Database & Dashboard Design of a CRM/BI Application for a State Export Agency

Bachelorarbeit

eingereicht von: **Tomas Sandor, Mag.**
Matrikelnummer: 51918792

im Fachhochschul-Bachelorstudiengang Wirtschaftsinformatik (0470)
der Ferdinand Porsche FernFH

zur Erlangung des akademischen Grades eines

Bachelor of Arts in Business

Betreuung und Beurteilung: Eszter Geresics-Földi, Bsc Msc

Wiener Neustadt, Mai 2023

Ehrenwörtliche Erklärung

Ich versichere hiermit,

1. dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.
2. dass ich diese Bachelorarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.

Michalovce, 11.05.2023

Unterschrift

Die Kapitel 1 bis 3 der vorliegenden Bachelorarbeit wurden im Rahmen der Lehrveranstaltung „Bachelor Seminar 1“ eingereicht und am 06.04.2023 als Bachelorarbeit 1 angenommen.

Abstract: Database & Dashboard Design of a CRM/BI Application for a State Export Agency

The main focus of this thesis is put on the topic of databases accompanied by a complementary theme dedicated to the dashboard design. Both themes are mirrored in the search for the fundamental and secondary aim of this thesis which were defined as follows: identification of the most appropriate database type and its consequent design accompanied by a dashboard design for the future CRM/BI application for the agency named SARIO – Slovak Investment and Trade Development Agency. The thesis structure consists of six bearing chapters divided into theoretical and practical part. The introductive part is devoted to the problem description, chosen methodologies and to the present state of affairs in the corresponding scientific field. The second section provides the core information on the relational and non-relational databases. The third chapter delivers knowledge on the CRM&BI systems. The fourth chapter represents the first practical one, and deals with a more detailed problem description, figures on the agency, defined requirements on the database and dashboard what resulted in the weighted scoring model elaboration. The fifth section is dedicated to the practical database and dashboard design. The research decisions, result and the final artifacts are recapitulated in the conclusive part of the thesis.

Keywords:

databases; relational databases; non-relational databases; design; dashboard; SQL

Kurzzusammenfassung: Datenbank- und Dashboard-Entwurf einer CRM/BI-Anwendung für eine staatliche Exportagentur

Der Schwerpunkt dieser Arbeit liegt auf dem Thema der Datenbanken, begleitet von einem ergänzenden Thema, das dem Dashboard-Design gewidmet ist. Beide Themen spiegeln sich in der Suche nach dem grundlegenden und sekundären Ziel dieser Arbeit wider, die folgendermaßen definiert wurden: Identifizierung des am besten geeigneten Datenbanktyps und der konsequente Datenbankentwurf, begleitet von einem Dashboard-Design für die zukünftige CRM&BI-Anwendung für die unternehmensunterstützende Agentur des Namens SARIO - Slowakische Agentur für Investitions- und Handelsförderung. Die Struktur der Arbeit besteht aus sechs Kapiteln, die sich in einen theoretischen und einen praktischen Teil gliedern. Der einführende Teil ist der Problembeschreibung, den gewählten Methoden und dem gegenwärtigen Forschungsstand gewidmet. Was die Theorie betrifft, so liefert der zweite und gleichzeitig der grundlegendste Teil die Kerninformationen über die Theorie der Datenbanken, einschließlich der relationalen und nicht-relationalen Datenbanken. Das dritte Kapitel liefert Wissen über CRM&BI-Systeme. Das vierte Kapitel stellt das erste praktische Kapitel dar und befasst sich mit einer detaillierteren Problembeschreibung, Informationen über die entsprechende Agentur, definierten Anforderungen an die Datenbank und dem Dashboard-Design, nach denen das „Weighted-Scoring-Modell“ erstellt wurde. Das fünfte Kapitel ist dem praktischen Entwurf der Datenbank und des Dashboards gewidmet. Die Forschungsentscheidungen, das Ergebnis und die endgültigen Forschungsartefakte werden im abschließenden Teil der Arbeit rekapituliert.

Schlagwörter:

Datenbanken; Relationale Datenbanken; Nicht-relationale Datenbanken; Entwurf; Dashboard; SQL

Contents

1. Introduction

The main aim of this bachelor thesis is to identify the most suitable database type and to design an appropriate database and a dashboard design for the future CRM/BI application for a business-oriented state agency named Slovak Investment and Trade Development Agency (SARIO). The need for a coherent, accessible and structured data pool stems from an increasingly higher volume of information flow the agency has been facing for the several past years on a daily basis. This fact appears to be in a close relation with the state of art in the field of a high-paced technology environment that is doubtlessly intertwined with a business or economic sphere. Both of the entities mentioned have a more-or-less direct impact on the volume of information that is to be processed by governmental bodies – in this case by the SARIO Agency. Similar to this thesis the state has its purpose(s) too. From an economic or a financial point of view one of its goals shall be security, maintenance and increase of the people´s economic welfare. This is certainly achievable if a state disposes of appropriate mechanisms and it catches up with the latest trends that enables it to administer the state welfare and public goods in an economically efficient way. This role or a function and means of a state links us to the term “cost-efficiency”. One can claim that this term is inherently interconnected with the birth of the electronic database systems. The roots of the today´s database systems are to be found in the 1950s and 1960s, in the years where computers and computer programs became economically effective and efficient for the business purposes (Foote 2021). Of course, the existence and an increasing usage of computers were not a straight cause of the phenomenon called database systems. It was rather the power of – previously left behind – data which induced the creation of the first databases and database management systems in the 1960s.

Whereas in the course of 1960s, a discussion on what type of a database shall be chosen for a company or a state agency would have been more-or-less unimaginable, regarding the size of the in that time possible and available opportunities, nowadays the companies and state entities must or at least should resolve the problem of selecting the right database from a wide range of types/products available. Our object of interest, the Slovak Investment and Trade Development Agency does not represent any exception. The agency represents a governmental body – working under the auspices of the Ministry of Economy of the Slovak Republic – that encompasses around 90 employees, most of whom are responsible for a development and enhancement of the economic status of the Slovak Republic and of the plenipotentiaries of the Slovak business sphere. Considering the aforementioned, the main aim of the agency is to promote advantageous economic environment of the Slovak Republic and simultaneously to support export activities of the entrepreneurs and companies based in Slovakia. Our main aim is to design the most appropriate database type and a dashboard for the future CRM/BI application for this agency based on certain key requirements that will be characterized in the corresponding chapter of this thesis, similarly as a considerably more-detailed description of the agency´s structure will be provided in the competent thesis chapter.

1.1 Methodology

Following the main aim of this thesis the following three methods were chosen to be applied: Comparative Analysis, Design Science Research, Weighted Scoring Model. It is also important to note, that other research methods could have been applied instead of the chosen ones. Admittedly, one could apply the “OSEMN” methodology and/or the “cross-industry standard process for data mining”, and/or also observation/monitoring method. However it was assumed that application of those methods would have required more space, time (regarding the intensively iterative character of the techniques mentioned) and even access to the exact constantly changing data (in real-time) of the object organization. The immediate text lines are dedicated to the methods of this bachelor thesis.

1.1.1 Comparative Analysis

A considerable part of this thesis will belong to several existing, appropriate and available database types. This database types represent our objects of interest and they are to be characterized and consequently compared. This comparison is impossible to be realized without a comparative analysis. The comparative analysis shall provide us with useful information on the characteristics of the selected database types, their main similarities, discrepancies and predominantly their most perceivable advantages and disadvantages relevant for the purposes of this thesis. However, comparative analysis shall also assist us to come to “a deeper understanding of a concept, problem or debate” (Comparative Analysis: Flipped Learning Module 2021), and thus it shall not be mistaken for a brief comparison. The quality and reliability of the information processed in the course of the comparative analysis is of high importance, therefore a special attention will be paid to the quality, relevance and balance of the scientific literature and other resources.

1.1.2 Design Science Research

The approach called “design science research” – based on the epistemological paradigm of design science – represents the second scientific method that will be used in order to achieve the main objective of the thesis. At this point it seems justifiable to provide the readers with a brief explanation on the difference between "design science" and "design science research". Whereas the term "design science" refers to "knowledge in the form of constructs techniques and methods, models, and/or well-developed theory for performing this mapping – the know-how for creating artifacts that satisfy given sets of functional requirements" (Vaishnavi and Kuechler 2004/21, 4), the term "design science research" can be defined as a "research that creates this type of missing knowledge using design, analysis, reflection, and abstraction" (Vaishnavi and Kuechler 2004/21, 4). This scientific approach may also be characterized as "a method that establishes and operationalizes research when the desired goal is an artifact or a recommendation" (Dresch et al. 2015, 67). In our case, the desired outcome or an artifact is the most appropriate database design for the future CRM&BI solution for the Slovak Investment and Trade Development Agency (SARIO). Specifically databases are mentioned by the academics as perspective artifacts that might be constructed based on the knowledge of the principles for the construction of data and visualization interfaces applying the data science research approach (Vaishnavi and Kuechler 2004/21, 26). The design science research method was also chosen since it "seeks to reduce the gap between theory and practice but maintains the necessary amount of rigor to ensure reliability of research results" (Dresch et al. 2015, 71) and, moreover, "the solutions generated by design science research should be liable to generalization for a specific class of problems" (Dresch et al. 2015, 68).

In order to bring a scientific research into effect according to the principles conveyed by the design science research a plenty of scientific approaches or methods were formalized by the plenipotentiaries of the academic sphere. However, regarding the aim and mainly the scope of this thesis it seems appropriate to present the only one, and simultaneously the most relevant approach for the thesis purpose. Paradoxically, the approach to be followed in this text is not considered as an approach or a method per se, and even the authors define it as a "set of guidelines for conducting and evaluating good design-science research" (Hevner et al. 2004, 77). Prior to introduction of the seven guidelines, that are to be itemized in a manner, whereby light will be shed on their relevance for the desired artifact – the database design – it is crucial to note that the main goal to be achieved by application of the seven rules is utility (Hevner et al. 2004, 80). In other words one can claim that the fundamental aim the aforementioned guidelines is an applied or an incorporated truth, what corresponds with our goal.

