Data Extraction and Integration for the Creation of a Database Describing Portuguese Unions and other Social Partners

João Manuel Ramires Machado

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Doutor Bruno Emanuel da Graça Martins
Prof. Doutor José Luís Brinquete Borbinha

Examination Committee

Chairperson: Prof. Doutor José Carlos Martins Delgado
Supervisor: Prof. Doutor Bruno Emanuel da Graça Martins
Member of the Committee: Prof. Doutor Alberto Manuel Rodrigues da Silva

December 2020

Acknowledgements

First, I would like to thank Professor Bruno Emanuel da Graça Martins and Professor José Borbinha for their guidance during this last year, through which they contributed very significantly to the work that we have developed together, with their knowledge and motivation.

Second, I would like to thank the DGERT institute (Direção-Geral do Emprego e das Relações de Trabalho) for making the information from their data sources available and more specifically to the Senior Informatics Technician Joaquim Félix, for his contribution in the development of the Web Service, in order to remotely obtain the information.

I would also like to thank the ICS institute (Instituto de Ciências Sociais da Universidade de Lisboa) for their contribution in the development of this work.

I have to thank my family, in particular my parents, for their constant support and for giving me the opportunity to learn in such a distinguished institute as Instituto Superior Tećnico.

Finally, I have to thank all my friends and colleagues for the constant support during the hard, although also amazing, time spent at Instituto Superior Tećnico.

João Manuel Ramires Machado

For my parents,

Resumo

O desenvolvimento de uma base de dados completa e consistente levanta muitos desafios, relacionados à coleta e integração de dados de várias fontes. A DGERT (Direção-Geral do Emprego e das Relações de Trabalho) é responsável pela publicação de um boletim oficial semanal com informações detalhadas sobre as organizações de trabalho e sobre as suas atividades. Este instituto também possui informações sobre organizações sindicais e patronais em diferentes fontes de dados.

No âmbito deste trabalho, foi desenvolvida uma base de dados sobre organizações sindicais e patronais portuguesas, juntamente com uma interface web de apoio à exploração dos dados. A base de dados desenvolvida contém informações obtidas a partir das diferentes fontes de dados disponibilizadas pela DGERT. A base de dados foi desenvolvida através de procedimentos de extração de informação e integração de dados, visando o apoio a diferentes estudos nas ciências sociais. Para o desenvolvimento da interface web, foi utilizada a framework Flask.

O desenvolvimento da base de dados envolveu múltiplas pequenas tarefas. Futuramente, é possível aprimorar essas tarefas, de forma a melhorar a quantidade de dados e a qualidade da base de dados.

Abstract

The development of a complete and consistent database raises many challenges, related to data collection and integration from multiple sources. The DGERT (Direção-Geral do Emprego e das Relações de Trabalho) institute is responsible for publishing a weekly official bulletim with detailed information on work-related organizations and on their activities. This institute also has information on trade-unions and employer organizations in different data sources.

In the context of this work, was developed a database on Portuguese labor unions and employer organizations, together with a web-based interface supporting the exploration of the data. The developed database contains information obtained from the different data sources available from DGERT institute. The database was developed through information extraction and data integration procedures, envisioning the support to different studies in the social sciences. For the development of the web-based interface, the Flask web framework was used.

The database development involved multiple small tasks. In the future it is possible to enhance those tasks, in order to improve the amount of data and the quality of the database.

Palavras Chave
Keywords

Palavras Chave

Extração de dados

Limpeza de dados

Integração de dados

Serviço web

Aplicação web

Keywords

Data extraction

Data cleaning

Data integration

Web service

Web application

Contents

List of Figures

List of Tables

1 Introduction

Empirical studies in the social sciences like Vandaele (2000) and Jensen (2020) frequently include or desire to include objective measures of unionization, including private and public sector labor union membership, density estimates, coverage, and representativeness. However, the development of a complete and consistent database focused on labor union membership raises many challenges, related to data collection and integration from multiple sources. In Portugal, the *Boletim do Trabalho e Emprego* (BTE) is a weekly official bulletim that publishes detailed information on work-related organizations like trade unions and employer organizations and on their activities. However, information extraction from the textual contents of the BTE is technically challenging, requiring the development of tailored approaches. Moreover, in support their day-to-day operations, the *Direção-Geral do Emprego e das Relações de Trabalho* (DGERT) institute that publishes the BTE also has information on trade-unions and employer organizations on relational databases and excel spreadsheets. Still, extracting the relevant information from these databases, and integrating it with additional contents extracted from the BTE, again requires the development of custom approaches. The main goals of this work were to create a database with all the information regarding portuguese unions and social partners and develop a web application where that database would be used and manipulated. In order to create a database with all the information obtained from the data sources, the data sources were analysed and data cleaning and integration techniques were performed. Regarding the BTEs, they were analysed and a part of the important information gathered was integrated in the database. In order to DGERT employees manipulate and use the database, a web application was developed using the Flask framework. This work was performed using information provided by DGERT within the scope of the REP (*Representatividade dos Parceiros Sociais e Impacto da Governança Económica*) project[1]that intends to contribute to a more informed decision-making process, more transparent and trustful organizations, and fairer labour relations.

[1] `https://rep.ics.ulisboa.pt/`

1.1 Motivation

In order to perform studies in the social sciences it's necessary that all the important information is integrated in a complete and consistent database focused on labor union membership. By having this, it is much easier to perform all kinds of studies. This was the main motivation for the development of this work.

Since DGERT doesn't have all the information in the same data source, there's the need to integrate the information from the different data sources into a common database. The extraction of information from the bulletins is a very difficult challenge, that needs to be performed using advanced data extraction techniques. Since these bulletins are PDF documents, there's the need to extract their textual contents in order to integrate them into the database with the rest of the information from DGERT databases and excel spreadsheets. Also, the information in the DGERT databases and excel spreadsheets is not available outside their local work network. It would be interesting to develop a method to get that information and share it with a number of selected members, only for research purposes. The different data sources have important information regarding labor union membership and it would be interesting to integrate the information into a common database. To ease the access to the data, it would be interesting to develop a web application in order to search and manipulate the database.

This work is part of the REP project. The REP project aims to understand the conditions and implications of the representativeness of social partners in economic governance. In democratic societies, they play a fundamental regulatory role of the labour market, representativeness providing them legitimacy to be consulted and to negotiate. However, representativeness is enveloped in a puzzle and scarce research tends to restrict representativeness to a membership rate and to focus on one side of labour relations. Assuming that representativeness is a multifaceted concept, the REP project would combine membership representativeness with composition and opinion congruence between representatives and represented. Also, it would focus on both trade unions and employers' associations, having into account their different collective interests. In summary, the REP project intends to contribute to a more informed decision-making process, more transparent and trustful organizations, and fairer labour relations. The REP project team consists of members of the INESC-ID (Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento) institute, the ICS (Instituto de Ciências Socias) institute and the DGERT (Direção-Geral do Emprego e das Relações de Trabalho) institute. The REP project is financed by the FCT (Fundação para a Ciência e a Tecnologia) institute.

1.2 Thesis Proposal

Regarding the development of this work, it was proposed the following approach:

- Integrate data from CSV files and the database provided by DGERT and do all the cleaning of the data, constructing in the end a relational database;
- Extract information from the BTE bulletins (PDF files) that contain detailed data related to the trade unions that appear in the DGERT database, and relate them with the relational database;
- Create a basic web application to manipulate data, to be used by researchers and employees of DGERT.

The first two steps involve exploring data extraction, cleaning and integration techniques.

The first involves creating a relational database and integrating the information from DGERT into that database.

The second involves parsing the PDF files content to corresponding text files. Data regarding boards lists needs to be collected and integrated into the database.

Regarding the third step, it involves creating a web application using Flask framework in order to manipulate data from the database. This web application would contain in its dashboard page statistics about the data represented by visualizations and a organizations search engine. The web application would also have a administration panel where the users can manipulate the database.

The goals of this work are:

- Study state-of-the-art data management and data integration techniques;
- Development of a relational model for the Portuguese labor unions database;
- Development of scripts for information extraction and integration;
- Development of a web-based interface for the Portuguese labor unions database.

1.3 Contributions

In brief, the main contributions of this thesis are as follows:

- a integrated database that aggregates all the information that is now available to investigators for social sciences' studies;
- a web application to explore and manipulate data from the database, improving the access to the information in a practical and intuitive way.

1.4 Structure of the Document

The rest of this document is organized as follows. Chapter 2 presents fundamental concepts as well as important related work in the areas of Information Extraction from Text and Data Integration supporting the Computational Sciences. Then, Chapter 3 details the followed solution design and describes the different steps involved in the project development, such as Extraction and Cleaning of Data, Integration of Data into a Common Database and Development of a Web Application for Data Exploration. Chapter 4 presents the solution demonstration, such as Statistical Characterization of the Dataset and Results. Finally, Chapter 5 concludes this document by summarizing the challenges found while doing this work, the contributions of this work and ideas for future work.

2 Concepts and Related Work

This chapter presents fundamental concepts and related work on information extraction and integration. First, the fundamental concepts are provided in Section 2.1. This subsection presents the most relevant concepts for the project developed. Then, important related work in the area is presented in Section 2.2, covering two main categories: Information Extraction from Text (Section 2.2.1) and Data Integration Supporting the Computational Social Sciences (Section 2.2.2).

2.1 Fundamental Concepts

This section starts to introduce and discuss the concepts of Data Profiling, Data Cleaning, Data Extraction and Data Integration. Data warehouses in Relational Data Management are also discussed. Finally, some Flask design principles in Web Data management are referred.

2.1.1 Data Profiling

Before initiating the data cleaning process, it is important to evaluate quality of data in a database and then assess its success. The task of evaluating data quality is named data profiling and typically involves gathering several aggregate data statistics which constitute the data profile. It's used to ensure that the values match up with the expectations (Ganti and Sarma, 2013). Normally, uses SQL queries to perform the computations. Data profiling can uncover data quality issues in data sources, and what needs to be corrected in ETL (Extract-Transform-Load).

Data profiling can identify data quality issues, which you can handle in scripts and data integration tools copying data from source to target. It also highlights data which suffers from serious or numerous quality issues, and the source of the issues.

Data profiling involves collecting statistics like *min*, *max*, *count* and *sum* and also collecting data types, length and recurring patterns. It also involves tagging data with keywords, descriptions and categories. It's also useful for performing data quality assessment, risk of performing joins on the data. Data profiling also involves discovering metadata, assessing its accuracy and

identifying distributions, key candidates and functional dependencies. By using profiling we can know the constraints of the data source, regarding integrity constraints, domain constraints and primary key constraints.

Regarding the steps for performing data profiling, Ralph Kimball[1], a father of data warehouse architecture, suggests a four-step process for data profiling:

1. Use data profiling at project start to discover if data is suitable for analysis and make a "go/no go" decision on the project;
2. Identify and correct data quality issues in source data, even before starting to move it into target database;
3. Identify data quality issues that can be corrected by Extract-Transform-Load (ETL), while data is moved from source to target. Data profiling can uncover if additional manual processing is needed;
4. Identify unanticipated business rules, hierarchical structures and foreign key/private key relationships, use them to fine-tune the ETL process.

Regarding data profiling, the techniques are:

- **Distinct count and percent** - identifies natural keys, distinct values in each column that can help process inserts and updates. It's very useful for tables without headers;
- **Percent of zero, blank and null values** - identifies missing or unknown data;
- **Minimum/maximum/average string length** - helps select appropriate data types and sizes in target database. Enables setting column widths just wide enough for the data, to improve performance;
- **Key integrity** - ensures keys are always present in the data, using zero/blank/null analysis. Also, helps identify orphan keys, which are problematic for ETL and future analysis;
- **Cardinality** - checks relationships like one-to-one, one-to-many, many-to-many, between related data sets. This helps BI tools perform inner or outer joins correctly;
- **Pattern and frequency distributions** - checks if data fields are formatted correctly, for example if emails are in a valid format. Extremely important for data fields used for outbound communications (emails, phone numbers, addresses).

[1] `https://panoply.io/analytics-stack-guide/data-profiling-best-practices/`

2.1.2 Data Cleaning

Data cleaning is the process of starting with raw data from one or more sources and maintaining reliable quality for your applications (Ganti and Sarma, 2013). It consists in a variety of tasks aimed at improving the quality of data.

2.1.2.1 Data Cleaning Tasks

This section refers to the main data cleaning tasks: Record Matching, Schema Matching, Deduplication and Data Standardization.

Record Matching

The goal of Record Matching is to match each record from a set of records with records in another table (Ganti and Sarma, 2013). This task needs to be accomplished when a new set of entities is imported to the target relation to make sure that the insertion does not introduce duplicate entities in the target relation. The goal of a record matching task is to identify record pairs, one in each of two input relations, which correspond to the same real-world entity. Challenges to be addressed in this task include **(i)** identification of criteria under which two records represent the same real-world entity, and **(ii)** efficient computation strategies to determine such pairs over large input relations. Figures 2.1 and 2.2 illustrate examples of Record Matching.

ID	Name	Street	City	Phone
r1	Sweetlegal Investments Inc	202 North	Redmond	425-444-5555
r2	ABC Groceries Corp	Amphitheatre Pkwy	Mountain View	4081112222
r3	Cable television services	One Oxford Dr	Cambridge	617-123-4567
s1	Sweet legal Invesments Incorporated	202 N	Redmond	
s2	ABC Groceries Corpn.	Amphitheetre Parkway	Mountain View	
s3	Cable Services	One Oxford Dr	Cambridge	6171234567

Figure 2.1: Two sets of customer records (Ganti and Sarma, 2013).

ID	Title
r1	Canon EOS 20D Digital SLR Body Kit (Req. Lens) USA
r2	Nikon D90 SLR
s1	Canon EOS 20d Digital Camera Body USA - Lens sold separately
s2	Nikon D90 SLR Camera

Figure 2.2: Product catalog with a new set of products (Ganti and Sarma, 2013).

Schema Matching

Schema Matching is the task of aligning attributes from different schemas (Ganti and Sarma, 2013). As an example, suppose the information from a warehouse was organized as a relation *R(Name, CityAddress, Country, Phone,...)*, which stores most of the address (except Country) in a single attribute in textual format. Now suppose there is another relation with data represented in the format *S(Company, Apt, Street, City, Zip, Nation, PhoneNumber)*, which breaks the address into individual components. To populate tuples in S into R, a process to convert each S tuple into the format of R is necessary.

Schema Matching provides **(1)** *attribute correspondences* describing which attributes in S correspond to attributes in R (Country corresponds to Nation, PhoneNumber corresponds to Phone, Company corresponds to Name, and City Address corresponds to the remaining four attributes in S) and **(2)** *transformation functions* that give concrete functions to obtain attribute values in R from attribute values in S, e.g., a transformation process gives a mechanism to concatenate all attributes to form City Address (or extract attributes like Zip Code when converting R to S).

Deduplication

The goal of deduplication is to group records in a table such that each group of records represent the same entity (Ganti and Sarma, 2013). This task is often needed when a database is being populated or cleaned for the first time. Deduplication and record matching are different because deduplication involves an additional grouping of "matching" records, such that the groups collectively partition the input relation.

