented in Figure 3.7.

total registration by major per year

		ACCOUNTING	CIVIL ENGINEERING (R)	COMPUTER ENGINEERING (R)	EIT GENERAL SUBJECTS	ELECTRONICS AND COMMUNICATIONS ENGINEERING (R)	ENGLISH
2002							
2003							
2004		88			345		
2005		189			376		
2006		243			143		15
2007		341			250		
2008	1	177			230	3	19
	2	198			73		139
	3	9					
2009		431	1		780		66
2010		233		10	397		217

Figure 3.7: Sample report snapshot [5]

3.1.4 Discussion

In this section, we presented three different articles regarding higher education decision support systems based on a DW. Section 3.1.1 described a decision support system for the academic information of Instituto Superior Técnico, known as SADIA [3]. Section 3.1.2 described the implementation of a DW, data mining and dashboards for the Bina Nusantara University [13]. Section 3.1.3 described a framework for designing higher education DWs [5], accompanied by a case study covering its use on the implementation of a DW for the University of Business and Technology in Jeddah.

From the aforementioned articles, it is possible to observe that the DW design was mostly guided by Kimball's methodologies (i.e., Kimball's bus architecture in [3] and the four-step dimensional design process in [3, 5]). The articles that describe the physical implementation (i.e., [13, 5]), used SQL Server as the underlying Relational Database Management System (RDBMS). The SADIA system [3] was not implemented, although a prototype was built [4], which used SQL Server as well.

Logical design decisions are detailed in [3]: to enhance the performance of user queries that imposed age and sex constraints, the authors used mini-dimensions for the students' age and sex, in addition to the student dimension; to enhance the performance of user queries that refer to a specific academic year, the authors introduced an academic year dimension, separated from the conventional time dimension. The latter is aligned with the context of our project, as several KPIs detailed in Section 4.2 analyze specific academic years.

The creation of an ETL process and reporting are covered in [5]. Since SQL Server was used as an underlying RDBMS, the tools provided by SQL Server were used (i.e., SQL Server Integration Services for ETL and SQL Server Reporting Services for reporting), albeit without enough detail regarding the

implementation. Additionally, dashboard generation was briefly covered in [13]. Implementation details are barely specified, as the authors focus on the purpose of the dashboards and the insights they provide.

In terms of validation methodology, all articles propose different approaches. In [3], the authors propose a verification of the models' compliance with the user analysis queries. In [13], the authors propose an interview based validation for ensuring the higher education standards are met, that is not detailed in the article: no results are presented; it is not detailed who the interviewees would be, or how the interviews would be conducted. In [5], the validation of the model was performed by generating reports to provide answers to strategic questions randomly asked by the executive managers of UBT.

The articles focus primarily on the logical design of a DW for higher education. However, despite their usefulness for a better understanding of the design process, the articles have an overall lack of detail regarding the implementation and validation methodologies.

3.2 Data Integration Software

In this section, the most prominent data integration software tools available are analyzed in terms of how they cover the development of the various components of the DW architecture presented in Figure 2.1. Additionally, we discuss whether and how the different software tools presented suits our development needs.

We based our choice of software on the Gartner Magic Quadrant for Data Integration Tools [17], an analysis report that evaluates data integration software in terms of the ability to execute, as well as the completeness of vision. The tools are divided into quadrants: the *Niche Players*, the *Visionaries*, the *Challengers* and the *Leaders*. The Leaders quadrant contains the software with the most complete vision and ability to execute, and it constitutes our focus for this section. From the leaders quadrant we focus on: Informatica PowerCenter, IBM InfoSphere Information Server, Oracle Data Integration and Talend Open Studio.

Additionally, this section covers the Pentaho and SQL Server add-on services, that are part of the Challengers and the Niche Players quadrants respectively. Our proficiency with Pentaho's product suite makes it a suitable candidate. SQL Server and its add-on services were the tools selected by all the articles detailed in Section 3.1, being a relevant inclusion as well.

3.2.1 Informatica PowerCenter

Informatica PowerCenter[1] is a tool used for building enterprise DWs, offering data integration capabilities through ETL processes. It is available for Windows and UNIX based systems. Additionaly, it can be used as a cloud-based service.

It is a powerful data integration solution, as it is able to deal with the extraction and transformation of large volumes of data. It also has the ability to connect to a vast set of different types of data sources

[1] https://www.informatica.com/products/data-integration/powercenter.html

(e.g., MySQL, SQL Server, Oracle, Mongo DB, Cassandra, SAP, Salesforce, Flat files, XML files).

Informatica PowerCenter offers a visual interface that enables the configuration of each step of the ETL process, using a drag and drop feature. Upon designing the ETL process, it can be scheduled to execute on a regular basis, according to the business needs.

The main focus of Informatica PowerCenter is data integration and, despite having data analysis capabilities, it is only suited for developing up to the DW layer of Figure 2.1. As such, other tools would need to be used alongside it, to be able to cover the OLAP Server and presentation layers.

3.2.2 IBM InfoSphere Information Server

IBM InfoSphere Information Server[2] is a data integration platform composed of a set of different tools. It is available for the Windows, Linux and AIX operating systems.

The tool used for data integration is the IBM InfoSphere DataStage. It provides ETL process designing capabilities, by offering a graphical interface that enables the creation of data integration transformations and jobs (i.e., sequences of transformations), through a drag and drop feature. It also enables scheduling the execution of ETL processes.

In addition to data integration, a tool used for reporting is included. The IBM InfoSphere FastTrack tool enables the creation of reports from a set of different report templates and to export them to different formats (e.g. PDF, HTML, XML). Reports may be generated manually, or they can be scheduled and generated periodically in an automated way.

The IBM InfoSphere Information Server does not include OLAP and Dashboard capabilities.

3.2.3 Oracle Data Integration

Oracle Data Integration[3] offers a set of four different data integration tools: Oracle GoldenGate, Oracle Data Integrator, Oracle Enterprise Data Quality and Oracle Enterprise Metadata Management.

The data integration solution more suited towards the development of a DW is Oracle Data Integrator. This tool uses an Extract, Load, Transform (ELT) approach, as an alternative to ETL. As such, data is extracted from multiple data sources and is immediately loaded into the target DW, before executing the transformation step, resulting in a better performance when dealing with large amounts of data. The data can be loaded into different database systems, not just Oracle Database. Using Oracle Data Integrator, it is possible to design data integration processes, through a graphical interface that enables a drag and drop approach to setup each step of the process.

In addition to data integration, Oracle also offers a suite of Business Intelligence tools that support reporting and dashboard generation capabilities. Oracle BI Answers provides a graphical interface that enables data exploration through ad-hoc queries. Data can be displayed in visual elements, which can then be used for building reports. Oracle BI Interactive Dashboards uses the visualizations produced by Oracle BI Answers to form dashboards.

[2] https://www.ibm.com/analytics/information-server
[3] https://www.oracle.com/middleware/technologies/data-integrator.html

3.2.4 Talend Open Studio

Talend Open Studio[4] is an open source platform for data integration and big data.

Talend Open Studio can connect to multiple heterogeneous data sources (e.g., MySQL, SQL Server, Oracle, Dynamo DB, Mondrian, CSV files). Through a drag and drop feature, it is possible to design ETL or ELT processes, which are converted to Java code, that can be executed in multiple operating systems (i.e., Windows, Linux, Mac). It is not possible to schedule ETL or ELT process executions in Talend Open Studio, although it is possible by using external tools (e.g., OS task scheduler).

As it is a data integration tool exclusively, it is not suited for covering the entire DW architecture represented in Figure 2.1. The OLAP server and presentation layers would require combining Talend with other tools.

3.2.5 Pentaho

Pentaho[5] consists of a collection of open source tools that encompass the capability of data integration, analysis and presentation. Developed using the Java language, the set of tools offered by Pentaho is supported by multiple operating systems.

Data Integration is achieved using Pentaho Data Integration (PDI). Providing a visual interface, this tool enables developing ETL processes using a drag and drop feature, to create transformations and jobs, that are used to extract, transform and load the data into the DW.

Using the Pentaho Schema Workbench (PSW), it is possible to define OLAP cubes in the form of XML schema files, by accessing the data in the DW. Pentaho Analysis Service (Mondrian), an OLAP server, is used to aggregate the data according to the XML schema files, which need to be provided to it. Making use of the Mondrian OLAP server, Pentaho offers two main tools for OLAP analysis, that provide a drag and drop web interface for the creation of MDX queries. The OLAP analysis tools are Pentaho Analysis and Saiku, available in the enterprise edition and the community edition respectively.

To generate reports, Pentaho Report Designer (PRD) or Saiku Reporting can be used. To create dashboards, Pentaho Dashboard Designer (PDD) is offered on the enterprise edition, whereas the community edition offers Community Dashboard Editor (CDE). All these tools provide a graphical interface for creating data visualizations and building reports or dashboards.

3.2.6 Microsoft SQL Server Add-on Services

Microsoft SQL Server[6] is an RDBMS developed by Microsoft, that includes a series of add-on services, that enable data integration, analysis and reporting. Despite the RDBMS being available for Windows, Linux and Mac (the former two via docker containers), the add-on services are exclusive to the Windows operating system.

SQL Server includes, as a data integration solution, SQL Server Integration Services (SSIS). This tool can integrate data from multiple sources (e.g., MySQL, SQL Server, Oracle, PostgreSQL, XML files,

[4] https://www.talend.com/products/talend-open-studio/
[5] https://www.hitachivantara.com/en-us/products/data-management-analytics/pentaho-platform.html
[6] https://www.microsoft.com/en-us/sql-server

CSV files), transforming the data into a suitable format and loading it into a DW. It enables creating ETL processes using a graphical interface, but can be complemented with coding for defining custom tasks and transformations.

Additionally, the Microsoft SQL Server license includes tools for analysis and reporting, that complement the architecture of Figure 2.1. These tools are SQL Server Analysis Server (SSAS) and SQL Server Reporting Services (SSRS). SSAS provides OLAP capabilities, enabling the creation of OLAP cubes, that offer enhanced analysis capabilities. SSAS supports multiple OLAP query languages (i.e., Language Integrated Query (LINQ) and Multidimensional Expressions (MDX)). Additionally, this tool offers built-in data mining features, enabling to identify patterns in the data. SSRS is a server-based application used as a solution for generating automated reports. The creation of dashboards is also possible, albeit with very limited interaction.

3.2.7 Discussion

For the development of our decision support system, the data integration software should ideally be open source and, if possible, multiple operating systems (OS) support would be preferable, as it would enable a more widespread usage.

While the software tools described in this section focused primarily on data integration, some were composed of sets of tools with additional purposes. The ability of covering all the components from Figure 2.1 is highly valued, as using a single tool is preferred over combining the use of multiple tools.

The various data integration tools considered were analyzed and their compliance with the aforementioned criteria was verified, as presented in Table 3.1.

Software	**Open Source**	**Multiple OS Support**	**ETL**	**OLAP**	**Reporting**	**Dashboards**
Informatica PowerCenter	No	Yes	Yes	No	No	No
IBM InfoSphere	No	Yes	Yes	No	Yes	No
Oracle data integration	No	Yes	Yes	No	Yes (Oracle BI)	Yes (Oracle BI)
Talend Open Studio	Yes	Yes	Yes	No	No	No
Pentaho	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft SQL Server Add-on Services	No	No	Yes	Yes	Yes	Yes

Table 3.1: Data integration software comparison

As expected, since most of the software described focused uniquely on data integration, they can only cover the ETL portion of the DW development, or briefly cover other aspects, such as reporting. Consequently, these tools will not be considered.

Pentaho and Microsoft SQL Server, possess a more comprehensive set of tools besides data integration, which are capable of fully covering the architecture presented in Figure 2.1. Pentaho has the advantage of being open source and supporting multiple operating systems (i.e., Windows, Linux and

Mac), but the decisive factor is our proficiency with Pentaho, which will enable starting the implementation phase immediately. For the aforementioned reasons, Pentaho will be used for the development of the DW.

Chapter 4

Business Requirements Specification

Kimball's Business Dimensional Lifecycle methodology identifies the Business Requirements Specification as a fundamental step in the creation of a data warehouse.

In this chapter we identify the business requirements of our system. First, we thoroughly describe the data that will serve as input to our system (Section 4.1) as well as the information to be extracted from that data, which we will address as Key Performance Indicators (KPIs) (Section 4.2).

4.1 Available Input Data

Over the years, the LEIC-T Coordinator has gathered, from the Fénix system, several sets of data regarding the LEIC-A and LEIC-T degrees' main areas, dated from 1989 to 2019. These data sets are organized in Excel sheets, detailed by the samples presented in Tables 4.1, 4.2, 4.3, 4.4 and 4.5. The data in these samples is anonymized, omitting the names and institutional numbers of students. Additionally, an overview of the various files is presented in Table 4.6. The data is grouped into six different categories, represented by differently structured files:

- **Degrees**, a single file that stores data about the degrees. Each row corresponds to a degree and contains the short and long name representations, as well as the number of ECTS and the number of years needed for its completion (i.e., degree years). A sample of this data is shown in Table 4.1.

Degree Acronym	Degree Name	ECTS To Complete	Years To Complete
LEIC-T	Licenciatura Bolonha em Engenharia Informática e de Computadores - Taguspark	180	3
LEIC-A	Licenciatura Bolonha em Engenharia Informática e de Computadores - Alameda	180	3

Table 4.1: Sample of the data on the degrees Excel file.

- **Departments**, a single file that stores data about the departments and scientific areas, as well as the courses they encompass. Each row represents the relationship between a course, department

and scientific area. It contains the course's short name representation, the departments' short and long name representations, as well as the scientific area's short and long name representations. A sample of this data is shown in Table 4.2.

Course Acronym	**Department Acronym**	**Department Name**	**Scientific Area Acronym**	**Scientific Area Name**
IA	DEI	Department of Computer Science and Engineering	IA	Artificial Intelligence
IAC	DEI	Department of Computer Science and Engineering	ASO	Architecture and Operating Systems
IAED	DEI	Department of Computer Science and Engineering	MTP	Programming Methodology and Technology
IEI	DEI	Department of Computer Science and Engineering	SI	Information Systems

Table 4.2: Sample of the data on the departments Excel file.

- **Curricular Plans**, a set of files that indicate which courses are part of the degree. Each file represents the curricular plan for a given degree of a given academic year, indicated in the file's name. Each row corresponds to a course and contains the course's short and long name representations, the ECTS obtained upon completion, as well as the degree year and semester it takes place. A sample of this data is shown in Table 4.3.

Course Acronym	**Course Name**	**ECTS**	**Degree Year**	**Semester**
AL	Linear Algebra	6.0	1	1
CDI1	Differential and Integral Calculus I	6.0	1	1
FP	Foundations of Programming	7.5	1	1
IAC	Introduction to Computer Architecture	7.5	1	1
IEI	Introduction to Information Systems and Computer Engineering	3.0	1	1

Table 4.3: Sample of the data from the excel file detailing the LEIC curricular plan in the academic year of 2015/2016.

- **Admissions**, a set of files that detail the student enrollment in the degree. Each file represents the admissions of a given academic year. Each row corresponds to a student and contains the student's institutional number, the admission state and the date when the admission state was modified. A sample of this data is shown in Table 4.4.

Student Number	Admission State	Admission State Modified Date
00001	Admitted	2015-09-10
00002	Admitted	2015-09-10
00003	Cancelled	2015-09-20
00004	Admitted	2015-09-10

Table 4.4: Sample of the data from the excel file detailing the admission phase in the academic year of 2015/2016.

- **Grades**, a set of files that store data about the students' performance on the courses. Each file represents the grades of a given academic year and semester, indicated in the file's name. The file has multiple sheets, each containing the grades of a given course. Each row represents an enrolled student's classification, containing the student's number and name, the grades obtained in the regular academic phase, the grade improvement phase and the special academic phase, as well as the final grade. A sample of this data is shown in Table 4.5.

Student Number	Student Name	Degree	Regular Grade	Improv. Grade	Special Grade	Final Grade
00001	Student Name	LEIC-T	NA	–	RE	RE
00002	Student Name	LEIC-T	10	NA	–	10
00003	Student Name	LEIC-T	NA	–	–	NA
00004	Student Name	LEIC-T	13	–	–	13

Table 4.5: Sample of the data from the excel file detailing the grades of the Linear Algebra course of LEIC.

Excel Files	Fields	Granularity
Degrees	Degree acronym	One row per degree
	Degree name	
	ECTS to complete	
	Years to complete	
Departments	Course acronym	One row per course
	Department acronym	
	Department name	
	Scientific area acronym	
	Scientific area name	
Curricular Plan	Degree acronym (from file name)	One row per course, year and semester
	Academic year (from file name)	
	Academic semester (from file name)	
	Course acronym	
	ECTS	
	Course degree year	

	Course semester	
Admissions	Degree acronym (from file name)	One row per student
	Academic year (from file name)	
	Student institutional number	
	Admission state	
	Admission state modified date	
Grades	Academic year (from file name)	One sheet per course, with one row per student
	Academic semester (from file name)	
	Course acronym (from sheet name)	
	Degree acronym	
	Regular grade	
	Improvement grade	
	Special grade	
	Final grade	

Table 4.6: Structure of the input excel files.

4.2 Key Performance Indicators

This section describes a series of relevant KPIs that were identified through interviews with the LEIC-T Coordinator, as well as by analyzing his semesterly reports.

The LEIC-T Coordinator has been monitoring the performance of two particular areas: the courses and the student generations. The LEIC-T Coordinator aims at answering several business questions regarding each area, which can be answered through certain KPI. Table 4.7 presents the business questions and respective KPIs.

1 - Course	
1.1 - What were the approval rates in a given semester?	
Q1.1.1 - Number of enrolled students	Quantifies the number of enrollments in a course
Q1.1.2 - Number of evaluated students	Quantifies the number of enrolled students that were evaluated in a course
Q1.1.3 - Number of approved students	Quantifies the number of enrolled students that obtained a positive grade in a course (value of 10 and higher)
Q1.1.4 - Ratio of approved students/ evaluated students	Indicates the proportion of evaluated students that were approved
Q1.1.5 - Ratio of approved students/ enrolled students	Indicates the proportion of enrolled students that were approved

1.2 - What were the approval rates in a given semester?	
Q1.2.1 - Number of students enrolled for the first time	Quantifies the number of first-time enrollments in a course
Q1.2.2 - Number of students approved in the first enrollment	Quantifies the number of students enrolled for the first time and approved in a course
Q1.2.3 - Ratio of students approved in the first enrollment / students enrolled for the first time	Indicates the approval rates for the students enrolled for the first time
2 - Generation of Students	
2.1 - How did a generation of students perform in a specific year / semester?	
Q2.1.1 - Final grade average obtained in a year / semester	Indicates the average of all final grades that students obtained in a specific year / semester
Q2.1.2 - Average ECTS percentage obtained in a year / semester	Indicates the average percentage of ECTS that students obtained in a specific year / semester
Q2.1.3 - Percentage of approvals by enrollments	Indicates proportion of approvals out of all enrollments in the courses of a specific year / semester
Q2.1.4 - Percentage of approvals by evaluations	Indicates proportion of approvals out of all evaluations in the courses of a specific year / semester
2.2 - Did students perform better when enrolling in fewer courses?	
Q2.2.1 - Number of students by number of enrollments in a year / semester	Indicates how many students enrolled in a certain number of courses in a year / semester
Q2.2.2 - Number of students by number of courses completed in a year / semester	Indicates how many students completed a certain number of courses in a year / semester
Q2.2.3 - Ratio of completed courses against enrolled courses	Indicates the success rate of the number of courses the students completed against the number of courses they were enrolled in
2.3 - Is there a most difficult course for a generation of students?	
Q2.3.1 - Number of students not passing a course and passing all others	Quantifies the number of students that have not passed a specific course having passed all other courses
2.4 - How did the students perform after the duration of their degree?	
Q2.4.1 - Number of students completing a degree, by number of semesters since admission	Quantifies the number of students in a generation that finished their degree, according to the number of semesters taken (since admission)
2.5 - Are the withdrawals significant in a generation of students?	
Q2.5.1 - Number of withdrawals	Quantifies the withdrawals of the generation on each semester since admission

Q2.5.2 - Number of comebacks	Quantifies the comebacks of the generation on each semester since admission
Q2.5.3 - Number of active students	Quantifies the students that were not enrolled or evaluated in at least one course on each semester since admission
Q2.5.4 - Number of inactive students	Quantifies the students that were not enrolled or evaluated in any course on each semester since admission

Table 4.7: Business questions and KPI

Chapter 5

The IST Degree Coordination Decision Support System

This chapter describes the development of SAD-CCIST, whose goal is to help the decision-making of degree coordinators, providing them with current and historic information regarding the performance of students in their degree. To achieve this goal, our system integrates academic data from a set of input Excel files gathered throughout the years, obtains KPIs related to the performance of students in their degree and displays them in the form of dashboards.