As promised in the lines above the following seven "guidelines for conducting and evaluating good design-science research" will be stated together with explanations mirroring their meaningfulness for our research:

<u>Guideline 1: Design as an Artifact.</u> The result of a design science research must be a viable artifact in the form of a construct, model, method or an instantiation (Hevner et al. 2004, 83). Our design science research-based developed artifact shall be embodied by

a viable product – a database design, accompanied by a dashboard design for the future CRM&BI application. Without any doubts, a viable database fulfils the conditions stipulated in the first guideline.

Guideline 2: Problem Relevance. The aim of a design science research is to develop technology-based solutions to important and relevant problems in the business sphere (Hevner et al. 2004, 83). Similar to the first criterion, the second one is fully 'represented' by the database and dashboard design for a state agency that aims to enhance the entrepreneurial environment and thus to boost the national economy. The status of the agency, together with the boarding of the industrial revolution "Industry 4.0" and increasingly huger data flow and their importance even strengthen the relevance of and a demand for quality database and dashboard design.

Guideline 3: Design Evaluation. The utility of the design science research result artifact, its quality and efficacy is to be demonstrated by a well-executed evaluation. (Dresch et al. 2015, 70). This criterion shall be also fulfilled in the course of this thesis, where various database types will be commented on and their efficacy will be evaluated. Of course an explanation or a justification for the choice of the final database design will be provided.

Guideline 4: Research Contributions. A design science research-based work must bring with itself verifiable contributions in the specific areas of the developed artifact and present clear grounding on the foundation of design and/or design methodologies. (Dresch et al. 2015, 70). As mentioned several times in the lines above the main artifact of this research is to be embodied by a database and dashboard design. The author of this thesis frankly wishes that the conclusions of this work and the knowledge produced in the course of this research will – to the most possible extent – enrich the knowledge base for the future research. On the other side one shall take the level or the character of the bachelor theses into account.

Guideline 5: Research Rigor. The research shall be conducted based on application of rigorous methods in both, the construction and the evaluation of artifacts. (Dresch et al. 2015, 70). As aforementioned, the key of the design science research is to accomplish the utility by the end of the research. This utility, which is – according to the majority of the relevant academics – equal to truth, shall stem from a solid knowledge base. Afterwards the knowledge base that stand at researcher's disposal shall be processed and applied accordingly. This means it should not be in correspondence exclusively with the scientific theory but also special conditions of the final artifact environment shall be taken seriously and properly into consideration. In our case the environment of the artifact is embodied predominantly by the requirements of the users and its application and usage by the users that are represented by the employees of the agency.

Guideline 6: Design as a Search Process. Following the formation of a definitive artifact a demand for implementation of all the resources available in pursuit of the pre-set goals is created, while the respect to the provisions and/or conveniences of the particular field is not left behind (Hevner et al. 2004, 83). In order to achieve the best, the most appropriate outcome – translated – the most suitable database design a considerable amount of the literature is to be used in order to broaden the author´s knowledge base. The laws of the environment, represented by rather technical factors and functioning, together with the practical usage requirements and/or conditions will be taken into consideration when choosing and creating the final artifact.

<u>Guideline 7: Communication of Research.</u> The artifact or the result of a design science research must be presented effectively both to technology-oriented as well as management-oriented audience (Hevner et al. 2004, 83). This requirement or a guideline for a good design science research will be fulfilled after the "production" of the final artifact. This is to be presented to the potential real users of the future application.

The afore-stated seven guidelines proposed by Alan Hevner should provide the researchers with an assistance in the design science research-based research heading towards the final artifact. However, this seven rules do not – in their own isolation – represent a sufficient base to start and convey the research. First, a solid and relevant knowledge base is of an inevitable character. By the adjectives solid and relevant it is meant that the knowledge shall be broad enough (enough to satisfy the needs of the research and in result to produce a quality artifact) and up-to-date. Therefore a number of sources will be applied for the purposes of this thesis, and majority of them is in accordance with the condition of the so-called "*up-to-dateness*".

Second, certain "*modus vivendi*" or "*modus operandi*" shall be coined when conveying the research and writing this particular thesis. Both *modi* can be found in the framework proposed by the authors of a paper called "Design Science in Information System Research" (Hevner et al. 2004, 79). The framework was illustrated for the application in the field of information systems, where two main scientific paradigms are taking place – the one of behavioural science on the one hand, and the one of design science on the other (Hevner et al. 2004, 79). The figures shows the information system research itself is placed in the centre of the framework. The research is formed by two "powers" – business needs and applicable knowledge. Those "powers" are results of the environment, regarding the business needs, and of the applicable knowledge when speaking about applicable knowledge. The business needs are presented as an outcome of various people, their organizations, and previous, concurrent and future technologies that are covered under the term "environment". The other side of the framework, the applicable knowledge is formed by the researcher´s theoretical foundation and methodologies covered under the term "knowledge base". As it is stipulated in the seven guidelines the research artifact shall represent a solution of a relevant business need produced under rigor and simultaneously it should be of a beneficial character to both – the environment, representing rather the behavioural science paradigm, and the knowledge base, which will be eventually able to produce future design science study enriched by the knowledge gained and mined in the course of the previous research.

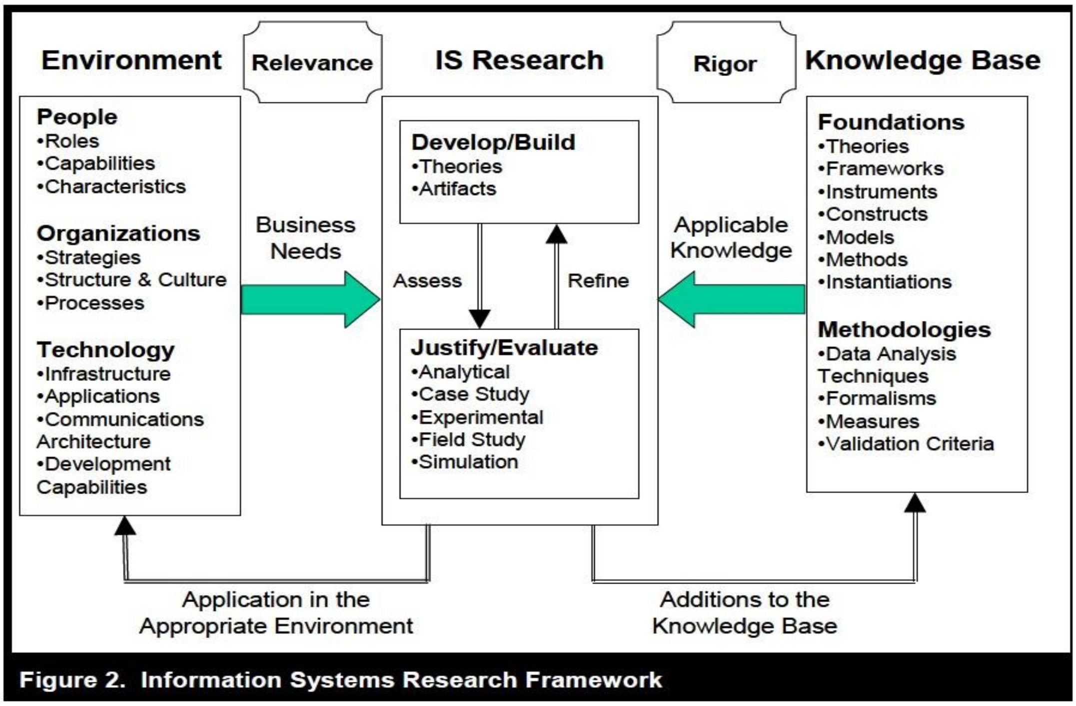


Figure 1: Information Systems Research Framework elaborated by Hevner et. al (Hevner et al. 2004, 80).

The figure pictured and described above represents a general prototype for the design science research. Therefore, in accordance with the principles of the design science research, considering the specific environment and of our research a customized figure is to be found and described in the lines below.

The first group of the left side framework is represented by the users of the artifact, which means by the relevant employees who will work with and use the database on more-or-less daily basis and so will they use the dashboard. Both artifacts are to be part of the future CRM&BI application. However, the structure of the application itself is not among the focuses of this thesis. The main agenda of the relevant employees/users is communication with the business and state circles, professional consultations with the relevant business companies/partners and state bodies, organization of economic-related fora, business missions and national stands at international trade-fairs. Therefore, we can claim that the main focus of the agenda is put on communication with and towards business and state spheres.

The information on the aforementioned briefly characterized agenda is to be processed for both internal and external purposes. Nowadays, most of the information is stored using high number of separate, incoherent excel table sheets with a considerably limited access for all relevant employees/users. Moreover, the individual tables are structured differently what prevents the users from immediate readability and comprehension of the stored information. The need for an advanced storage of the information or data flow is the increasingly huger amount of the information units. A proper storage, structure and generally organization of the information gained will provide the users, hence the employees with a faster availability, enhanced clarity/transparency and history of the information. The information properly stored in a database shall also ameliorate, accelerate the time-management of the entire organization, of the agency. A tailor-made database shall enable the organization to achieve a substantial enhancement in the issue the references were made to.

The right side of the framework made of knowledge base is represented by the theoretical foundations and methodologies. The first element, the theoretical foundations of this research are to a large extend dependent on the up-to-dated or concurrent relevant literature that was gathered, filtered and researched for the purposes of this work and that will continuously and constantly be re-evaluated and repeatedly researched in order to achieve the most appropriate final artifact. Most of the literature is thus concentrated on the current existing and available databases, database management systems, database types and of course of on their design. Naturally, the literature on the dashboard structure and design will not be omitted.

The methodological background of this thesis is represented in the subparts of this chapter. The needs or the requirements of the relevant employees thus of the users will create an important base for the actual form or resemblance of this thesis. On the other side the relevant available database, database systems will be characterized, compared, evaluated and finally weighted by of the weighted scoring model method that will be developed/designed according to the relevant business needs. The author of this thesis will endeavour to satisfy and fulfil the business requirements by the mentioned methodologies, the methods applied in the relevant literature sources and by his already gained databases-related knowledge. The business requirements will be derived from the structure, agenda and function of the Slovak Investment and Trade Development Agency (SARIO), and from the information and relevant demands and remarks provided by the employees of the agency. Due to a substantially high sensitivity degree of the original data the final artifact and potential database designs and evaluations will be exclusively populated by unreal, fabricated data.