For example, consider a enterprise data warehousing scenario. When the data warehouse is first populated from various feeds, it is possible that the same customer could be represented by multiple records in one feed, and even more records across feeds. So, it is important for all records representing the same customer to be reconciled. In figure 2.3, records {g11, g12, g13} are "duplicate" records of each other while {g21, g31} is another set of duplicate records.

ID	Name	Street	City	Phone
g11	Sweetlegal Investments Inc	202 North	Redmond	425-444-5555
g12	Sweet legal Invesments Incorporated	202 N	Redmond	
g13	Sweetlegal Inc	202 N	Redmond	
g21	ABC Groceries Corp	Amphitheatre Pkwy	Mountain View	4081112222
g31	Cable television services	One Oxford Dr	Cambridge	617-123-4567

Figure 2.3: Shows records with {g11, g12, g13} being one group of duplications, and {g21, g31} another set of duplicate records (Ganti and Sarma, 2013).

In figure 2.4, {g21, g22, g23, g24} all represent the same entity, a Nikon DSLR camera.

ID	Title
g1	Canon EOS 20D Digital SLR Body Kit (Req. Lens) USA
g21	Nikon D90 SLR
g22	Nikon D90 SLR Camera
g23	Nikon D90
g24	D90 SLR

Figure 2.4: Shows records with {g21, g22, g23, g24} all representing the same entity, a Nikon DSLR camera (Ganti and Sarma, 2013).

Sometimes when there is a large number of records representing the same entity, it is difficult to decide what record to use in order to represent the entity. Some criteria may be applied, like the record which contains bigger detail and information about the entity or the record that is more correctly written.

When the data source has a huge number of records, sometimes performing deduplication takes a lot of time. One possible improvement can be made by using clustering. By using clustering, records can be grouped based on a specific characteristic and deduplication can be performed in those clusters. This way, performance is improved and the data is also more organized.

Data Standardization

Data Standardization is the task of ensuring that all attribute values are "standardized" as per the same conventions (Ganti and Sarma, 2013). It's a critical operation required before record matching or deduplication. Standardizing the format and correcting attribute values leads to significantly better accuracy in other data cleaning tasks such as record matching or deduplication.

2.1.2.2 Technological Approaches

Regarding technological approaches, there are many that enable the development and deployment of effective solutions for data cleaning.

The first category consists of Domain-Specific Verticals like Trillium[2]that provides data cleaning functionality for specific domains (Ganti and Sarma, 2013). Since this technological approach understands the domain where the vertical is being used it can tune its solution for the

[2]www.trillium.com

given domain. The main advantage of these domain-specific solutions is that they can incorporate knowledge of the domain while developing the solution. Since the domain is known, the flow of operations to be performed are often decidable upfront. These solutions can be comprehensive and are easier to deploy. The disadvantage of developing domain-specific solutions is that they are not generic and cannot be ported to other domains. Solutions are also sensitive to sub-categories within a domain.

The second category of approaches relies on horizontal ETL Platforms such as Microsoft SQL Server Integration Services[3]and IBM Websphere Information Integration[4](Ganti and Sarma, 2013). These platforms provide a suite of operators including relational operators such as select, project and equi-join. A feature that is common in these frameworks is that applications can plug in their own custom operators. A data transformation and cleaning solution is built by composing the default and custom operators to obtain an operator tree or a graph. The downside of this approach is the limitation that most of the data cleaning logic potentially needs to be incorporated as custom components. Creating those custom components is nontrivial so this approach requires a big amount of effort from developers.

The third approach builds upon the extensible ETL platform by extending their repertoire of the default operators beyond traditional relational operators with a few core data cleaning operators such that with much less extra effort and code, a rich variety of efficient and effective data cleaning solutions can be obtained (Ganti and Sarma, 2013). The advantages of this approach include those of retaining much of the flexibility of the generic ETL platforms while also having the heavy lifting done by the optimized but flexible data cleaning operators. This means that this solutions can be easily developed. However, still have to be developed for any given domain and scenario. This approach, the operator-based approach, is similar to query processing which derives its power from compositionality over a few basic operators and is in sharp contrast with the earlier approaches which focused on the development of monolithic data cleaning solutions.

2.1.2.3 Generic Data Cleaning Operators

It is also important to talk about some critical primitive operators. This operators can be used (along with standard relational operators) to build fairly general and accurate data cleaning solutions (Ganti and Sarma, 2013). The similarity join is a very important data cleaning operator that is responsible for "joining" similar data. It's very useful in record matching and

[4]SSIS - `http://msdn.microsoft.com/sql`

[4]`https://www.ibm.com/developerworks/data/newto/db2ii-getstarted.html`

also deduplication. For example, consider a sales data warehouse. Owing to various errors in the data due to typing mistakes and differences in conventions, product names and customer names in sales records may not match exactly with master product catalog and reference customer registration records respectively. In these situations, it would be desirable to perform similarity joins. For instance, two products can be joined (respectively, customers) if the similarity between their part descriptions (respectively, customer names and addresses) is high. This operation is a fundamental operation to identify approximate duplicate entities in databases and to identify for a given record the best few approximate matches from among a reference set of records. The current approaches exploit similarity between attribute values to join data across relations. A variety of string similarity functions have been considered, such as edit distance, jaccard similarity, cosine similarity and generalized edit distance, for measuring similarities. However, no single string similarity function is know to be the overall best similarity function, and the choice usually depends on the application domain. The soundex function is used for matching person names. There's the need for a similarity join operator that employs customizable similarity functions.

Other critical operation is clustering that is useful in many data cleaning tasks. Clustering refers to the operation of taking a set of items, and putting them into smaller groups based on "similarity" (Ganti and Sarma, 2013). For example, a list of restaurants may be clustered based on similar cuisines, or based on their price, or some combination of price and cuisine. Clustering is often used in a pre-processing step of deduplication called blocking. When the set of records to be deduplicated is very large, blocking performs a crude clustering to bring similar "blocks" of records together, and a finer-grained pairwise comparison is only performed within each block. Another application of clustering is in deduplication itself. Once there are pairwise similarities between pairs of records in a block, clustering based on the pairwise similarities is used to obtain the final deduplicated set of records. In addition to the similarity measure, clustering may be guided by constraints that restrict which set of items may or may not be grouped together and by an objective function that determines the best possible clustering among all that satisfy the constraints. There are two main approaches to clustering: **(1)** *a hash-based approach* where each item is placed in a cluster based on the value it produces based on some hash function; **(2)** *a graph-based approach* that translates the clustering problem into finding structures in a graph.

The differences in schema between the source and destination databases often makes the data cleaning operation such as record matching and deduplication challenging. Due to those differences, an attribute at the source may actually correspond to a concatenation of attribute values at the destination schema. It becomes important to "parse" the attribute values from the

source into the corresponding attribute values at the destination. Consider a scenario where a customer relation is being imported to add new records to a target customer relation. Suppose the address information in the target relation is split into its constituent attributes [street address, city, state, and zip code] while in the source relation they are all concatenated into one attribute. Before the records from the source relation can be inserted in the target relation, it is essential to segment each address value in the source relation to identify the attribute values at the target (Ganti and Sarma, 2013). For example, an input address string "15633 148th Ave Bellevue WA 98004" has to be split into the following sub-components before populating the destination table:
House Number : 15633
Street Name : 148th Ave.
City : Bellevue
State : WA
Zip : 98004
The goal of a parsing task is to split an incoming string into segments each of which may be inserted as attribute values at the target. A challenge to be solved by this task is to identify sub-strings of an input string which form the attribute values of the destination schema.

Parsing Definition: Given a string text T and a relation schema S={A1,...,An}, the goal of parsing is to construct a tuple t with schema S from T subject to the constraint that t is considered a valid and accurate tuple in target relation (Ganti and Sarma, 2013).

2.1.3 Data Extraction

Data extraction can be seen as a task who consists of filling slots in a database from sub-segments of information. Data Extraction and Data Integration can be performed using different ETL (Extract-Transform-Load) tools, that have specific functionalities useful to achieve this goal. It can also be defined as a family of techniques: *segmentation* + *classification* + *association* + *clustering*. Segmentation is also known as named entity extraction where we extract from sub-segments of text all the named entities present. Classification consists in separating the entities found into names, companies and positions, e.g. (Microsoft Corporation, CEO, Bill Gates). Different entities can refer to the same thing and association is used in order to separate those who refer to the same thing. Association consists in separating the entities found into associations between the entities found. The last step is clustering where the information is grouped based

on the entities that are found (Mooney, 1999). The majority of data extraction comes from unstructured data sources and different data formats. This unstructured data can be in any form, such as tables, indexes and analytics. Sometimes, data sources like documents have a non regular structure and a close domain, making it difficult to extract information (Mooney, 1999). Within the scoped of this work, there's the need to extract information based on PDF files. There is also possible to extract information based on a joint approach between data in text and data in tables as will be described in the related work.

Data in a warehouse may come from different sources, a data warehouse requires three different methods to use the incoming data. These processes are known as Extraction, Transformation and Loading (ETL). The process of data extraction involves retrieval of data from disheveled data sources. The data extracts are then loaded into the staging area of the relational database. Here extraction logic is used and source system is queried for data using application programming interfaces. Following this process, the data is now ready to go through the transformation phase of the ETL process (Mooney, 1999).

2.1.4 Data Integration

Data integration can be defined as the set of techniques that enable a uniform access to a set of autonomous and heterogeneous data sources, controlled by different people, through a common schema (Doan et al., 2012). Some reasons to perform data integration are: create a website for tracking services, collaborate with third party (the scope of this work) and comply with government regulations and business intelligence.

The ultimate goal of data integration is to generate valuable and usable information to help solve problems and gain new insights (Doan et al., 2012). Some reasons why it's hard to perform data integration are: **systems reasons** (SQL support across multiple systems not exactly the same, managing different platforms and distributed query processing), **logical reasons** (schema and data heterogeneity) and **social and administrative reasons** (locating and capturing relevant data in the enterprise and also convincing people to share the information) (Doan et al., 2012).

Data integration systems can be warehoused (a data warehouse) or virtual.

Building a Data Integration System

The first step is creating a middleware mediator or data integration system over the sources

which can be warehoused (data warehouse) or virtual. This system presents a uniform query interface and schema. Also abstracts away multitude of sources and consults them for relevant data (unifying different source data formats and possibly schemas). The sources are generally autonomous instead of designed to be integrated. The sources may be local databases or remote web sources/services and may require certain input to return output (Doan et al., 2012).

Regarding the architecture of a generic Data Integration System: First is decided what sources will be used and by using a wrapper/extractor over each one of the sources the information required is extracted and translated into a relational form. Then, by using source descriptions/transforms the information is inserted into the mediated schema or warehouse. In the mediated schema or warehouse it is possible to perform query reformulation and queries over the materialized data (Doan et al., 2012).

Virtual Data Integration System

Logical components of a virtual data integration system (Doan et al., 2012):

- Mediated Schema: it is built for the data integration application and contains only the aspects of the domain relevant to the application. Will contain a subset of the attributes seen in sources.
- Source Descriptions: specify the properties of the sources the system needs to know to use their data. Main component are semantic mappings that specify how attributes in the sources correspond to attributes in the mediated schema, how to resolve differences in how data values are specified in different sources.
- Wrapper: programs whose role is to send queries to a data source, receive answers and possibly apply some basic transformation to the answer. Other info is whether sources are complete.

Regarding query processing in a virtual data integration system, there are three phases: Query reformulation, query optimization and query execution (Doan et al., 2012).

Query reformulation consists in rewriting the user query that was posed in terms of the relations in the mediated schema, into queries referring to the schemas of data sources. The result is called a logical query plan that is a set of queries that refer to the schema of the data sources and whose combination will yield the answer to the original query.

Query optimization accepts a logical query plan as input and produces a physical query plan that specifies the exact order in which sources are accessed, when results are combined, which algorithms are used for performing operations on the data and the amount of resources allocated to each operation.

Query execution is responsible for the execution of the physical query plan. It dispatches the queries to the individual sources through the wrappers and combines the results as specified by the query plan. It may also ask the optimizer to reconsider its plan based on its monitoring of the plan's progress.

It's important to distinguish between virtual data integration systems and materialized data integration systems. Virtual data integration systems are systems where new tables are not created. Instead, when there are two tables and information is needed from both tables, materialized view that uses information of those tables is created. This way a virtual schema - a mediated schema - is created, that will model the kind of answers the users will want when querying the database. Virtual data integration systems also allow getting information from tables in a remote system and a local system. A table that is located in a remote system can be accessed and used when creating a view to be used with a table on the local system (Doan et al., 2012).

Data Warehoused Integration System

Regarding offline replication (Doan et al., 2012):

1. Define a database with a specified schema;
2. Define procedural mappings using a ETL (Extract, Transform, Load) tool to import the data and clean it;
3. Periodically copy all of the data from the data sources (sources and the warehouse are independent at this point).

Advantages and Disadvantages of Data Warehouses

Advantages:

- Queries over the warehouse don't disrupt the data sources;
- Can run very heavy-duty computations, including data aggregation, data mining and data cleaning.

Disadvantages:

- Need to spend time to design the physical database layout, as well as logical which takes a lot of effort;
- Data is generally not up-to-date (lazy or offline refresh).

Regarding materialized data integration systems, these systems create new tables that have joint information of other tables. In order to create/define the new table, the attributes, their types and the primary and foreign keys have to be defined. Then, the inserts are performed in the table.

When joining the data, there can be some issues. There can be problems with the domain from where the join condition is being done. The columns need to have the same pattern so the join can be successfully made. Duplicate detection also needs to be done because some records may represent the same entity and that issue needs to be addressed.

Within the scope of this work, there are three data sources: CSV files, a database and BTEs (PDF documents). BTEs are PDF files that are distributed as a newspaper and contain information about Portuguese unions. The goal is to integrate all this data in a unique database, that will be used by a private institute to ease the search for information.

2.1.5 Relational Data Management

Data warehouses contain consolidated data from many sources, augmented with summary information and covering a long time period. Warehouses are much larger than other kind of databases; sizes ranging from several gigabytes to terabytes are common. Typical workloads involve ad hoc, fairly complex queries and fast response times are important. A distributed DBMS with good scalability and high availability (achieved by storing tables redundantly at more than one site) is required for very large warehouses. An organization's daily operations access and modify operational databases. Data from these operational databases and other external sources are extracted using external interfaces supported by the underlying DBMS (database management system) (Ramakrishnan et al., 2003). Data warehouses are normally implemented over relational databases or RDBMS (relational database management systems) being designed for query and analysis rather than transaction processing (Kimball and Ross (2013) and Inmon (2005)). It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables

an organization to consolidate data from several sources. In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools and other applications that manage the process of gathering data and delivering it to business users. A typical data warehousing architecture is illustrated in figure 2.5.