The various steps of the design and implementation of our system are described in this chapter. Section 5.1 describes the general architecture of our system. Section 5.2 describes the creation of the DSA. Section 5.3 describes the creation of the dimensional model of the DW. Section 5.4 describes the ETL processes that populate our system's underlying DW. Section 5.5 describes the OLAP cubes created over the implemented dimensional model. Finally, Section 5.6 describes the end-product of our system, the dashboards.

5.1 System Architecture

As a DSS, our system follows the architecture described in Section 2.1. The layers that compose the architecture are: the data sources, the DSA, the DW, the OLAP server and the presentation layers. The overall architecture of our system is presented in Figure 5.1.

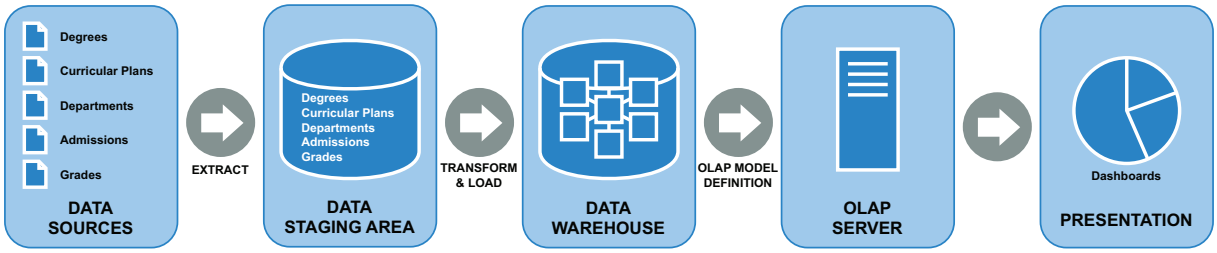

Figure 5.1: Architecture of the developed system

Our system is designed to integrate academic data regarding the IST LEIC-A and LEIC-T degrees, previously detailed in Section 4.1. The input data comes from the IST Fénix system, and includes data of each academic year ever since the creation of the degree (i.e., 1989) until today. For organizational reasons, it was not possible to access the Fénix system directly, so the data was requested to the Administrative Services of the Department of Computer Science and Engineering (DEI in portuguese), that provided it as Excel files. The exported Excel files are grouped into four categories (i.e., degrees, curricular plans, admissions, grades), each detailing a different aspect that helps determining the performance of students of our degrees of reference.

We opted to include a DSA, which is detailed in Section 5.2. The DSA was built to match the exact structure of the data exported from the Fénix system, so that the data can be directly exported from it, facilitating a future integration with the Fénix system.

The DW is the central piece of the architecture and is described in Section 5.3. The DW was designed as a relational database and was structured as a set of dimensions linked to a set of facts (i.e., dimensional model), a structure optimized for analytical querying. The data from the DSA is loaded into the DW, but first it has to match the structure imposed by the DW, meaning that it has to undergo the transformation step, which also ensures the quality of the data.

The ETL processes, described in Section 5.4, are responsible for extracting data from the input data sources, storing it into the DSA and then transforming and loading it into the DW. The data is extracted directly from the input Excel files into the DSA. Once in the DSA, the data is transformed so that it fits the DW dimensional model and is loaded into the latter upon being transformed.

An OLAP server, described in Section 5.5, was introduced to further increase the efficiency of analytical queries. The OLAP server uses the data from the DW and performs a series of aggregations and precalculations over it, forming multidimensional structures called OLAP cubes. The OLAP cubes enable the use of OLAP operations (i.e., roll up, drill down, slice and dice).

Finally, the end users are presented with dashboards, which are described in Section 5.6. The dashboards display information regarding the degrees of reference (i.e., LEIC-A and LEIC-T) and enable the end users to choose the appropriate time frame for the visualizations. The goal is to provide useful insights to the end users, to help them making decisions accordingly.

5.2 Data Staging Area

This section describes the creation of the DSA used to store the data extracted from the input Excel files.

As described in Section 4.1, the data that serves as input to our system comes from the IST academic data information system, known as the Fénix system, albeit in the form of Excel files.

Since the data in the Excel files was directly extracted from the Fénix system, it means that the way the data is structured in the Excel files is the same as in the Fénix system. If the DSA follows that same structure, it makes it possible to use the Excel files as input for now and, in the future, extracting the data directly from the Fénix system in a fully automated way.

The input Excel files are grouped into four categories (i.e., degrees, curricular plans, admissions, grades), meaning that the DSA requires four tables to store data. To match the structure of the Excel files, the tables have the same fields as their corresponding group of Excel files.

The data detailing the *degrees* is contained in a single Excel file with a single sheet. Each row contains the degree acronym, degree name, number of ECTS credits to complete the degree and number of years to complete the degree. Each of these fields is obtained directly from the Excel file, with each row of the Excel file resulting in a row in the corresponding DSA table (i.e., degrees table). The mapping between the source and target fields is presented in Table 5.1.

Source Field	**Source Field From**	**Target Field**	**Extraction**
degree_acronym	Columns	degree_acronym	Direct
degree_name	Columns	degree_name	Direct
degree_ects_to_ complete	Columns	degree_ects_to_ complete	Direct
degree_years_to_ complete	Columns	degree_years_to_ complete	Direct

Table 5.1: Degree file extraction

The data detailing the *departments* is contained in a single Excel file with a single sheet. Each row contains the course acronym, department acronym, department name, scientific area acronym and scientific area name. Each of these fields is obtained directly from the Excel file, with each row of the Excel file resulting in a row in the corresponding DSA table (i.e., departments table). The mapping between the source and target fields is presented in Table 5.2.

Source Field	**Source Field From**	**Target Field**	**Extraction**
course_acronym	Columns	course_acronym	Direct
department_acronym	Columns	department_acronym	Direct
department_name	Columns	department_name	Direct
scientific_area_acronym	Columns	scientific_area_acronym	Direct
scientific_area_name	Columns	scientific_area_name	Direct

Table 5.2: Departments file extraction

The data detailing the *curricular plans* is contained in multiple Excel files (i.e., one file per degree and per academic year), each containing a single sheet. Each row contains the course acronym, course name, number of ECTS obtained upon completion, course year and course semester. The name of each file contains the degree acronym and the academic year for which the curricular plan is valid (e.g., LEICT_Plano_2009). These fields are extracted from the name using regular expression capture groups. Each row of each Excel file, together with the degree acronym and academic year indicated in the filename, results in a row in the corresponding DSA table (i.e., curricular_plans table), and the mapping between the source and target fields is presented in Table 5.2.

The data detailing the *admissions* is contained in multiple Excel files (i.e., one file per degree and per academic year), each containing a single sheet. Each row contains the student institutional number, admission state and admission state date. The name of each file contains the degree acronym and the academic year in which the admissions took place (e.g., LEICT_CNAES_2018_2019). Each row of each Excel file, together with the degree acronym and academic year indicated in the filename, results in a

Source Field	Source Field From	Target Field	Extraction
degree_acronym	Filename	degree_acronym	Regex Capture Group
academic_year	Filename	academic_year	Regex Capture Group
course_acronym	Columns	course_acronym	Direct
course_name	Columns	course_name	Direct
course_ects	Columns	course_ects	Direct
course_year	Columns	course_year	Direct
course_semester	Columns	course_semester	Direct

Figure 5.2: Curricular plan files extraction

row in the corresponding DSA table (i.e., admissions table), and the mapping between the source and target fields is presented in Table 5.3.

Source Field	Source Field From	Target Field	Extraction
degree_acronym	Filename	degree_acronym	Regex Capture Group
academic_year	Filename	academic_year	Regex Capture Group
student_ institutional_number	Columns	student_ institutional_number	Direct
admission_state	Columns	admission_state	Direct
admission_state_ date	Columns	admission_state_ date	Direct

Figure 5.3: Admission files extraction

The data detailing the *grades* is contained in multiple Excel files (i.e., one file per degree, per academic year and per academic semester), each containing multiple sheets (i.e., one sheet per course). Each row contains the student institutional number, student name, regular grade, improvement grade, special grade and final grade. The name of each file contains the degree acronym, academic year and academic semester in which the courses were taught (e.g., LEICA_2011_2012_2S). The name of each sheet contains the course acronym (e.g., AL, FP, LP). Each row of each Excel file, together with the degree acronym and academic year indicated in the filename and the course acronym indicated in the sheetname, results in a row in the corresponding DSA table (i.e., grades table), and the mapping between the source and target fields is presented in Table 5.4.

Source Field	Source Field From	Target Field	Extraction
degree_acronym	Filename	degree_acronym	Regex Capture Group
academic_year	Filename	academic_year	Regex Capture Group
academic_semester	Filename	academic_semester	Regex Capture Group
course_acronym	Sheetname	course_acronym	Direct
student_ institutional_number	Columns	student_ institutional_number	Direct
student_name	Columns	student_name	Direct
regular_grade	Columns	regular_grade	Direct
improvement_grade	Columns	improvement_grade	Direct
special_grade	Columns	special_grade	Direct
final_grade	Columns	final_grade	Direct

Figure 5.4: Grade files extraction

The DSA was implemented as a relational database, using MySQL as the RDBMS. We created all the tables described (i.e., degrees, curricular plans, admissions, grades) along with their respective

attributes.

5.3 Dimensional Modeling

This section describes the creation of the dimensional model, according to the methodologies proposed by Kimball [1, 2], namely the Kimball Bus Architecture and Kimball Four-Step Dimensional Design Process methodologies.

The dimensional modeling activity begins with the application of the Kimball Bus Architecture methodology [1]. This methodology decomposes the dimensional model planning process, focusing on finding the relationships between the business processes and the associated conformed dimensions. The business processes and conformed dimensions had to be identified, through a careful analysis of the input data sources and KPIs respectively detailed in Sections 4.1 and 4.2.

The business processes focus on the evaluation of students throughout the duration of their degree. There is a need to know when the students were admitted, what their grades were on each course of a semester, how they performed in their degree in a given semester and when they graduated. Therefore, the business processes identified were: *student admission*, *student evaluation*, *student activity* and *student graduation*. Associated to the business processes, we could identify four different dimensions. These were: the *degree*, *course*, *student* and *time*. The course dimension is part of a *scientific area*, which in turn is part of a *department*, forming a hierarchy of three levels. The time dimension has two different semantics: *admission time* and the *evaluation time*. The admission time marks the instant in which the student was admitted to a degree, whereas the evaluation time marks the instant in which an evaluation took place. The logical relationships between the identified business processes and conformed dimensions are represented in the bus architecture matrix in Figure 5.5.

	COMMON DIMENSIONS				
BUSINESS PROCESSES	**Degree**	**Course**	**Student**	**Admission Time**	**Evaluation Time**
Student Admission	X		X	X	
Student Evaluation		X	X	X	X
Student Activity	X		X	X	X
Student Graduation	X		X	X	X

Figure 5.5: Bus architecture matrix

The identification of the business processes and associated conformed dimensions is followed by the logical design of the dimensional model. The logical design is guided by the Kimball Four-Step Dimensional Design Process methodology [2], which tackles each business process in the bus architecture matrix individually, identifying the granularity, dimensions and factual measures.

1. **Selecting the Business Process**

Each business process identified in the bus matrix will be selected individually to result in different fact tables. In that sense, the business processes selected are: the *student admission*, *student evaluation*, *student activity* and *student graduation*.

2. **Declaring the Granularity**

   The granularity defines the lowest level of detail of a single row in each fact table originated by the business processes.

   In terms of the *student admission* process, it happens only once per student and per degree and in a specific time instant.

   In terms of the *student evaluation* process, a student is evaluated in a given course in a given semester.

   In terms of the *student activity process*, we want to know the performance of each student in a given degree in a given semester.

   Finally, in terms of the *student graduation*, it happens only once per student, per degree and in a specific time instant.

3. **Identifying the Dimensions**

   With the bus architecture matrix, the candidate dimensions were identified.

   For the *student admission* process, the *student*, *degree* and *admission time* dimensions were identified.

   For the *student evaluation* process, the *student*, *course*, *admission time* and *evaluation time* dimensions were identified.

   For the *student activity* process, the *student*, *degree* and *admission time* and *evaluation time* dimensions were identified.

   Finally, for the *student graduation* process, the *student*, *degree* and *admission time* and *evaluation time* dimensions were identified.

4. **Identifying the Facts**

   The factual measures were chosen based on the KPIs the system must answer.

   The *student admission* process registers an admission event, making it a factless table, therefore with no factual measures.

   The *student evaluation* process focuses on the evaluation of a student in a course, namely the student's grades (i.e., regular, improvement, special and final) and his/her evaluation status (i.e., approved, failed) and also whether it is the student's first enrollment in the course and whether the student has already passed the course and is enrolling to improve his/her grade. The factual measures are *is first enrollment* (boolean), *is improvement* (boolean), *evaluation status* (string), *regular grade* (integer), *improvement grade* (integer), *special grade* (integer) and *final grade* (integer).

The *student activity* process focuses on the activity of a student in terms of his/her degree, namely the number of courses enrolled, the number of courses completed, the number of ECTS credits obtained, the maximum number of ECTS credits the student can obtain and his/her grade average. It also focuses on whether the student was active (i.e., was evaluated in at least one course), if he/she made a comeback (i.e., was inactive in the previous semester and became active in the current semester) or a withdrawal (i.e., was active in the previous semester and became inactive in the current semester), as well as if the student failed only one course and which course that was. The factual measures are *is active* (boolean), *is comeback* (boolean), *is withdrawal* (boolean), *courses enrolled* (integer), *courses completed* (integer), *courses evaluated* (integer), *ects obtained* (decimal), *ects possible* (decimal), *grade average* (decimal), *only course failed id* (integer).

The *student graduation* process focuses on the activity of the student during the entire duration of his/her degree, such as the time taken to complete the degree, the number of semesters the student was active and inactive, the number of courses enrolled, completed and evaluated, as well as the grade average. The factual measures are *number of academic years since admission* (integer), *number of academic semesters since admission* (integer), *number of academic periods since admission* (integer), *courses enrolled* (integer), *courses completed* (integer), *courses evaluated* (integer), *grade average* (decimal).

Through the Kimball Four-Step Dimensional Design Process methodology we are able to create the dimensional model, as we have identified the fact tables, their granularity, associated dimensions and measures, which are presented in Table 5.3.

The dimensional model resulted in a fact constellation composed of four fact tables and eight normalized dimensions (i.e., . Each dimensional table contains a set of descriptive attributes that characterize the dimensions. Each fact table contains a set of factual measures registered in the events of the modeled business processes. Figure 5.6 presents the dimensional model created. The primary keys are in bold and underlined, while the foreign keys are only in bold. The relationship between tables is expressed with a connecting line and the cardinality of the relationship is expressed by the symbols at the end of the lines. The relationships between a fact table and its associated dimensions are relationships of *one-to-many*: a single row of the dimensional table can be associated with multiple rows of the fact table (e.g, student dimension and student admission fact tables). The same cardinality applies to dimension hierarchies: a single row of a level of the hierarchy can be associated with multiple rows of the level below (e.g., department and scientific area dimension tables).

The DW was also implemented as a relational database, using MySQL as the RDBMS. We created all the dimension and fact tables described, along with their attributes.

Business process	**Granularity**	**Linked dimensions**	**Measures**
Student Admission	One row per Student, per Degree (on the time of admission)	Student Degree Admission Time	—-
Student Evaluation	One row per Student, per Course, per Semester	Student Course Admission Time Evaluation Time	• Is first enrollment (0/1) • Is improvement (0/1) • Evaluation status • Regular grade • Improvement grade • Special grade • Final grade
Student Activity	One row per Student, per Degree, per Semester	Student Degree Admission Time Evaluation Time	• Is active (0/1) • Is withdrawal (0/1) • Is comeback (0/1) • Number of courses enrolled • Number of courses completed • Number of courses evaluated • Number of ECTS obtained • Number of ECTS possible • Grade average • Only course failed ID
Student Graduation	One row per Student, per Degree (on the time of graduation)	Student Degree Admission Time Evaluation Time	• Number of academic years since admission • Number of academic semesters since admission • Number of academic periods since admission • Number of courses enrolled • Number of courses completed • Number of courses evaluated • Grade average

Table 5.3: Result of Kimball Four-Step Design Process methodology

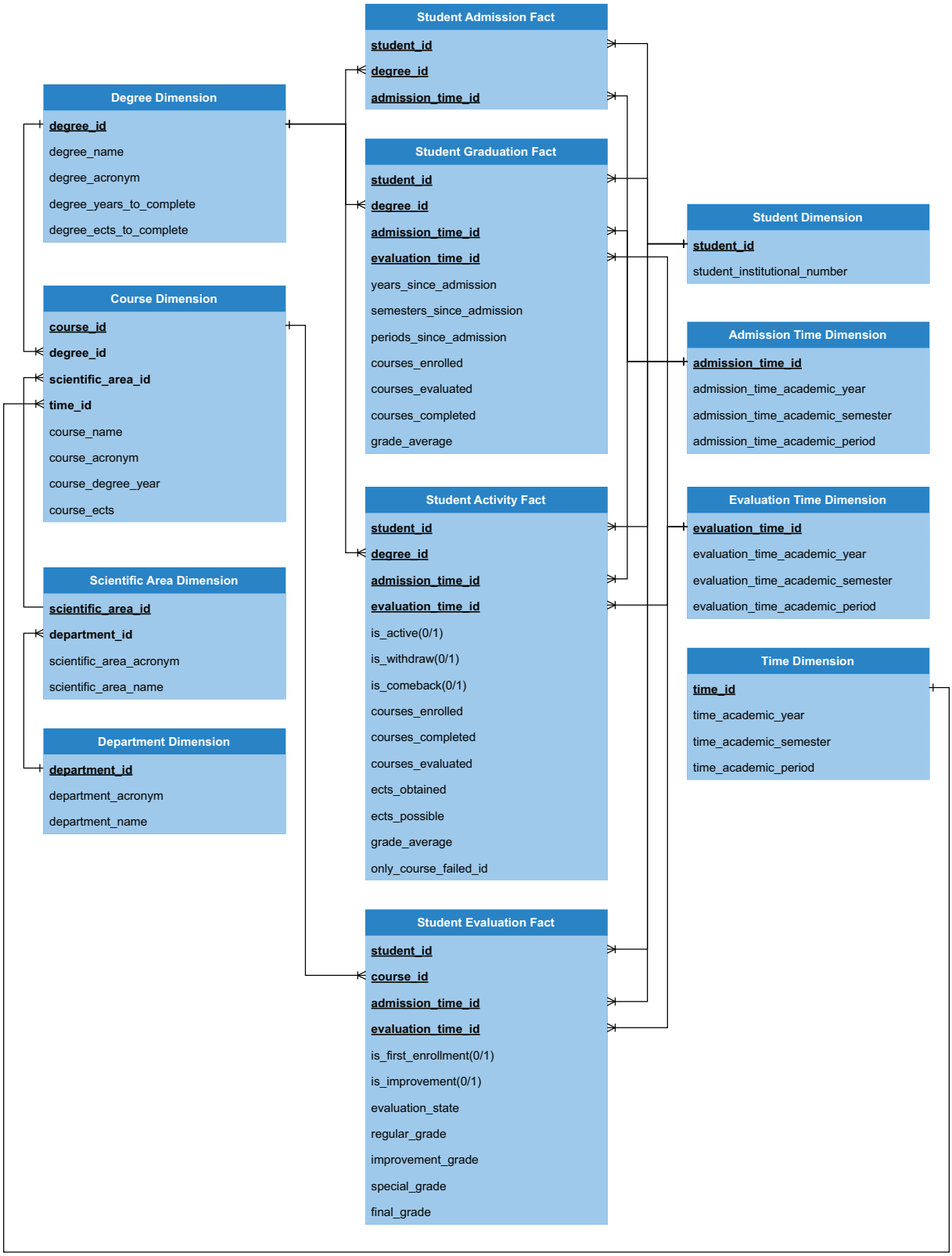

Figure 5.6: Proposed dimensional model

5.4 Extract-Transform-Load Processes

The Extract-Transform-Load (ETL) processes are responsible for populating the DSA using the set of input data, and then populating the DW using the data in the DSA.

The ETL processes were created using Pentaho Data Integration (PDI). With PDI it is possible to create jobs, which define a sequence of transformations. The transformations are, in turn, a sequence of operations, such as reading and storing data in files and databases, or applying calculations to data (e.g., sort, group by, filter, join, append).