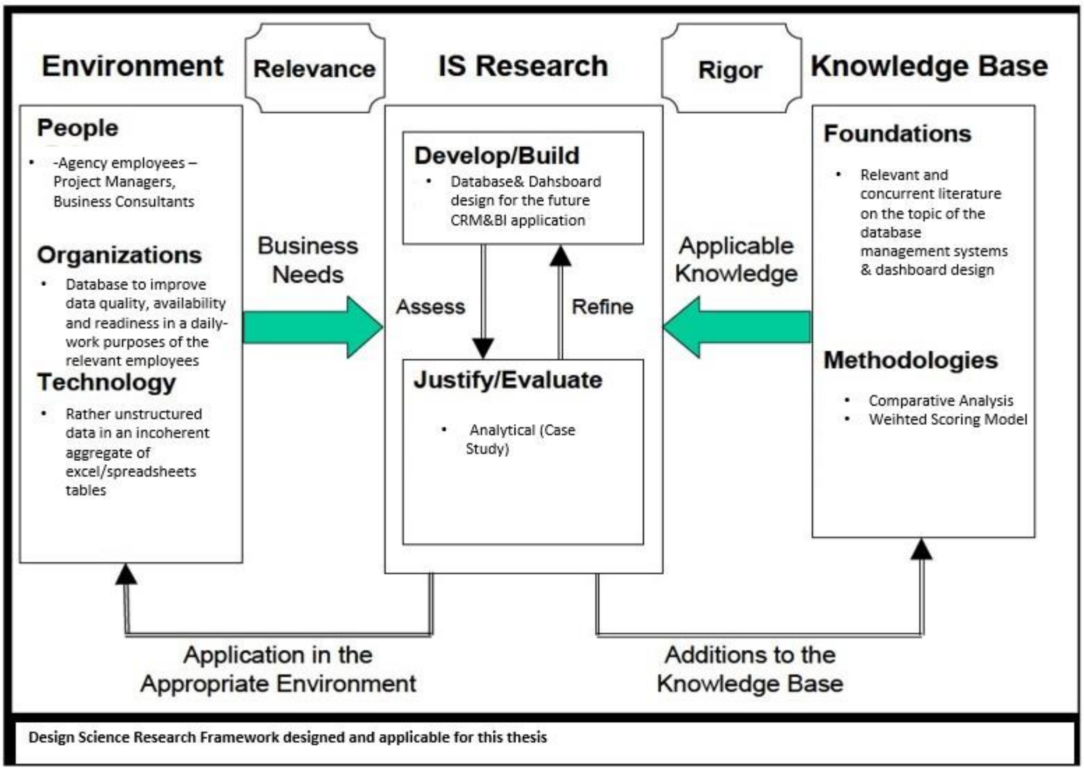


Figure 2: Design Science Research Framework designed and applicable for this thesis based on the example presented by Hevner et al. (Hevner et al. 2004, 80).

Withing the boundaries of the aforementioned framework and by assistance the selected methods and knowledge base, the thesis shall provide us with a satisfying answer to the following main research question: “What is the most appropriate and applicable database type for a CRM/BI application for a state export agency which disposes of a rather unstructured data storage?”. Our hypothesis to this question sounds as follows: “MySQL relational database is the most appropriate database type to be applied in this case.” The theoretical and practical findings gained in the process of this thesis shall provide us relevant information on the basis of which we will be able to confirm or to disconfirm the expressed hypothesis.

Furthermore, the findings to be gained and elaborated in the course of the research shall provide us with answer on the fractional questions or sub-questions which shall lead to a qualified and quality answer to the main research question. The fractional questions/sub-questions were formed as follows:

SRQ1: What relevant database management systems are currently available?

The field of data science, an inherent part of which are the database management systems, is one of the most prospective fields of the revolution (Industry 4.0 and the Fourth Industrial Revolution Explained n.d.), thus it is evolving at fast-pace. This statement is also applicable to the database management systems. Therefore, it is crucial to identify the relevant available and potentially appropriate database management systems that may satisfy the needs of our specific case, since limited scope of this thesis will not be sufficient to a deeper study on all of the available database management systems.

SRQ2: What are the relevant perceivable differences between the relational and non-relational database management systems?

Based on the data model the database management systems are divided into two main classes – relational and non-relational databases. The comparison of this two basic classes of database management systems shall bring us closer to the final evaluation, where, on the basis of the knowledge base and business needs, a decision on the character of the final artifact shall be derived.

SRQ3: What are the relevant perceivable differences among the relational database management systems?

The answer on this sub-question of the research shall be derived from characteristic of relevant relational database management systems. Therefore, several relevant relational database management systems types will be listed, introduced, characterized and consequently compared in order to answer the subjected question. This process is to be based mainly on the existing and available knowledge base derived from scientific literature on data science and database management systems in particular. The answer to this research question shall be replied by identification of the most appropriate relational database type for our specific case.

SRQ4: What are the relevant perceivable differences among the non-relational database management systems?

Similar to the previous question, the answer on the fourth fractional or sub-question shall compare the perceivable differences, this time of the relevant non-relational database management systems. Analogically, relevant literature used in the course of this research will create us substantial knowledge base in order to identify the most appropriate, applicable and of course available database type (out of the non-relational database management systems) for the Slovak Investment and Trade Development Agency (SARIO).

After searching for and finally finding the answer to the sub-research questions of this thesis we shall be able to design the main artifact – the database type for the future CRM/BI application of the Slovak Investment and Trade Development Agency(SARIO). This final artifact will be accompanied by the elaboration of the application dashboard design. Since the dashboard design shall create a graphical interface of the CRM/BI application, the relevant CRM/BI systems will be introduced and compared in the subjected chapter of this thesis. Both, the main artifact – the database type – and the dashboard design are to be designed according to the best author´s knowledge, literature, methodologies and the business needs of the environment described above in this chapter.

1.1.3 Weighted Scoring Model

The weighted scoring model method, popular mainly in the project management field, was selected as the third and simultaneously the last method to be used for the purposes of this bachelor thesis. This method – also known as a decision matrix (Romanelli n.d.) – will provide us an assistance in the decision-making process which is to be executed mainly on the fundamentals that are to be found in the process of comparative analysis and design science research. Therefore, the weighted scoring model method is to be of a supportive character in relation to the first two aforementioned methods. In order to use this method in an appropriate manner, certain steps shall be undertaken. These are the following:

- Criteria identification relevant to the project (in our case database design)
- Weight assignment to the identified criteria
- Input record
- Results analysis (Weller 2021)

Since the main priority of this thesis is the database design for a future CRM&BI application for the Slovak Investment and Trade Development Agency, the criteria and their corresponding weight are to reflect the requirements and needs of the agency. Therefore, the criteria and their weight will be shaped according to the aforementioned, and they will be introduced in the corresponding chapter of this text.

1. 2 State of Research on Similar Topics

The importance of a comprehensive and efficient database systems has certainly been a vivid theme the state authorities have to deal with for the several past decades. Majority of the state services, at least, in the so-called more economically developed countries are provided to the state citizens via electronic access what results in the need for a high quality databases and/or database management systems. On the other hand it can be generally anticipated and understood that the process of the concrete state body-related database represents a considerably sensitive process, mainly from the security point of view. Therefore, in the process of the literature research dedicated to or relevant for this particular thesis, one can register a substantial shortage of available researches and scientific contributions presenting a more concrete approach in the process of a suitable database type selection and design, and similar can be claimed on the dashboard design concerning the so-called state-affairs. However the following lines are to mention some of the available and relevant works.

Regarding the fact that this thesis is dedicated to the research of the most appropriate database type for the state agency it is justifiable to take a glance at how the state or government circles have reacted to the spread of electronic databases and database-related affairs. Therefore, a brief insight into the government- or public sphere-related articles on data management and databases seems more than relevant. According to the study elaborated by the British Academy and the Royal Society the most important and by their own words "*overarching principle*" (The British Academy and The Royal Society 2017, 9) of the data governance is the human flourishing promotion (The British Academy and The Royal Society 2017, 9). Although this statement is of a rather sociological character, it is applicable mainly for the public sector. The study provides the readers with several more principles, out of which we can mention "*Anticipate, monitor and evaluate*" principle which includes or which should lead to "*Scrutinise the status quo*" subprinciple (The British Academy and The Royal Society 2017, 60). The application of the right database type shall be of a beneficial character when re-evaluating the current state of affairs and thus to provide a structured information pool leading to the status quo-amelioration. The aforementioned study is rather legal or sociological-centred and thus it lacks the technical or technological aspects of the data management.

The study named "New Horizons for a Data-Driven Economy" edited by José M. Cavanillas, E. Curry and W. Wahlster offers a vast overview on the data storage systems even for the purposes of the public sector. However, the study´s main focus is big data-oriented, what goes beyond the scope of this thesis. Nevertheless, it outlines the NoSQL Databases, NewSQL Databases, Big Data Query Platforms and Cloud Storage possibilities. In relation to the public sector the author of the subjected chapter, R. Munné, points out the economic efficiency and personalisation of public services of the (big) data technologies (Cavanillas et al. 2016, 197).

The importance of the data processing and data management has been of course recognised by the official authorities of several countries. This fact is supported by official guidelines dedicated to the data usage, design and management. Certain countries have even elaborated and are implementing their national data-dedicated strategies. For instance, the United Kingdom has developed its own "National Data Strategy" with an aim to "strengthen our economy and create big opportunities" (The Government of The United Kingdom Department for Digital, Culture, Media & Sport 2020) for the UK. One of the mission mentioned in this official document transformation of the governmental data handling in order to boost efficiency and to enhance the level of the services provided to the public (The Government of The United Kingdom Department for Digital, Culture, Media & Sport 2020), which calls upon creation of an appropriate data infrastructure, what partially corresponds to this thesis. The strategy outlines crucial issues that shall be in the centre of attention when it comes to data, especially data quality, and differently measured benchmarks for data applied in the course of the whole data lifecycle (The Government of The United Kingdom Department for Digital, Culture, Media & Sport 2020). Thus the main emphasis in this UK national strategy is put rather on data than on the data infrastructure – database management system itself. However it provides a valuable insight into the data-related management in the public sector.