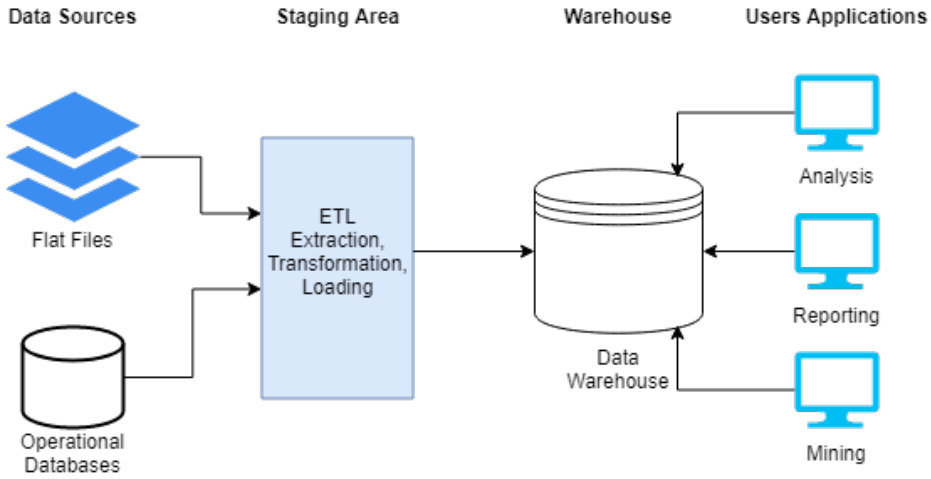


Figure 2.5: A typical data warehousing architecture.

To create and maintain a warehouse, many challenges must be met. A good database schema must be designed to hold an integrated collection of data copied from diverse sources. Since the source databases are often created and maintained by different groups, there are a number of semantic mismatches across these databases, such as different currency units, different names for the same attribute and differences in how tables are normalized and structured. These differences must be reconciled when data is brought into the warehouse (Ramakrishnan et al., 2003).

Data is extracted from operational databases and external sources, cleaned to minimize errors and fill in missing information when possible and also transformed to reconcile semantic mismatches (Ramakrishnan et al., 2003). Transforming data is typically accomplished by defining a relational view over the tables in the data sources. Loading data consists of materializing such views and storing them in the warehouse. Unlike a standard view in a relational DBMS, therefore, the view is stored in a database (the warehouse) that is different from the databases containing the tables it is defined over.

The cleaned and transformed data is finally loaded into the warehouse. Additional pre-processing such as sorting and generation of summary information is carried out at this stage. Data is partitioned and indexes are built for efficiency. Due to the large volume of data, loading is a slow process. Loading a terabyte of data sequentially can take weeks, and loading even a gigabyte can take hours. Parallelism is therefore important for loading warehouses (Ramakrishnan

et al., 2003).

Alter data is loaded into a warehouse and additional measures must be taken to ensure that the data in the warehouse is periodically refreshed to reflect updates to the data sources and periodically purge old data (perhaps onto archival media) (Ramakrishnan et al., 2003). An important task in maintaining a warehouse is keeping track of the data currently stored in it. This is done by storing information about the warehouse data in the system catalogs. The system catalogs associated with a warehouse are very large and often stored and managed in a separate database called a metadata repository.

The value of a warehouse is ultimately in the analysis it enables. The data in a warehouse is typically accessed and analyzed using a variety of tools, including OLAP query engines, data mining algorithms, information visualization tools, statistical packages and report generators (Ramakrishnan et al., 2003).

OLAP applications are dominated by ad hoc, complex queries. In SQL terms, these are queries that involve group by and aggregation operators. In a multidimensional data model, the focus is on a collection of numeric measures. Each measure depends on a set of dimensions. Let's suppose an example based on sales data. The measure attribute in this example is sales. The dimensions are Product, Location and Time. Given a product, a location, and a time, there is at most one associated sales value. Identifying a product by a unique identifier *pid* and, similarly, identifying location by *locid* and time by *timeid*, the sales information can be arranged in a three dimensional Sales array (Ramakrishnan et al., 2003). This array is shown in Figure 2.6.

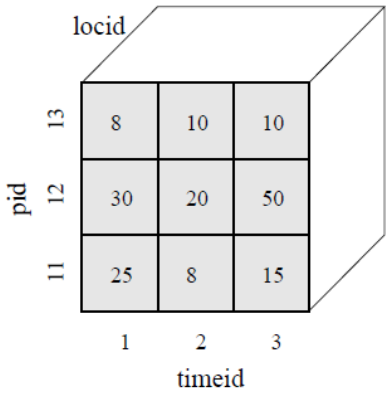


Figure 2.6: Example of a multidimensional dataset (Ramakrishnan et al., 2003).

The data in a multidimensional array can also be represented as a relation, as illustrated in Tables 2.1, 2.2 and 2.3, which show the same data as in figure 2.6, with additional rows corresponding to the slice *locid*=2. This relation relates the dimensions to the measure of interest and is called the fact table.

locid	***city***	***state***	***country***
1	Madison	WI	USA
2	Fresno	CA	USA
5	Chennai	TN	India

Table 2.1: Locations represented as Relations (Ramakrishnan et al., 2003).

pid	***pname***	***category***	***price***
11	Lee Jeans	Apparel	25
12	Zord	Toys	18
13	Biro Pen	Stationery	2

Table 2.2: Products represented as Relations (Ramakrishnan et al., 2003).

pid	***timeid***	***locid***	***sales***
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
11	2	2	22
11	3	2	10
12	1	2	26
12	2	2	45
12	3	2	20
13	1	2	20
13	2	2	40
13	3	2	5

Table 2.3: Sales represented as Relations (Ramakrishnan et al., 2003).

Regarding dimensions, each one can have a set of associated attributes. The location dimension is identified by the *locid* attribute, used to identify a location in the *Sales* table. Also has attributes, country, state, and city. The same happens to the other dimensions. For each dimension, the set of associated values can be structured as a hierarchy (Ramakrishnan et al., 2003). For example, cities belong to states, and states belong to countries. Dates belong to weeks and months, both weeks and months are contained in quarters, and quarters are contained in years. Some of the attributes of a dimension describe the position of a dimension value with respect to

this hierarchy of dimension values. Figure 2.7 represents the hierarchies for the attribute values of Product, Location and Time dimensions.

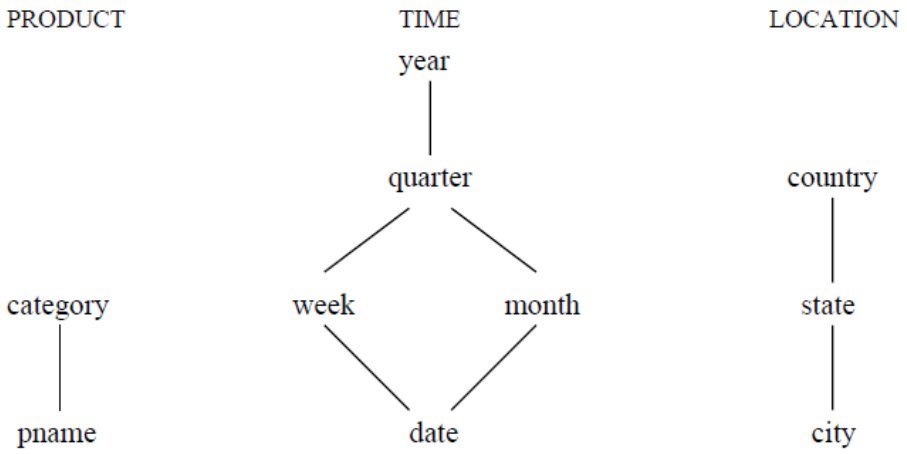


Figure 2.7: Dimension Hierarchies (Ramakrishnan et al., 2003).

Information about dimension can also be represented as a collection of relations (Ramakrishnan et al., 2003):

Locations(*locid*: integer, *city*: string, *state*: string, *country*: string)

Products(*pid*: integer, *pname*: string, *category*: string, *price*: real)

Times(*timeid*: integer, *date*: string, *week*: integer, *month*: integer, *quarter*: integer, *year*: integer, *holiday_flag*: boolean)

These relations are much smaller than the fact table in a typical OLAP application and they are called dimension tables. The *Times* table illustrates the attention paid to the Time dimension in typical OLAP operations. SQL's date and timestamp data types are not adequate; to support summarizations that reflect business operations, information such as fiscal quarters, holiday status, and so on is maintained for each time value (Ramakrishnan et al., 2003). Figure 2.8 shows the tables of the sales example.

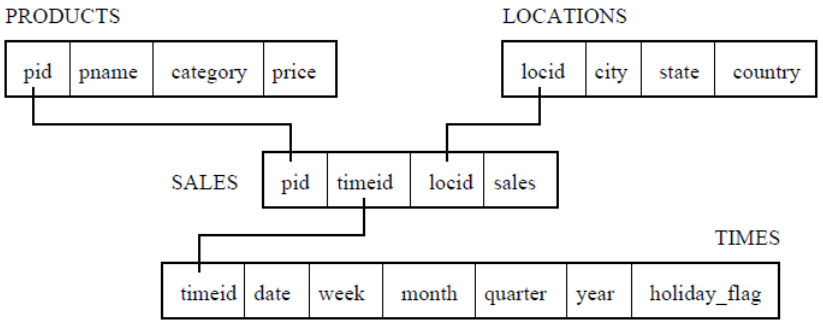


Figure 2.8: Star Schema Example (Ramakrishnan et al., 2003).

It suggests a star, centered at the fact table Sales. The combination of a fact table and dimension tables is called a star schema. This star schema is very common in databases designed for OLAP. The bulk of the data is typically in the fact table, which has no redundancy; it is usually in BCNF (Boyce-Codd Normal Form). In fact, to minimize the size of the fact table, dimension identifiers (such as pid and timeid) are system-generated identifiers (Ramakrishnan et al., 2003).

After having a multidimensional model of data, queries and manipulation of data can be performed. The operations supported by this model are strongly influenced by end user tools such as spreadsheets. The goal is to give end users who are not SQL experts an intuitive and powerful interface for common business-oriented analysis tasks. A very popular operation is aggregating a measure over one or more dimensions. Examples:
Find the total sales.
Find total sales for each city.
Find total sales for each state.

These queries can be expressed as SQL queries over the fact and dimension tables. When a measure is aggregated on one or more dimensions, the aggregated measure depends on fewer dimensions than the original measure. When the total sales by city is computed, the aggregated measure is total sales and it depends only on the location dimension, whereas the original sales measure depended on Location, Time and Product Dimensions. These queries can also be used to summarize information at different levels of a dimensions hierarchy. If the total sales per city are giver, it can be aggregated on the location dimension to obtain sales per state. This operation is called rollup (Ramakrishnan et al., 2003). The inverse of roll-up is drill-down: Given total sales by state, it is possible to get a more detailed presentation by drilling down on location.

2.1.6 Web Application Development

Regarding the web application, it will be developed using the Flask framework. Flask is a web microframework written in Python. Flask has some design principles[5]that will be referred in this section.

The Explicit Application Object A Python web application based on WSGI has to have one central callable object that implements the actual application. In Flask this is an instance

[5]`https://flask.palletsprojects.com/en/1.1.x/design/`

of the Flask class. Each Flask application has to create an instance of this class itself and pass it the name of the module.

The Routing System Flask uses the Werkzeug routing system which was designed to automatically order routes by complexity. This means that routes can be declared in arbitrary order and will still work as expected. This is a requirement for the users that want to properly implement decorator based routing since decorators could be fired in undefined order when the application is split into multiple modules. The routes try to ensure that URLs are unique, automatically redirecting to a canonical URL if a route is ambiguous.

One Template Engine Flask uses the Jinja2 template engine. It is also possible to use a different template engine, although Flask will still configure Jinja2 for the user. Template engines are like programming languages and each of those engines has a certain understanding about how things work. Jinja2 has an extensive filter system, a certain way to do template inheritance, support for reusable blocks (macros) that can be used from inside templates and also from Python code, uses Unicode for all operations, supports iterative template rendering, configurable syntax and more.

Micro with Dependencies Flask is a framework that takes advantage of the work already done by Werkzeug to properly interface WSGI (which can be a complex task at times). Thanks to recent developments in the Python package infrastructure, packages with dependencies are no longer an issue and there are very few reasons against having libraries that depend on others.

Thread Locals Flask uses thread local objects for request and session objects. Thread locals cause troubles for servers that are not based on the concept of threads and make larger applications harder to maintain. Flask is not designed for those applications since its' goal is to quickly and easily write a traditional web application.

2.2 Related Work

2.2.1 Information Extraction from Text

Regarding this subsection, it's important to talk about studies made that are related to the project we are going to develop.

Extracting Information from PDF documents

Regarding information extraction and integration, Zhang et al. (2017) developed a framework named DeepDive that combines database and machine learning ideas to help solve the problems

of developing KBC systems. Knowledge base construction (KBC) is the process of populating a knowledge base with facts extracted from unstructured data sources such as text, tabular data expressed in text and in structured forms, and even maps and figures. KBC has a problem when we try to populate a SQL database with information from unstructured data sources including emails, web pages and PDF reports, quite similar to what we are going to do in this project. The framework uses a standard execution model in which programs go through two main phases: Grounding, in which one evaluates a sequence of SQL queries to produce a data structure called a factor graph that describes a set of random variables and how they are correlated. Every tuple in the database or result of a query is a random variable (node) in this factor graph. The inference phase takes the factor graph from the grounding phase and performs statistical inference using standard techniques. The output of inference is the marginal probability of every tuple in the database.

The authors illustrate the use of the framework with an example related to paleontology, a domain based on the description and biological classification of fossils. The authors used an existing knowledge base compiled by human volunteers as a test bed for the KBC research they made. PaleoDB, one of the largest such knowledge bases, took more than 300 paleontologists and 11 human years to build over the last two decades. The authors constructed a prototype called PaleoDeepDive that takes in PDF documents and attacks challenges in optical character recognition, natural language processing, information extraction, and integration.

To validate the system, the authors performed a double-blind experiment to assess the quality of the system versus the PaleoDB. The results are shown in figure 2.9. They found that the KBC systems built on DeepDive achieved better quality and accuracy than a knowledge base built by human volunteers over the last decade, being also cheaper and less time-consuming. The success of PaleoDeepDive motivated a series of other KBC applications in a diverse set of domains including both natural and social sciences.

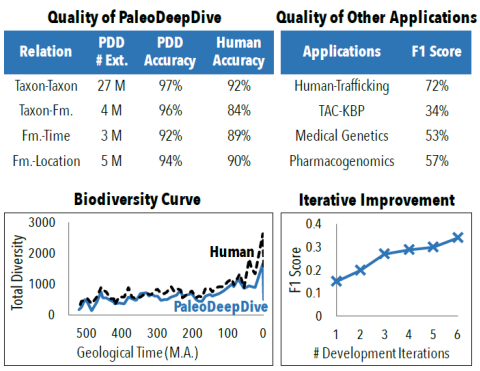


Figure 2.9: Quality of KBC systems built with DeepDive (Zhang et al., 2017).

Extracting Information from Tables

Also regarding information extraction, more precisely extracting information from text documents, Govindaraju et al. (2013) reported the importance of extracting information from documents having an approach that joins the inferences done across tabular and text information. Prior approaches used textual and tabular features separately. The authors approach uses both types of features for relation extraction. The authors used standard NLP (Natural Language Processing) features, such as dependency paths, parts of speech and named entity recognition, etc. They believed that a deeper understanding of the text in which a table is embedded will lead to a higher quality table extraction. Their probabilistic model jointly uses both tabular and textual features. One of the advantages of a joint approach is that one can predict portions of the complicated predicate that is buried in a table.