We defined two main jobs for building and populating the DSA and DW:

- *Extraction*, a sequence of transformations responsible for reading all groups of input Excel files (i.e., admissions, curricular plans, degrees, grades) and populating the respective tables in the DSA.

- *Transformation and Loading*, a sequence of transformations responsible for accessing the tables from the DSA and processing their data, structuring it into a suitable format for storing it in the DW.

All the jobs and transformations created are presented in Appendix A.

Extraction

The extraction step starts by ensuring that the DSA tables exist. If the tables that should compose the DSA do not exist, they are created.

As described in Section 5.2, there are five different groups of input Excel files (i.e., degrees, departments, curricular plans, admissions, grades) and the DSA is composed of five tables to store data of each group of Excel files. As such, the extraction step is divided into five sub-processes, each handling the extraction of data of a particular group of input Excel files.

The extraction is done sequentially and, in this case, the order of extraction is not important. As such, first the admission files are extracted, followed by the curricular plan files, followed by the departments file, followed the degree file and finally the grade files. The extraction process is presented in Figure 5.7.

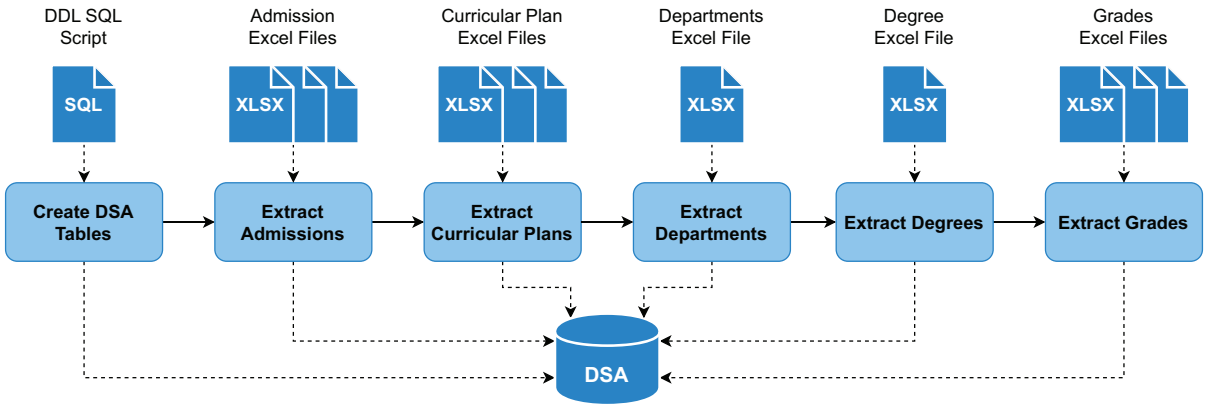

Figure 5.7: Extraction process

The content of each Excel file from each group of files is read into memory. Additionally, the names of the admissions, curricular plans and grades files indicate the degree acronym, academic year and/or academic semester, which are extracted from the filenames using regular expression capture groups and then converted to the appropriate data type. The data is then stored in the respective table in the DSA.

Transformation and Loading

Similarly to the extraction step, we start by ensuring that the DW tables exist. If the tables that should compose the DW do not exist, they are created.

Unlike the extraction step, in this case, the order by which the dimension and fact tables are populated matters, as the facts are linked to several dimensions and some dimensions are linked to other dimensions. The transformation and loading of each dimension and fact table is done in a single sub process. The overall transformation and loading process is presented in Figure 5.8.

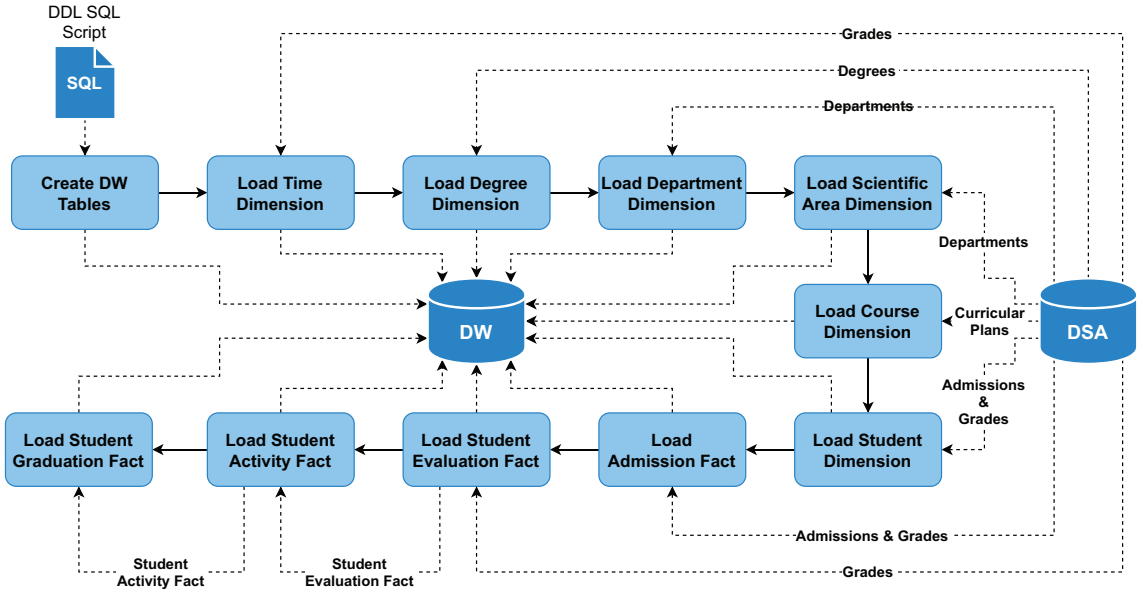

Figure 5.8: Transformation and loading process

The first dimension loaded is the time dimension. The grades table in the DSA is the only table that contains the academic year and academic semester fields, both needed in the time dimension. We select, from the grades table, every unique academic year and academic semester combination, which are subsequently used to populate the time dimension. The process of loading the time dimension table is presented in Figure 5.9.

Figure 5.9: Time dimension table loading process

The second dimension loaded is the degree dimension. The DSA degree table and the DW degree

dimension table follow the same structure, so the data can be loaded directly from the former into the latter. The process of loading the degree dimension table is presented in Figure 5.10.

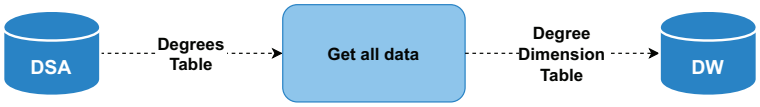

Figure 5.10: Degree dimension table loading process

The third dimension loaded is the department dimension. It consists of getting all the unique combinations of department acronyms and names, then loading them into the department dimension table. The process of loading the department dimension table is presented in Figure 5.11.

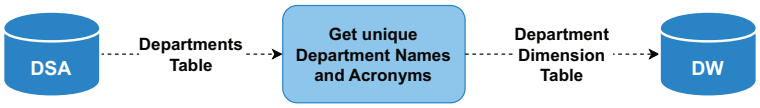

Figure 5.11: Department dimension table loading process

The fourth dimension loaded is the scientific area dimension. It consists of getting all the unique combinations of scientific area acronyms and names, then obtaining the ID of the department the scientific area belongs to, and finally loading them into the scientific area dimension table. The process of loading the scientific area dimension table is presented in Figure 5.12.

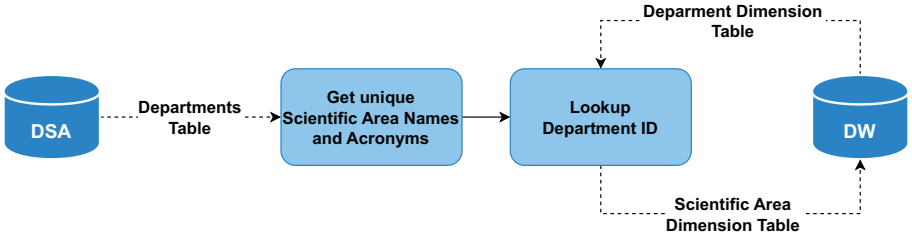

Figure 5.12: Scientific area dimension table loading process

The fifth dimension loaded is the course dimension. First, we get all records from the DSA curricular plan table. Since the course dimension is linked to the time dimension and degree dimension, a lookup must be performed for both dimensions, to obtain the time ID and degree ID of the records to which the course record is linked to. The academic year and semester are the fields used to lookup the time ID. The degree acronym is the field used to lookup the degree ID. The IDs obtained and the remaining fields from the DSA curricular plan table are used to populate the DW course dimension table. The process of loading the course dimension table is presented in Figure 5.13.

The sixth and final dimension loaded is the student dimension. The student dimension uses data from the DSA admissions and grades tables, since there can be students enrolled in courses, without having a registered admission. We get all unique student institutional numbers (i.e., IST IDs) from both tables and join them. We have to remove the duplicate records (since the students may have records in both tables) and then populate the student dimension table. The process of loading the student dimension table is presented in Figure 5.14.

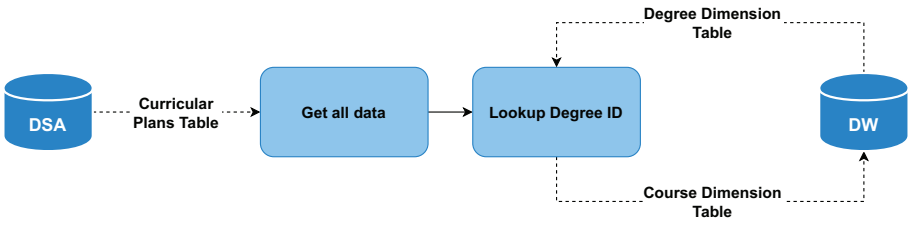

Figure 5.13: Course dimension table loading process

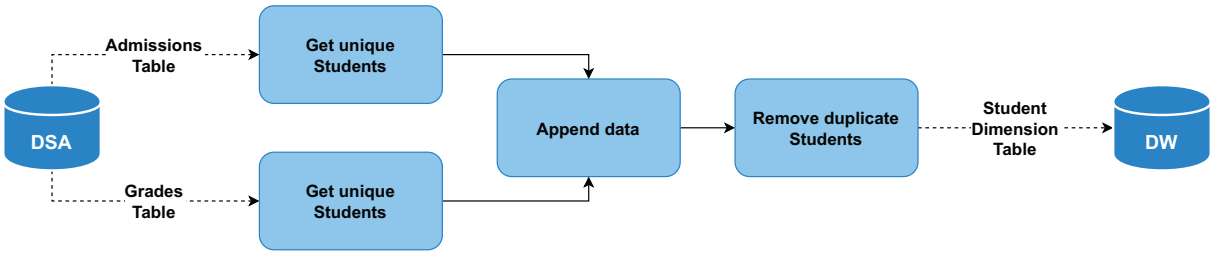

Figure 5.14: Student dimension table loading process

Upon loading the dimensions, we can load the fact tables. We start with the admission fact table. The admission fact table uses data from the DSA admissions and grades tables, since there is a need to know which admitted students were enrolled in all the courses of their admission semester. Using the admissions data, we determine and exclude all cancelled admissions. Using the grades data, we determine the students that were enrolled in all the first semester courses. The data is joined using student, degree and time as keys, and then the data is used to populate the DW admission fact table. The process of loading the admission fact table is presented in Figure 5.15.

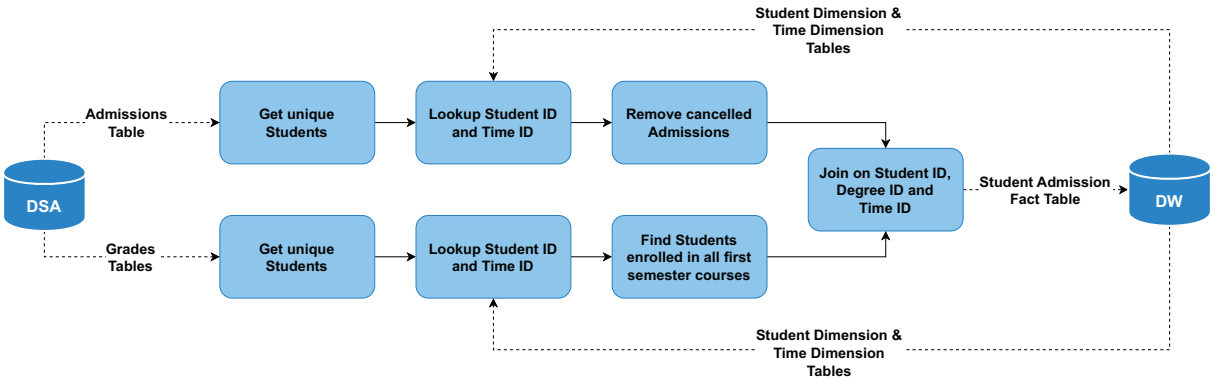

Figure 5.15: Admission fact table loading process

The second fact table loaded is the student evaluation fact table. The student fact table uses data from the DSA grades table. The final grade field is used to determine the evaluation status (i.e., ”AP”, ”RE”, ”NA”). By analyzing all enrollments of a student in a course, it is possible to determine whether it is his/her first enrollment in the course, or if he/she has been previously approved (i.e., improvement). The *evaluation status*, *first enrollment* and *improvement*, as well as the *regular grade*, *improvement grade*, *special grade* and *final grade* fields are used to populate the DW student evaluation table. The process of loading the student evaluation fact table is presented in Figure 5.16.

The third fact table loaded is the student activity fact table. The data used to populate the student

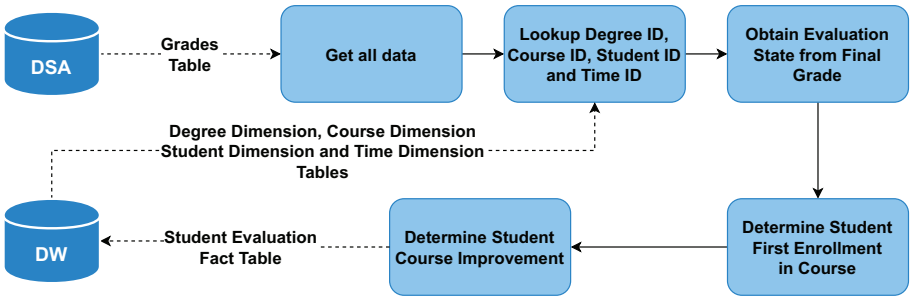

Figure 5.16: Student evaluation fact table loading process

activity fact table is obtained by aggregating the data from the student evaluation fact table on a semester basis. Using the data from the student evaluation fact table, it is possible to determine how many courses a student was evaluated, enrolled and approved in and, consequently, the number of ECTS obtained. We also determine whether the student failed only one course in that semester. The number of evaluated courses in a semester determines if the student was active (i.e., true if evaluated in at least one course, false otherwise) and a modification of his/her activity status in consecutive semesters determines if the student had a comeback or withdrawal (i.e., comeback when going from inactive to active, withdrawal when going from active to inactive). The *number of courses evaluated, enrolled and approved*, *ECTS obtained*, *only course failed*, *activity status*, *withdrawal and comeback status* fields are used to populate the DW student activity table. The process of loading the student activity fact table is presented in Figure 5.17.

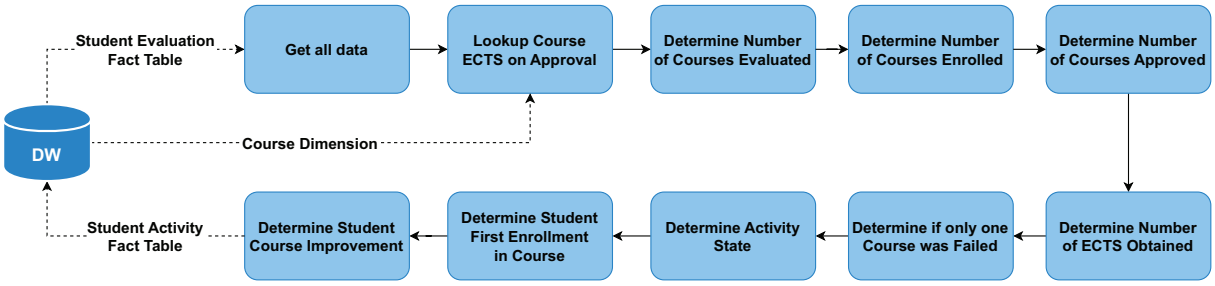

Figure 5.17: Student activity fact table loading process

The final fact table loaded is the student graduation fact table. To obtain the data that populates the student graduation fact, we perform aggregations over the data from the student activity fact table, to determine if a student has graduated, by determining if the number of ECTS obtained has reached the number of ECTS required to complete the degree. Due to modifications to the LEIC curricular plan (i.e., such as 2012/2013), a tolerance of 1.5 ECTS is given, to consider eventual course equivalences. Additionally, data from the student admission fact table is also used to calculate the elapsed time since the admission until the graduation. The *number of years, semesters and periods since admission* fields are used to populate the DW student graduation fact table. The process of loading the student graduation fact table is presented in Figure 5.18.

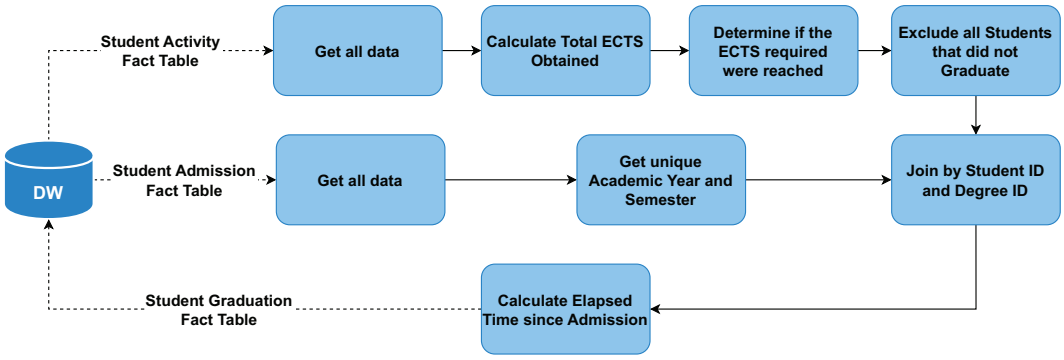

Figure 5.18: Student graduation fact table loading process

5.5 Online Analytical Processing

This section describes the creation of an OLAP model over the dimensional model created in Section 5.3, to increase the analytical query efficiency, by performing calculations and aggregations beforehand, therefore reducing the number of operations performed when accessing the data for presentation purposes.

The OLAP model is a logical model defined over a DW, that facilitates a multidimensional analysis. It is composed of *cubes*, *measures*, and *dimensions*. The cubes are collections of dimensions and measures related to a particular subject. The measures are indicators the users are interested in analyzing. Each dimension is constituted by a set of attributes that help dividing the measures into sub-categories (e.g., number of enrollments by course or by semester). The dimensions form hierarchies that are composed of multiple levels and they enable computing intermediate sub-totals (e.g., number of enrollments by academic year or by semester).

We used Pentaho Schema Workbench to create the OLAP model. First, we defined the dimensions used by the cubes. Each dimension table from the dimensional model was defined as a dimension in the OLAP model. Some dimensions form hierarchies (e.g., degree and course), which had to be specified when creating the OLAP dimensions. Figure 5.19 shows the hierarchies present within the dimensions. The *course* dimension is part of two hierarchies (i.e., degree hierarchy and department hierarchy), since a course can be categorized as a part of a *degree*, or, alternatively, as a part of a *scientific area*, which in turn belongs to a *department*. The *degree* itself can also be considered as a single level hierarchy, since the student activity and student graduation facts are connected to the degree dimension, but are not directly connected to the course dimension. The *time* dimensions also form hierarchies, since an *academic period* is part of an *academic semester* and an academic semester is a part of an *academic year*. Finally, the *student* dimension is considered a single level hierarchy.

Once the dimensions are created, we create the OLAP cubes. Each fact table from the dimensional model originated a cube in the OLAP model, with the main difference between the two being the fact that the cubes use aggregations (i.e., count, sum, average, maximum, minimum) of the measures from the fact tables. The aggregated measures of each cube are presented in Table 5.4.