The study conveyed by researches F. Sá, P. Martin and Maryam Abbasi named as "Portuguese Local E-government: A Study on the Most Adopted Databases in 2019" offers an overview on the database systems used by more than 100 public sector entities – city halls – in Portugal in 2019. The survey shows that a vast majority of the Portuguese town halls adopted the relational databases. The most frequent used database among the survey respondents was stated to be the "*SQL Server Express*" applied by more than 68% of the (Sá et al 2021, 7). This was followed by the "*Informix*" database, a database developed by IBM (Sá et al 2021, 6). However, the authors notes that "*Informix*" made its mark mainly thank to the fact that the software used by the selected town halls administrations requires application of this particular database (Sá et al 2021, 6). The last medal position belongs to "*MySQL*" database system (Sá et al 2021, 6). Of course, the study mentions other reason of the information technology competent employers of the city halls when making decision for a certain database than just a more-or-less compulsory requirement as it was in case of "*Informix*". According to the answers of the respondents the most frequent and decisive factors were as follows: technical assistance, security, performance (Sá et al 2021, 9). Regarding the "*winner*" of the research, "*SQL Server Express*", the users emphasized price (in connection to the database system), security and the system as such "*global evaluation*" (Sá et al 2021, 8).

To continue within the frame of the public sector sphere, a study "E-Government Databases: A Retrospective Study" composed by Indian researches A.J. Singh and R. Chauchan provides us with an overview and evaluation of fourteen databases belonging to the same number of e-government applications that were developed by the authorities of the Indian State of Himachal Pradesh (Singh and Chauhan 2010, 67). The authors aimed to evaluate the database systems based on their adherence to the core database designing principles – primary key, foreign key, data redundancy, documentation, constraints, transactions handling, E-R design and master data (Singh and Chauhan 2010, 67). This adherence is, according to the authors, crucial for the level of robustness and proliferation of the e-government services (Singh and Chauhan 2010, 72). All of the subjected databases were based on the MS-SQL server. The main insufficiencies revealed by the data were particularly the following: shortage of documentation; shortage of master data; isolation and inconsistency of the data and databases (Singh and Chauhan 2010, 72). However, it is important to note, that these "malfunctions" were not caused by the database type itself. Similarly to the aforementioned databases, the relational model of a database was proposed to be used for the purposes of the Nationwide Information System of India by S.K. Jain, G. Singh and M.M. Gore in their paper called “Database Design for Nationwide Information System to Govern Country Effectively” (Jain et al. 2003, 1).

2. DATABASES

2.1 Databases Introduction

To begin with the second chapter of this bachelor thesis it is more than justifiable to provide an introduction to the databases. Therefore the following line are to outline the databases fundamentals and the fundamental terms that are inevitably and closely connected to the databases. Firstly, it shall be defined what the term “database” represents itself. C.J. Date claims, that “a database is a collection of persistent data that is used by the application systems of some given enterprise” (Date 2004, 11) . Another definition can be derived from the book called “An Introduction to Database Systems” written by B. C. Desai, where he defines a database as appropriate models capturing the diverse classes of objects of interest and relationships among them (Desai 1992, 2) . We can claim that the both aforementioned definitions of a database are based on truth and they refer to the more-or-less modern databases used on certain computer systems. However, from a more historical point of view, the computer-managed database had its predecessors in more traditional forms. Since a database can be simply characterized as an organized data collection, the first predecessors of the today´s modern “organized collection of data” were embodied by the books consisting of a structured and organized facts – data. Fair examples of these predecessors are dictionaries, encyclopaedias, family-trees and event atlases providing graphic storage of data. The next and more cultivated form of databases were born by “the adoption of punched cards and other physical media that could store information in a form that could be processed mechanically”(Harrison 2015, 5). In this case, G Harrison offers an examples stemming from the 19th century as he mentions loom cards that were used to program fabric looms to generate complex fabric patters, and also the perforated paper strips that represented certain melodies (Harrison 2015, 5). As it was already mentioned in the first lines of this thesis, the modern, electronic form of database was adopted in the course of 1950s and 1960s. The two definitions and this brief historical entrée to the world of databases have mentioned, whether directly or indirectly mentioned certain crucial terms that are necessarily intertwined with databases.

There is no database without any data. Therefore, we can claim that data form a fundamental component of a database, since this is an organized collection of data. The term “data” itself is defined by R. Elmasri and S.B. Navathe as “facts that can be recorded and that have implicit meaning” (Elmasri and Navathe 2016, 4). This definition might be widened by a broader data “limitation” claiming that “data are observations or measurements … represented as text, numbers, or multimedia” (United States Geological Survey n.d.). On the other side it is crucial to note, that there is a substantial difference between “data” and “information”. The first term represents raw and unprocessed facts, whereas the latter is “the result of processing the raw data to reveal its meaning” (Coronel and Morris 2016, 4). Those measurements represented as texts, numbers or multimedia are thus processed and stored in databases. However, the organization of a data collection might not be a sufficient criterion for a database, since, according to R. Elmasri and S. B. Navathe, this shall also fulfil the following requirements or so-called “implicit properties”:

- A database depictures certain characteristics of the existing world, that could be referred to as a world prototype in the horizontal or event in the vertical sense. Every single occurrence and/or possible change occurred in this world-prototype is to be appropriately registered in a database
- A database does not represent any coincidental aggregation of data. On the contrary, the term database is considered to “personify” a logical set of data inherently conveying or possessing certain meaning
- A database is structured, constructed and filled up with data in pursuit of a particular intention. It disposes of a “pre-determined” group(s) of users and certain related functioning software solutions used or intended to be used by the same group(s) of users. (Elmasri and Navathe 2016, 4)

Regarding the afore-stated implicit principles, one might claim that, to a certain extent, a database may be regarded as a structured, logically organized information digital twin of the real world. There are not any prescribed or pre-defined size and complexity for a database to be qualified for a database. The decisive qualification for a logically structured collection of data to become or to be regarded as a database is the real world representation by the means of logically structured and organized data, certain communication or information exchange mainly in direction “from real world to data” and of course a database shall be administrated and used by real or computer-aided users in order to fulfil or to follow certain objectives.

2.1.1 Database Management System – “a database approach” and its components

Increasingly larger amount of data and simultaneously larger amount of databases presumably created specific necessity – mother of invention (Plato, 2007) – to effectively, and particularly efficiently, process, control and manage data and databases. Naturally, a manual database creation followed by a consistent manual maintenance and/or management of a database shall be regarded as a feasible option and this had been a reality before the modern computer-aided or computer-based databases arrived on the scene. In this case the mother of invention, necessity, resulted in the “invention” and application of the database management systems generally referred to under the known abbreviation – “DBMS”. The necessity for the invention or the evolution of the database management systems stemmed from the insufficient or from the rather inappropriate (however this statement shall not be regarded as unconditionally applicable for all cases) performance or qualities of their predecessors – the files systems. As its name invoke the principal responsibility of a database management system is to secure and to execute the management of databases. A database management system thus represents a computer-based program or a "collection of programs that manages database structure and controls access to the data stored in the database" (Coronel and Morris 2016, 6). Together with the other fundamental elements – data, hardware, procedures, other software programs and users – a database management systems constitutes an organizational unit called "database system". The first database management systems were evolved from the traditional file systems. Their first application for the business purposes dates back to the 1960´s. (Garcia-Molina et al. 2009, 2). The first business pioneers who decided to apply the DBMS for their purposes are to be found among the representatives of the banking houses, airline reservation systems and also other sphere of businesses and entrepreneurs who wished to store and manipulate their corporate records in this – at that time – innovative way. (Garcia-Molina et al. 2009, 2) Having applied this approach the disadvantages stemming from the conventional file system should have been eliminated, what was and still is the added value of the database management systems. However, the initial appearance or the structure of the DBMS differs from the today´s "DBMS offer". The first years of the DBMS existence were coined by a predominantly application of two data models – the "hierarchical" or tree-based model and the graph-based "network" model, what was changed in the year of 1970.(Garcia-Molina et al. 2009, 2). This is the year when a British scientist called Edgar F. Codd came with his work named as "A Relational Model of Data for Large Shared Data Banks" (Codd 1970), thus provided a "modus operandi" relevant and related to the today´s database systems. The relational data model found its competition in other, "younger" or more recently described or developed data model, which will be briefly described and commented on later in this introduction.

Regarding the fact, that even manual management of a database disposes of certain advantages it seems justifiable to take an insight into the main functions and benefits with which the so-called "database approach" (Elmasri and Navathe 2016, 4) might contribute. The function of the core – the DBMS – of a database system can be characterized as follows:

- To allow the users to create new databases and specify their schemas (logical structure of the data), using a specialized data-definition language
- to allow the users to query and modify the data using a query language or data-manipulation language
- to support the storage of very large amount of data – terabytes or more – over a long period of time, allowing efficient access to the data for queries and database modifications

- to enable durability, the recovery of the database in case of failures, errors or intentional misuse
- to control access to data from many users at once, without allowing isolation and atomicity (Garcia-Molina et al. 2009, 1-2)

Regarding the DMBS functions outlined in the previous lines, it is obvious, as it was already stated, that the main aim of the DBMS is an effective and efficient database management. The “struggle” aiming to achieve this purpose shall be perceived in the course of the database life along the following states of affairs that were defined by C. Coronel and S. Morris as the advantages of the DBMS:

- improved data sharing
- improved data security
- better data integration
- minimized data inconsistency
- improved data access
- improved decision making
- increased end-user productivity (Coronel and Morris 2016, 8)

Presumably these advantages are the main reason why the organizations make decisions to implement the so called “database approach” and to keep aloof from the traditional “file system approach”. These advantages can be executed thank to the two distinct types of commands – DDL data definition language, and DML data manipulation language (Garcia-Molina et al. 2009, 5). Data definition language are used by a database administrator to modify the database schema, whereas data manipulation language affects the database content (Garcia-Molina et al. 2009, 5). However, to provide a more complex view, it is of high importance to mention, that the file system approach disposes of its justification, considering the disadvantages of the “database approach” or the situations and/or circumstances under which it shall be more beneficial to apply the file system approach. For that reason R. Elmasri and S.B. Navathe introduced the following “overhead costs” of the DBMS application resulting from the following substances:

- high initial costs, investment in hardware, software and training
- a wide and generous space provided by the DBMS for the data definition and processing
- high costs and overhead for providing security, concurrency control, recovery and integrity functions (Elmasri and Navathe 2016, 27)

Therefore, the managers or the people in control of or responsible for shall consider the pros and cons of the both database and file system approaches, considering the specific conditions and environment of a particular project. Otherwise, a nonideal decision might lead at least to a considerably heightened cost volume.