The authors considered three domains: Petrology, Finance and Geology. For each domain, they built a system to extract relations from text, tables or both. They concluded that a join inference system that uses standard NLP features can significantly improve the quality of the extracted relations, and that this result holds consistently across all three domains.

The authors did an experiment to test their hypothesis. They selected 100 geology journal articles and asked three geoscientists to annotate these journal articles manually to extract a specific relation (1.5K tuples). They processed each document using Stanford CoreNLP, PDFto-HTML and PDF2table. For extraction of features they used the techniques shown in Table 4. Regarding the approaches used, the authors implemented four systems, each of which has access to different types of data:

- TABLE. This approach only used the tables in a document;
- TEXT. This approach only had access to the text in a document;
- MERGE. Using TABLE and TEXT, they extracted all facts and their associated probability. Then they combined those two probabilities using a linear combination;
- JOINT. They built a join approach that uses information from both tables and text. This approach was a large factor graph in which they embedded the CRFs developed in TABLE and TEXT. Additionally, they allowed JOINT to predict projections of each relations.

The features used in TABLE, TEXT and JOINT approaches are shown in Table 2.4.

The authors validated that the JOINT approach achieved higher quality than the other

three approaches considered and concluded that using deeper NLP features combined with a join probabilistic model has a statistically significant impact on recall and precision.

TASK	TEXT	TABLE	JOINT
NER	POS tags Stanford NER Regular Expression Dictionary	PDFtotable NER of neighbor cells Regular expression Dictionary #columns	Whether a mention in table also appears in the text
EL	POS tags Bing query results Freebase Stanford Parser	PDFtotable Bing query results Freebase	Subjective mentions in the sentence near a table
RE	Dependency path Term proximity Word sequence	Table headers Table subheaders RE of neighbor rows	Join between relations

Table 2.4: List of features used in TABLE, TEXT and JOINT approaches. **NER**, **EL**, and **RE** refer to named-entity recognition, entity linking, and relation extraction, respectively (Govindaraju et al., 2013).

2.2.2 Data Integration Supporting the Computational Social Sciences

Regarding the creation of a database, Visser (2015) refers how they created a database on Institutional Characteristics of Trade Unions, Wage Setting, State Intervention and Social Pacts (ICTWSS) with information between 1960 and 2014. This database covered four key elements of modern political economies: trade unionism, wage setting, state intervention and social pacts, and contained annual data on 51 countries.

There were 194 variables, organized in different groups:

- Rights (6);
- Wage Setting (14);
- Social Pacts and Agreements (29);
- Work Council and employee representation in the enterprise (5);
- Union Authority (12);
- Employer organization (2);
- Number and membership of unions and confederation (23);

- Union density and bargaining coverage (18);
- Union concentration and centralisation (15);
- Membership composition and union density by categories (70).

The development of the database, took place in steps. They begun with the sections on social pacts and union organization, membership and authority. The section on social pacts was developed in the framework of the NEWGOV project, financed under the EU FP7 research framework, on "Distributive Politics, Learning and Reform: National Social Pacts". Their database contained information on the negotiation and signing of pacts, the actor combinations involved, whether these were wage pacts or pacts dealing with other issues, whether they were broad or single-issue pacts. In addition the database also contained entries on the existence of bipartite agreements between unions and employers, distinguishing between wage and non-wage agreements, and between autonomous agreements and agreements sponsored by the state or depending on legislation.

The database also covered the existence of bipartite and tripartite councils or bodies for social economic policy making, advice and forecasting. The part on wage setting was focused on features such as bargaining coverage, division between enterprise and sectoral bargaining; level and type (or mode) of coordination, predominant level of bargaining, the frequency or scope of additional enterprise bargaining within sectoral or cross-sectoral agreements, articulation of multi-level bargaining, legal or contractual basis for derogation; the use of opening clauses in agreements, the average length of agreements, the intensity of government intervention, types and grades of administrative extension of agreements, minimum wage setting, employer organization and union centralisation.

3 Problem Analysis and Solution

This chapter details the solution architecture and its components. First, Section 3.1 describes the project methodology followed and the solution designed for the developed work based on the different types of sources. Section 3.2 describes how the extraction and cleaning of data was performed and details how the Web Service was developed. Then, the steps for integration of data into a Database, such as Definition of the Database Schema and Integration of different types of data, are presented in Section 3.3. Finally, the Development of the Web Application for Data Exploration, such as the Development of the Dashboard Page and Development of the Administration Panel, are described in Section 3.4.

3.1 Solution Design

Regarding the project methodology used, a "plan, run, test, improve" process was followed. This type of process is common in agile methodology. The agile methodology is based on the concept of ongoing waves or sprints of project planning and execution, enabling to continuously adapt and mature the plan, scope and design throughout the project. This methodology relies on trusting employees and teams to work directly with customers to understand the goals and provide solutions in a fast and incremental way. During the meetings that occurred throughout the development of the project, reviews of progress were made. This section presents the solution design and describes the steps performed. Figure 3.1 shows the solution design followed.

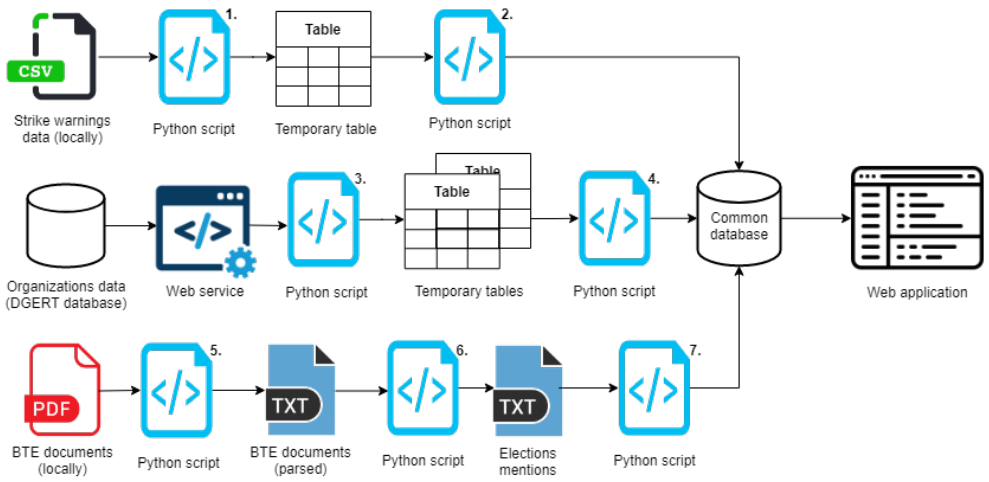


Figure 3.1: Solution Design.
1 - script responsible for importing the contents of the CSV files containing information about strike warnings into a temporary table.
2 - script responsible for cleaning the information and integrating it into the common database.
3 - script responsible for calling the methods of the web service and importing the obtained information into temporary tables.
4 - script responsible for performing data cleaning of the information and integrating it into the common database.
5 - script responsible for parsing the BTE documents (PDF files) into text files.
6 - script responsible for searching elections mentions in the content of the text files. The mentions found were then stored in corresponding text files.
7 - script responsible for cleaning the information from the mentions text files and integrating the elected managers lists into a common database.

Regarding the strike warnings data, the information was already stored in CSV files. DGERT provided the files and they were saved locally in the project workspace. Then, the files were read using a Python script and their content imported into temporary tables. The Python script was also responsible for cleaning the information and integrating it into a common database.

Regarding the organizations data, the information was stored in DGERTs' database. In order to remotely have access to the information, a web service was developed in collaboration with DGERT. The Python script makes calls to the methods of the web service (through developed web service PHP scripts) and imports the obtained information into temporary tables. Then, the Python script performs data cleaning of the information from the temporary tables and integrates the information into the common database.

Regarding the BTE documents, the information was already stored in PDF files. These documents were obtained from DGERT and the Portal do Governo's website. The PDF files were parsed into text files using a Python script in order to ease the search for the information. Then, a Python script was used that was responsible for reading the text files and searching for the desired

information (trade union organizations elections mentions). The elections mentions were stored in text files with the following name format "organizationName_bte[number]_bteYear.txt". These text files were then read in a Python script responsible for cleaning the information from the text files and integrating the elected managers lists into the common database.

In parallel, a web application was being developed in order to manipulate and search the created database. The web application was developed using Flask framework. The web application consists of a dashboard and an administrative panel. The dashboard has graphic representations of data from the database and an organizations' search engine that are directly connected (it dynamically updates the graphic representations). The administrative panel allows performing queries and searching the database.

3.2 Data Extraction and Cleaning

In order to develop this work, the different types of information had to be extracted and cleaned. This data would then be integrated into a database. There were three sources of information such as CSV files (strike warnings) , PDF files (the bulletins) and DGERTs' database (data from organizations). In order to extract and clean the information, different approaches were followed.

3.2.1 Extraction and Cleaning of Data from Organizations

Regarding the data from organizations, the data was located in a operational relational database from DGERT, that is used by DGERT employees.

In order to extract information about organizations, it was decided to develop a Web Service. This Web Service would be used to remotely extract the content from the DGERT tables.

In collaboration with DGERT (Direção-Geral do Emprego e das Relações de Trabalho), more specifically with DGERT worker' Joaquim Félix, was developed a Web Service responsible for providing the content of the tables to be later used in the development of the database. Since the tables with information from organizations are constantly being updated, it's easier to get their content remotely with a defined periodicity.

In a first phase, Materialized Views over the tables to be used and Stored Procedures were created. These Stored Procedures have a defined periodicity. Since some tables were very big and

in order to not overload the DGERT servers, each Stored Procedure was filtered by year. Doing this, prevented the server from going down and being unavailable to other DGERT services.

After completing the first phase, a Web Service using Java was developed in order to have access to the information from DGERT database. Then, some PHP Scripts were created that would be used to remotely obtain the content of the tables (JSON format) from the Web Service. In order to control the calls to the PHP scripts, it was created an authentication mechanism based in HTAccess that asks for username and password. Each PHP Script makes a call to the auxiliary class for the creation of a SOAP Client, used to access the Web Service. For each PHP Script, the method that corresponds to the table (in JSON format) to be returned is invoked.

Regarding the hierarchical structure of JSON data, the data is stored in a dictionary, where each key represents a concatenation of the table name with the number of the row returned and each value corresponds to a dictionary that contains the attribute value pairs of the row returned. The listing 3.1 shows an example of the JSON data structure.

```
1         {"alteracao_1":
2             {"tipo": 2, "especie": 4,
3             "subEspecie": 14, "numero": 1031,
4             "ano": 1996, "controlo": 5,
5             "servico": "DRCOT", "codEntG": 1,
6             "codEntE": 157, "numAlt": 2,
7             "numBTE": 14, "dataBTE": "1988-07-30",
8             "serieBTE": 3, "ambitoGeografico": "NACIONAL"}
9         }
```

Listing 3.1: Example of the JSON data structure

The Web Service was developed using the system/methodology that was adopted at DGERT. The reasons for this have to do with the bureaucracies with the II (Instituto de Informática), who is responsible for implementing the work, security and logistic methodologies at DGERT. The Web Service was made in Java because the II demands all internal services to be made in Java. Furthermore, the II configures each IP address in order to access the Oracle database. The database can't be available to the exterior so the PHP Scripts act as a wrapper to the Web Service, in order to have access to the data.

There are six PHP Scripts to return information from six different tables: Statute Changes,

Elections of Management Bodies, Entities, Collective Agreements, Grantors and Processes.

The Statute Change's table contains information about processes of entities that had statutes change and specifies the BTE where that change is depicted. It also specifies the geographic scope of that change.

The Elections of Management Bodies' table contains information about elections that occurred in entities, specifying the date, number of candidates, number of months in office and number of voters. It also specifies the BTE where that election is depicted.

The Entities' table contains information about entities, specifying the id, acronym, name, postal code, place, phone number, fax number, address, district and status of the entity (active or cancelled).

The Collective Agreements' table contains information about collective agreements between entities, specifying the id, the type, the entities involved and the geographic scope of the collective agreement. It also specifies the BTE where that collective agreement is depicted and the CAE (Classificação Portuguesa de Actividades Económicas) code associated to that collective agreement.

The Grantors' table contains information about entities involved in collective agreements, specifying the id, the type of the collective agreements and which entities are involved in the collective agreements.

The Processes' table contains information about all the processes like elections, statute changes and collective agreements. Specifies the subject, title, designation and the date when the process was opened.

In order to dynamically extract the content of the tables from the Web Service, a Python script was developed. This Python script remotely calls the PHP scripts created with the Web Service and imports the content of the tables returned into temporary tables. This script uses an authentication mechanism in order to control the calls to the PHP scripts from the Web Service, denying access to unauthorized persons. The returned content is used in the creation of the database. The Python script is responsible for remotely obtaining the content of the tables and creating the database, using SQLite3 Python module.

Regarding the cleaning of the extracted data, it was also performed in the previously referred Python script. The Entities temporary table was the only table that needed to be cleaned. Some spelling errors in the name of the entity were fixed and the abbreviations were replaced with the respective meaning. Regarding the acronym of the entity, some acronyms that weren't correct

were fixed and acronyms were added to the entities that contained the acronym in the entity name itself.

3.2.2 Extraction and Cleaning of Data from Strike Warnings

Regarding the extraction of data from Strike Warnings, the developed Web Service was not used because the information about strike warning was not stored in the same database. The strike warnings information was stored in CSV files that were provided by DGERT. Since the Strike Warnings would not be updated anymore, the information stored in the CSV files were used as a basis for the extraction. These CSV files specify the organizations who had strikes and the period where that strike occurred. Figure 3.2 shows an example of the content of the CSV files.