While the OLAP model could prove beneficial as it is meant to increase the analytical query efficiency as opposed to the relational dimensional model, our solution had limitations. With the OLAP model we

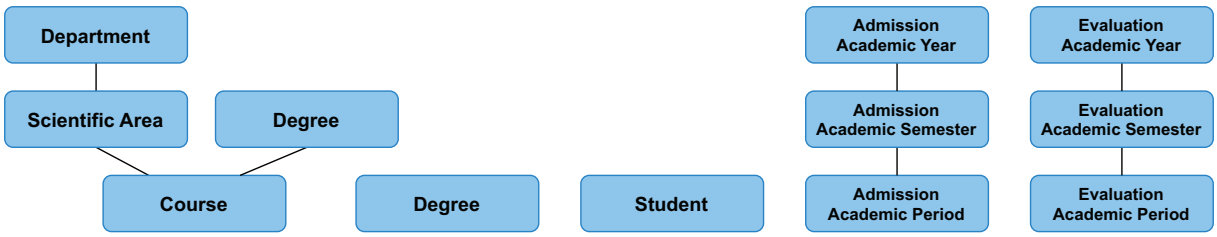

Figure 5.19: OLAP Dimension Hierarchies

Cube	Dimensions	Measures	Aggregations / Calculations
Student Admission	Student Degree Admission Time	Total Admissions	COUNT(student_id)
Student Evaluation	Student Course Admission Time Evaluation Time	Total Enrollments	COUNT(student_id)
		Total Evaluations	COUNT (evaluation_state!=”NA”)
		Total Approvals	COUNT (evaluation_state=”AP”)
		Approvals/Enrollments	Total Approvals / Total Enrollments
		Approvals/Evaluations	Total Approvals / Total Evaluations
		Average Regular Grade	AVG(regular_grade)
		Average Improvement Grade	AVG(improvement_grade)
		Average Special Grade	AVG(special_grade)
		Average Final Grade	AVG(final_grade)
Student Activity	Student Degree Admission Time Evaluation Time	Total Enrollments	COUNT(student_id)
		Total Evaluations	COUNT(evaluation_state!=”NA”)
		Total Approvals	COUNT(evaluation_state=”AP”)
		Approvals/Enrollments	Total Approvals / Total Enrollments
		Approvals/Evaluations	Total Approvals / Total Evaluations
		Average Regular Grade	AVG(regular_grade)
		Average Improvement Grade	AVG(improvement_grade)
		Average Special Grade	AVG(special_grade)
		Total Active	SUM(is_active)
		Total Withdraws	SUM (is_withdraw)
		Total Comebacks	SUM(is_comeback)
Student Graduation	Student Degree Admission Time Evaluation Time	Total Graduations	SUM(student_id)
		Average Enrollments	AVG(courses_enrolled)

Table 5.4: OLAP Cubes and Aggregated Measures

proposed, we were unable to obtain some of the required KPIs from Section 4.2. Q2.1.3 and Q2.1.4 measure the percentage of approvals by enrollments and by evaluations respectively, which are not possible to obtain with this OLAP model, since there is no way to group a measure by other measures. Furthermore, it was not possible to group measures by all the academic years and semesters at the same time, which is crucial for Q1.2.1 and Q1.2.2, that indicate the evolution of the course approval rates throughout all the semesters of all academic years.

As the proposed OLAP model is limited in terms of its capacity for obtaining all the required KPIs, we decided not to use it in the final solution and we opted by performing direct queries to the DW instead.

5.6 Dashboards

Our DSS presents the data in the form of interactive dashboards. To create the dashboards, we used the Pentaho Business Analytics (BA) platform. Through a plugin, called CTools[1], it is possible to create different dashboards, using HTML, Javascript and CSS. We used a Javscript library called C3.js[2] that extends the capabilities of the CTools, by offering a vast selection of interactive charts.

Pentaho BA allows the dashboards to access the DW and perform various different queries, that are responsible for obtaining the KPIs, which are displayed in the various visualizations that compose the dashboards. The goal is to answer the business questions of the LEIC-T Coordinator, detailed in Section 4.2. The business questions were grouped semantically into two categories (i.e., courses, generation of students) and several KPIs were identified to answer each question.

5.6.1 Student Generation Performance

The LEIC-T Coordinator has analyzed the performance of student generations that were admitted since 2007. Usually the LEIC-T Coordinator focuses on understanding how students have performed on a semesterly and yearly basis, as well as the overall performance of students over the entire duration of their degree. These are two different facts in the dimensional model from Section 5.3 (i.e., student activity fact, student generation fact), which have different granularity and measures. For that reason, we created one dashboard for *student activity* and another for *student graduation*.

Student Activity

Every semester, the LEIC-T Coordinator analyzes how the students have performed in the courses they were enrolled to. The first thing that comes to mind when thinking of student performance is how successful the students were in the courses, for instance, the approvals and grades. The approvals grant the students a certain amount of ECTS, which are also used to measure the performance of students (i.e., number of ECTS obtained in a semester).

The LEIC-T Coordinator, however, uses some other important measures, such as comebacks and withdrawals, which are determined by the activity of a student. An active student is a student that was

[1] https://help.pentaho.com/Documentation/9.1/Products/CTools
[2] https://c3js.org/reference.html

evaluated in at least one course in a given semester. A withdrawal happens when a student becomes inactive after being active in the previous semester, whereas a comeback happens when a student becomes active after being inactive in the previous semester.

All these measures are presented in the student generation activity dashboard, whose layout is presented in Figure 5.20, that displays information about the first semester of the generation of 2009 from LEIC-T.

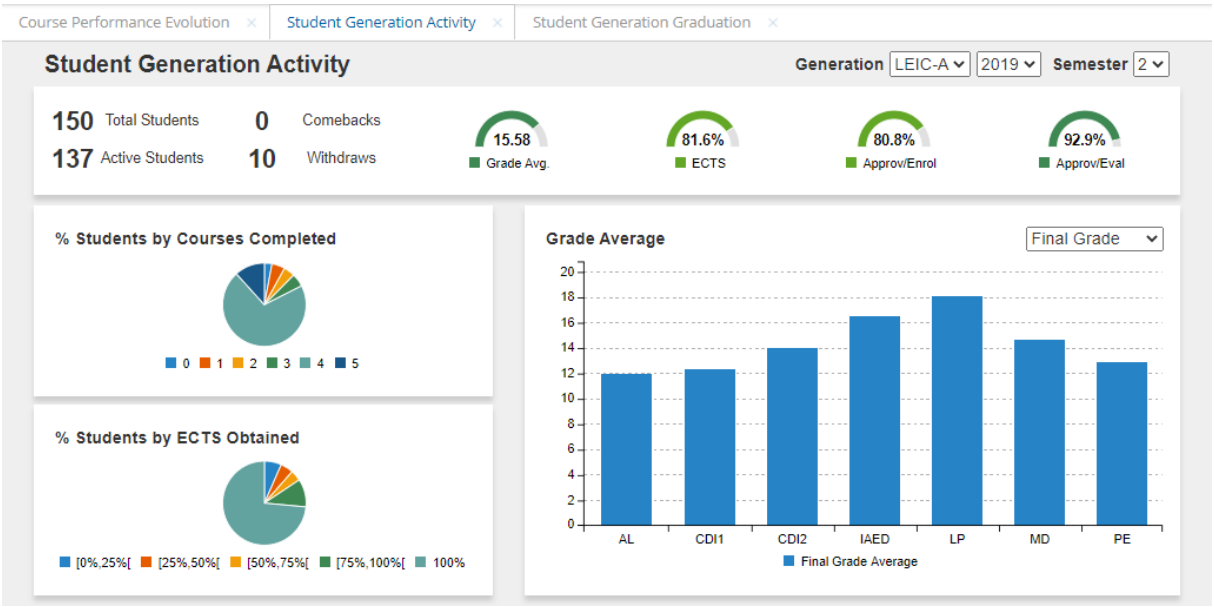

Figure 5.20: Student generation activity dashboard layout

The layout of the student generation activity dashboard is the following:

1. Dashboard Filters

   Located at the top-right side of the dashboard. Responsible for selecting the data to be displayed by the dashboard.

   1.1. *Degree*, an option that allows the user to select the degree of admission of a generation of students, out of a list of all degrees.

   1.2. *Year*, an option that allows the user to select the academic year that corresponds to the admission year of a generation of students, out of a list of all academic years.

   1.3. *Semester*, an option that allows the user to select the semester (since the admission) in which the evaluations took place, out of a list of all semesters in which there has been at least one active student.

2. General Indicators Panel

   The first panel located at the top of the dashboard. Displays indicators about the general performance of the generation of students selected.

   2.1. *Total Students*, a textual indicator of the total number of students initially admitted in the generation of students selected.

2.2. *Active Students*, a textual indicator of the number of active students from the generation of students selected and in the semester selected.

2.3. *Comebacks*, a textual indicator of the number of comebacks from the generation of students selected and in the semester selected.

2.4. *Withdraws*, a textual indicator of the number of withdraws from the generation of students selected and in the semester selected.

2.5. *Grade Average*, a gauge indicator of the final grade average obtained by the generation of students selected and in the semester selected.

3. Students by Courses Completed Panel

   The first panel located at the bottom-left side of the dashboard.

   3.1. *Students by Courses Completed*, presented as a pie chart where each section indicates the number and percentage of students from the student generation selected, that completed a certain number of courses in the semester selected.

4. Students by ECTS Obtained Panel

   The second panel located at the bottom-left side of the dashboard.

   4.1. *Students by ECTS Obtained*, presented as a pie chart where each section indicates the number and percentage of students from the student generation selected, that obtained a certain percentage of ECTS (i.e, [0%,25%[, [25%,50%[, [50%,75%[, [75%,100%[, 100%) in the semester selected.

5. Course Grade Average Panel

   The panel located at the bottom-right side of the dashboard.

   5.1. *Grade Type (panel filter)*, an option that allows the user to select the type of grade that the chart should display (i.e., Regular Grade, Improvement Grade, Special Grade, Final Grade).

   5.2. *Grade Average by Course*, presented as a bar chart where each bar indicates the grade average of a course from the semester selected and based on the grade type selected.

Student Graduation

In terms of graduation, the LEIC-T Coordinator analyzes the number of students that successfully obtained the total number of ECTS required to complete the degree. The LEIC-T Coordinator was also interested in finding out how many students completed their degree after its expected duration (i.e., 6 semesters), as well as the number of students that remain active after that duration. Additionally, the biggest difficulties of the generations are highlighted, by showing the number of withdraws and comebacks registered in each semester since admission, as well as the number of students that failed a certain course but were able to pass all others in a semester.

All these measures are presented in the student generation graduation dashboard, whose layout is presented in Figure 5.21, that displays information about the generation of 2015 from LEIC-A.

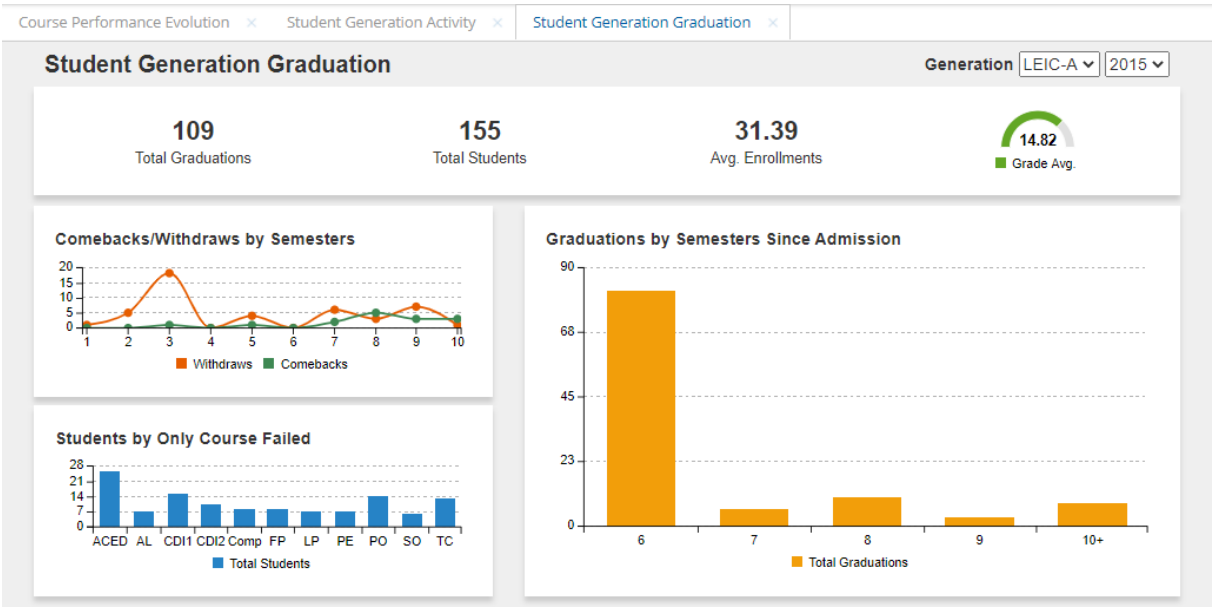

Figure 5.21: Student generation graduation dashboard layout

The layout of the student generation graduation dashboard is the following:

1. Dashboard Filters

   Located at the top-right side of the dashboard. Responsible for selecting the data to be displayed by the dashboard.

   1.1. *Degree*, an option that allows the user to select the degree of admission of a generation of students, out of a list of all degrees.

   1.2. *Year*, an option that allows the user to select the academic year that corresponds to the admission year of a generation of students, out of a list of all academic years.

2. General Indicators Panel

   The first panel located at the top of the dashboard. Displays indicators about the general performance of the generation of students selected.

   2.1. *Total Graduations*, a textual indicator of the total number of students from the generation of students selected that has already graduated.

   2.2. *Total Students*, a textual indicator of the total number of students initially admitted in the generation of students selected.

   2.3. *Average Enrollments*, a textual indicator of the average number of enrollments that the generation of students selected needed to graduate.

   2.4. *Grade Average*, a gauge indicator of the final grade average obtained by the graduates of the generation of students selected.

3. Comebacks/Withdraws by Semester Panel

   The first panel located at the bottom-left side of the dashboard.

   3.1. *Comebacks/Withdraws by Semester*, presented as a line chart where two lines indicate the percentage of students from the student generation selected that registered a withdraw/comeback in each semester since admission.

4. Students by Only Course Failed Panel

   The second panel located at the bottom-left side of the dashboard.

   4.1. *Students by Only Course Failed*, presented as a bar chart where each bar indicates the percentage of students from the student generation selected that failed the course and passed all the other courses in which they were enrolled to in a semester.

5. Graduations by Semesters Since Admission Panel

   The panel located at the bottom-right side of the dashboard.

   5.1. *Graduations by Semesters Since Admission*, presented as a bar chart where each bar indicates the percentage of students from the student generation selected that graduated in that semester.

5.6.2 Course Performance

In terms of courses, the LEIC-T Coordinator was particularly interested in analyzing the evolution of a course throughout the years. The LEIC degree has been functioning since 1989 and several courses have existed since then and until today, albeit with updated contents and names. As the courses are constantly being updated, it is fundamental to be able to compare the current edition of a course with its previous editions, to understand whether the changes affected the performance of the course. To do so, the most prominent measure to analyze is how successful were the students in that edition of the course, which can be measured through the approval rate. The LEIC-T Coordinator is interested in measuring the overall approval rates, as well as the first enrollment approval rates.

All these measures are presented in the course performance evolution dashboard, whose layout is presented in Figure 5.22, that displays information about the Probabilities and Statistics course.

The layout of the course performance evolution dashboard is the following:

1. Dashboard Filters

   Located at the top-right side of the dashboard. Responsible for selecting the data to be displayed by the dashboard.

   1.1. *Course*, an option that allows the user to select a course, out of a list of all courses.

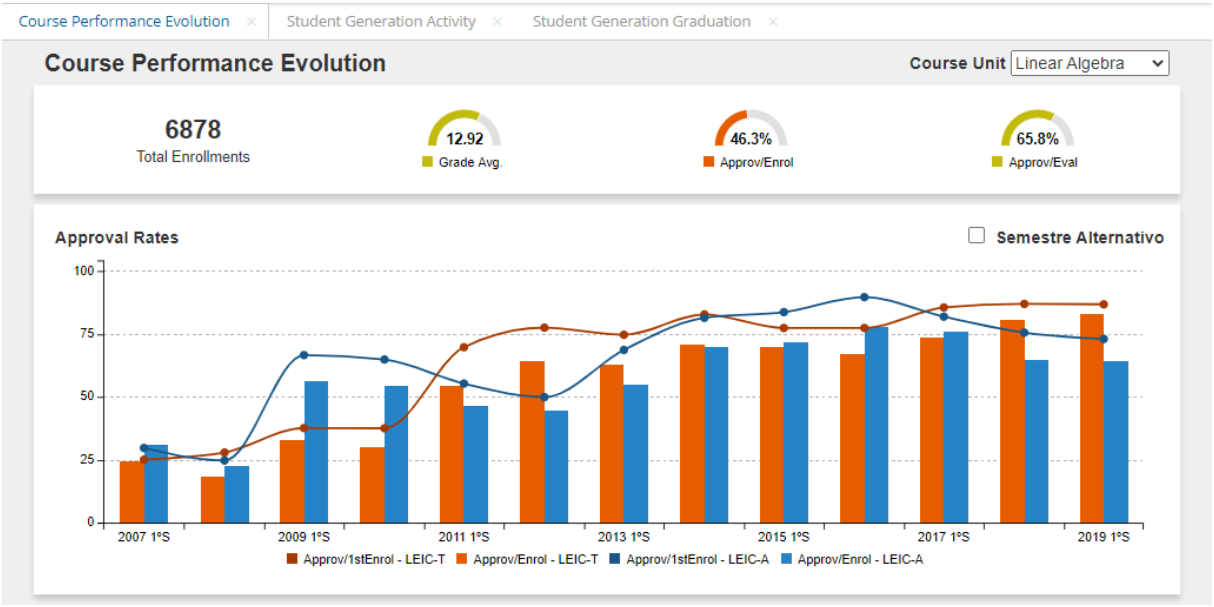

Figure 5.22: Course performance evolution dashboard layout

2. General Indicators Panel

   The first panel located at the top of the dashboard. Displays indicators about the general performance of the course selected.

   2.1. *Total Enrollments*, a textual indicator of the total number of enrollments the course selected had, in all years.

   2.2. *Grade Average*, a gauge indicator of the final grade average obtained in the course selected, in all years.

   2.3. *Approvals by Enrollments*, a gauge indicator of the approval/enrollments rate obtained in the course selected, in all years.

   2.4. *Approvals by Evaluations*, a gauge indicator of the approval/evaluations rate obtained in the course selected, in all years.

3. Approval Rates Panel

   The panel located at the bottom of the dashboard.

   3.1. *Alternate Semester (panel filter)*, an option that allows the user to select which semesters are displayed (i.e., regular semesters or alternate semesters).

   3.2. *Approval Rates by Semester*, presented as a bar/line chart in which the bars indicate the overall approval rates and the lines indicate the first enrollment approval rates of the course selected, in the selected semesters of all years.

Chapter 6

Experimental Validation

This chapter describes the experimental validation of the proposed solution and the results obtained from it. Section 6.1 describes a validation of the dimensional model using validity matrices. Section 6.2 describes a validation of data integrity, by comparing the data stored in our DW against the data stored in the Excel files currently produced by the LEIC-T Coordinator. Section 6.3 describes the validation of the dashboards in terms of usability, through a series of usability tests with different users.

6.1 Dimensional Model Validation

In this section, we describe the validation of the dimensional model, in terms of its capability for obtaining the required KPIs. To determine whether the various dimensions and facts involved in the dimensional model can obtain the KPIs identified in Section 4.2, we created validity matrices.

A *validity matrix* is a tabular structure that contains in its rows the dimensions and factual measures and, in its columns the KPIs that must be obtained. Each cell is marked with an ”X”, whenever the dimension or fact is involved in obtaining a KPI. This validation was performed after the logical design of the dimensional model, to ensure it complies with the business requirements before implementing the physical model.

Table 6.1 presents the validity matrix for the course related KPIs. All the course related KPIs can be obtained from the measures within the student evaluation fact table, which means that the student, course, evaluation time and admission time dimensions are required. Q1.1.1, the number of students enrolled to a course, needs no factual measures, since it is an event that occurs whenever a student is associated to a course and a time instant. Q1.1.2 to Q1.1.5 and Q1.2.2 to Q1.2.3 are related to the evaluation of students in a course and, therefore, require the *evaluationStatus* measure. Q1.2.1, to Q1.2.3 are related to first enrollments, which require the *isFirstEnrollment* measure.