Shall a decision in favour of the database approach be made, than a “confrontation” with an entity named as “database system environment” is unavoidable. This term was already mentioned in the lines of this thesis, referred to as “database system”. Its substantial parts or components are data, hardware, procedures, software and users also referred to as “people”, whereas the users are characterized as a sub-group of people. Also this distinction will be visible in the following lines where the components and their possible sub-groups will be introduced:

- Data – already characterized as raw material, thus different from information, since this raw material – data – are used for the information generation
- Hardware – this term refers to all system-related physical units and/or devices including computers, storage devices, printers, network devices (hubs, fiber optics, etc.) and other physical devices (automated teller machines, ID-readers, etc.)
- Procedures – described as instructions and rules of a subjected organization that govern the design and use of the applied DBMS
- Software – under this term the three types or categories of software are covered, firstly the “operating system software” managing all hardware components, such as Linux, Mac OS, Microsoft Windows etc.; second the “database management system software” bearing responsibility for the data management, such as DB2, MySQL, Oracle Database PostgreSQL or SQLite; the third group is represented by programs that might be characterized as “utility software” which bear responsibility for providing access to the data stored in the DBMS and also for provision of tools dedicated to the data manipulation, therefore they highly contribute to the management of the DBMS environment
- People – this component is further divided by C. Coronel and S. Morris into 4 subsections: system administrators – accountable for the database system´s general operations; database administrators – responsible for a smooth functioning of the database system; database designers – in charge of the database design structure; system analysts and programmers – bearing responsibility for the creation of the interface for the end-users in order to be able to manipulate with the database content, with data; end users – the people for whom the database was made and who use the application program perform the specific daily operations (Coronel and Morris 2016, 23-24)

2.1.2 Data Model(ling)

The so-called raw material stored in another component of a database system – the data – is put into the database or databases according to certain pattern or specific prescribed, exerted practices. These steps are to be executed in the phase called “database design”. To be concrete the process where these steps are being executed is referred to as a “data modelling”. Data modelling can be defined as an iterative and progressive process the result of which is a creation of a specific data model providing a solution to a specific problem domain. (Coronel and Morris 2016, 36) One shall bear in mind, that the aforementioned specific problem domain shall be rigorously defined in order to create the most accurate data model possible that would reliably display the real environment of a real problem domain. Therefore, there is an assumption that a problem domain shall dispose of its concrete and defined environment, its scope and its limitations. Thus, data model can be defined a mostly graphical representation of a complex real world data structures (Coronel and Morris 2016, 36). The main purpose of the data modelling and of the data models is to provide an understandable view on a real environment with a higher degree of complexity. A high-quality data model is considered to be a “blueprint” to build a proper database system meeting all the customers´ requirements (Coronel and Morris 2016, 36). Another important aspect of a data model is its feasibility to provide and complex overview of the situation and thus to encompass or to create a consolidate model based on higher number of view or angles. We have already mentioned the first data models that were applied just after the “inventions” of the DBMS, and to provide a more complex view there is a list of the relevant data models known today: hierarchical, network, relational, object-oriented, object-relational, XML hybrid DBMS, key-value store, column store (Coronel and Morris 2016, 43). Disregarding which model is being applied, the data designer will have to cope with the following terms: attributes, constraints, relationships and entities. These terms are also named as “basic building blocks” of a data model (Coronel and Morris 2016, 37-38). Since as it was mentioned, these terms are inherent for to all data models, we will define and/or discuss these terms in a specific part of this text that will be dedicated to the relational data model, where also more concrete information related to relational databases will be provided.

2.1.3 Sets of logical operations and their “ACID” role

The data model basic building blocks and the databases as such are necessarily related to operations, more concretely to the logical operations. The logical operations are also called “CRUD” operations according to the activity they execute. “CRUD” is an acronym of the for principal logical operations that are used to modify the database content – create, read, update and delete. The “CRUD” operations provide a basis for a transaction process. In the context of databases a transaction can be regarded as a set of database logical operations (Juba et al. 2016, 6) and/or also as a logical unit of work that is to be entirely completed or entirely aborted (Coronel and Morris 2016, 485). Transactions are especially important for the transactional databases, as for instance a database of a successful web-shop, that is expected to be under a constant change or modification processes. An example of a transaction process might be embodied by a single ticket purchase for an event via website, which triggers certain modifications in the subjected database, such as change of the tickets (products) available or adding a new customers to the database. It may be regarded, that transactions shall fulfil the role of a database-stabilisation or consistency mechanism, but this is only one out of four qualities or principles prescribed for transactions. These principles are known under the "ACID" abbreviation. The four letters depicture the same number of the transaction principles or properties – atomicity, consistency, isolation and durability (Date 2015, 295). Atomicity stands for the "all or nothing", what evokes that all the transaction parts or steps are to be executed correctly, otherwise a transaction is regarded to be failed. The consistency transaction property stipulates that a transaction transforms a consistent state of the database into another consistent state, so one may claim that this property could be named also as an integrity (Date 2015, 295). Regarding the consistency or integrity property, a decisive role is assigned to the database constraints definition which will be discussed in more detail in a part dedicated to the relational databases. Under isolation we understand the fact that a subjected transaction results are isolated from all other transactions, until the time when the subjected transaction commits (Date 2015, 295). The last property – durability – stands for persistence of the successfully executed transactions, even if any crash shall take place, the transaction result shall be stored in a database, what is secured by a technique named as "write-ahead log" also abbreviated as "WAL" (Juba et al. 2016, 6).

2.1.4 Classification of Databases

Considering the need for a provision of a more complex overview to the database systems it seems appropriate to provide the readers with a basic classification of the databases or database types that can be managed by the database management system. As it was several times mentioned, the modern database management systems had emerged in the 1960s for the first time, and it is natural that since that time a solid number of database patterns (in the sense of "types") have been "invented" and applied. This is also applicable for the number of criteria according to which the databases can be classified today.

Firstly, it is fundamental to realize that the databases are being structured to serve the end-users purposes. Therefore, the first, basic criterion of classification is the number of users that are permitted/ able to use the database at the same point of time. In this case there is a simple division into two subgroups – single-user databases – where only one user is permitted/is able to use the database and the other potential users have to wait until the first actual user ends his activity, and multiuser databases. A multiuser database remains available for a simultaneous use by two and more users. This database type is divided into two subgroups – workgroup database, usually used by a specific organization department up to 50 persons/users, and enterprise database, usually simultaneously used by more or all organization departments (Coronel and Morris 2016, 8).

The second classification of database types relies on the number of places or locations where the data stored and managed in the database is stored. From this point of view we classify a centralized database, where the data is stored in one location, and a decentralized database supporting data being distributed across several locations (Coronel and Morris 2016, 8).

The character of data stored and managed in a database is decisive when it comes to division into “general-purpose database” and “discipline-specific database” (Coronel and Morris 2016, 9). As the names indicate, the former stores general information or information mixed from various resources, whereas the letter is filled with data on a specific topic, as for instance a meteoroid observation research.

The data stored in the databases can be regarded as the core of the databases. Since the data character, their structure and storage can dispose of different forms, there is a classification of databases based on the structure of the data stored in the particular database. In consonance with this classification we can divide databases on the ones with unstructured data, structured data and semi-structured data(Coronel and Morris 2016, 9). The first and the second group stemming from this classification, the unstructured data and the semi-structured data are in the centre of the interest of the new databases generation named as “extensive markup language databases”, or shortly “XML databases” (Coronel and Morris 2016, 10). Regarding the fact that the XML databases use different approaches when requesting and manipulating the data from what is used to in case of the relational models, the XML databases constitutes a group of the so called “Not only SQL” systems, known also as “NoSQL”.

Last but not least, according to some authors the most popular databases classification is simultaneously based on a database purpose and on the time sensitivity of the information processed from a database (Coronel and Morris 2016, 9). In this regard we divide databases into two following subgroups: transactional database, well-known under the name of “OLTP database”, online transaction processing database, and analytical database, primarily using the “OLAP tools”, online analytical processing tools. For the first subgroup the real-time transactions are of cardinal importance, since these databases are used in the fields as financial transactions processing or purchase processing. Therefore, the data have to be processed promptly, ideally without any time delays, and with the highest precision possible. The latter subgroup, the analytical databases are usually composed of two parts – data warehouse and online analytical processing front end (Coronel and Morris 2016, 9). The data are stored in the warehouse, which provide a basis for a decision support, and the online analytical processing front end provides tools for an advanced data analysis, thus constituting a new scientific field named “business intelligence” (Coronel and Morris 2016, 9).

2.2 Relational Databases

In correspondence to the name of this chapter this section of the bachelor thesis is to widen the information or the knowledge on the general topic of databases by providing information related to the relational model of data or on the relational databases.