Nº Entrada	Data entrada	Entidade Sindical	Entidade Patronal	CAE	Designação	Início	Fim	Observações	ANO	MÊS
1	02-Jan	FECTRANS - Federação dos Sindicatos de Transportes e Comunicações	Metropolitano de Lisboa - EPE	49310	Empresa	15-Jan		Greve entre as 6 e a	2013	1
2	02-Jan	SMAQ - Sindicato Nacional dos Maquinistas dos Caminhos de Ferro por	MTS - Metro, Transportes do Sul, SA	49310	Empresa	16-Jan	15-Feb		2013	1
3	02-Jan	SENSIQ - Sindicato de Quadros e Técnicos	Metropolitano de Lisboa - EPE	49310	Empresa	15-Jan		Entre as 8:30 e as 12	2013	1
9	02-Jan	SFRCI - Sindicato Ferroviário da Revisão Comercial Itinerante	CP - Comboios de Portugal, EPE	49100	Empresa	20-Jan			2013	1
10	03-Jan	SITE SUL - Sindicato dos Trabalhadores das Indústrias Transformadoras	Lisnave - Estaleiros Navais, SA / Lisnave	33150	2 Empresas	10-Jan	10-Jul		2013	1
Nº 9	02-Jan-13	STSTA - Sindicato dos Trabalhadores do Sector Têxtil de Aveiro	Royal Label - Têxteis, Unipessoal Lda	14131	Empresa	10-Jan-13			2013	1
17	07-Jan	SINFB - Sindicato Independente Nacional dos Ferroviários	REFER - Rede Ferroviária Nacional, EPE /	52211 / 49	3 Empresas	18-Jan	28-Feb		2013	1
Nº 33-2	04-Jan-13	STRUN - Sindicato dos Trabalhadores de Transportes Rodoviários e Urb	Transportes Silva Marques, Ldª.	49410	Empresa	20-Jan-13	22-Jan-13		2013	1
Nº 33-2	04-Jan-13	STRUN - Sindicato dos Trabalhadores de Transportes Rodoviários e Urb	TRACAR - Transportes de Carga e Comér	49410	Empresa	20-Jan-13	22-Jan-13		2013	1
24	08-Jan	FECTRANS - Federação dos Sindicatos de Transportes e Comunicações	Metropolitano de Lisboa - EPE	49310	Empresa	22-Jan		Greve entre as 6 e as	2013	1
Nº 39	07-Jan-13	SITE-NORTE - Sindicato dos Trabalhadores das Indústrias Transformadc	FPS - Fábrica Portuguesa de Segmentos,	78300	Empresa	15-Jan-13	31-Dec-13		2013	1
Nº 57	10-Jan-13	SINTAP - Sindicato dos Trabalhadores da Administração Pública	Câmara Municipal de Coimbra	84113	FP-Câmara	25-Jan-13	30-Jun-13		2013	1
28 e 112	09-Jan	SITE-CSRA - Sindicato dos Trabalhadores das Indústrias Transformador	Sorel, S.A. / Stand Moderno	45110	2 Empresas	17-Jan	TI	A partir das 15:30	2013	1
40 e Nº 66	10-01-2013 e 11-	STMT - Sindicato Têxtil do Minho e Trás-os-Montes	ATP Associação Têxtil e Vestuário de Por	n	Sectores	19-Jan	01-Jan		2013	1
41	10-Jan	SITE SUL - Sindicato dos Trabalhadores das Indústrias Transformadoras	SINDUFLEX-COMERCIALIZAÇÃO DE CON	43290	Empresa	17-Jan	TI	Trab. Colocados na	2013	1
42	10-Jan	SIESI - Sindicato das Indústrias Eléctricas do Sul e Ilhas	KEMET ELECTRONICS PORTUGAL, SA	27900	Empresa	17-Jan	18-Jan		2013	1
54	14-Jan	SENSIQ - Sindicato de Quadros e Técnicos	Metropolitano de Lisboa - EPE	49310	Empresa	29-Jan		Das 8:30 às 12:30	2013	1
Nº 60	10-Jan-13	SITE Centro Norte - Sindicato dos Trabalhadores das Indústrias Transfo	Fábrica Papel Cartão Zarrinha, S.A	17211	Empresa	07-Feb-13	28-Feb-13	Greve 7, 14, 21 e 28	2013	1
55	14-Jan	FECTRANS - Federação dos Sindicatos de Transportes e Comunicações	Metropolitano de Lisboa - EPE	49310	Empresa	29-Jan		Greve entre as 6 e as	2013	1
61 e nº 80	16-Jan	STSSSS - Sindicato dos Trabalhadores da Saúde, Solidariedade e Segurar	Centro Social Infantil da Cruz de Pau	94991	FP - Câmara	21-Jan		Entre as 14:00 e as 1	2013	1
62	16-Jan	FIEQUIMETAL - Federação Intersindical das Indústrias Metalúrgicas, Qu	Thyssenkrupp, Elevadores, SA	28221	Empresa	23-Jan			2013	1
66	17-Jan	FEPCES - Federação Portuguesa dos Sindicatos do Comércio, Escritório	Confederação do Comércio e Serviços d	n	Sectores	15-Jan	30-Jun		2013	1
73-5	18-Jan	SNTSF - Sindicato Nacional dos Trabalhadores do Sector Ferroviário	REFER - Rede Ferroviária Nacional, EPE	52211	Empresa	01-Feb	28-Aug		2013	1
73-5	18-Jan	SNTSF - Sindicato Nacional dos Trabalhadores do Sector Ferroviário	CP - Comboios de Portugal, EPE	49100	Empresa	01-Feb	28-Aug		2013	1
73-5	18-Jan	SNTSF - Sindicato Nacional dos Trabalhadores do Sector Ferroviário	CP Carga - Logística e Transportes Ferro	49200	Empresa	01-Feb	28-Aug		2013	1
73-5	18-Jan	SNTSF - Sindicato Nacional dos Trabalhadores do Sector Ferroviário	EMEF - Empresa de Manutenção e Equip	33171	Empresa	01-Feb	28-Aug		2013	1
73-5	18-Jan	SNTSF - Sindicato Nacional dos Trabalhadores do Sector Ferroviário	CP Carga - Logística e Transportes Ferro	49200	Empresa	01-Feb	28-Aug		2013	1
63	16-Jan	STAD - Sindicato dos Trabalhadores de Serviços de Portaria, Vigilância, I	APFS - Associação Portuguesa de Facility	812 / 801	Sectores	16-Jan	31-Dec		2013	1

Figure 3.2: Example of the CSV files content.

There were two CSV files, one with information ranging between 2013 and 2014 and other ranging between 2015 and 2019. The information was imported into two temporary tables using the Python Script responsible for the creation of the database.

The table with information between 2013 and 2014 specifies the entrance number, entrance date, trade union organizations involved, employer organizations involved, CAE code, tax identification number, designation, strike start date, strike end date and observations.

The table with information between 2015 and 2019 specifies the strike number, entrance date, trade union organizations involved, employer organizations involved, CAE code, tax identification number, activity sector, days of strike, strike duration, strike start date, strike end date and observations.

Regarding the cleaning of data from Strike Warnings, the information from both tables

was joined into another temporary table. This temporary table would contain the Trade union organization ID, the strike start year, the strike start month, the trade union organizations involved, the employer organizations involved, the strike end year, the strike end month and the strike duration. At this phase, the Trade union organization ID was set to NULL since there was no information about the trade union organization ID at that time.

In order to insert the information regarding the years between 2013 and 2014 into the new temporary table, the date related attributes were processed in order to separately extract the year and month. This approach was used in order to had more flexibility to the queries to be posterior performed in the database. Although the table already had the year and month attributes, it was decided that in specific cases, the best approach was to extract that information from the date related attributes because they would be more trustworthy. Regarding the strike duration attribute, the date related attributes were used to calculate it. Sometimes, the attributes were represented using the month name instead of the month number. To solve this, a temporary table was created, that maps the month number to the month name. This temporary table was used in order to calculate the strike duration, while also using information from the date related attributes. Regarding the trade union organization name attribute and more specifically the case where the strike warning had more than one involved (several names in the same attribute field separated by semi-commas), the semi-commas were replaced by slash symbols. This modification was made in order to posterior perform a recursive split.

After cleaning the information regarding the years between 2013 and 2014 and inserting it to the new temporary table, the records where the strike duration was negative were updated. In some records, the employers who inserted them, swapped the values of the date related attributes causing the calculation of the strike duration to be negative. To solve this, the attribute values between the strike start and end dates had to be switched.

To clean the information regarding the years between 2015 and 2019 and insert it to the new temporary table, was used the same approach as in the first table.

In order to separate the records regarding strike warnings with more than one trade union organization involved into several records regarding the same strike warning, a recursive function was created. This recursive function is responsible for splitting by slash symbol the records with more that one trade union organization involved into several records. The recursive function was called in a insert statement into another temporary table. A excerpt of the script that performs this steps is showed in figure 3.3.

After the split, the Entidade_Sindical attribute was updated in order to fix spelling errors

```
cursor.execute("""WITH RECURSIVE split(id, ano_in, mes_in, ent_sind, ent_pat, ano_f, mes_f, dur, rest) AS
(SELECT Id_Entidade_Sindical, Ano_Inicio, Mes_Inicio, '', Entidade_Patronal, Ano_Fim, Mes_Fim, Duracao, Entidade_Sindical || ' / ' FROM TEMP_AVISOS_GREVE
    UNION ALL
SELECT  id, ano_in, mes_in, SUBSTR(rest,0,INSTR(rest,'/')-1), ent_pat, ano_f, mes_f, dur, SUBSTR(rest, INSTR(rest,'/')+2) FROM split WHERE rest LIKE "% / %"
AND INSTR(rest, "(") = 0 AND rest <> '')
INSERT INTO TEMP_AVISOS_GREVE_2 SELECT DISTINCT id, ano_in, mes_in, ent_sind, ent_pat, ano_f, mes_f, dur FROM split WHERE ent_sind <> '';""")
```

Figure 3.3: Excerpt of the script responsible for the strike warnings data integration.

and replace the abbreviations with the respective meaning. This updates were made in order to improve the number of matches to obtain the Id_Entidade_Sindical attribute.

As previously referred, the temporary table has the Id_Entidade_Sindical attribute defined as NULL. In order to get the values for this attribute, the Org_Sindical table was used. This table will be referred to in section 3.3.2 because it is part of the developed database. For the records where the Entidade_Sindical attribute from the temporary table matched the Nome attribute from the Org_Sindical table, the Id_Entidade_Sindical attribute from the temporary table was set as the ID attribute from the Org_Sindical table.

Since some records had attributes switched like the name of trade union organization with the name of the employer organization, those records were updated and the names were switched back.

After performing this update and for the records where the Id_Entidade_Sindical was still NULL, the SQL statement to get the trade union organization ID was executed again.

This approach was followed because it was decided to associate to the Strike Warning the Id of the trade union organization involved. For the strike warnings where there are more than one trade union organization involved, there will be several records describing the same strike warning.

The resulting temporary table would then be integrated in the database, described in section 3.3.3.

3.2.3 Extraction and Cleaning of Data from BTEs' PDF documents

The bulletins were PDF documents that needed to be parsed in order to extract the information. Regarding data from the bulletins, a Python script was created to parse the information from the PDFs and store it in text files. This step was performed after having a part of the database script already created, since information from a specific table was needed in order to know the bulletins mentions regarding the trade union organizations.

The goal regarding the BTEs, was to extract information about elected managers lists and integrate it in the database. In order to do this, a Python script was created, that searches

for those lists in the text files and stores the obtained results in text files corresponding to the mentions. This script iterates over the text files containing the information from the BTEs documents and searches the occurrence of elections of each organization, were the elected managers lists is described. This occurrences are saved into text files corresponding to the mention of the organization in a specific bulletin. These text files would then be used in order to integrate the information into the database, described in section 3.3.4.

3.3 Integration of Data into a Common Database

After performing the extraction and cleaning steps, the already cleaned data was integrated into a common database, using the same Python Script previously referred.

3.3.1 Definition of the Database Schema

The database defined was a relational database consisting of 12 tables. This type of database has logical connections between the tables and allows performing operations in order to manipulate the data. This subsection describes the structure of the database. Figure 3.4 shows the conceptual UML model of the developed database.

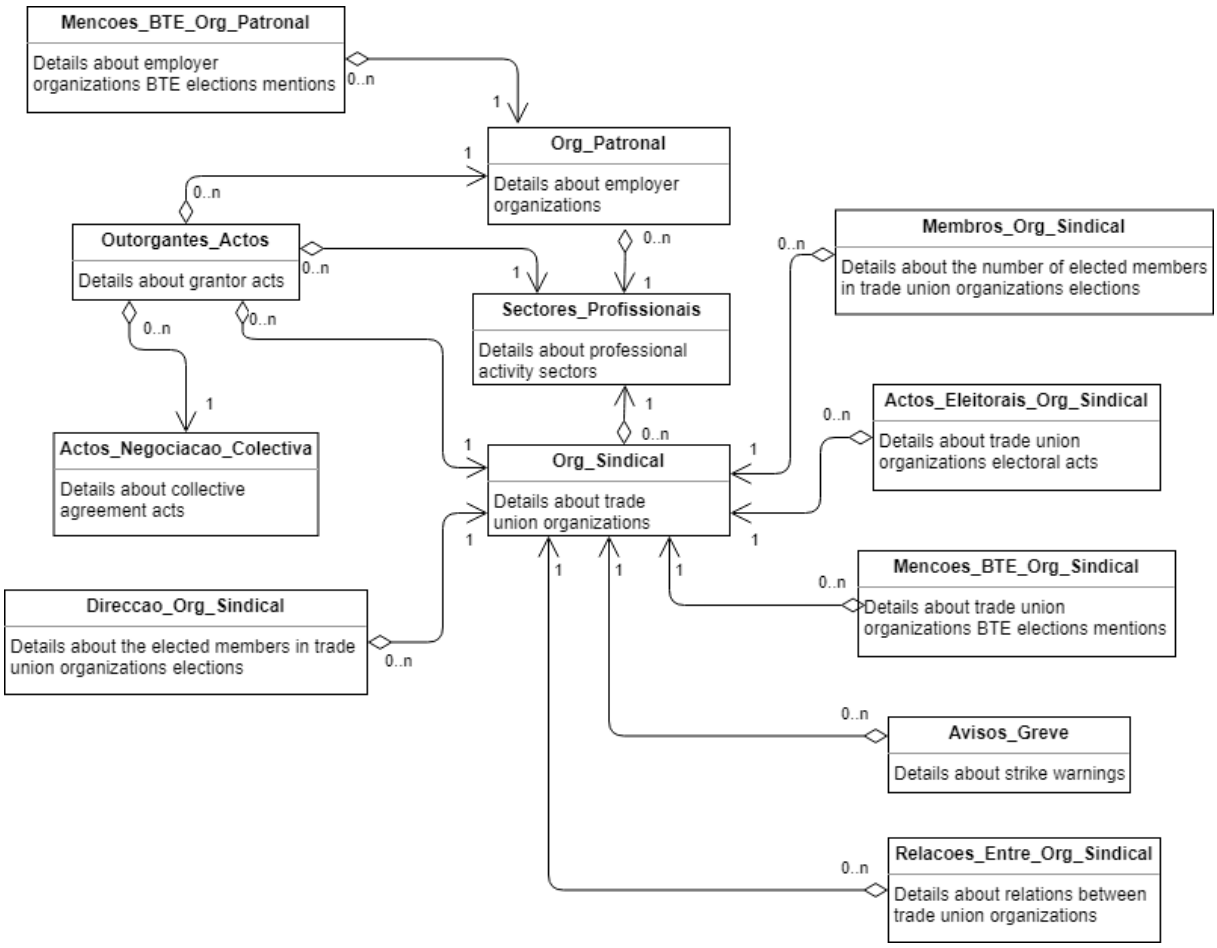


Figure 3.4: Conceptual UML model of the developed database.

Actos_Negociacao_Colectiva table

Regarding this table, it contains information about collective negotiation acts, describing the Act ID, the Act sequential ID, the year when the Act was celebrated, the Act name, the Act type, the Act nature, the BTE number where the act is described, the BTE series number, the Date when the Act was celebrated, the URL corresponding to the BTE that describes the act and the Act geographic scope. The primary key of this table is the set of the act ID, the act sequential ID and the year when the act was celebrated attributes. It was decided to keep information regarding the BTE where the act is mentioned. By doing this, it is possible for the user to search in that BTE for more information.