Table 6.2 presents the validity matrix for the student generation related KPIs. The student generation related KPIs are obtained from the student activity and graduation fact tables, which means they require the student, degree, evaluation time and admission time dimensions. Q2.1.1 is obtained directly by the *finalGradeAverage*. Q2.1.2 measure the percentage of ECTS obtained out of all ECTS possi-

	Q1.1.1	Q1.1.2	Q1.1.3	Q1.1.4	Q1.1.5	Q1.2.1	Q1.2.2	Q1.2.3
Dimensions								
Degree								
Course	X	X	X	X	X	X	X	X
Student	X	X	X	X	X	X	X	X
Admission Time	X	X	X	X	X	X	X	X
Evaluation Time	X	X	X	X	X	X	X	X
Student Evaluation Fact Measures								
firstEnrollment(0/1)						X	X	X
firstImprovement(0/1)								
evaluationStatus		X	X	X	X		X	X
regularGrade								
improvementGrade								
specialGrade								
finalGrade								

Table 6.1: Course KPIs Validation Matrix

ble, which can be obtained with the *numberOfECTSObtained* and *numberOfECTSPossible* measures. Q2.1.3 and Q2.1.4 are ratios of approvals against enrollments and evaluations respectively; both require the *numberOfCoursesApproved*, with the former requiring the *numberOfCoursesEnrolled* and the latter requiring the *numberOfCoursesEvaluated*. Q2.2.1 and Q2.2.2 indicate the number of enrollments and approvals respectively, which require the *numberOfCoursesEnrolled* and *numberOfCoursesCompleted* measures, whereas Q2.2.3 requires both measures since it is a ratio of the two. Q2.3.1 indicates the number of students that did not pass a course, having passed all others, which requires the *onlyFailedCourseID* measure. Q2.4.1 indicates the number of graduations by number of semesters since admission, which requires the measure *numberOfSemestersSinceAdmission* from the student graduation fact table. Q2.5.1 and Q2.5.2 indicate the number of withdrawals and comebacks respectively, which can be obtained through the *isWithdrawal* and *isComeback* measures from the student activity fact table. Finally, Q2.5.3 and Q2.5.4 indicate the number of active and inactive students respectively, which can be obtained through the *isActive* measure (i.e., true for active, false for inactive) from the student activity fact table.

Having created the validity matrices, we have verified that the dimensional model created is able to provide answers to all the business questions identified in Section 4.2.

6.2 Data Integrity Validation

A fundamental aspect of a decision support system is the integrity of the data within it. To guarantee that the data is accurate, the data stored in our DW was validated against the data contained in the Excel files created by the LEIC-T Coordinator.

For visualization purposes, the LEIC-T Coordinator possesses another group of Excel files, that apply formulas to the Excel files described in Section 4.1, to obtain various metrics related to the performance of the generations of students. This group of files consisted of one file per degree and academic year. Each Excel file contained, for every student and semester, metrics related to student evaluation (i.e.,

	Q2.1.1	Q2.1.2	Q2.1.3	Q2.1.4	Q2.2.1	Q2.2.2	Q2.2.3	Q2.3.1	Q2.4.1	Q2.5.1	Q2.5.2	Q2.5.3	Q2.5.4
Dimensions													
Degree	X	X	X	X	X	X	X	X	X	X	X	X	X
Course													
Student	X	X	X	X	X	X	X	X	X	X	X	X	X
Admission Time	X	X	X	X	X	X	X	X	X	X	X	X	X
Evaluation Time	X	X	X	X	X	X	X	X	X	X	X	X	X
Student Activity Fact Measures													
isActive(0/1)												X	X
isWithdraw(0/1)										X			
isComeback(0/1)											X		
numberOfCourses Enrolled			X		X		X						
numberOf CoursesCompleted			X	X		X	X						
numberOf CoursesEvaluated				X									
numberOf ECTSObtained		X											
numberOf ECTSPossible		X											
finalGradeAverage	X												
onlyFailedCourseID								X					
Student Graduation Fact Measures													
numberOfYears SinceAdmission													
numberOfSemesters SinceAdmission									X				
numberOfPeriods SinceAdmission													
numberOfCourses Enrolled													
numberOf CoursesCompleted													
numberOf CoursesEvaluated													
finalGradeAverage													

Table 6.2: Student Generation KPIs Validation Matrix

final grade) and student activity metrics (i.e., only course failed, approvals, enrollments, ECTS obtained, ECTS possible, withdraws and comebacks). Student graduation metrics are not obtained by these Excel files, so they can not be compared.

The LEIC-T Coordinator warned that these files were not always populated with the most up-to-date data possible. In some cases, usually in the first semesters, the grade files used by the LEIC-T Coordinator were missing the special grades. This happened due to the grade files being extracted from the IST Fénix system after the end of the semester, but before the special exams took place. This was not the case when it comes to our system: all the files used as input already include the special grades. Due to this slight discrepancy, minor deviations are expected.

Another issue the LEIC-T Coordinator pointed out is the fact that these Excel files may be incorrect in terms of the sets of students considered for the student generation analysis, since they were manually selected. The students should be selected according to the following criteria: the students had to be listed in the admission files and had to be enrolled to all the first semester courses. The manual selection of the students is error prone, so there are cases in which some students do not meet at least one of the criteria, whereas the set of students we obtained always complied with the aforementioned criteria. Therefore, the two sets of students do not match, which may cause even more discrepancies when performing the comparisons.

To produce trustworthy comparisons, this group of Excel files takes as input should be updated with the special grades and the set of students should be updated according to the aforementioned criteria. To perform these comparisons, we considered only a portion of the available Excel files, since correcting the files has to be done manually and correcting all of them would require a considerable effort. As such, we selected only the files related to the LEIC-T degree, which include data from 2007 to 2019.

To perform the comparison, we created a Python script (Appendix B) that accesses the DW and reads the Excel files. The script obtains data from both data sources, for each academic year, then compares all the records obtained, and outputs a percentage of matching records. Whenever a student is present in one of the data sources but not on the other, we consider it a mismatch. We compared the data stored in our DW against the original Excel files and against the Excel files upon correcting their input data (i.e., updating the grades data and updating the set of students). The results yielded by these comparisons are presented in Table 6.3.

	Student Evaluation	**Student Activity**	**Overall**
Original Files	96,26%	77,88%	90,05%
Files with updated Grades	96,65%	79,10%	90,72%
Files with updated Grades and Students	100,00%	100,00%	100,00%

Table 6.3: Results of comparing data from SAD-CCIST against LEIC-T Coordinator Excel files

The lower accuracy values obtained in the first two comparisons (i.e., original files and files with updated grade data) are explained by the fact that there was a considerable difference between the sets of students being considered in the student generation analysis. Upon a detailed analysis of the Excel files, we determined that they considered a total of 1223 students, whereas in our system, only 1051

students were being considered part of the student generations admitted between 2007 and 2019. We then decided to go through each Excel file and determined which students fulfilled the criteria for being considered a part of the student generation. There were several students in the Excel files that had cancelled their admission, which was causing mismatches in terms of student activity (i.e., withdrawals), as most of these students had not enrolled to all first semester courses. As such, these students were being counted as first semester withdrawals, when they should have been excluded in the first place. After removing all students that did not fulfill the criteria, we were left with the 1051 students that matched the ones in our system, and the comparison yielded 100% of matches.

During this validation process, we came across a minor issue when calculating withdraws/comebacks. In the ETL processes (described in Section 5.4), we fill in the gaps of student activity, which are caused when students do not enroll to courses for at least one semester. Thanks to filling the gaps, there was an issue when calculating the previous semester, which is needed to determine if there was a modification in student activity, which determines if a withdraw/comeback occurred. This meant that in some cases, there was a slight discrepancy in the withdraw/comeback values when comparing the data. The result of 100% was obtained upon correcting this issue and repopulating the DW.

6.3 Usability Validation

In this Section, we describe the evaluation of the dashboards created in Section 5.6. To evaluate user interfaces such as dashboards, their usability must be measured. Usability is typically measured by having a set of users performing usability tests. These tests encourage the users to use the system and complete several tasks. The users that participate in the usability tests should be as representative as possible of the intended end users [18].

6.3.1 Test Users

We invited a set of users comprised of current and former degree coordinators, as well as administrative staff from the IST Department of Computer Science and Engineering (DEI). We considered these users the most representative possible, since they are the most likely to benefit from a system such as SAD-CCIST.

In total, there were 14 users participating in the usability tests. The users belong to different categories (i.e., Full Professor (PAC), Associate Professor (PAS) and DEI Administrative Services (SADEI)) and to different scientific areas of DEI (i.e., Architecture and Operating Systems (ASO), Programming Methodology and Technology (MTP), Computer Graphics and Multimedia (IG), Artificial Intelligence (IA), Information Systems (SI) and DEI Administrative Services (SADEI)) and half of them are or have been degree coordinators before, as seen in Figure 6.1.

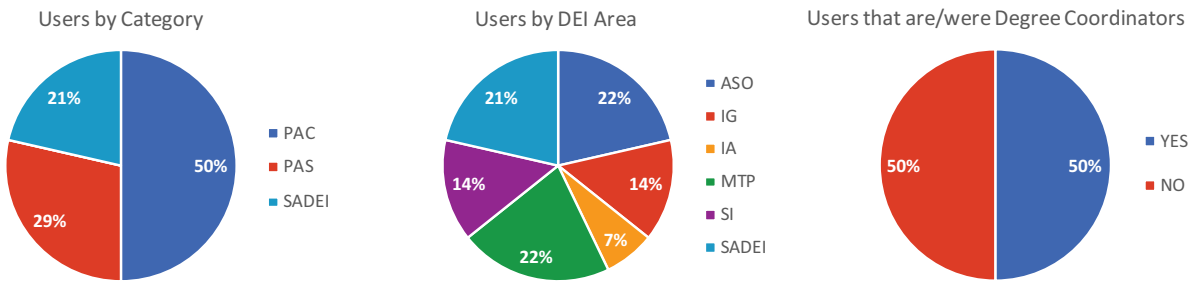

Figure 6.1: Profile of the test users

6.3.2 Usability Test

The users were introduced to a set of tasks that encouraged them to use the dashboards to obtain specific data, by interacting with the various visual components (i.e, filters, graphics, tooltips). The goal is to assess whether the design of the dashboards allows the users to understand, at first glance, what information they are being presented with and if they can easily obtain certain information from the dashboards. We created three groups of tasks, one for each dashboard (i.e., student generation activity, student generation graduations, course performance evolution). The tasks are the following:

1. Observe the performance of the generation of students of 2019 from LEIC-A in their 2nd semester.
   1.1. Indicate the course in which the students had a better final grade average.
   1.2. Indicate the number of students that obtained 100% of the ECTS.
2. Observe the graduations of the generation of students of 2015 from LEIC-A.
   2.1. Indicate the number of semesters that the majority of students took to graduate.
   2.2. Indicate the semester with the most withdrawals.
   2.3. Indicate the course that the students have struggled with the most.
3. Observe the approval rate evolution of the *Probabilities and Statistics*[1] course, throughout the years.
   3.1. Indicate the time instant in which the approval rate is the lowest for LEIC-T.
   3.2. Indicate the degree in which the approval rates are generally higher.
   3.3. Indicate the alternate semester in which the approval rate is the highest for LEIC-A.

The users were encouraged to use the system on their own, as much as possible, though they may require assistance to complete the tasks. We opted by not letting the users fail the tasks, no matter how long they may take, and instead of measuring their success through task completion, we decided to record the following measures:

- *Elapsed time*, the time taken by the users to complete the task.

[1] The course of Probabilities and Statistics was chosen to ensure that the users interact with the dashboard filter, that presents all courses in alphabetic order.

- *Number of errors*, considering an error an incorrect result in the task.
- *Number of assistances*, the times the users asked for the assistance of the moderator.

Right upon completing each group of tasks, users were asked a *single ease question* [19], to assess the difficulty of each group of tasks, on a scale of 1 to 7 (i.e., very difficult to very easy). When the users considered the task difficult (i.e., above 5), they were asked to justify their answer. Not only does this help us measure the satisfaction of the users, but it also allows us to identify whether the users struggle with a particular dashboard.

Finally, after completing all the tasks, the users were asked to evaluate the overall usability of the dashboards. We used the *System Usability Scale* (SUS), which is a 10 item questionnaire, with response options that range from 1 to 5 (i.e., strongly disagree to strongly agree) [20]. The questions that compose SUS are the following:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

SUS yields a score that ranges from 0 to 100, that represents a composite measure of the overall usability of the system. For a system to be considered acceptable in terms of usability, it should have a SUS score of at least 68 [6] (i.e., average SUS score). Table 6.4 indicates how SUS scores should be interpreted.

SUS Score	**Grade**	**Adjective Rating**
80.3 – 100	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
0 – 51	F	Awful

Table 6.4: SUS scores meaning [6]

All the materials used for conducting the usability tests are presented in Appendix C, which includes the session guide and the forms the users had to fill in.

6.3.3 Test Results

The results obtained from the usability test sessions provided an overview of the overall usability of our system and helped identifying some areas of improvement.

Figure 6.2 presents the average of all the measures recorded for each task. The three tasks were considered relatively easy by the users, with an average of *5.64*, *5.86* and *4.93*, which are all values above average (i.e., on a scale of 1 to 7, that means very difficult and very easy respectively). Despite this fact, the third task, that consisted of using the course performance evolution dashboard, presented less desirable results than the first two. Users took more time performing this task, with an average of *03:38* minutes spent. This task also had the highest average number of errors and assistances, which, unsurprisingly, led to being considered the most difficult of the three. The users struggled with this dashboard in particular, due to the approval rates chart consisting of bars and lines of similar colors, which made the users confused, since it was not clear what each meant at first glance.

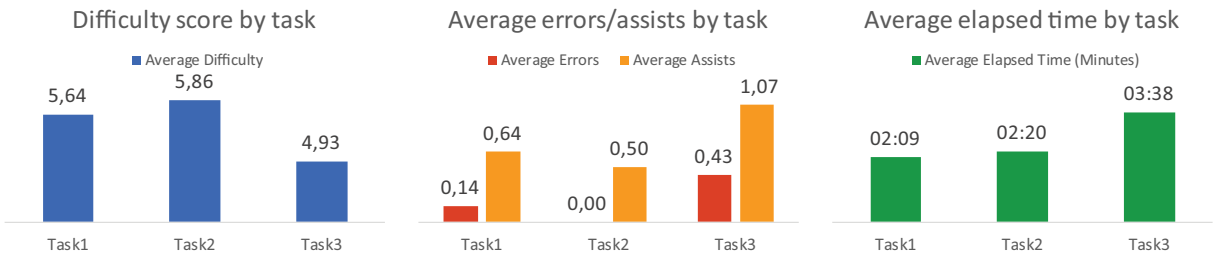

Figure 6.2: Performance measures of each set of tasks

Based on the users' answers to the SUS questionnaire, we computed the average, standard deviation, maximum and minimum SUS scores, which are presented in Table 6.5.

Average	Std. Deviation	Maximum	Minimum
85.18	17.74	100.00	42.50

Table 6.5: SUS average, standard deviation, maximum and minimum score values

The average SUS score was *85.18*, which indicates an overall excellent usability score according to this scale. The standard deviation value of *17.74* is slightly high, which indicates that, despite most users giving the system a very good score, a few users did not consider the system as good, as also suggested by the maximum and minimum scores of *100* and *42.5*. For this reason, we decided to further investigate how the users evaluated the system in terms of usability. Figure 6.3 shows the number of evaluations by each SUS score grade.

It is possible to confirm that the majority of the users did, indeed, give our system a good score. A total of 11 out of the 14 test users considered the system *excellent* and 1 user considered the system *good*, with all these scores being of *80* and higher. Only 2 users considered the system as *awful*, giving it scores of *42.5* and *50*. Users that struggle during the usability tasks tend to give a lower grade to the system [20], which was exactly what happened in this case. The time taken by these users, as well as the number of errors and assistances was always close to or above average for each task, which may explain why they ranked the system so low. Nonetheless, this indicates that while the system was

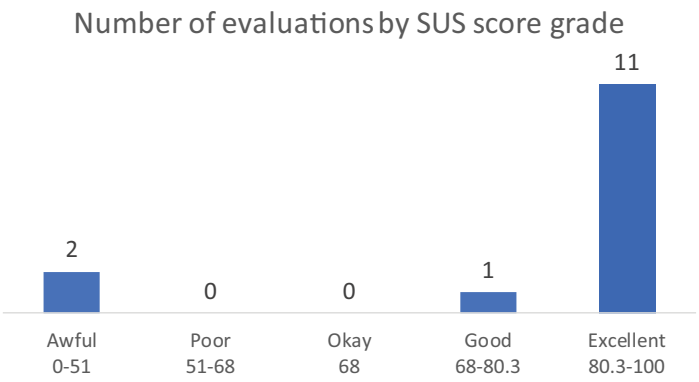

Figure 6.3: Evaluations by SUS score grade

regarded as very good by most users, it can still improve in terms of usability.

6.3.4 User Feedback

During the session, the users provided spoken feedback as well as written feedback, as the questionnaire included an optional field in which users were encouraged to write what they believed that SAD-CCIST could do to provide a better experience.

Overall, the users found SAD-CCIST to be usable, but also useful. Most users considered that the information displayed by the dashboards provides useful insights about the performance of the students and courses of the LEIC degrees.

The users proposed various suggestions that could improve the overall design of the dashboards, as well as the design of each dashboard individually.

1. Overall Design
    1.1. Remove the title from the dashboard, since it is redundant, as it appears in the dashboard tabs.
    1.2. The filters should be placed on the top-left side of the dashboard (where currently the title is placed), instead of the top-right, to be perceived better.
2. Student Generation Activity Dashboard
    2.1. For better perception of the pie charts, display the values over each section, and not only when placing the mouse over one of the sections of the chart.
3. Student Generation Graduation Dashboard
    3.1. Display the metrics as percentages instead of absolute values, as percentages are more appropriate for comparing the graduation measures of different student generations.
4. Course Performance Evolution Graduation Dashboard
    4.1. Make different visualizations for the approval rates and the first enrollment approval rates, since the two metrics in the same chart is not well perceived by the users.

4.2. Improve the alternate semester filter, so that users may also be able to view the approval rates in regular and alternate semesters at the same time and not just one or the other.

Some of these suggestions have already been added to the dashboards, namely removing the titles and placing the filters on the top-left side of all dashboards (i.e., 1.1 and 1.2) and replacing all absolute values with percentages on the student generation graduation dashboard (i.e., 3.1).

Some suggestions were not possible to implement at the moment, as they require a more careful planning. 2.1 suggests that each section of the pie charts should display their value, but when including the values, the labels overlapped other sections and left the pie charts almost unreadable. 4.1 involves creating a new visualization, which requires an analysis of how the new visualization would fit the current layout. 4.2 involves changing the filter and its functionality, which requires a modification of the filter's code and the query responsible for obtaining the data.

Chapter 7

Conclusions

In this document we present the proposal of a DSS for IST degree coordination, known as SAD-CCIST. This system arises from the need for automating the current manual process performed by the LEIC-T Coordinator, that consists of extracting data related to the performance of students and courses of a degree, processing the data and generating visualizations from it.

7.1 Summary

An initial study regarding the concepts of a DSS was performed in Chapter 2, which details the architecture of its main component, the DW, as well as its standard design methodologies. We also analyzed the concepts specific to the higher education domain, especially applicable to the reality inside IST.

To understand how DSSs have been implemented in the higher education context, in Chapter 3, we analyzed relevant works in the area. We analyzed implementations of DSSs for three different higher education institutions, and with different scopes, such as admissions and academic performance. This analysis provides an overview of the methodologies used to implement the DSS solutions, as well as the validation methodologies, when included. We also assessed some of the most prominent business intelligent stacks, to determine which would better suit our system's needs.

The system should be tailored to the LEIC-T Coordinator's needs, which led to the specification of the business requirements, presented in Chapter 4. We thoroughly described the input data available and its structure. Based on the input data and interviews with the LEIC-T Coordinator, we determined the KPIs that the system should be able to obtain.

Considering the methodologies and implementation details from the previous analysis, as well as the business requirements specified we began the implementation of SAD-CCIST, described in Chapter 5. We identified the various layers that compose the architecture of our system, whose implementation details are then described. The available input data is the first layer and is composed of several Excel files with academic data. Through the extraction processes, the input data is stored in the DSA, a relational data base that follows the structure of the input data, the second layer of the architecture. The data from the DSA is used by the transformation and load processes, that transform the data and use it to

populate the DW, the third layer of the architecture. The fourth layer is the OLAP server, which introduces an OLAP model that performs precalculations and aggregations to the data from the DW, to make the data even more suited for fast analytical querying. Though an OLAP model was created, its inability for obtaining all the required KPIs led to its exclusion. As such, the final layer of the presentation layer, obtained the data directly from the DW. The presentation of the data is achieved through dashboards, that provide insights to the end users on three main subjects: student activity, student graduation and course performance.