The relational model had caught the light of the world thank to the British scientist, at that time an employee of the IBM company, Edgar F. Codd in 1970 when he published his paper named “A Relational Model of Data for Large Shared Data Banks” (Codd 1970). In this work he had put the fundamentals of the relational model, a model which is based on two main mathematical theories – predicate logic and set theory(Coronel and Morris 2016, 73). We can claim that these parental theories gave birth to the three fundamental components of the relational model:

- A logical data structure represented by relations
- An Integrity rules set that shall secure the data consistency over the database life-cycle
- Operations set that defines the data manipulation (Coronel and Morris 2016, 73)

The aforementioned components, together with the two formal query languages connected to the relational model – the relational calculus and the relational algebra, were firstly put into commercial practice in the 1980s, by the companies called IBM and Oracle (Elmasri and Navathe 2016, 149). The relational calculus is regarded as a fundament of the SQL language, whereas the latter, the relational algebra is applied in databases for the purposes of query processing and optimization (Elmasri and Navathe 2016, 149).

In general, a relational database system consists of databases constituted of logical concepts – tables – representing a relations collection (Elmasri and Navathe 2016, 149; Coronel and Morris 2016, 8) .

2.2.1 Basic “Relational” Terminology (Visualized)

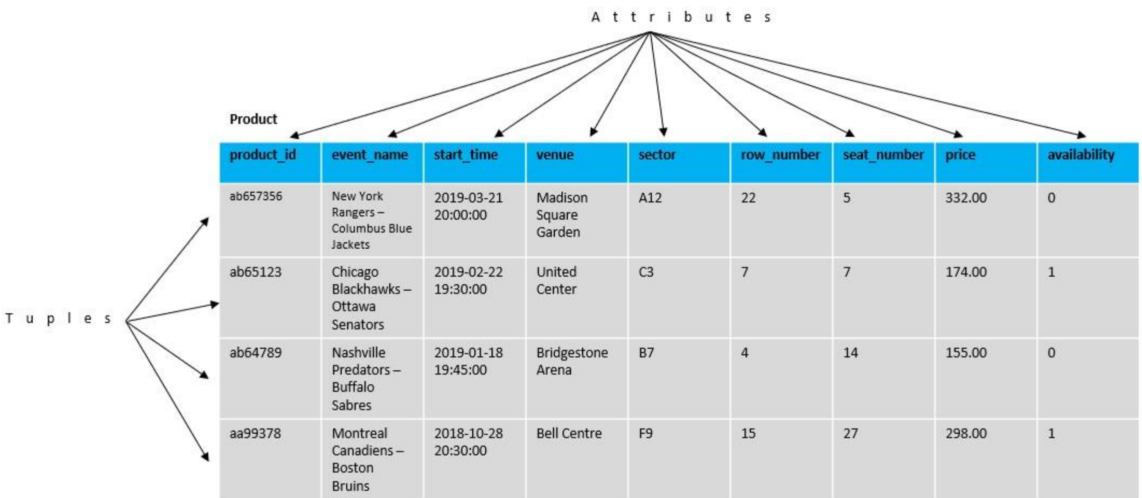
The introductive part on databases of this thesis provided a list of term related to the databases as such, what is, of course, applicable also for the relational ones: attributes, constraints, relationships and entities. The following lines will be dedicated to a more detailed view or a description not only of the aforementioned terms, but also of a terms that are even more associated with relational databases.

The first so-called basic building block of (not only) relational data model is entity. An **entity** may represent a person, place, event or any other object that can be stored in a database (Coronel and Morris 2016, 37). Disregarding if we are talking about person, place, event or other object, or other entity, in the database terminology, we can state, that every entity disposes of certain qualities that makes it different from other entities, therefore an entity are of a unique nature. These qualities are – in case of databases – embodied by **attributes**. The attributes represent certain qualities of an entity. For instance a database storing and manipulating data of any e-commerce entrepreneur certainly stores entity named "customer". Conventionally a customer represents a person who disposes of a first name and a surname. These two data on customer, his/her first name and surname represents in this case the attributes of the customer entity. The entities might be somehow connected between each other. These connection, or associations are called **relationships**, or sometimes referred to as relations (Coronel and Morris 2016, 38). A relationship or a relations can be illustrated with the help of a customer and a product – for instance a ticket for an event. Whereas a customer might be able to buy as many tickets as there are at disposal, a concrete ticket can be purchased exactly by only one customer. Of course, we could elaborate more examples, where the associations will be of a various character. Therefore, there are three types of relationships, when it comes to (relational) databases: one-to-many (displayed as "1:M "or "1..*"); many-to-many (displayed as "M:N" or "*..*"); one-to-one (displayed as "1:1" or "1..1") (Coronel and Morris 2016, 38). The examples for the two remaining relationship types might be formulated as follows: M:N – a customer can attend many events and a single event can be attended by many customers; 1:1 – a player of a hockey team disposes of his/her specific contract, and the same is applicable for the contract since this can be signed with/by or related to only one player. Besides the afore-stated relationship types it is vital to note, that all of this relationships are bidirectional (Coronel and Morris 2016, 38), that means the relationship may be "pursued" in both directions, as it was illustrated using the concrete examples for each relationship types. In order to keep certain database integrity of a constraint or constraints shall be defined on the particular data. A constraint can be characterized as a restricting rule assigned to certain data (Coronel and Morris 2016, 37). As an example we can mention a rule stipulating that a primary key value must not equal NULL. The previous sentence contains at least two terms that also belong to the inherent terminology of the relational databases and that will be introduced together with the other essential terms.

The relational model is based on the existence of tables, that are defined or named as relations, since the author of the relational model, Edgar F. Codd used these two terms as synonyms (Coronel and Morris 2016, 74). Moreover, the tables are also referred to as "entity set", since a table, or a relations represents a group of related entity occurrences. (Coronel and Morris 2016, 74) The tables of the relational model are two-dimensional. Naturally these relations or – for a better imagination – these tables consist of rows and columns. However, in the relational modelling terminology the table rows and columns are called substantially differently, as they represent certain values. The rows of a relations contain the aforementioned entity occurrences, also defined as records, and "more officially" as tuples (Coronel and Morris 2016, 74; Elmasri and

Navathe 2016, 151; Garcia-Molina et al. 2009, 5). In order to achieve certain terminological consistency and/or integrity of these thesis we will define a row of a relation as a tuple. A column of a relation is defined as an attribute, that was already characterized. The aforementioned tuples and attributes are populated by the values of a certain domain (Elmasri and Navathe 2016, 151). The domains shall secure the atomicity quality of a relation, since o domain represents a set of atomic values, what implies indivisibility of the domain values (Elmasri and Navathe 2016, 151). In order to define a domain one shall specify a data type of the values that are to be included in a particular domain (Elmasri and Navathe 2016, 151). That means that the values of the tuples must satisfy specific criteria of a certain elementary data type (Garcia-Molina et al. 2009, 23). Besides the data type, a domain values can dispose of a specific prescribed data format. All the information, besides concrete tuples, on the aforementioned terms are to be represented in a relation schema. This can be simply defined as a set of attributes of a relation (Garcia-Molina et al. 2009, 23), since a relation schema consists of a relation name and a list of its attributes (Elmasri and Navathe 2016, 151). Thus a relational schema describes a relation, its attributes that are part of a certain domain (Elmasri and Navathe 2016, 151). The number of the attributes defined in a relation schema represents the degree or arity of the subjected relation (Elmasri and Navathe 2016, 151).

In full consonance with the international proverb sounded as "It´s better to see something one, than to hear about it a thousand times", in our case to read about it a thousand times, and in order to provide a better understanding of the afore-stated terms and their definitions a "humble" visualization accompanied by a description is to be provided.



product_id	event_name	start_time	venue	sector	row_number	seat_number	price	availability
ab657356	New York Rangers – Columbus Blue Jackets	2019-03-21 20:00:00	Madison Square Garden	A12	22	5	332.00	0
ab65123	Chicago Blackhawks – Ottawa Senators	2019-02-22 19:30:00	United Center	C3	7	7	174.00	1
ab64789	Nashville Predators – Buffalo Sabres	2019-01-18 19:45:00	Bridgestone Arena	B7	4	14	155.00	0
aa99378	Montreal Canadiens – Boston Bruins	2018-10-28 20:30:00	Bell Centre	F9	15	27	298.00	1

Figure 3: A relation named "Product" with its tuples and attributes. Inspired by a table elaborated by Elmasri and Navathe (Elmasri and Navathe 2016, 153).

This picture displays a relation called “Product”, and the relation as such represents a product of imagination of the author of this thesis and an example provided by R. Elmasri and S.B. Navathe. The table shall represent one relation of a database system that deals with the products – the tickets for the National Hockey League games. Of course, the data were fabricated and as it was written, they represent a product of imagination exclusively, and thus are not based on any real data. The relation displayed obeys the terminological conventions for relations and attributes of the relational model. These stipulates that the relation name shall begin with a capital letter, whereas the attributes names shall be written using the lower case (Garcia-Molina et al. 2009, 23). This relation provides data that generate information on the product. There are nine attributes in total, what implies, the degree or the arity of the relations that equals the same number – nine. The relation contains four rows (besides the first row filled by the attributes´ names) what indicates that the Product relation is consisted of four tuples, and each of them is made of nine attributes, and they can be presented as follows:

(ab657356, New York Rangers – Columbus Blue Jackets, 2019-03-21 20:00:00:00, Madison Square Garden, A12, 22, 5, 322.00, 0)

The example of the first tuple shows, that the tuples can be represented surrounded by the parentheses, or round brackets, whereas their components are separated by commas (Elmasri and Navathe 2016, 154). The tuples of the Product relation are populated and thus created by certain values of the attributes. In this case we can pretend that certain domains were assigned to the attributes. For instance it can be claimed the following:

- “product_id”: a data type “”VARCHAR(8)” was assigned to the attribute
- “event_name”:a data type “VARCHAR(50)” was assigned to the attribute
- “start_time”: a data type “DATETIME(fsp)” was assigned to the attribute
- “venue”: a data type “VARCHAR(50)” was assigned to the attribute
- “sector”: a data type “VARCHAR(5)” was assigned to the attribute
- “row_number”: a data type “SMALLINT()” was assigned to the attribute
- “seat_number”: a data type “SMALLINT()” was assigned to the attribute
- “price”: a data type “DECIMAL(7,2)” was assigned to the attribute
- “availability”: a data type “BOOLEAN” was assigned to the attribute

Based on this data type assignment we can enrich the following Product relation schema:

Product(product_id, event_name, start_time, venue, sector, row_number, seat_number, price, availability)

… by the information of the data types assigned to the domains of the attributes, what would result in the following form:

Product(product_id:varchar, event_name:varchar, start_time:datetime, venue:varchar, sector:varchar, row_numbet:smallint, seat_number:smallint, price:decimal, availability:boolean)

The example provided by a relational table also show us how the relation, or the table "Product" represents facts about certain entities, in this case about products, and eventually the next table, the next relation could provide us the information on the customers who bought a particular product. This would mean, that the table would provide us information on the product relationship to a certain customer. This short excurse was just to demonstrate how the relational model provides information on both facts and relationships on/of entities, uniformly named as relations (Elmasri and Navathe 2016, 156). It is also fair to note, that the data types imaginarily assigned to the attributes are being used in the MySQL database, and the data types of the other relational databases indicates and/or can indicate slight differences.