Sectores_Profissionais table

This table contains information about professional activity sectors, describing the Activity sector name, the abbreviated activity sector name and the medium wage associated to the activity sector. This table can be used to perform different researches using the medium wage of each activity sector as a comparison method. The primary key of this table is the activity sector name attribute.

Org_Sindical table

This table contains information about trade union organizations, describing the organization ID, the organization type, the organization name, the organization acronym, the above organization name, the headquarters county, the headquarters district, the organization postal code, the organization address, the organization address local, the organization postal area, the organization phone number, the organization fax number, the organization geographic scope, the activity sector to which the organization belongs, the number of members of the organization, the date of the organization's first activity, the date of the organization's last activity, the organization status and the organization website. The organization type attribute has four possible values: Union Confederation, Union Federation, Union or Union Union. The organization status has two possible values: Active or Cancelled. The primary key of this table is the trade union organization ID attribute. This table has a foreign key in the activity sector attribute that references the activity sector attribute at the Sectores_Profissionais table.

Org_Patronal table

This table follows exactly the same structure used in the Org_Sindical table but this time for employer organizations.

Actos_Eleitorais_Org_Sindical table

This table contains information about elections regarding trade union organizations, describing the ID from the trade union organization, the election date, the number of members of the electoral roll, the number of registered members, the number of voting members, the number of months of term and the number of competing lists. The primary key of this table is the set of the trade union organization ID and the election date attributes. This table has a foreign key in the trade union organization ID attribute that references the trade union organization ID attribute at the Org_Sindical table.

Avisos_Greve table

This table contains information about strike warnings, describing the ID from the trade union organization that was involved in the strike, the strike start year, the strike start month, the trade union organizations involved, the employer organizations involved, the strike end year, the strike end month and the strike duration. The primary key of this table is the set of all the attributes. This table has a foreign key in the trade union organization ID attribute that references the trade union organization ID attribute at the Org_Sindical table.

Direccao_Org_Sindical table

This table contains information about trade union organization's boards, regarding the board members. Describes the trade union organization ID, the board member name, the board member position, the board start date and the board end date. The primary key of this table is the set of the trade union organization ID, the board member name, the board start date and the board end date attributes. This table has a foreign key in the trade union organization ID attribute that references the trade union organization ID attribute at the Org_Sindical table.

Membros_Org_Sindical table

This table contains information about the number of trade union organization boards members, describing the trade union organization ID, the board start date, the board end date and the number of board members. The primary key of this table is the set of the trade union organization ID, the board start date and the board end date attributes. This table has a foreign key in the trade union organization ID attribute that references the trade union organization ID attribute at the Org_Sindical table.

Mencoes_BTE_Org_Sindical table

This table contains information about mentions of trade union organizations in BTEs, describing the trade union organization ID of the organization mentioned in the BTE, the URL

corresponding to the BTE, the BTE year, the BTE number, the BTE series, the mention description, the boolean Statute Change attribute that says if the mention corresponds to a Statute Change or not, the boolean Elections attribute that says if the mention corresponds to a Election or not and the level of confidence of the mention. The primary key of this table is the set of the trade union organization ID, the BTE year, the BTE number and the BTE series attributes. This table has a foreign key in the trade union organization ID attribute that references the trade union organization ID attribute at the Org_Sindical table.

Mencoes_BTE_Org_Patronal table

This table follows exactly the same structure used in the Mencoes_BTE_Org_Sindical table but this time for employer organizations.

Outorgantes_Actos table

This table contains information regarding Grantors Acts, describing the Act ID, the Act sequential ID, the year when the Act was celebrated, the trade union organization ID of the trade union organization involved in the act, the employer organization ID of the employer organization involved in the act and the activity sector associated to the act. The primary key of this table is the set of all the attributes.

This table has four foreign keys:

- The set of the act ID, the act sequential ID and the year when the Act was celebrated attributes reference the same attributes at the Actos_Negociacao_Colectiva table;
- The trade union organization ID attribute references the same attribute at the Org_-Sindical table;
- The employer organization ID attribute references the same attribute at the Org_Patronal table;
- The sector attribute references the same attribute at the Sectores_Profissionais table.

Relacoes_Entre_Org_Sindical table

This table contains information about relations between trade union organizations, describing the trade union organizations, the type of relation between them and the date when that association was made. Regarding the type of relation attribute, it can have two possible values: Contains or Contained In. The primary key of this table is the set of the trade union organization ID attributes. This table has two foreign keys, the trade union organization ID attributes both reference the trade union organization ID attribute at the Org_Sindical table.

3.3.2 Integration of Data from Organizations

In order to integrate the data from organizations, the JSON content remotely obtained from the six tables were imported into temporary tables in the database creation Python script.

Regarding the statute changes information, it was used together with information from entities and integrated into the Org_Sindical and Org_Patronal tables in order to set the geographic scope of the organizations. It was also used together with the entities information and integrated into the Mencoes_BTE_Org_Sindical and Mencoes_BTE_Org_Patronal tables in order to obtain the information about organizations statute changes mentions.

Regarding the information about electoral management bodies, it was used together with the entities information and integrated into the Mencoes_BTE_Org_Sindical and Mencoes_-BTE_Org_Patronal tables in order to obtain the information about organizations electoral acts mentions. The electoral management bodies were also used together with the entities information and integrated into the Actos_Eleitorais_Org_Sindical table in order to obtain the information about all the carried out electoral acts.

The information about entities was also integrated into the Org_Sindical and Org_Patronal tables, because it contained most of the information regarding trade union and employers' organizations.

The information about collective agreements was integrated into the Actos_Negociacao_-Colectiva table because it contained all the information regarding collective agreement acts. The collective agreements information was also used together with the grantors information and integrated into the Outorgantes_Actos table in order to obtain the grantors acts and the activity sectors associated to each act.

The processes information was used together with the statute changes information and the electoral management bodies information in order to create a view named Datas_Entidades that maps to each organization ID the first and last activity dates. This view was used together with the entities information and integrated into the Org_Sindical and Org_Patronal tables.

After doing all this steps, it was suggested by the members of the REP Project that would be interesting to have information regarding the organizations websites. Using the same Python Script, was developed a function that searched the web and stored for each organization name, the most suitable returned website. This information was stored in a text file and then integrated into the Org_Sindical and Org_Patronal tables.

3.3.3 Integration of Data from Strike Warnings

Regarding information about strike warnings, it was decided to create a table named Avisos_Greve with the following information: Trade union organization ID, the strike start year, the strike start month, the trade union organizations involved, the employer organizations involved, the strike end year, the strike end month and the strike duration. It was considered that those attributes were the ones that would be interesting to have in this table.

This table was filled with the information present in the temporary table used to extract and clean the data from strike warnings described in section 3.2.2.

Finally, the temporary tables used in this extraction, cleaning and integration process were dropped.

3.3.4 Integration of Data from BTEs' PDF documents

Regarding the integration of data from BTEs PDF documents, a portion of the text files containing the information about the elected managers lists were used in the Python script that creates the database. Only a portion of the files was used because most of the bulletins had different layout formats and the extraction of information had various problems. The files were read and their content was inserted into a specific table in the database, the Direccao_Org_Sindical table. This table would contain information about the trade union organization to which the election is associated, the management body member name, the management body member gender, the management body member position, the election date and the term start and end dates. Most of the attributes of the table were filled based on the elected managers list information. Regarding the id of the trade union organization, the Org_Sindical table was used in order to obtain the organization id associated to the name of the organization the election was about. In order to define the gender attribute, two CSV files were used. One of the files contained Portuguese male first names and the other contained Portuguese female first names. For each elected member name, we checked if the first name was in any of the files, setting the gender attribute respectively (Male or Female). For the cases where the first name had no match, the gender attribute was defined as NULL.

It was also decided to store information in another table, the Membros_Org_Sindical table. This table would contain information about the trade union organization to which the election is associated, the number of management body members, the election date and the term start and end dates. This table was filled by performing a group by query in the Direccao_Org_Sindical

table, that counts the number of members associated to each election date, each term (start and end dates) and each organization (Id). Figure 3.5 shows the queries used in the integration into the database.

```
cursor.executemany("""INSERT OR IGNORE INTO Direccao_Org_Sindical(ID_Organizacao_Sindical, Nome_Pessoa, Genero_Sexo, Cargo, Data_Eleicao, Data_Inicio, Data_Fim)
    VALUES(?,?,?,?,?,?,?);""", to_db)

cursor.execute("""INSERT INTO Membros_Org_Sindical SELECT
    ID_Organizacao_Sindical,
    Data_Eleicao,
    Data_Inicio,
    Data_Fim,
    COUNT(*) AS Numero_Membros
    FROM Direccao_Org_Sindical GROUP BY ID_Organizacao_Sindical, Data_Eleicao, Data_Inicio, Data_Fim;""")
```

Figure 3.5: Queries used to integrate the information from the BTEs into the database.

3.4 Web Application Development for Data Exploration

In parallel with the steps previously referred, it was being developed a Web Application for Data Exploration. The Web Application was developed using Python's Flask framework, containing a dashboard page and a administration panel.

3.4.1 Development of the Dashboard Page

It was decided to first develop a dashboard page, containing a login bar for authentication in the web application as also representations of statistics about data from the database and a search bar in order to search for organizations.

Regarding the dashboard page, there is a login bar for authentication by password in order to concede access to the administration panel.

Then, four representations with statistics about data from the database were added. All the representations were created using Chart.js. It was created a bar chart representing Active Union Organizations per Type and Year (Stacked Bar Chart). It was also created a Choropleth map representing Active Union Organizations per District with a color scale. The map was created using an existing open-source Chart.js module [1] for charting Choropleth maps with legends. It was developed a bar chart representing Strike Warnings per Year. As final representation, it was created a horizontal bar chart representing Active Union Organizations per Activity Sector. All the created data representations allow performing mouse hover, displaying specific information.

[1] `http://github.com/sgratzl/chartjs-chart-geo`

Posteriorly, a search bar to search for organizations by name or acronym was added, with the results found being listed as a table on the dashboard page. By clicking the button to the left of the search bar, the user can choose if he wants to search for trade union organizations or employers' organizations. After a search is performed, the four data representations dynamically update regarding the results of the search performed.

To enable even more functionalities, a download button was added, to enable the possibility to download the search results into a excel file. The users of the web application can then use that excel file to analyse and filter the results.

3.4.2 Development of the Administration Panel

Regarding the administration panel, it was used an SQLite data manipulation tool written in Python. The index page shows some basic information about the database, including the number of tables and indexes, as well as its size on disk. In the web application, the user can create, rename or drop columns and indexes. The user can also sort the content of the tables by a given attribute. It's also possible to query the tables and export their content to JSON or CSV format. The web application allows importing CSV and JSON files into a table.

4 Demonstration

Regarding the developed work, there are two main results:

- The creation of a database describing Portuguese unions and other social partners;
- The development of a web application to be used in order to access and manipulate the created database.

The integration of data from the different data sources into a common database was successfully performed and the information was distributed among the 12 tables in the database.

This section describes some details regarding the results of this work. Section 4.1 describes some statistics about the data from the resulting dataset. The results regarding the developed Web Application and the results of the BTEs' information extraction in Section 4.2.

4.1 Statistical Characterization on the Resulting Dataset

The developed database contains 12 tables, with information about Portuguese organizations and other social partners. Figure 4.1 shows content from the Actos_Negociacao_Colectiva table. This table was created using information contained in DGERTs database. Figure 4.2 shows the content of the Avisos_Greve table. This table was created based in information contained in two CSV files that were provided by DGERT.

ID	ID_SEQUENCIAL	Nome_Acto	Tipo_Acto	Natureza	Ano	Numero	Serie	Data	URL	Ambito_Geografico
6804	1	ACORDO COLETIVO ENTRA A BRISA - AUTO ESTRADAS DE P ...	Acordo coletivo de trabalho	Rev. Global	2020	36	0	2020-09-29	https://github.com/bgmartins/rep-database/raw/master/BTE-data/bte36_2020.pdf...	Continente
6405	1	ACORDO DE EMPRESA ENTRE A TOMAZ DO DOURO - EMPPREE ...	Acordo de empresa	Consolidação	2020	12	0	2016-03-29	https://github.com/bgmartins/rep-database/raw/master/BTE-data/bte12_2020.pdf...	Nacional
6261	1	CONTRATO COLECTIVO ENTRE A APHP-ASSOCIAÇÃO PORTUGU ...	Contrato coletivo de trabalho	Rev. Global	2020	15	0	2010-04-22	https://github.com/bgmartins/rep-database/raw/master/BTE-data/bte15_2020.pdf...	Nacional
6494	1	CONTRATO COLETIVO ENTRE A ANCIPA - ASSOCIAÇAO NACI ...	Contrato coletivo de trabalho	Rev. Global	2020	6	0	2016-02-15	https://github.com/bgmartins/rep-database/raw/master/BTE-data/bte6_2020.pdf...	Nacional
6581	1	CONTRATO COLETIVO ENTRE A AES - ASSOCIAÇAO DE EMPR ...	Contrato coletivo de trabalho	Rev. Global	2020	38	0	2017-10-15	https://github.com/bgmartins/rep-database/raw/master/BTE-data/bte38_2020.pdf...	Nacional
6180	1	AE ENTRE A LUSA - AGÊNCIA DE NOTICIAS DE PORTUGAL, ...	Acordo de empresa	Rev. Global	2020	15	0	2009-04-22	https://github.com/bgmartins/rep-database/raw/master/BTE-data/bte15_2020.pdf...	Nacional

Figure 4.1: Content from the Actos_Negociacao_Colectiva table.

Id_Entidade_Sindical	Ano_Inicio	Mes_Inicio	Entidade_Sindical	Entidade_Patronal	Ano_Fim	Mes_Fim	Duracao
2.50.0	2013	1	FECTRANS - FEDERAÇÃO DOS SINDICATOS DE TRANSPORTES ...	Metropolitano de Lisboa - EPE	2013	1	NULL
1.344.2	2013	1	SMAQ - SINDICATO NACIONAL DOS MAQUINISTAS DOS CAMI ...	MTS - Metro, Transportes do Sul, SA	2013	2	30
1.404.1	2013	1	SENSIQ - SINDICATO DE QUADROS E TÉCNICOS	Metropolitano de Lisboa - EPE	2013	1	NULL
1.506.1	2013	1	SFRCI - SINDICATO FERROVIÁRIO DA REVISÃO COMERCIAL ...	CP - Comboios de Portugal, EPE	2013	1	NULL
1.593.0	2013	1	SITE SUL - SINDICATO DOS TRABALHADORES DAS INDÚSTR ...	Lisnave - Estaleiros Navais, SA / Lisnaveyards - N ...	2013	7	181

Figure 4.2: Screenshot of content from the Avisos_Greve table.

4.2 Results

The developed database contains information from various sources like CSV files, PDF files and a relational database. The PDF files contained the bulletins with information about elections and statutes changes. Regarding the BTEs' information extraction, the information from the Mencoes_BTE_Org_Sindical table was used in order to obtain the bulletin mentions to trade union organizations elections. Figure 4.3 shows the content of the Mencoes_BTE_Org_Sindical table. Table 4.1 shows statistics about the trade union organization BTE elections mentions.