The solution was evaluated in Chapter 6. We performed a validation of the dimensional model using validity matrices, that indicate which factual measures are needed to obtain each of the KPIs. The validity matrices proved our solution to be capable of obtaining all the required KPIs. The data stored in the DW was validated as well, by comparing it with the data processed by the LEIC-T Coordinator. There were slight discrepancies, due to an inconsistent selection of students in the LEIC-T Coordinator's files, that when manually corrected, resulted in a *100%* match between the two sets of data. Finally, we conducted usability tests with 14 test users. Using the SUS methodology, the users performed a series of usability tasks and were asked to fill in a questionnaire. SAD-CCIST obtained an average score of *85.18* out of 100, which the methodology considers to be an *excellent* usability score.

7.2 Future Work

While SAD-CCIST meets the specified business requirements, it could include more features and some of the current features could also be improved. In this section we analyze the limitations of our system and what can be added to improve our solution.

7.2.1 Dimensional Model

The dimensional model designed in the scope of this project, contains data about the performance of students and courses of the LEIC degree. The LEIC-T Coordinator has also been experimenting with data related to the the IST Course Unit Quality (QUC [1] in portuguese). This data could also be featured in our dimensional model, provided that new dimensions and facts related to that data are added. Since the dimensional model was designed according to Kimball's methodologies, it is possible to add new dimensions and facts in an iterative way, without having to modify the dimensions and facts already included.

7.2.2 ETL Processes

Currently, every time a new generation of students is admitted (i.e., at the beginning of each academic year) or every time the grades are released (i.e., at the end of each semester), the ETL loads all the input data to the DSA and DW. A full load is not the most efficient approach, especially when the input data grows every academic year/semester. The solution would be creating incremental ETL processes,

[1] https://quc.tecnico.ulisboa.pt/en/

to load new files incrementally, without having to replace all the data previously stored in the DSA and DW.

In the academic year of 2013/2014, the curricular plan was modified. When a curricular plan changes, usually the ECTS awarded by certain courses also change. When students complete a course in the years before the change, they are awarded with a certain amount of ECTS. If they have not graduated when the change takes place, the ECTS they were awarded before are subjected to change if the course now awards a different amount of ECTS. There is currently no way to deal with these changes, which is causing an incorrect number of graduations in the years close to 2013.

Starting in the 2021/2022 academic year, the curricular plan will be restructured. This change introduces courses that are taught on a quarter basis (i.e., academic periods), rather than being taught only on a semester basis. We have prepared the time dimension to handle these academic periods, but it was a feature left unexplored, as no actual data was available at the time being. This change would also imply that some dashboards need to be changed, as they are working only on a semester basis.

Whenever the data provided as input to the system does not meet certain requirements, a warning should be issued. Duplicate data on any files should always issue a warning. When loading the grade data, there are certain events that should trigger a warning, such as empty course sheets, or sheets with no evaluation data (i.e, grades or evaluation state). When loading the admission data, a warning should be issued when a student has already been admitted to another degree. These warnings would help the degree coordinators to keep track of errors and outliers.

Instead of using Excel files extracted from the IST Fénix system, an integration with Fénix would be beneficial for our system. Currently, the Excel files must be manually extracted and provided to the system as input, which is error prone. If the system was able to access the data directly, it would be possible to make the system less dependent of human action. Furthermore, it would be possible to obtain data for all other degrees from IST, enabling a more widespread usage of our system.

7.2.3 OLAP Model

The OLAP model created in the scope of this project was limited in its capability for obtaining all the required KPIs, which eventually led to its exclusion. The system would benefit from including a fully functional OLAP model, capable of obtaining all KPIs, since it increases the efficiency of analytical queries such as the ones performed by our system. With time, the data will grow considerably and, while currently there are no noticeable performance issues, the efficiency of the system will become an increasing concern.

With an OLAP model, the system would also be able to include an ad-hoc querying feature. The Pentaho BA platform, the tool used for visualization, is capable of integrating with Saiku[2], and offers the end user the possibility of performing ad-hoc queries over an OLAP model. Currently, the system provides the end users with fixed information, suited to the LEIC-T Coordinator's needs. The inclusion of ad-hoc queries would allow the end users to freely obtain information that lies within the system, but is not currently possible to visualize.

[2]https://www.meteorite.bi/products/saiku/

7.2.4 Dashboards

SAD-CCIST currently offers three different dashboards, that provide useful information to the end users. The information provided by these dashboards is only a portion of what the system has yet to offer. With that said, more dashboards could be created using the data currently stored in our DW. The LEIC-T Coordinator expressed interest in a dashboard that allows observing the evolution of the number of ECTS obtained by all generations of students, in each semester and degree years. Though our system includes a dashboard for analyzing the performance of the generations of students in each semester since admission, it focuses only in a specific generation of students. This new dashboard would allow allow a broader view of the performance of the student generations, as well as enabling a direct comparison of the performance of all student generations.

Additionally, the suggestions given by the test users that were not possible to include, should be implemented to improve the dashboards in terms of usability.

Bibliography

[1] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons, Inc., 3rd edition, 2013. ISBN 1118530802, 9781118530801.

[2] Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy, and Bob Becker. *The Data Warehouse Lifecycle Toolkit*. John Wiley & Sons, Inc., 2nd edition, 2008. ISBN 0470149779, 9780470149775.

[3] Elsa Cardoso, Helena Galhardas, Maria Trigueiros, and António Rito Silva. A decision support system for IST academic information. *Informatica (Ljubljana)*, (3), 2003.

[4] Elsa Cardoso. Sistema de apoio à decisão para a informação académica do Instituto Superior Técnico. Master's thesis, Instituto Superior Técnico, 2003.

[5] Isam M Aljawarneh. Design of a data warehouse model for decision support at higher education: A case study. *Information Development*, 32(5), 2016.

[6] Jeff Sauro. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC, 2011.

[7] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of Data Integration*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2012. ISBN 0124160441, 9780124160446.

[8] Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1), 1997.

[9] Ralph Kimball and Joe Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data*. John Wiley & Sons, Inc., USA, 2004. ISBN 0764567578.

[10] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN 0123814790, 9780123814791.

[11] Ralph Kimball. Letting the users sleep, part 1. *DBMS*, 9(13), December 1996. ISSN 1041-5173. URL `http://dl.acm.org/citation.cfm?id=247582.247594`.

[12] Ralph Kimball. Letting the users sleep, part 2. *DBMS*, 10(1), January 1997. ISSN 1041-5173. URL http://dl.acm.org/citation.cfm?id=265117.265120.

[13] Eka Miranda, Eli Suryani, et al. Implementation of datawarehouse, datamining and dashboard for higher education. *Journal of Theoretical & Applied Information Technology*, 64, 2014.

[14] Global design of the national higher education information system (SINAS-DIKTI). Technical report, Directorate General of Higher Education, 1990.

[15] K Ram Ramamurthy, Arun Sen, and Atish P Sinha. An empirical investigation of the key determinants of data warehouse adoption. *Decision support systems*, 44, 2008.

[16] Henry Y Zheng. Business intelligence as a data-based decision support system and its roles in support of institutional research and planning. *Institutional Research and Planning in Higher Education: Global Contexts and Themes*, 2015.

[17] Ehtisham Zaidi, Eric Thoo, and Nick Heudecker. Magic quadrant for data integration tools. *Gartner*, 2019.

[18] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.

[19] Jeff Sauro and Joseph S Dumas. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1599–1608, 2009.

[20] John Brooke. Sus: a “quick and dirty’usability. *Usability evaluation in industry*, page 189, 1996.

Appendix A

Pentaho Jobs and Transformations

This appendix contains all the jobs and transformations that compose the ETL processes. The jobs and transformations are files created using Pentaho Data Integration.

A.1 Extraction Processes

The following figures represent the job and transformations that compose the Extraction processes. Figure A.1 is a job and Figures A.2, A.3, A.4, A.5 and A.6 are transformations that are a part of that job.

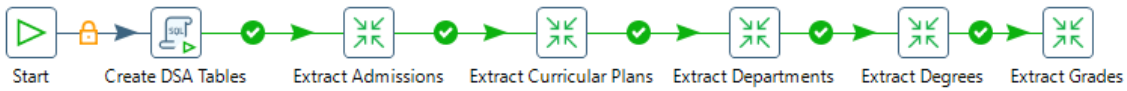

Figure A.1: Extraction job

Figure A.2: Extract admissions transformation

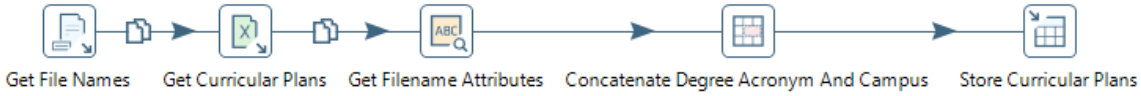

Figure A.3: Extract curricular plans transformation

Figure A.4: Extract departments transformation

Figure A.5: Extract degrees transformation

Figure A.6: Extract grades transformation

A.2 Load-Transform Processes

The following figures represent the job and transformations that compose the Load-Transform processes. Figure A.7 is a job and Figures A.8, A.9, A.10, A.11, A.12, A.13, A.14, A.15, A.16 and A.17 are transformations that are a part of that job.