2.2.2 Essential SQL Statements

The lines above were devoted to the terms as relation/table, domain and attributes which play a crucial role in the relational databases. However, form a practical point of view – of course after a rigorous analysis – firstly, it is essential to create a relation/table. This is possible thank to the "CREATE TABLE" SQL statement. This is to be demonstrated using the example of a table displayed on page 30:

```
CREATE TABLE Transfer (
transfer_id VARCHAR(8),
transfer_date DATE(),
team_id VARCHAR(1)
transferred_to VARCHAR(40),
player_id VARCHAR(6)
);
```

Using the aforementioned statement one is able to create a relation named "Transfer" with the displayed attributes of respective data types. Taking a glance at the table we can see, that it disposes of a unique identificator "transfer_id" called as "primary key". This is to be declared as follows:

```
transfer_id VARCHAR(8) PRIMARY KEY;
```

If there was a wish to define a primary key, the value of which could not equal "NULL", the following statement would be applied:

```
transfer_id VARCHAR(8) NOT NULL PRIMARY KEY;
```

Shall a new attribute be added to the relation, one shall use “ALTER” and “ADD” statements. The following example demonstrates adding an attribute called “manager_id”.

```
ALTER TABLE Transfer ADD manager_id VARCHAR(1);
```

Similarly, one may be interested in deleting certain attribute what is possible thank to “DROP” statement:

```
ALTER TABLE Transfer DROP manager_id VARCHAR(1);
```

Shall we just delete the data entries, and not the table/relation itself, “TRUNCATE” statement shall be applied:

```
TRUNCATE TABLE Transfer;
```

Shall a table as a whole be deleted, “DROP” statement is to be applied again:

```
DROP TABLE Transfer;
```

2.2.3 Fundamental Data Types

Since the aforementioned example demonstrates datatypes that are typical of MySQL database, it is beneficial to provide a short introduction on the general data types that shall be assigned to the values of the relation attributes using the SQL language. Therefore, in general the SQL data types can be divided into the following three groups:

- Numeric – intended for integer number of various size, such as “INTEGER” or “INT” and “SMALLINT”, further intended for floating-point numbers of various precision such as “FLOAT”, “REAL”, “DOUBLE PRECISION”, whereas it is possible to declare the formatted numbers by using “DECIMAL (i,j)”, “DEC(i,j)” or “NUMERIC(i,j)”, where “i” refers to precision (total number of decimal digits) and “j” stands for the scale – the number of digits after the decimal point
- Character-string – can be of fixed and of varying length. The examples of the former are “CHAR(n)”, “CHARACTER(n)”, and the latter is represented by “CHAR VARYING(n)” or “CHARACTER VARYING(n)”. The “n” withing the brackets represents the number of characters in the first case, whereas in the second case it refers to the maximum number of characters.
- Bit-string – similar to the character-string data types, this can be also of fixed length – “BIT(n)”, and of varying length – “BIT VARYING(n)”.
- Boolean – the values TRUE and FALSE are present. However also a type “UKNOWN” is possible.
- Date – the basic DATE data type consists of eight positions “YYYY-MM-DD”, the “extended” DATETIME is comprised of “YYYY-MM-DD HH:MM:SS” (Elmasri and Navathe 2016, 183)

The afore-stated SQL data types shall not be considered as an exhausted list of the SQL data types, since SQL offers a higher number of data types that are beyond the scope of this text.

The terminological basis introduced, described and demonstrated by a simple example shall not be missed in the knowledge base of any database administrator, designer, analyst or programmer who deals with relational databases. This statement is also applicable for the term or better said a technique named as "normalisation", the form of which will be introduced in the next sub-chapter.

2.2.4 Normalization

The term "normalization" represents a technique used in the process of a database design that is common to the relational model, but as mentioned, normalisation is rather connected to the database design than to the relational model (Date 2015, 292). The aim of a normalisation is to minimize the data redundancy and dependency and/or to eliminate the three database anomalies – insertion, update and deletion anomaly (Oppel [2005] 2012, 8). We can simply characterize the normalization as "the process of replacing duplicate things with a reference to the original thing." (Date 2012, 5) It shall be considered as important to note that several different theories and/or discussions on normalization can be find among the books and other scientific pieces dedicated to the databases and to the database design theory. Therefore the readers may might be confronted with a higher number of normal forms (as for instance seven) (Elmasri and Navathe 2016, 476), than it is intended to be presented in the following lines. These will be devoted to a three-scale normalization process identified by Boyce and Codd, thus its name "Boyce-Codd normal form" or shortly "BCNF" (Elmasri and Navathe 2016, 476).

<u>Zero and The First Normal Form</u>

A relation or a table is usually "born" or at least firstly designed in the so called "zero" normal form. That means the subjected relation is not considered to accomplish the conditions or rules of the first normal form. These can be formulated as follows: A relation is in the first normal form if and only if all its tuples contain exactly one value for each attribute (Date 2012, 37). That means an attribute, in other word a column cannot contain multiple values. An example of a zero-normal form relation is provided by the following table:

Transfer		
transfer_date	transferred_ to	player_id
12.10.2020	Pittsburgh Penguins	pc5678, pc8761
15.10.2020	Winnipeg Jets	pc9870, pc2341

Figure 4: A relation named "

Transfer"demonstrating the "zero-normal form".

The table proposed in the example named "Transfer" is in the so-called zero form. The reason is hidden under its attribute "player_id" which stores multiple/redundant values. Therefore, in order to "move" this table to the first normal form it is necessary to eliminate the multiple/redundant values of the attribute the following actions are to be undertaken:

- Create a unique identificator, named as "(primary) key" for each data set
- Remove the repeating set of data or a multiple-value attribute to a new table (however, also a conversion of a multiple-value attribute to a higher number of rows – tuples is possible) (Oppel [2005] 2012, 18)

Transfer			
transfer_id	transfer_date	transferred_to	player_id
tn438743	12.10.2020	Pittsburgh Penguins	pc5678
tn445763	12.10.2020	Pittsburgh Penguins	pc8761
tn567543	15.10.2020	Winnipeg Jets	pc9870
tn569321	15.10.2020	Winnipeg Jets	pc2341

Figure 5: A relation named "Transfer" as a demonstration of the third and thus also of the second and the first normal form.

Having created this relation we achieved the first normal form, since the multiple or redundant values of the attribute "player_id" were eliminated, or one may say they were "converted" into more tuples, more rows.

The Second Normal Form

The next step in the process of normalization shall be achievement of the second normal form. However, regarding the previous relation example this step is not really needed, since the subjected table already achieved the second normal form, and we are to explain why and how. The whole theory of normalization is connected to the terms of functional dependencies, and therefore the same can be claimed on the second normal form (Date 2012, 40). The term "functional dependency" can be described as follows: Let "A" and "B" be the attributes of a relation "R", then the attribute "B" is functionally dependent on the attribute "A" if and only if the attribute "A" determines the attribute "B", so it can be claimed that the attribute “B” is determined (only) by the attribute “A”, or the attribute “A” determines the attribute “B” (Oppel [2005] 2012, 20). In other word a relation or a table has to dispose of a unique identificatory, a unique, so called “primary key” which determines other attributes. In case of our table the unique identificatory is embodied by the “transfer_id”, since this determines the whole tuple, and we can unambiguously claim, that for instance transfer_id of value “tn438743” stands for the transfer dated to October, 12th 2020 of a player with player_id “pn5678” to the team called “Pittsburgh Penguins”. It would be different if we had a relation called “Contracts”, which would contain the attributes “contract_id”, “copy_id”, “transfer_date”, “transferred_to”, “player_id”, since the concrete copy could not be defined only by the attribute “contract_id”. This would be possible only in “cooperation” with the attribute “copy_id”, which would be functional dependent on the attribute “contract_id”. It is also crucial to note that if a relation is in the second normal form, it is simultaneously also in the first normal form. Analogically, this is also applicable for the relationship “third normal form -> second normal form -> first normal form”.

The Third Normal Form

The third normal form is again based on a dependency. However, this time the main focus is put on the “transitive dependency”. An attribute “C” is transitively dependent if it is dependent on other attribute which is not defined as a primary key of the relation (Oppel [2005] 2012, 20). If we take a look on the table mentioned in the part of the text dedicated to the first normal form, we, as it was stated clearly see, that this table is already in the second normal form, since the attributes are determined by the primary key “transfer_id”. The question is if a transitional dependency is also presented in the table or not. And the answer is negative, in fact the relation “transfer_id” does not contain any attribute that would be dependent on any other attribute apart from the primary key. Therefore, the relation mentioned is also in the third normal form, since it fulfils the conditions adhered to the first, second and third normal form. However, to provide a more complex view, an example of the second normal form state was elaborated:

Transfer				
transfer_id	transfer_date	team_id	transferred_to	player_id
tn438743	12.10.2020	1	Pittsburgh Penguins	pc5678
tn445763	12.10.2020	1	Pittsburgh Penguins	pc8761
tn567543	15.10.2020	2	Winnipeg Jets	pc9870
tn569321	15.10.2020	2	Winnipeg Jets	pc2341

Figure 6: A relation named “Transfer” demonstrating the second and thus simultaneously the first normal form.