ID_Organizacao_Sindical	URL	Ano	Numero	Serie	Descricao	Mudanca_Estatuto	Eleicoes	Confianca
1.101.0	https://github.com/bgmartins/rep-database/raw/mast ...	1979	9	2	NULL	False	True	1
1.101.0	https://github.com/bgmartins/rep-database/raw/mast ...	2011	37	0	NULL	True	False	1
1.103.2	https://github.com/bgmartins/rep-database/raw/mast ...	1998	9	3	NULL	True	True	1
1.103.2	https://github.com/bgmartins/rep-database/raw/mast ...	2002	5	1	NULL	True	True	1
1.103.2	https://github.com/bgmartins/rep-database/raw/mast ...	2006	12	1	NULL	False	True	1
1.103.2	https://github.com/bgmartins/rep-database/raw/mast ...	2010	1	0	NULL	True	True	1

Figure 4.3: Content from the Mencoes_BTE_Org_Sindical table.

Statistic	**Value**
BTE elections mentions (1977-2019)	1939
BTE elections mentions used (2008-2018)	901
BTE elections mentions that found a match with the organization name	686
BTE elections mentions that found a match with the elected managers lists	400

Table 4.1: Statistics about the trade union organizations BTE elections mentions.

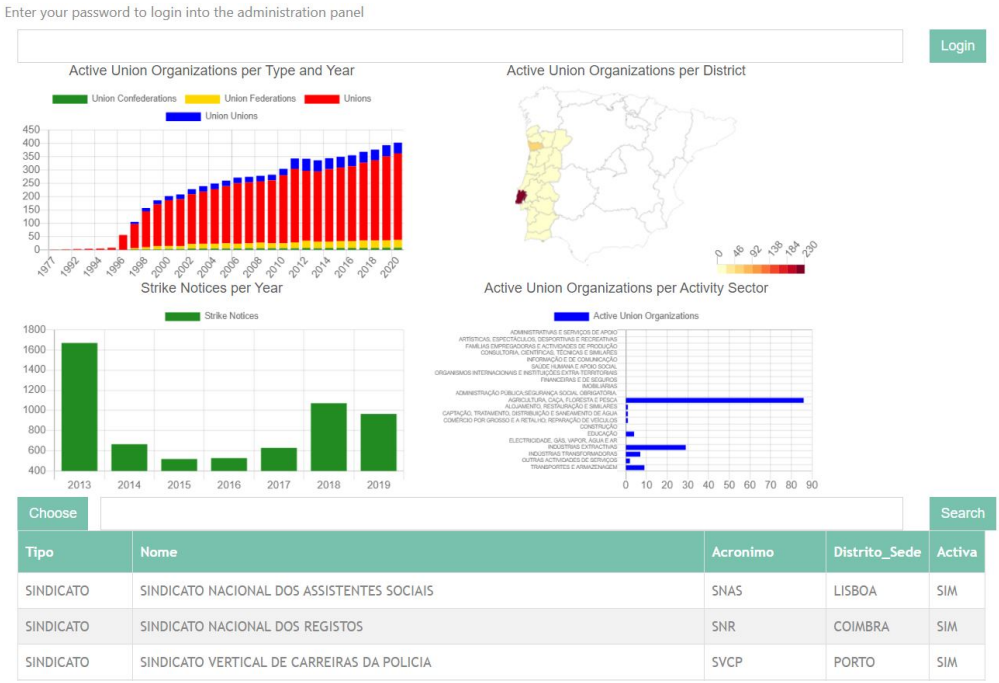
From the Mencoes_BTE_Org_Sindical table, the trade union organizations elections mentions corresponding to the years range from 2008 to 2018 were selected. The reason for using only this range was because the mentioned bulletins were already parsed into text files and in a format that allowed extracting the elected management bodies lists. This range contained 901 elections mentions. The bulletins described in those mentions were searched in order to find the elected management bodies lists associated to the elections mentioned. From those 901 mentions, 686 found a match with the trade union organization name and 215 did not. Regarding the bulletins where a match was found, they were analysed to see if the match contained the elected management bodies list which was the goal of the extraction. From the 686 mentions that found a match with the trade union organization name, 400 found a match that contained the list of the elected members (corresponding to 58%).These 400 mentions were used and the information was integrated into the Direcao_Org_Sindical and Membros_Org_Sindical tables. The Direcao_Org_Sindical table contains personal information. Figure 4.4 shows content from the Membros_Org_Sindical table. In the appendix, A and B show examples of the bulletins files.

ID_Organizacao_Sindical	Data_Eleicao	Data_Inicio	Data_Fim	Numero_Membros
1.108.3	2010-06-12	2010	2013	5
1.117.0	2008-03-19	2008	2010	14
1.117.0	2010-03-17	2010	2012	7
1.117.0	2011-05-17	2011	2013	14
1.117.0	2014-03-31	2014	2017	14
1.117.0	2017-02-15	2017	2020	18
1.119.1	2010-05-26	2010	2014	17
1.119.1	2015-01-14	2015	2019	17
1.122.1	2010-05-31	2010	2012	9
1.122.1	2014-12-13	2014	2016	9
1.122.1	2016-12-19	2016	2018	9
1.129.1	2007-12-07	2007	2010	6
1.129.1	2010-12-17	2010	2013	7
1.129.1	2014-01-31	2014	2017	11
1.129.1	2017-03-24	2017	2020	7

Figure 4.4: Content from the Membros_Org_Sindical table.

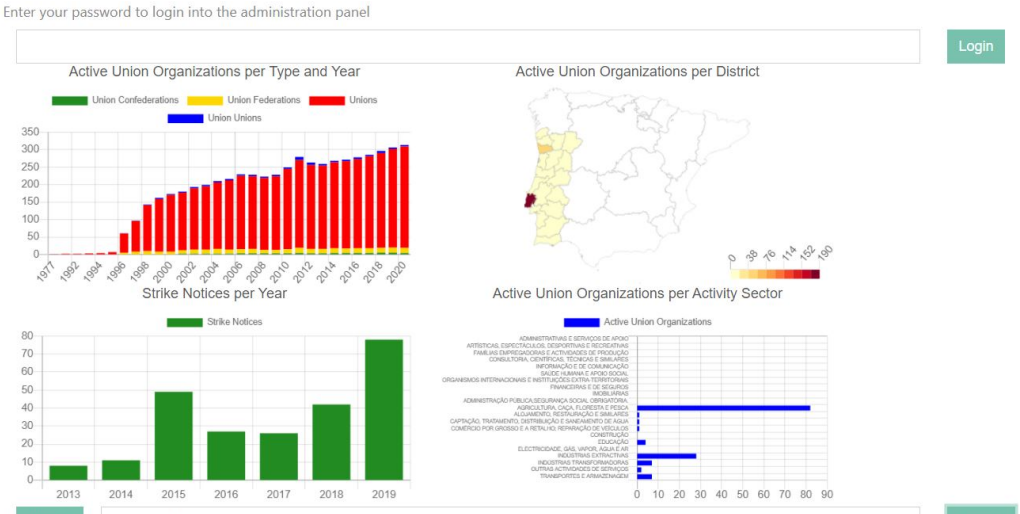
The developed web application consists of two main parts: the dashboard page and the administration panel.

Regarding the dashboard of the web application, it's possible to login into the administration panel, see some statistics about data from the database through charts and also query organizations by name or acronym. The results of the search are displayed in the dashboard and the user can export the results into an excel file. After a search is performed, the four data representations dynamically update regarding the results of the search performed. Figures 4.5 and 4.6 show examples of that dynamism.



Tipo	Nome	Acronimo	Distrito_Sede	Activa
SINDICATO	SINDICATO NACIONAL DOS ASSISTENTES SOCIAIS	SNAS	LISBOA	SIM
SINDICATO	SINDICATO NACIONAL DOS REGISTOS	SNR	COIMBRA	SIM
SINDICATO	SINDICATO VERTICAL DE CARREIRAS DA POLICIA	SVCP	PORTO	SIM

Figure 4.5: Dashboard page in the initial stage.



Tipo	Nome	Acronimo	Distrito_Sede	Activa
SINDICATO	SINDICATO NACIONAL DOS ASSISTENTES SOCIAIS	SNAS	LISBOA	SIM
SINDICATO	SINDICATO NACIONAL DOS REGISTOS	SNR	COIMBRA	SIM
SINDICATO	SINDICATO VERTICAL DE CARREIRAS DA POLICIA	SVCP	PORTO	SIM

Figure 4.6: Dashboard page after performing a search by organizations.

Regarding the administration panel of the web application, there is an index page and different tabs that perform different functionalities. The index page shows some basic information about the database listing all the tables and including the number of tables and indexes, as well as its size on disk. The structure tab displays information about the structure of the selected table, including columns, indexes, and foreign keys (if any exist). From this page you can also create, rename or drop columns and indexes. The content tab displays all the table data. Links in the table header can be used to sort the data. The query tab allows you to execute arbitrary SQL queries on a table. The query results are displayed in a table and can be exported to either JSON or CSV. In order to refresh the information from the database, an option of updating the database was added to this tab. The import tab supports importing CSV and JSON files into a table, that the user can use for research purposes. There is also the option of automatically create columns for any unrecognized keys in the import file. Figure 4.7 shows the content of the structure tab of the administration panel regarding the Actos_Negociacao_Colectiva table.

Figure 4.7: Example of the administration panel page regarding the Actos_Negociacao_Colectiva table.

5 Conclusions and Future Work

As said in the Demonstration section, the main goals of this work were achieved although there were some challenges during the work performed. The pandemic caused many inconveniences while developing this work and some of the tasks had to be done remotely. The Web Service had to be finished working remotely and there were some problems with DGERTs' remote access that delayed its' conclusion.

Initially, when the web service was made, the returned information would retrieve all data from each table and this caused an issue. DGERT's server would frequently go down because of all the information and would be unavailable to all the other DGERT's services. To fix this problem, each call to the web service would be made for a specific year, reducing the data load on the server. In order to enrich the quality of data, the web service had to be updated several times because during the project meetings was suggested to have more detailed information about the organizations.

Regarding the information about the bulletins, only a part of the elections mentions was used since there were format problems in most of the bulletins and would be very difficult to extract the information. The search for the elections mentions was firstly made using a script that searched the bulletins by organization name and the "ELEICAO" keyword but I noticed that there weren't much matches in the search performed. After analysing a portion of the bulletins, I decided to use the "MANDATO" keyword instead. Doing this significantly improved the number of matches regarding elections mentions and those results were locally saved in text files. In the database creation script, I had to parse the elections dates into a SQLite compatible format. To do this, I performed a mapping between the month number and the month name in order to have the date in a compatible format. The references to the elected managers lists had to be validated in order to ensure that they were correct and with results that could be grouped in the database.

The conclusion of the developed web application took a lot of time since it was shown to all the members of the project and improvements were discussed and suggested. Since the dashboard page would be predominantly used by users without experience with SQL, it was important to

discuss with the members of the project the most adequate layout. The administration panel is a more technical tool that allows performing SQL queries and needs to be used by an experienced user. Initially, the data representations were static and after meeting with the project members, it was agreed to relate the data representations with the organizations' search engine. This way, the representations would be dynamic, more interactive and user friendly.

In this dissertation, I discussed the methodology followed to perform this work, as well as the results and challenges.

This section describes the main conclusions of this work. Section 5.1 describes the contributions of this work and Section 5.2 describes ideas and suggestion for future work.

5.1 Overview on the Contributions

The most important contributions of this M.Sc. thesis are as follows:

- The data quality is better, since data cleaning techniques were performed;
- Some info from the BTEs PDF documents is now integrated in the common database making the database richer in information. The information on who was elected as a result of the elections held for the positions of union organizations was added to the database;
- The access to the information is now made through the developed Web Application, where the users can make searches, modify and insert new records to tables;
- The Web Application can be used by various organizations from different areas for study purposes and can also be integrated into projects of those same organizations. Thanks to the authentication mechanism, it is possible to give access to specific organizations in order to let them use the application for research purposes and create statistics about the data.

5.2 Future Work

One of the limitations while performing this work were the challenges to extract information from the bulletins (PDF files). The bulletins have different writing formats over the years, which makes it very difficult to extract information programmatically. In some cases the boards lists appear in the form of a table, causing problems in the extraction process. Other problem was

related to the older bulletins (older than 2008) that weren't original PDF files and their content couldn't be parsed to text files using the same method performed for the newer bulletins.

For future work it would be interesting to extract the information from the older bulletins into text files. This could be performed by parsing the PDF files using an OCR (Optical Character Recognition) tool.

The dashboard created in the web application has graphic representations of data and a organizations search engine. For future work, the dashboard could have more information like adding more graphic representations and other types of search.

The developed database can be used by different institutions in order to perform studies about the social sciences. For future work, it would be interesting to have someone managing the access to the web application in order to allow different institutions to make use of the developed web application in order to manipulate and search the database.

Bibliography

Doan, A., Halevy, A., and Ives, Z. (2012). *Principles of data integration*. Elsevier.

Ganti, V. and Sarma, A. D. (2013). *Data cleaning: A practical perspective*. Morgan & Claypool Publishers.

Govindaraju, V., Zhang, C., and Ré, C. (2013). Understanding tables in context using standard nlp toolkits. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 658–664.

Inmon, W. H. (2005). *Building the data warehouse*. John wiley & sons.

Jensen, C. S. (2020). Trade unionism in europe: Are the working class still members? *European Journal of Industrial Relations*, 26(1):107–120.

Kimball, R. and Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling*. John Wiley & Sons.

Mooney, R. (1999). Relational learning of pattern-match rules for information extraction. In *Proceedings of the sixteenth national conference on artificial intelligence*, volume 334.

Ramakrishnan, R., Gehrke, J., and Gehrke, J. (2003). *Database management systems*, volume 3. McGraw-Hill New York.

Vandaele, K. (2000). Bleak prospects: Mapping trade union membership in europe since 2000. *Bleak Prospects: Mapping Trade Union Membership in Europe Since*.

Visser, J. (2015). Data base on institutional characteristics of trade unions, wage setting, state intervention and social pacts, 1960-2014 (ictwss). *Institute for Advanced Labour Studies AIAS*.

Zhang, C., Ré, C., Cafarella, M., De Sa, C., Ratner, A., Shin, J., Wang, F., and Wu, S. (2017). Deepdive: Declarative knowledge base construction. *Communications of the ACM*, 60(5):93–102.