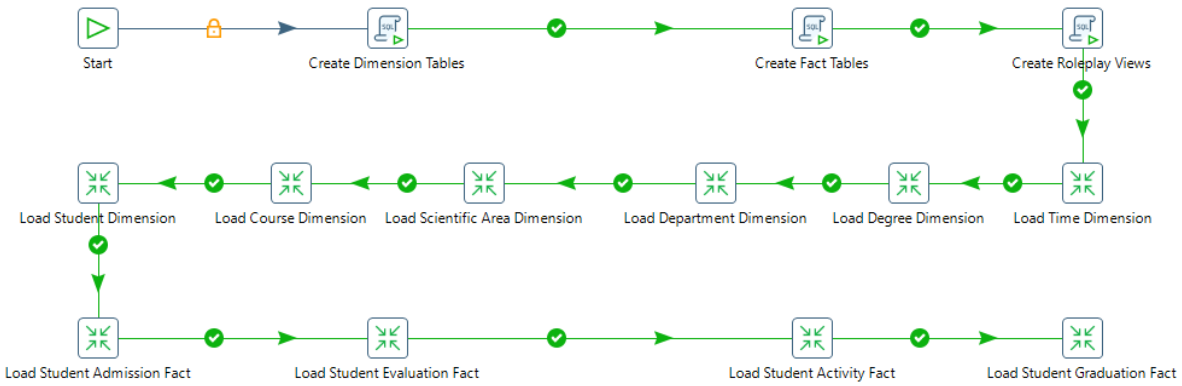

Figure A.7: Transform-load job

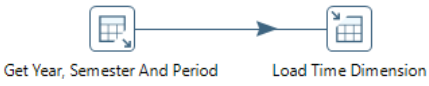

Figure A.8: Load time dimension transformation

Figure A.9: Load degree dimension transformation

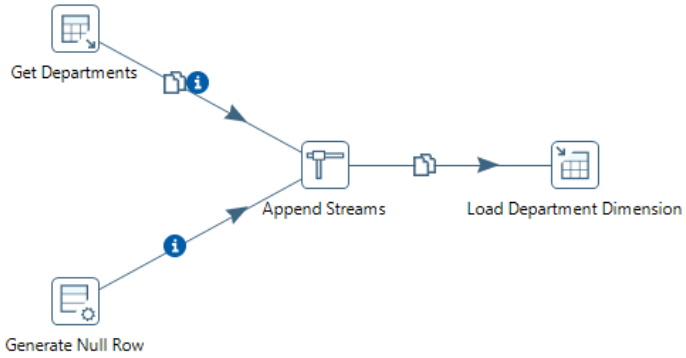

Figure A.10: Load department dimension transformation

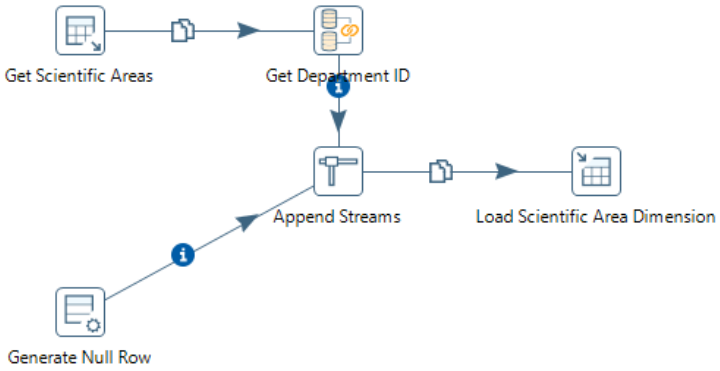

Figure A.11: Load scientific area dimension transformation

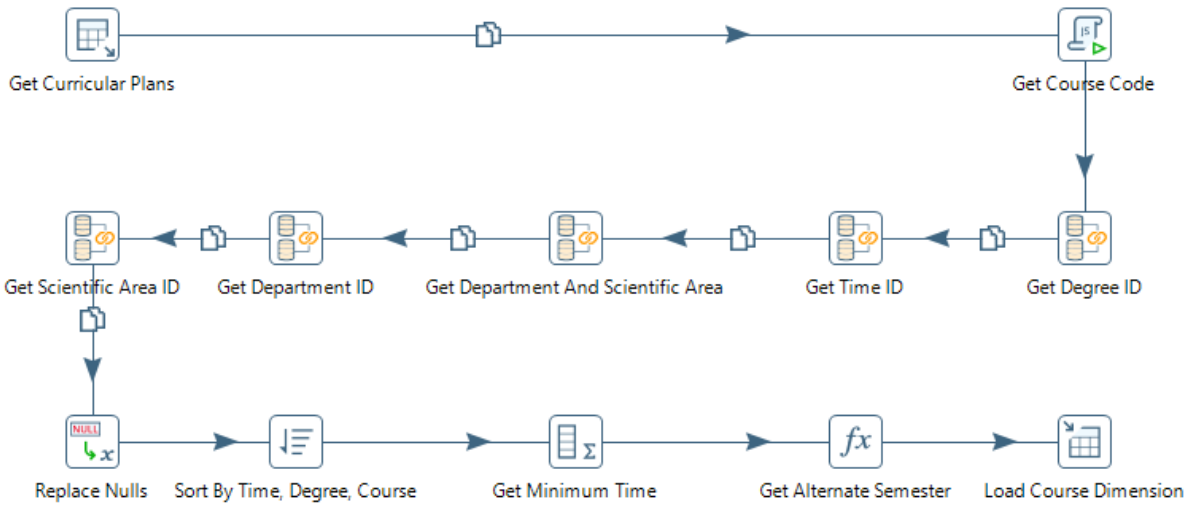

Figure A.12: Load course dimension transformation

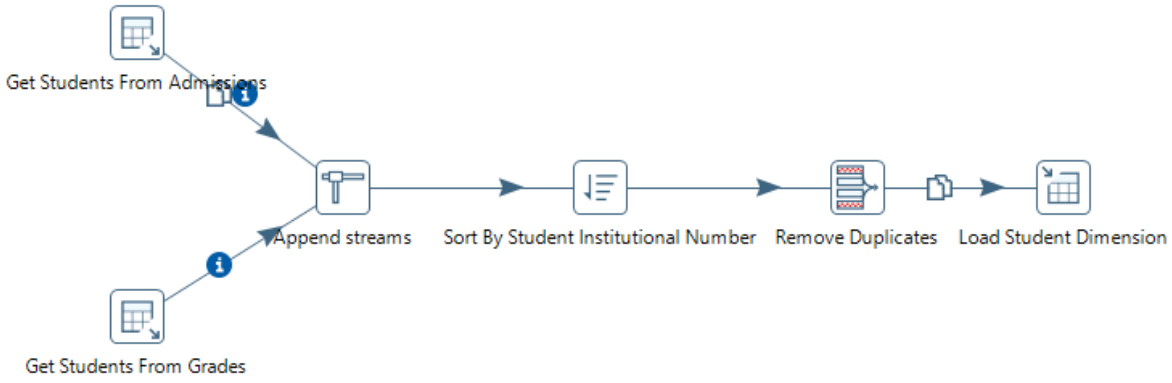

Figure A.13: Load student dimension transformation

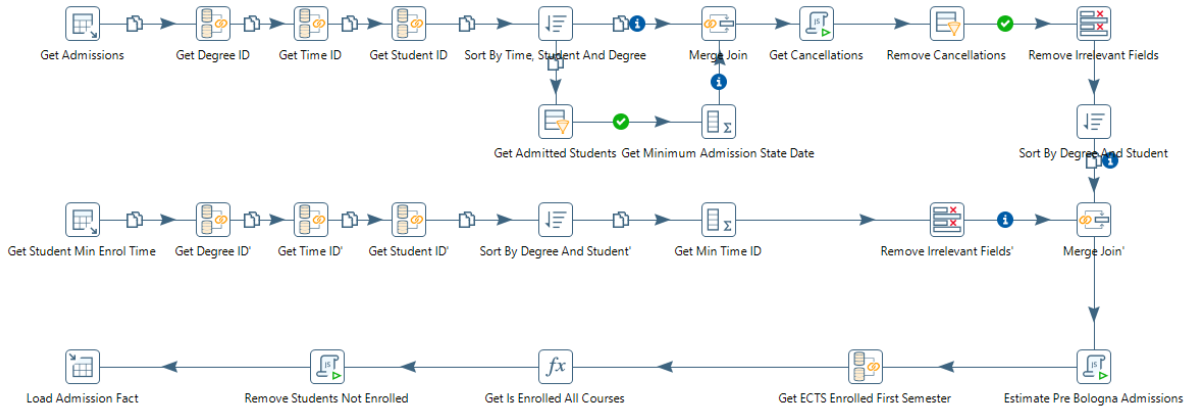

Figure A.14: Load student admission fact transformation

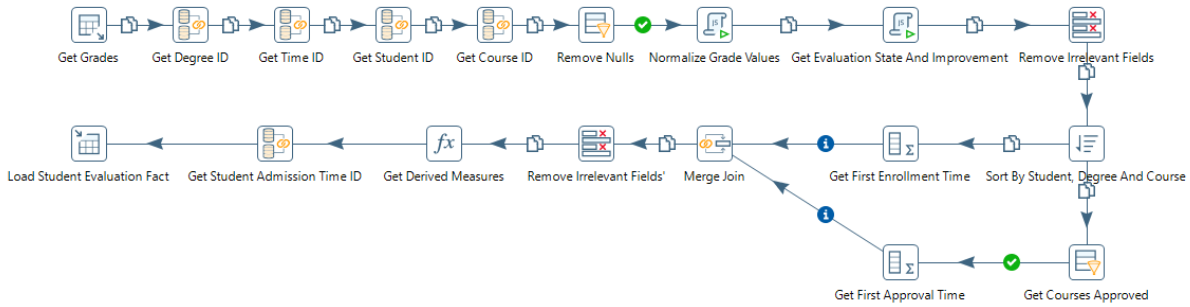

Figure A.15: Load student evaluation fact transformation

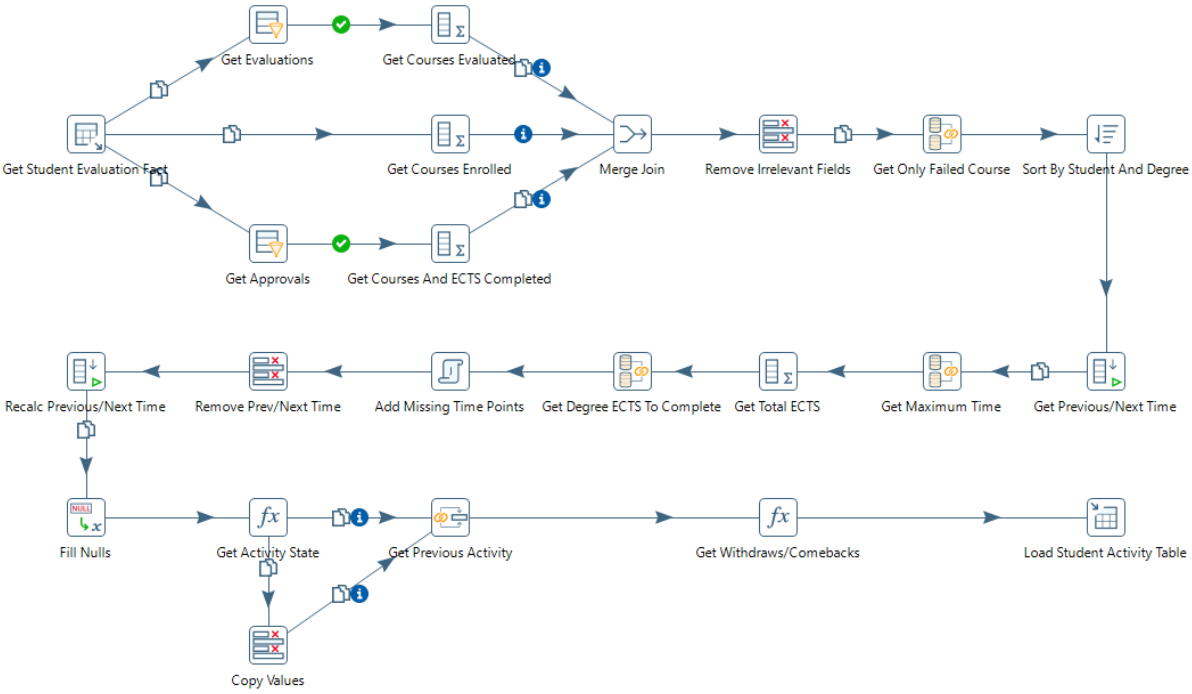

Figure A.16: Load student activity fact transformation

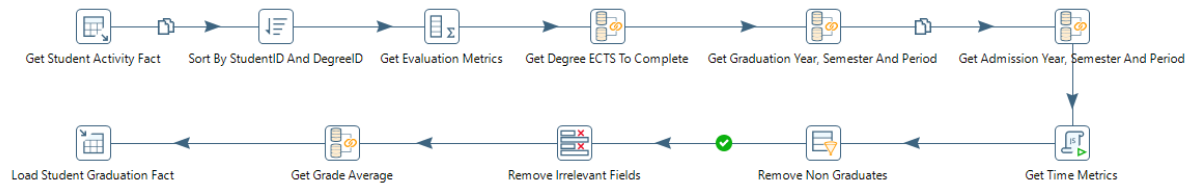

Figure A.17: Load student graduation fact transformation

Appendix B

Data Integrity Validation

This appendix presents the Python script used to evaluate the integrity of the data stored by our DW. This script accesses the DW and reads the Excel files produced by the LEIC-T Coordinator (and the corrected versions of these files), comparing grades, activity measures, withdraws and comebacks. It outputs a percentage of matching records for each type of measures compared and an overall percentage of matching records. The script is presented in Listing B.1.

```
import pickle
import numpy as np
import pandas as pd
import mysql.connector

def evaluate_student_grades(data, degree, year):
    errors = 0
    fields = 0
    for semester in range(1, 7):
        sheet = data[str(semester) + "ºS"]

        query_filename = "../../sql/data_warehouse/dql/evaluation/
            query_student_generation_final_grades_by_course.sql"
        query_file     = open(query_filename,"r")
        query          = query_file.read()
        query_file.close()

        result = pd.read_sql(query%(degree, year, semester), con=con)
        dw_data = pd.DataFrame(columns=result["course_acronym"].unique())

        for student in result["student_institutional_number"].unique():
            row = pd.Series(index=dw_data.columns, dtype=object)
```

```
        row["ALUNO"] = student
        for i, grade in result.loc[result["student_institutional_number
            "] == student].iterrows():
            row[grade["course_acronym"]] = grade["final_grade"]
        dw_data = dw_data.append(row, True)
    dw_data = dw_data.rename(columns={"APSEI": "CS"})
    if dw_data.empty:
        continue
    dw_data["ALUNO"] = dw_data["ALUNO"].astype(int)
    dw_data = dw_data.set_index("ALUNO")
    dw_data = dw_data.astype(str)
    dw_data = dw_data.replace({"nan": "NI"})

    last_col = sheet.columns[sheet.isin(['Fase']).any()][0]
    last_row = sheet.index[sheet.iloc[:, 0]=="LAST"][0]

    ex_data = sheet.iloc[3:last_row, 0:last_col]
    ex_data.columns = list(sheet.iloc[2, 0:last_col])
    ex_data = ex_data.rename(columns={"APSEI": "CS"})
    ex_data["ALUNO"] = ex_data["ALUNO"].astype(int)
    ex_data = ex_data.set_index("ALUNO")
    ex_data = ex_data.loc[:, dw_data.columns]
    ex_data = ex_data.replace({np.nan: "NA", "NI": np.nan})
    ex_data = ex_data.dropna(thresh=1)
    ex_data = ex_data.astype(str)

    for column in ex_data.columns:
        ex_data[column] = ex_data[column].str.replace("\.0", "")

    ex_data = ex_data.replace({"nan": "NI"})

    if ex_data.shape[0] != dw_data.shape[0]:
        intersect = dw_data.index.intersection(ex_data.index).unique()
        difference = dw_data.index.difference(ex_data.index)
        errors += len(difference)*dw_data.shape[1]
        difference = ex_data.index.difference(dw_data.index)
        errors += len(difference)*ex_data.shape[1]
        dw_data = dw_data.loc[intersect]
        ex_data = ex_data.loc[intersect]
```

```
        diff = ex_data.compare(dw_data)
        errors += diff.notna().sum().sum() / 2
        fields += ex_data.shape[0] * ex_data.shape[1]

    #print("  Grades: %.2f%% accuracy."%(100*(fields-errors)/fields))
    return errors, fields

def evaluate_student_activity(data, degree, year):

    errors = 0
    fields = 0
    for semester in range(1, 7):
        sheet = data[str(semester) + "ºS"]

        query_filename = "../../sql/data_warehouse/dql/evaluation/
            query_student_generation_semester_activity.sql"
        query_file     = open(query_filename,"r")
        query          = query_file.read()
        query_file.close()

        dw_data = pd.read_sql(query%(degree, year, semester), con=con)
        dw_data["ALUNO"] = dw_data["ALUNO"].astype(int)
        dw_data = dw_data.set_index("ALUNO")
        dw_data = dw_data.replace({None: np.nan, "-": np.nan})
        dw_data = dw_data.rename(columns={"MÃ©dia AP": "Média AP"})
        dw_data = dw_data.dropna(thresh=2).reset_index(drop=True)
        dw_data = dw_data.round(2)

        first_col = sheet.columns[sheet.isin(['FALTA']).any()][0]
        last_col  = sheet.columns[sheet.isin(['Média AP']).any()][0] + 1

        last_row = sheet.index[sheet.iloc[:, 0]=="LAST"][0]

        ex_data = sheet.iloc[3:last_row, first_col:last_col]
        ex_data.columns = list(sheet.iloc[2, first_col:last_col])
        ex_data["ALUNO"] = sheet.iloc[3:last_row, 0].astype(int)
        ex_data = ex_data.set_index("ALUNO")
        ex_data = ex_data.replace({None: np.nan, "-": np.nan})
```

```
        ex_data = ex_data.dropna(thresh=1).reset_index(drop=True)
        if not ex_data["FALTA"].isna().all():
            ex_data["FALTA"] = ex_data["FALTA"].str.replace("-II", "2")
            ex_data["FALTA"] = ex_data["FALTA"].str.replace("-I", "1")
            ex_data["FALTA"] = ex_data["FALTA"].str.replace("-1", "1")
            ex_data["FALTA"] = ex_data["FALTA"].str.replace("-2", "2")
        ex_data[["ECTSAP", "ECTSIN"]] = ex_data[["ECTSAP", "ECTSIN"]].
            astype(float)
        ex_data = ex_data.round(2)

        if ex_data.shape[0] != dw_data.shape[0]:
            intersect  = dw_data.index.intersection(ex_data.index)
            difference = dw_data.index.difference(ex_data.index)
            errors += len(difference)*dw_data.shape[1]
            difference = ex_data.index.difference(dw_data.index)
            errors += len(difference)*ex_data.shape[1]
            dw_data = dw_data.loc[intersect]
            ex_data = ex_data.loc[intersect]

        diff = ex_data.compare(dw_data)
        errors += diff.notna().sum().sum() / 2
        fields += ex_data.shape[0] * ex_data.shape[1]

    #print("  Activity: %.2f%% accuracy."%(100*(fields-errors)/fields))
    return errors, fields

def evaluate_student_withdraws_comebacks(data, degree, year):
    sheet = data["Pivot 6Sem"]

    query_filename = "../../sql/data_warehouse/dql/evaluation/
        query_student_generation_withdraws_comebacks.sql"
    query_file     = open(query_filename,"r")
    query          = query_file.read()
    query_file.close()

    dw_data = pd.read_sql(query%(degree, year), con=con)
    dw_data = dw_data.rename(columns={"Novas DesistÃªncias": "Novas Desistê
        ncias"})
    dw_data = dw_data.astype(int)
```

```
        dw_data["Semestre"] = dw_data["Semestre"].apply(lambda x: str(x) + "°␣
            sem")

        ex_data = sheet.iloc[21:27, 1:5].reset_index(drop=True)
        ex_data.columns = list(sheet.iloc[19, 1:4]) + ["Semestre"]
        ex_data = ex_data.replace({"-": 0})
        ex_data = ex_data.round(2)
        ex_data = ex_data.loc[(ex_data!=0).sum(axis=1)>1]
        ex_data = ex_data.loc[ex_data["Ativos"]!=0]

        diff = ex_data.compare(dw_data)
        errors = diff.notna().sum().sum() / 2
        fields = ex_data.shape[0] * ex_data.shape[1]

        #print("  Withdraws/Comebacks: %.2f%% accuracy."%(100*(fields-errors)/
            fields))
        return errors, fields
    pd.set_option('display.max_rows', 500)
    pd.set_option('display.max_columns', 500)
    pd.set_option('display.width', 1000)

    config_filename = "dbconfig.p"
    config_file     = open(config_filename,"rb")
    config          = pickle.load(config_file)
    config_file.close()

    con = mysql.connector.connect(
        host     = config["host"],
        port     = config["port"],
        user     = config["user"],
        passwd   = config["password"],
        database = config["database"]
    )

    errors = np.array([0, 0, 0])
    fields = np.array([0, 0, 0])
    for degree in ["LEIC-T"]:
        for year in range(2007, 2020):
```

```
        data = pd.read_excel("../../graphs/generational_analysis_updated_/%
            s_AdmitidosSET%d.xlsx"%(degree.replace("-", ""), year),
            sheet_name=["1°S","2°S","3°S","4°S","5°S","6°S","Pivot 6Sem"],
            header=None)

        print("%s %d"%(degree, year))
        grades_errors, grades_fields        = evaluate_student_grades(data,
            degree,year)
        activity_errors, activity_fields    = evaluate_student_activity(data
            ,degree,year)
        with_come_errors, with_come_fields =
            evaluate_student_withdraws_comebacks(data,degree,year)

        errors[0] += grades_errors
        errors[1] += activity_errors + with_come_errors
        errors[2] += grades_errors + activity_errors + with_come_errors

        fields[0] += grades_fields
        fields[1] += activity_fields + with_come_fields
        fields[2] += grades_fields + activity_fields + with_come_fields

print(100*(fields-errors)/fields)
```

Listing B.1: Python script for data integrity validation

Appendix C

Usability Tests

This appendix contains all the materials used when conducting the usability test sessions. During the usability tests, the users were given a session guide (C.1), a form with usability tasks (C.2) and a form with a usability questionnaire (C.3).

C.1 Session Guide

The users were given a session guide, right before the usability tests, to help them understand the purpose and goals of the session. The session guide is presented in Listing C.1.

The goal of the Decision Support System for IST Degree Coordination (SAD–CCIST) is to guide the decision making of the IST Degree Coordinators, by providing them with useful insights about the performance of the degree, through interactive dashboards.

SAD–CCIST receives sets of data, related to the various areas that compose the degree, which detali the courses, curricular plans, admissions and grades. The data received as input is processed and transformed, so that they can be stored in a data warehouse, a database whose structure is suited for quickly obtaining the data that enhances the decision making process. This data warehouse is designed according to the Degree Coordinators' most relevant questions.

To answer the Degree Coordinators' answers, three dashboards were created. Each dashboard displays informations about three different areas that detail the performance of the degree: i. – the course performance throughout the years, ii. – the semesterly activity of a student generation and iii. – the overall performance of a student generation from admission until graduation.

```
This session will evaluate the dashboards in terms of usability. For this
    purpose, we ask you to perform a set of tasks for each dashboard. The
    tasks consist of real use cases, in which you are encouraged to interact
     with the dashboards' visual components, with the purpose of obtaining
    certain measures or informations. After each set of tasks, you will be
    asked to assess their difficulty. The tasks are available through the
    following link: https://forms.gle/5im3ddJBYPK1ZMEe9

After all tasks are concluded, you will be able to evaluate the overall
    experience of using the SAD-CCIST system. To do so, we ask you to fill
    in a usability questionnaire. The questionnaire is composed of 10 items,
     with response options that range from 1 to 5. At the end of the
    questionnaire, you may leave your suggestions and feedback. The
    usability questionnaire is available through the following link: https
    ://forms.gle/imnwUfhUNAHUoFzQ8

Thank you for your help!
```

Listing C.1: Usability session guide

C.2 Usability Tasks Form

When conducting the usability tests, the users were presented with a set of tasks, available through a form. Figures C.1, C.2 and C.3 present each page of the usability tasks form.