A

Excerpt of a bulletin file mentioning the Sindicato dos Enfermeiros Portugueses election

José Gerando de Freitas Oliveira, agente principal, bilhete de identidade n.º 138992.
João Manuel Pina Almeida, agente principal, bilhete de identidade n.º 141332.
Manuel Mário Silva Pereira, agente principal, bilhete de identidade n.º 139895.
António Manuel da Silva Freitas, agente principal, bilhete de identidade n.º 143381.
Ricardo Manuel Sá Pinto, agente, bilhete de identidade n.º 152324.
Luís Miguel de Sousa Martins, agente, bilhete de identidade n.º 150956.
Sérgio Carlos Lopes Marques, agente, bilhete de identidade n.º 153613.
José Carlos Ferreira Balbino, agente, bilhete de identidade n.º 149755.
Luís Miguel Jorge Gomes, agente, bilhete de identidade n.º 152847.
Carlos Manuel Pereira, agente, bilhete de identidade n.º 151906.
Vítor Manuel de Sousa Magalhães, agente, bilhete de identidade n.º 155074.
Luís Filipe Costa Pinto, agente principal, bilhete de identidade n.º 145097.
Geraldo Gerónimo Amiguinho Ferreira, agente principal, bilhete de identidade n.º 145545.

Suplentes:

Carlos Diogo Ribeiro Pimenta, agente, bilhete de identidade n.º 154141.
Nelson Emanuel Lorenço dos Santos, agente, bilhete de identidade n.º 148145.

Associação Sindical Autónoma de Polícia — ASAP

Eleição em 5 de maio de 2012 para mandato de dois anos.

Direção Nacional

Presidente — Delmino de Abreu Farinha.
1.º vice-presidente — Manuel dos Santos Quinó.
2.º vice-presidente — Joel Leandro Martins Ferreira.
Tesoureiro — Edmundo Ramos Alves.
Secretário — Luís Filipe Pinto Teixeira.
Vogal — Francisco António Santos Ferreira.
Suplente — Rodrigo Manuel Remuge Teixeira.

Sindicato dos Enfermeiros Portugueses

Eleição em 29 de novembro de 2011 para mandato de quatro anos.

Direção Nacional

Alfredo Manuel Botelho Gomes, 8110766, Viseu.
Ana Margarida Brissos Santos Mendes, 12608916, Lisboa.
Ângela Manuela Sousa Moreira, 3587107, Lisboa.
António Almeida Matias, 414833, Lisboa.
Carlos Dias Barata, 04421731.
Carlos Manuel Oliveira Neves, 6814863, Aveiro.
Celso Filipe Boto Silva, 9897794, Lisboa.
Daniela Martins Braz Santos, 12180160, Lisboa.
Dina Maria Silva Mendonça, 8452278, Leiria.
Edgar dos Santos, 7543908, Beja.
Elisabete Oliveira Ferreira Amoedo, 10763097, Lisboa.
Fernando Manuel Pereira Pais, 8079512, Coimbra.
Francisco Hermínio Meneses Branco, 7069784, Angra do Heroísmo.
Helena Isabel Domingos Jorge, 09631895, Santarém.
Ilda Maria Silva Bernardo, 06631169.
Isa Girão Domingos Pereira, 12558004.
Isabel Maria Lopes Barbosa, 11982525.
João Fernando Duarte Lopes Damásio, 7834873, Santarém.
João Luís Barbadães Morais Pereira, 8665418, Vila Real.
Joaquina Roque Duarte, 7542163, Ponta Delgada.
Jorge Manuel Silva Rebelo, 2358831.
José Carlos Correia Martins, 6977296, Lisboa.
José Dias Tavares, 6666302, Aveiro.
José Domingos Nunes Afonso, 11263069, Porto.
José Manuel Dias Pinto, 8736941, Braga.
José Manuel Santos Araújo, 2047160, Lisboa.
Margarida Maria de Jesus Costa, 5324486, Lisboa.
Maria Antónia Alves Rodrigues, 3017266, Vila Real.
Maria da Conceição Rodrigues Santos Sousa, 4405446, Castelo Branco.
Maria de Fátima Teixeira Gomes Monteiro, 370310.
Maria de Guadalupe Miranda Simões, 7113237, Lisboa.
Maria do Céu Coelhas Rodrigues, 10274932, Évora.
Maria do Rosário Serra Martins Carvalho, 5201654.
Maria João Oliveira Simões Alves, 6212943, Coimbra.
Maria José Birrento Simões, 9955050.
Maria Paula Barroso Vilas Boas Miranda, 6888187, Porto.
Maria Teresa Almeida Faria, 06923515.
Marlene Isabel Lopes Viegas, 12764458, Faro.
Nuno Miguel Dias Manjua, 11226623, Faro.
Nuno Miguel Figueiredo Zambujal, 11570685.
Patrícia Henriques Fonseca Barbosa, 10538553.
Paula Maria Magueijo Lisboa, 4475742, Castelo Branco.
Paulo Jorge Reis Anacleto, 6992479, Lisboa.
Pedro Miguel Teixeira Frias, 11025463.
Rui Manuel Castro Marroni, 4316181.
Sérgio Bruno Santos Sousa, 11432593.
Sérgio Miguel Matias Silva, 11546511.
Susana Alexandra Fonseca Teixeira, 11707865.
Zoraima Arminda Clemente Cruz Prado, 11037975, Lisboa.

Suplentes:

Abel António Varela Rebeca, 10614583.
Ana Clara Vitória Félix, 12247250.
Ana Maria Gaspar Alves, 8695088, Castelo Branco.
Ana Paula Plácido Pais Santos, 10667188, Setúbal.

António Artur Querido Mendes, 332116.
Artur Jorge Correia Almeida, 11534289.
Carolina Galinholas Lopes Ribeiro, 12498026, Beja.
Cristina Mariana Soares Barros Alves, 11114741, Porto.
Fernando Mendes Dias Ferreira, 10366088.
Laura Lorenzo Vazquez, 71011496, Espanha/Zamora.
Marco Aurélio Ferreira Pinto, 10790883, Aveiro.
Paulo Renato Pereira Gomes, 11449413, Leiria.
Ricardo Daniel Serra, 11456416, Portalegre.
Rui Miguel Correia Martins, 11263152.
Vera Lúcia Gomes Ramos, 11847431.

Sindicato Nacional dos Engenheiros, Engenheiros Técnicos e Arquitetos (SNEET)

Eleição em 15 de março de 2012 para mandato de quatro anos.

Direção

Presidente: João Lourenço Martins de Oliveira Pinto, portador do bilhete de identidade n.º 1926677, de 20 de setembro de 2004.

Vice-presidentes:

Augusto Ferreira Guedes, portador do bilhete de identidade n.º 7526592, de 3 de setembro de 2007.

Evaristo de Almeida Guerra de Oliveira, portador do bilhete de identidade n.º 315258, de 1 de setembro de 2005.

Efetivos:

Luís Filipe da Costa Pico Adão, portador do cartão de cidadão n.º 4884142, válido até 6 de março de 2014.

António Manuel Rodrigues Marques, portador do cartão de cidadão n.º 4884239, válido até 16 de dezembro de 2014.

Nuno Maria de Figueiredo Cabral da Câmara Pestana, portador do bilhete de identidade n.º 2021742, de 31 de março de 2006.

Manuel Luís Gomes Vaz, portador do cartão de cidadão n.º 02067634, válido até 19 de setembro de 2016.

Célia Sofia de Almeida Maia, portadora do cartão de cidadão n.º 12986808, válido até 12 de dezembro de 2016.

Vanda Teresa Rogado Medeiro Pereira da Cruz, portadora do bilhete de identidade n.º 11304053, de 4 de dezembro de 2006.

Maria Helena Lopes Francela Capelo, portadora do bilhete de identidade n.º 9274543, de 17 de maio de 2007.

Hugo Miguel França Deodato, portador do cartão de cidadão n.º 11543626, válido até 4 de agosto de 2015.

Suplentes:

Carlos Fernão Gomes Pereira, portador do cartão de cidadão n.º 7635494, válido até 8 de fevereiro de 2015.

José Faustino Fraga Amaral, portador do cartão de cidadão n.º 5666881, válido até 15 de fevereiro de 2017.

José Luís Gonçalves Coelho, portador do cartão de cidadão n.º 6911276, válido até 12 de junho de 2015.

Rui António Pires Pereira, portador do cartão de cidadão n.º 7974982, válido até 5 de junho de 2015.

Paula Cristina Martins Rolo, portadora do cartão de cidadão n.º 8222102, válido até 9 de janeiro de 2014.

Nuno Miguel Matias Tempera, portador do cartão de cidadão n.º 8962717, válido até 7 de setembro de 2014.

Anabela Guerreiro Mestre, portadora do bilhete de identidade n.º 6074872, de 17 de abril de 2007.

Alfredo Manuel da Silva Rocha, portador do bilhete de identidade n.º 2215586, de 11 de outubro de 2005.

Hélder de Sousa Valério, portador do cartão de cidadão n.º 377299, válido até 21 de novembro de 2013.

José Manuel Mendes Delgado, portador do bilhete de identidade n.º 5522790, de 28 de julho de 2005.

ASOSI — Associação Sindical dos Trabalhadores do Sector Energético e Telecomunicações

Eleição em 11 de maio de 2012 para mandato de quatro anos.

Direção

Efetivos:

António Fernando Capinha S. Roque, bilhete de identidade n.º 7195148, EDP — Distribuição Energia, S. A.

José Gonçalves Mendes, bilhete de identidade n.º 4071572, EDP — Distribuição Energia, S. A.

Elísio Lopes da Cruz, bilhete de identidade n.º 7632412, EDP — Distribuição Energia, S. A.

António José dos Santos, bilhete de identidade n.º 6046345, EDP Valor — Gest. Int. Serv.

Isidro Batista Santos, bilhete de identidade n.º 7956645 EDP — Distribuição Energia, S. A.

Suplentes:

Fernando Pedro C. Bernardes, bilhete de identidade n.º 4244018, EDP — Distribuição Energia, S. A.

Tomás Baiano Rebelo, bilhete de identidade n.º 6539943, EDP — Distribuição Energia, S. A.

José Mateus Esteves, bilhete de identidade n.º 4256309, EDP — Distribuição Energia, S. A.

Fernando João Alves Saraiva, bilhete de identidade n.º 7796333, EDP — Distribuição Energia, S. A.

António Augusto Beselga Pais, bilhete de identidade n.º 4307474, EDP — Distribuição Energia, S. A.

Armando Diamantino Cardoso Peneda, bilhete identidade n.º 3154435, EDP — Distribuição Energia, S. A.

José Joaquim Ferreira Pereira, bilhete de identidade

B Excerpt of a bulletin file mentioning the Sindicato da Energia election

Suplentes:

Abilio Marques Duarte, bilhete de identidade n.º 9069076.

Rui Manuel Ferreira Sousa, bilhete de identidade n.º 5396947.

Paulo Alexandre Custodia Lopes, bilhete de identidade n.º 10583690.

Rui Miguel Abreu Duque Aveiro, bilhete de identidade n.º 18317574.

Josué Abel Bandola Martins, bilhete de identidade n.º 6093087.

Sindicato Independente dos Ferroviários e Afins - SIFA

Eleição em 30 de outubro de 2013, para o mandato de três anos.

Secretário geral - José Marques Maia Lindo, cartão de cidadão n.º 6616342.

Vice-secretário - José Guilherme Sequeira Braz, cartão de cidadão n.º 5520493.

Vice-secretário - João Pedro Lopes da Silva, cartão de cidadão n.º 11064937.

Tesoureiro - Jorge Paulo da Conceição Pereira, cartão de cidadão n.º 7466782.

Secretário - António Augusto Batista Margarido, cartão de cidadão n.º 5563625.

Secretário - José Manuel da Costa Lima, bilhete de identidade n.º 6597729.

Secretário - José Manuel da Conceição Lopes, bilhete de identidade n.º 7750953.

Secretário - Nuno Miguel Ferreira Marques, bilhete de identidade n.º 10047530.

Secretário - Jorge Francisco Moreira Raimundo, bilhete de identidade n.º 1650620.

Secretário - Jorge Dias Silva, bilhete de identidade n.º 6489599.

Secretário - Bruno Luis Louro Raimundo, cartão de cidadão n.º 11956726.

Secretário - João Nuno Rodrigues Bernardo, cartão de cidadão n.º 12084195.

Secretário - Daniel Antonio da Costa Domingos, cartão de cidadão n.º 11261322.

SINERGIA - Sindicato da Energia

Eleição em 29 de abril de 2014, para mandato de três anos.

Presidente: Afonso Henrique de Almeida Cardoso, cartão de cidadão n.º 5807513 5ZZ6, data de validade 2/9/2014.

Vice-presidente: Manuel José Martins Pacheco, bilhete de identidade n.º 6069200, data de validade 20/1/2017, pelo arquivo de identificação de Braga.

Vice-presidente: Carlos Manuel Paiva Anselmo, bilhete de identidade n.º 7409822, data de validade 19/3/2015, pelo arquivo de identificação de Ponta Delgada.

Vice-presidente: Emanuel Alberto Mendes Vieira, cartão de cidadão n.º 12569671 0ZZ2, data de validade 4/2/2016.

Tesoureiro: Manuel Luís Figueiredo Alves Silva Fafiães, cartão de cidadão n.º 3817764 1ZZ2, data de validade 22/8/2015.

Vogal: António Manuel Carita Franco, cartão de cidadão n.º 5399968 1ZZ3, data de validade 5/11/2015.

Vogal: António Rodrigues Antunes, bilhete de identidade n.º4085483, data de validade - Vitalicio, pelo arquivo de identificação da Guarda.

Vogal: Isabel Maria Silva Jourdan, bilhete de identidade n.º 6626859, data de validade 29/12/2018, pelo arquivo de identificação de Braga.

Vogal: Joaquim Cardoso Santos, cartão de cidadão n.º 4011193 8ZZ7, data de validade 20/10/2015.

Vogal: Joaquim Coelho Marqueiro, cartão de cidadão n.º 3848932 5ZZ5, data de validade 19/2/2015.

Vogal: José Carlos Marques Palma, cartão de cidadão n.º 06960455 0ZZ3, data de validade 30/11/2016.

Vogal: José Carlos Marques Rodrigues, cartão de cidadão n.º 07790779 5ZZ1, data de validade 28/6/2015.

Vogal: Rosa Maria Valente Pinho Lopes, cartão de cidadão n.º 06502448 6ZY2, data de validade 3/3/2019.

União Geral de Trabalhadores - UGT - Braga

Eleição em 15 de março de 2014, para o mandato de quatro anos.

Presidente:

Sindicato: SBN.

Nome: César Alberto Rodrigues Campos, bilhete de identidade/cartão de cidadão n.º 6649864, profissão: bancário, entidade empregadora: Banco Santander Totta.

Sindicato: SPZN.

Nome: Manuel António Esteves, bilhete de identidade/cartão de cidadão n.º 3495999, profissão: professor, entidade empregadora: Agrupamento de Escolas D. Maria II.

Sindicato: SINDEL.

Nome: Paulo José Sousa Soeiro Gandra, bilhete de identidade/cartão de cidadão n.º 3847592, profissão: técnico administrativo, entidade empregadora: EDP - Gestão da produção de energia SA.

Sindicato: SINTAP.

Nome: Ana Laura Ribeiro Campos Cunha, bilhete de identidade/cartão de cidadão n.º 9885870, profissão: técnico administrativo, entidade empregadora: Município de V. N. de Famalicão.

Sindicato: STE.

Nome: Carlos Eurico Dourado Teixeira Leite, bilhete de identidade/cartão de cidadão n.º 7799541, profissão: economista, entidade empregadora: Segurança Social de Braga.