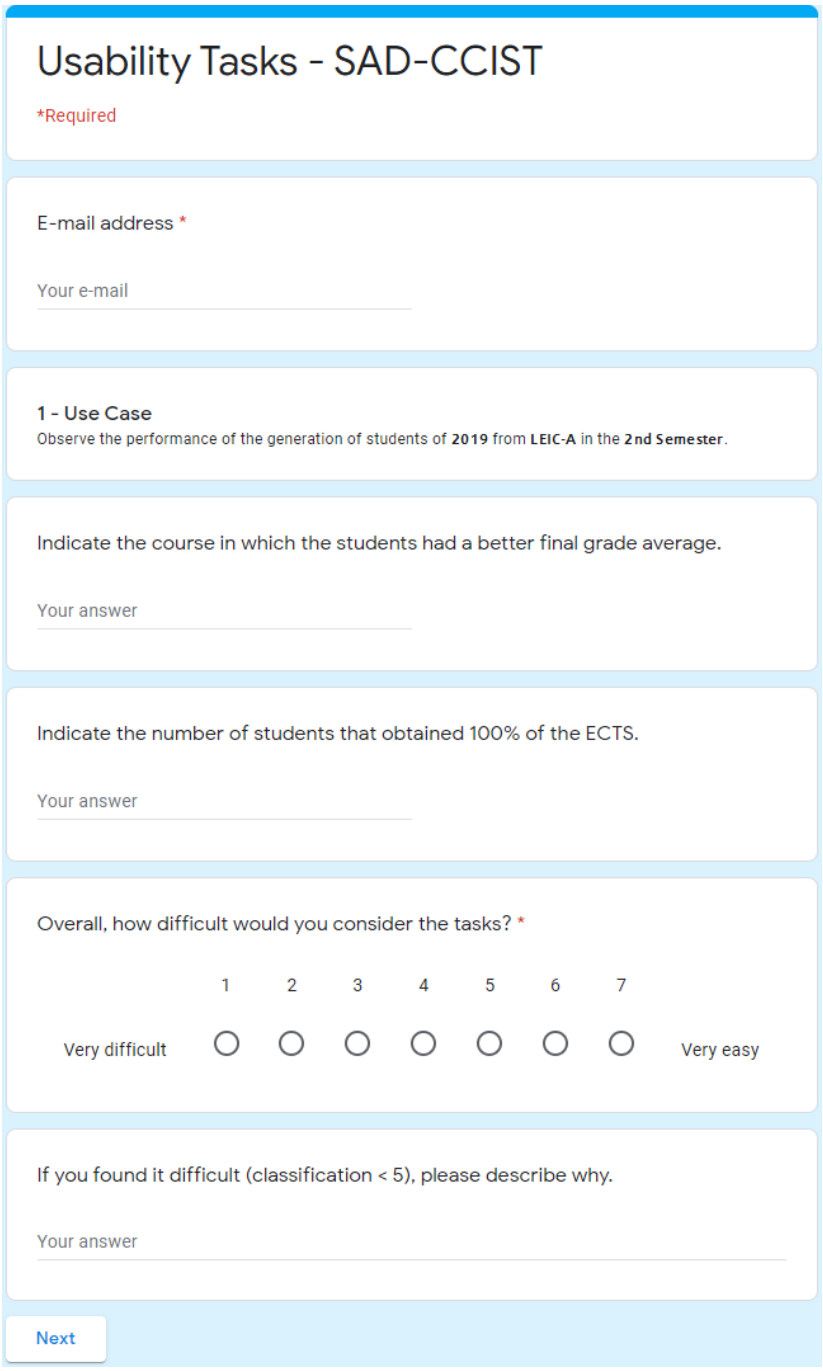

Figure C.1: Page 1 of the usability tasks form

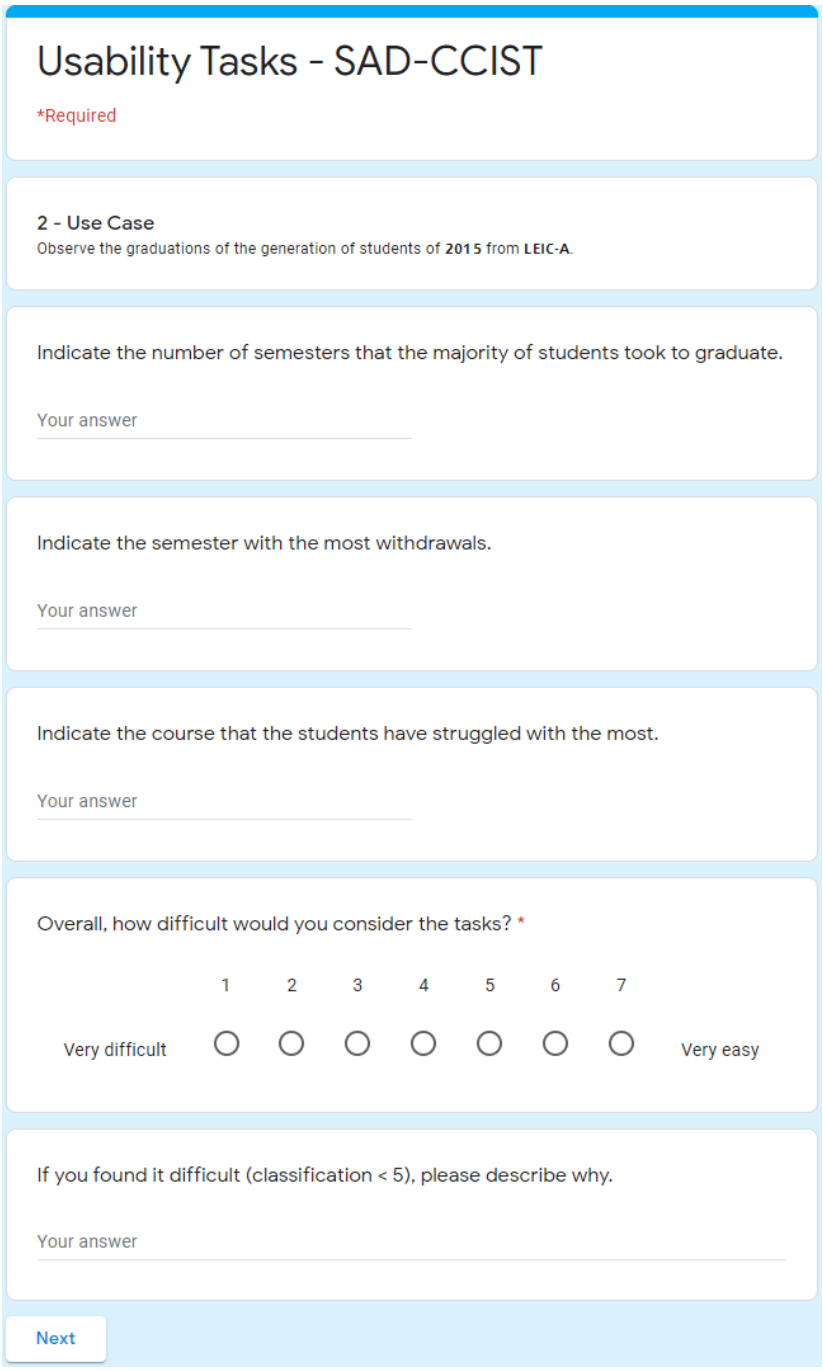

Figure C.2: Page 2 of the usability tasks form

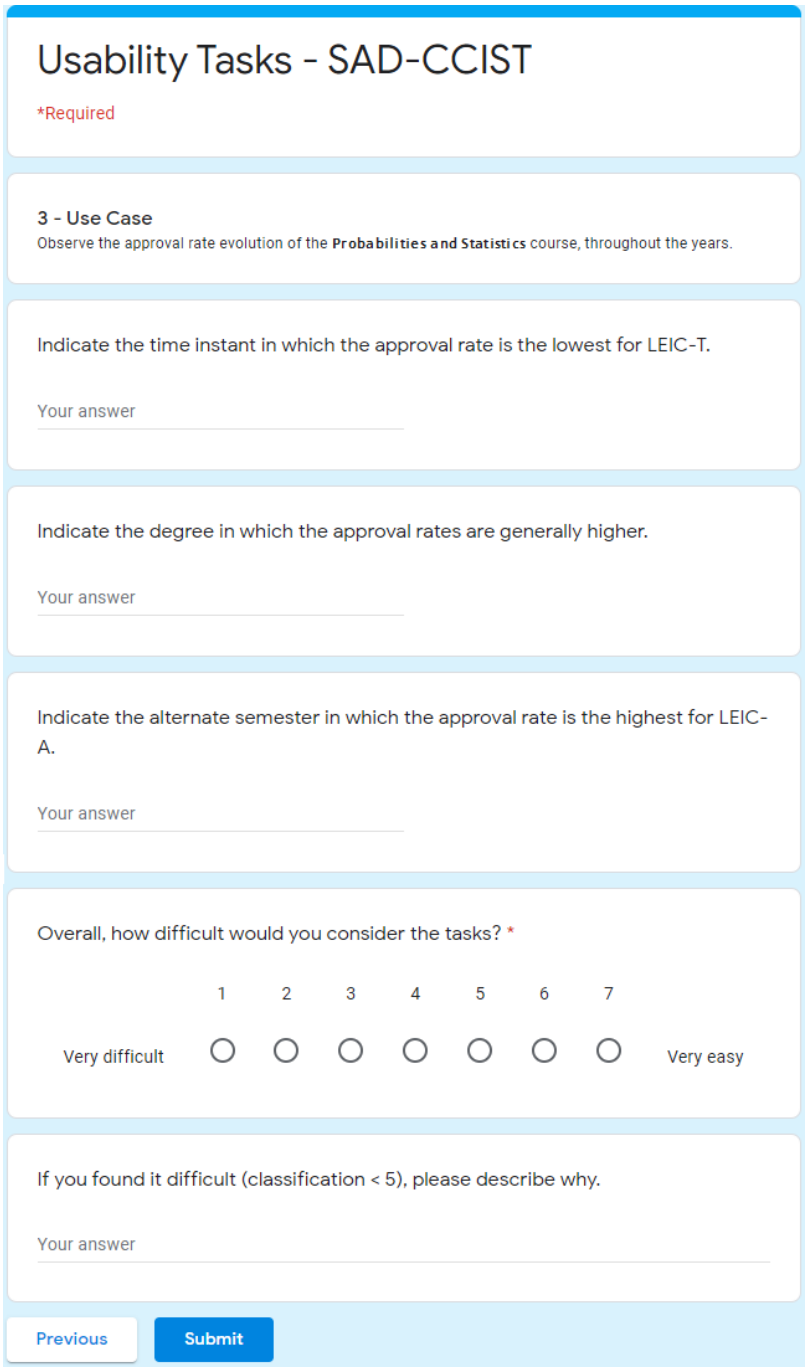

Figure C.3: Page 3 of the usability tasks form

C.3 Usability Questionnaire Form

After completing the usability tasks, the users were asked to fill in a usability questionnaire, to assess their overall experience. Figures C.4 and C.5 present each page of the usability questionnaire form.

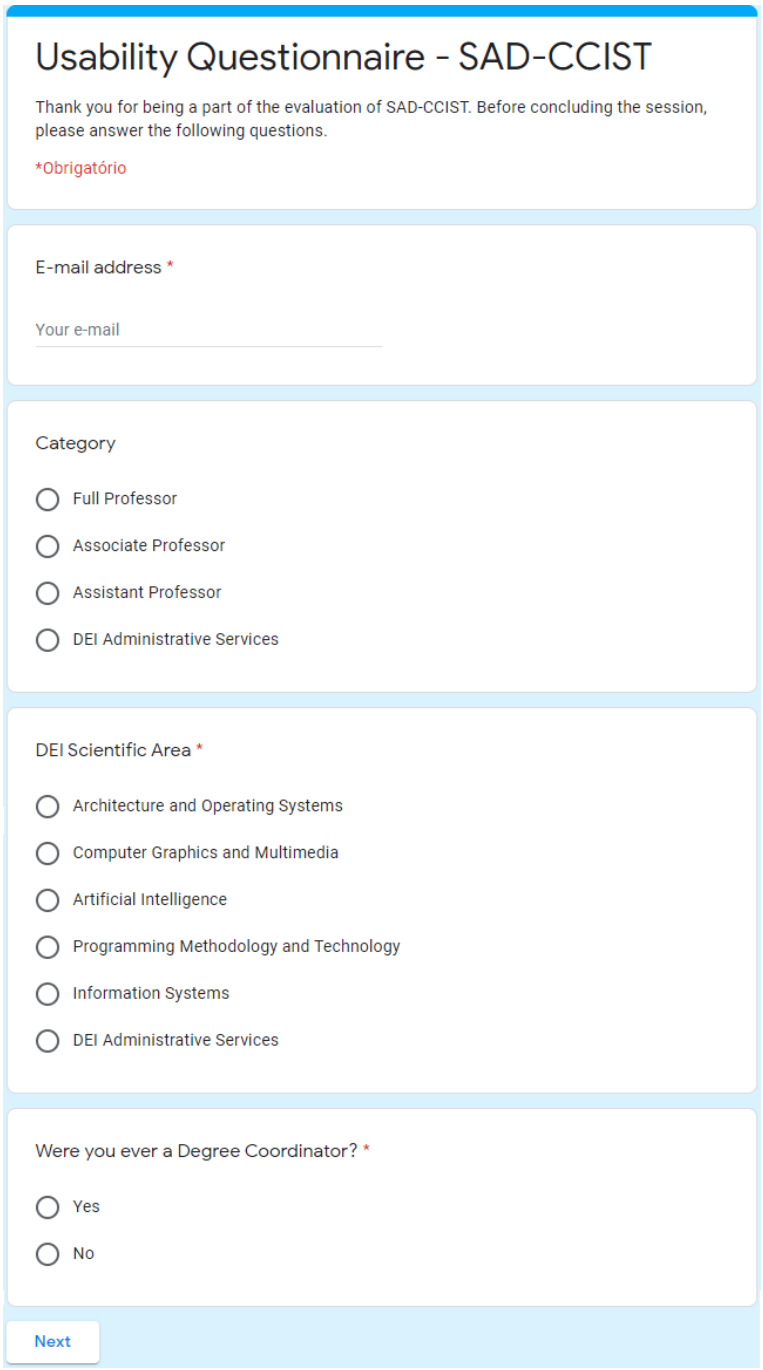

Figure C.4: Page 1 of the usability questionnaire form

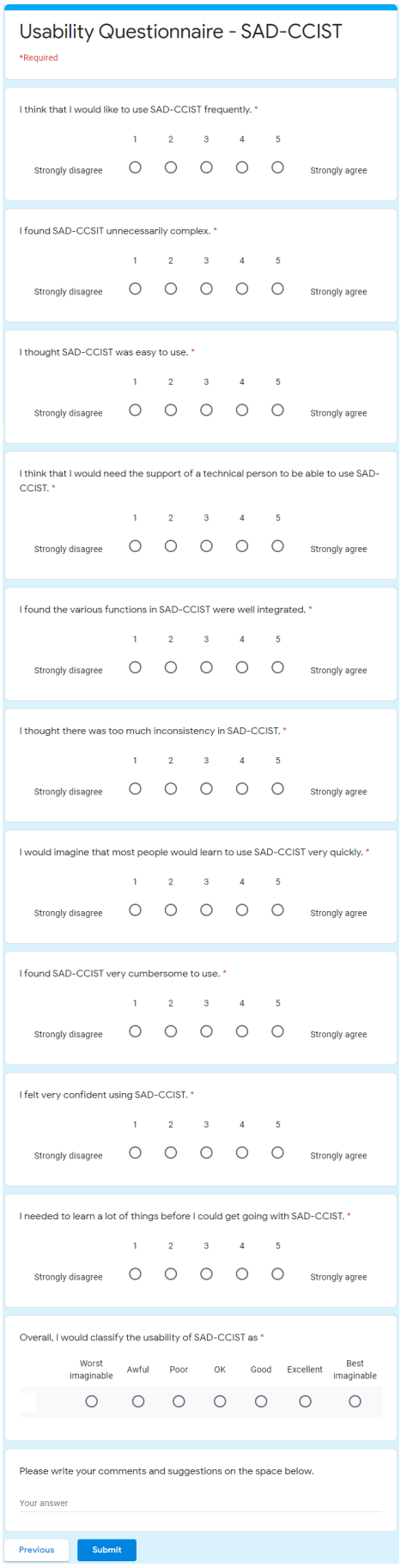

Figure C.5: Page 2 of the usability questionnaire form

Appendix D

Software Tools Installation Guide

This appendix describes the installation of the software tools used in this project. We cover the installation of the Pentaho software tools (D.1) and MySQL (D.2).

Before installing the tools, we must create a working directory and download all assets of the project.

1. Create our project's working directory:

```
mkdir /sad-ccist
cd /sad-ccist
```

2. Update the repositories:

```
sudo apt-get update
```

3. Install Git:

```
sudo apt-get install git
```

4. Verify the version of Git:

```
git --version
git version 2.30.0
```

5. Clone the repository with all the assets:

```
git clone https://git.rnl.tecnico.ulisboa.pt/ist194127/sad-ccist
```

6. Create an *input* directory, where the Excel files should be stored:

```
mkdir input
```

   Note: the repository does not include the input Excel files, because of limitations on file uploads. If you are trying to setup this environment on a remote server, use an FTP software tool (e.g., WinSCP, FileZilla) to transfer the files to the *input* directory.

D.1 Pentaho Software Tools

The Pentaho software tools require the installation of Java. A 64-bit version of Java 8 is recommended, so we used OpenJDK 8. The following steps detail how to install it:

1. Update the repositories:

```
sudo apt-get update
```

2. Install OpenJDK:

```
sudo apt-get install openjdk-8-jdk
```

3. Verify the version of the JDK:

```
java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b09)
OpenJDK 64-Bit Server VM (build 25.242-b09, mixed mode)
```

   If the correct version of Java is not being used, use the alternatives command to switch it:

```
sudo update-alternatives --set java /usr/lib/jvm/jdk1.8.0_version/
    bin/java
```

4. Setup the JAVA_HOME and PENTAHO_JAVA_HOME variables:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PENTAHO_JAVA_HOME=$JAVA_HOME
```

Before the installation of the Pentaho software tools, the */pentaho* directory should be created:

```
mkdir /pentaho
```

D.1.1 Pentaho Data Integration

PDI will be installed in the */pentaho/data-integration* directory. The following steps detail how to install it:

1. Access the *pentaho* directory:

```
cd /pentaho
```

2. Download the zip file with the PDI assets:

```
wget https://sourceforge.net/projects/pentaho/files/Pentaho%209.1/
    client-tools/pdi-ce-9.1.0.0-324.zip/download
```

3. Unzip the file to the current directory:

```
unzip ./pdi-ce-9.1.0.0-324.zip
```

D.1.2 Pentaho Business Analytics

Pentaho BA will be installed in the */pentaho/pentaho-server* directory. The following steps detail how to install it:

1. Access the *pentaho* directory:

```
cd /pentaho
```

2. Download the zip file with the Pentaho BA assets:

```
wget https://sourceforge.net/projects/pentaho/files/Pentaho%209.1/
    server/pentaho-server-ce-9.1.0.0-324.zip/download
```

3. Unzip the file to the current directory:

```
unzip ./pentaho-server-ce-9.1.0.0-324.zip
```

D.2 MySQL

The installation of MySQL will allow the creation of the DSA and DW. To setup a MySQL connection from the Pentaho software tools, we will need to install a MySQL connector. The following steps detail how to install it:

1. Update the repositories:

```
sudo apt-get update
```

2. Install MySQL:

```
sudo apt-get install mysql-server
```

3. Run the security script:

```
sudo mysql_secure_installation
```

   You will be asked to set a password for the *root* user. Set it to *rootroot*.

4. Download the zip file with the MySQL connector:

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-
    connector-java-5.1.49.zip
```

5. Unzip the file:

```
unzip ./mysql-connector-java-5.1.49.zip
```

6. Copy the connector file to the *lib* directories of PDI and Pentaho BA:

```
cp ./mysql-connector-java-5.1.49/mysql-connector-java-5.1.49.jar /
    pentaho/data-integration/lib
cp ./mysql-connector-java-5.1.49/mysql-connector-java-5.1.49.jar /
    pentaho/pentaho-server/tomcat/lib
```

Appendix E

User Guide

This appendix describes how to use the various components of the system.

E.1 Populate the DSA and DW

To populate the DSA and DW, we must run the ETL processes. The execution of these processes may take a while, typically from 15 to 20 minutes.

Before running the processes, please make sure that all Excel files were placed in the *input* directory.

1. Access the */pentaho/data-integration* directory:

```
cd /pentaho/data-integration
```

2. Run the extract process, to build the DSA:

```
kitchen.sh -file=/sad-ccist/etl/data_staging_area/load.kjb -level=
    Minimal
```

3. Run the transform-load process, to build the DSA:

```
kitchen.sh -file=/sad-ccist/etl/data_warehouse/full_load.kjb -level
    =Minimal
```

E.2 Dashboard Usage

To use the dashboards, the Pentaho BA server application must be running:

1. Access the */pentaho/pentaho-server* directory:

```
cd /pentaho/pentaho-server
```

2. Start the server application:

```
start-pentaho.sh
```

Once the server application is running, we can access it from `http://localhost:8080/pentaho`. Accessing this url, will lead us to the Pentaho User Console (PUC), which will ask for a login, as seen in Figure E.1. The default username and password are *admin* and *password*.

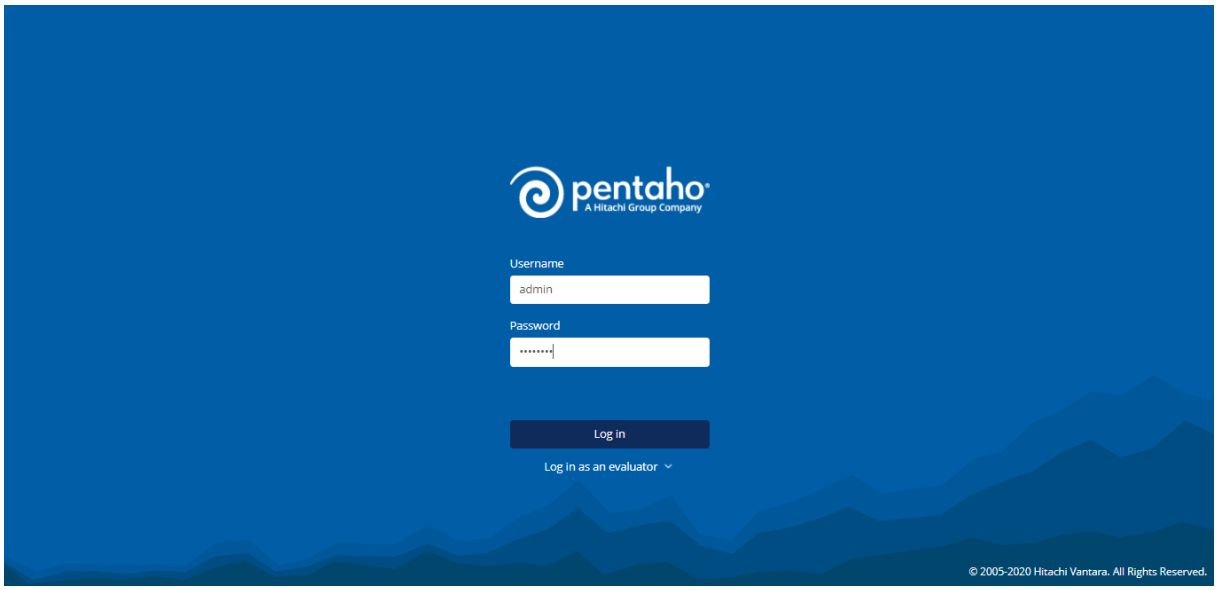

Figure E.1: Pentaho User Console - Login

Once the authentication is successful, we are redirected to the home page, presented in Figure E.2. If this is the first time accessing the PUC, we must setup a connection to our DW and we must upload the dashboard files to the server.

Figure E.2: Pentaho User Console - Home Page

First we will setup the DW connection:

1. Press the *Manage Data Sources* button. A new window will be opened.

2. In the new window, press the *New Data Source* button. Another window will be opened.

3. In the new window, set the *Source Type* option to *Database Table(s)*.

4. Click the *+ icon* to add a new *Connection*. Yet another window will be opened.

5. Configure your connection as suggested by Figure E.3:

   5.1. Set the *Connection Name* to *Degree Coordination*.

   5.2. Set the *Database Type* to *MySQL*.

   5.3. Set the *Host Name* to *localhost*.

   5.4. Set the *Database Name* to *degree_coordination_dw*.

   5.5. Set the *Port Number* to *3306*.

   5.6. Set the *User Name* to *root*.

   5.7. Set the *Password* to *rootroot*.

6. Press the *Test* button and confirm your connection is valid.

   A window will appear with the message *Connection to database [degree_coordination_dw] succeeded*.

7. Close all windows to return to the home page.

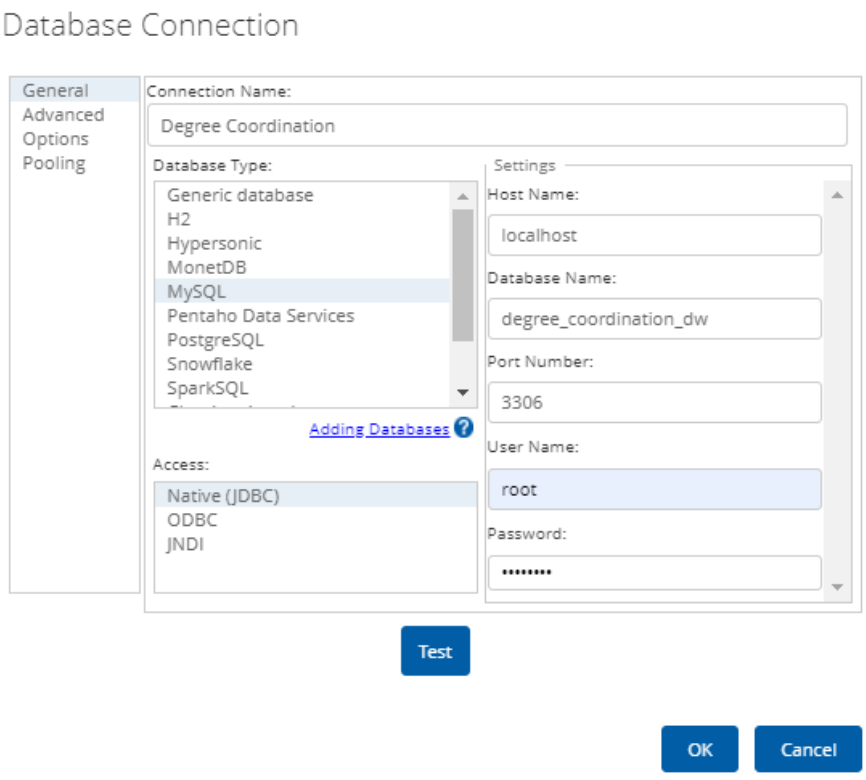

Figure E.3: DW connection configuration

Now that the DW connection is created, the dashboards will be able to get the data they need to generate their visualizations. We will now upload the dashboard files to the server:

1. Press the *Browse Files* button. A new page will be opened.
2. Press the *Public* folder. The *Folder Actions* will appear on the right side of the page.
3. Select the *New Folder...* option.
4. Set the *Name* option to *Degree Coordination*.
5. Press the *Public* folder. The *Folder Actions* will appear on the right side of the page.
6. Select the *Upload...* option. A new window will be opened.
7. In the new window, press the *Browse...* button.
8. Select the dashboard files from the dashboard folder downloaded from the SAD-CCIST git repository. These are files with the extensions *.cda*, *.cdfde* and *.wcdf*.

   Note: only one file can be submitted at a time.

After these configuration steps, the dashboards are ready to be used. We will be able to open them by browsing files inside PUC and selecting files with the *.cda* extension. We can add these files to the favourite list, to be able to access them directly from the home page, for simplicity purposes.