As we can see a column, or an attribute called “team_id” was added to the original version of the “Transfer” relation. This new attribute brought the element of transitive dependency, more concretely to the attribute “transferred_to” which is now transitively dependent on the newly added attribute “team_id”. In this case the transitive dependency can be now eliminated by a simple removal of the both subjected attributes – “transferred_to” and “team_id” to their own table/relation. The mode or the order of the tables representing the respective normal forms by this thesis might be confusing. However, this was only to demonstrate that the third normal form is achievable with a single step.

2.2.5 Concrete Relational Databases

Since the final solution of the most suitable database type shall be of as economically effective and efficient nature, we free or also called open-source databases will be introduced. The first one is represented by “MySQL”, whereas the second is known under the name “PostgreSQL”.

2.2.5.1 MySQL

MySQL represents currently the most popular open-source relational database management system developed and distributed under General Public License by Oracle Corporation (MySQL 2022). However, if needed, a commercial license is also available. MySQL constitutes a client/server environment with a multi-layered server design composed of independent modules, it is written in “C” and “C++” and disposes of “a very fast thread-based memory allocation system” (MySQL 2022). Its high speed shall be also fostered by a fast B-tree disk tables (MyISAM) with index compression, execution of very fast joins via application of optimized nested-loop joins, optimized and fast class library and also by the possibility of application of the temporary in-memory hash tables (MySQL 2022). Reliability of MySQL was tested by several compilers such as “Purify” and “Valgrind”. Security is supported by an encrypted password while connecting to the server, by a flexible privilege and password system enabling host-based verification (MySQL 2022). MySQL can be run on a wide spectrum of various platforms (and their modifications), as for instance the platforms of Apple OS X, Linux, Microsoft Windows and Ubuntu. MySQL is mainly designed for a high volume of data, vast databases containing even 50 million records and supporting up to 64 indexes per table (MySQL 2022). Connectivity is also provided by APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby and Tcl so that the clients can be written in many languages (MySQL 2022). MySQL offers several client and utility command-line (such as “mysqldump” and “mysqladmin”) and graphical (for instance “MySQL Workbench”) programs. Support, help or assistance are provided by for instance in the “mysqlcheck” client (MySQL 2022).

2.2.5.2 PostgreSQL

PostgreSQL represents an object-relational database management system written in C that holds its name since 1996 (Juba et al. 2016, 32). Its popularity among the relational DBMS is not coined by a vast application only, but also by a statement that PostgreSQL “is quite possibly the finest open source RDBMS available”(Perkins et al. 2018, 9). The roots of PostgreSQL come from University of California and currently it is being developed and distributed by the PostgreSQL Global Development Group (Perkins et al. 2018, 10). Similar to MySQL, PostgreSQL represents a client/server environment, but on the contrary to MySQL, besides relational – SQL querying it support non-relational – JSON querying too (AWS Amazon Web Services n.d.). The structure of PostgreSQL consists of a shared memory, background processes and a data directory (Kinsta 2022). It is also of a multi-platform-character, thank to which it can be used on various platforms such, as for instance Linux, Microsoft Windows, Apple OS X and Solaris. PostgreSQL supports multiple programming languages such as C, C++, JavaScript, Python and Ruby. It disposes of a high number of features and supports various data types, and both of them can be extended, since PostgreSQL is of extendible character, what in connection to its “total” open source character means, that everyone can create his or her own features and own data types. Nevertheless, PostgreSQL itself supports advanced data types. PostgreSQL is ACID-compliant, it offers fair recovery possibilities – “PITR” Point-In-Time Recovery that enables to restore data to a concrete moment, and every database change is recorder by “WAL” – write ahead log (IBM n.d.). Data inconsistency is fought against using a multiversion model “MVCC” multiversion concurrency control (PostgreSQL 2022). Restoring capabilities are supported also by a tool called “pgBackRest”. Security of PostgreSQL is based on the following three pillars: network-level security, transport-level security and database-level security (Satori Cyber 2021).

2.2.5.3 MySQL vs PostgreSQL Comparison

Considering the fact that the number of comparison studies devoted to differences between the introduced relational DBMS, MySQL and PostgreSQL is fairly limited and/or fairly outdated (some comparative studies were made in the beginning of this millennium and certain articles also emerged in the recent years, but they provide rather an exclusive focus on performance) while the both DBMS have been constantly evolving, and the second fact that the data experience of the thesis author is not sufficient, the following content displayed mainly in tables will be largely based on the information gained from the websites focused on data and data management.

Aspect	MySQL	PostgreSQL
Provider	Oracle Corporation	PostgreSQL Global Development Group
License	Open source	Open source
Language(s)	C, C++	C
Compatible operation systems	FreeBSD, Linux, MacOS, Open Source Build, Oracle Solaris, Windows	FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows
Support for APIs	Ada, C, C#, C++, D, Delphi, Eiffel, Erlang, Haskell, Java, JavaScript, Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme, Tcl	.Net, C, C++, Delphi, Java, JavaScript, Perl, PHP, Python, Tcl
ACID compliance	Yes, partial	Yes
SQL compliance	Yes, partial	Yes
Extensibility	No	Yes
Data Types	SQL - Standard	Advanced Data Types (e.g. arrays, user-defined data types)
Queries	SQL-Standard based	SQL-Standard based
Indexes	Hash Indexes, B-Tree Indexes, R-Tree Indexes, Inverted Indexes	Hash Indexes, B-Tree, R-Tree, Partial Indexes
Performance	Dependent; OLTP-read-heavy-processes-appropriate	Dependent; OLAP-concurrent-write-processes-appropriate
Referral Integrity	Foreign Keys	Foreign Keys
Replication	Multisource replication, source-replica replication	Source-replica replication
In-memory capability	Yes	No
Consistency	Immediate	Immediate

XML support	Yes	Yes
Storage	Multiple	Single
Security	Access Control Security (ACL), Encrypted Connections	Access Control Security (ACL) Multiple Encrypted Connections
Popularity	Stack overflow: 656 649 questions till 11.01.2023 Google: 44 500 000 results in 2022	Stack overflow: 164 836 questions till 11.01.2023 Google: 15 400 000 results in 2022
User-Friendliness	Suitable even for less-experienced users	Suitable for rather more experienced users
User Support Availability	Vast Possibilities of Free User-Support by Community; Vendor Support	Limited but also Vast Possibilities of Free User-Support by Community

The comparison table is focussed on the newest versions of the compared entities and it was elaborated based the following resources:

(DB-Engines 2022a; Devart n.d.; Luzanov 2016; Smallcombe 2022; Chen 2021; Revoof 2022; Phaujdar 2021; EnterpriseDB 2019; Vinchin 2022; Kakarla 2020).

2.3 “NoSQL Movement”

This chapter is to provide information on the “opponent” of the relational databases, the databases that can poses a threat for, or definitely at least are able to question the relevance of the relational databases application in certain cases – the non-relational databases. The or the outbreak of the “NoSQL” movement and/or databases was an answer to the increasingly higher demands of the commercial players involved in the “Web 2.0” mainly after the beginning of this millennium. Those demands were certainly affected by an unprecedent massive flow of rather unstructured data, the processing of which could be of a considerably higher time- and effort-consuming character what led to the adoption of the NoSQL distributed systems or databases. However the term “NoSQL” was coined by Carlo Strozzi in 1998 as he created his “Strozzi NoSQL Open Source Relational Database”(DataStax 2018). Nevertheless, the first application of a non-relational database, “officially” referred to as “NoSQL” was introduced in 2009 under name “Dynamo” which was developed by one of the leading e-commerce companies called Amazon.(Harrison 2015, 45) Whereas the relational approach is known for its defined relational model, it is not possible to define one all-encompassing “non-relational” or “NoSQL” model. On the other side we can divide the NoSQL databases into following four groups: column databases; document databases; graph stores; key-value stores (Coronel and Morris 2016, 54). Moreover, Elmasri and Navathe added three additional categories of the NoSQL databases: hybrid systems; object databases; XML databases (Elmasri and Navathe 2016,888). Due to limited space of this thesis, this text will provide a closer introduction to the three most popular NoSQL databases – column databases, document databases and key-value stores/databases. But firstly let´s take a brief overview of the characteristics that is more-or less common for all of the subjected NoSQL databases.

Firstly, the existence of the NoSQL databases is intertwined with the so called “CAP Theorem”, which was introduced by Eric Brewer in 2000 (Harrison 2015, 43). The letters “CAP” represents abbreviations of the words or the qualities “consistency”, “availability” and “partition tolerance” connected to the distributed database systems. The “CAP Theorem” stipulates that it is impossible for a distributed database systems to simultaneously dispose of all of the qualities, and thus shall be possible for a distributed database system to simultaneously dispose of the two out of the three mentioned qualities at most (Coronel and Morris 2016, 581). It make sense to define the mentioned properties of the CAP theorem which may have a different meaning or a connotation in relation to the distributed database systems. Consistency shall ensure that all users of the particular database dispose of an identical view of the data at any given instant (Harrison 2015, 44). The second term – availability – means that the systems write and read operations are operational, available, even in case of a failure (Elmasri and Navathe 2016, 889). The last one – partition tolerance – “means that the database can maintain operations in the event of the network´s failing between two segments of the distributed systems” (Harrison 2015, 44). As mentioned, this theorem supposes a dropout of one of the properties. In case of the distributed database systems and thus in case of NoSQL databases consistency was “victimized”. Therefore, in relation with these non-relational database systems the term of “eventual consistency” proved its worth (Harrison 2015, 45). In practice this term is being projected as a certain state of data that is eventually visible to all of the users (Harrison 2015, 45). However this state of date is not visible synchronously to all of the users, but possibly with certain time-delay. The concept of “eventual consistency” was given a birth thank to the Amazon´s Dynamo key-value store revealed in 2007.(Harrison 2015, 45)

Secondly, all of the NoSQL databases dispose, besides the concept of eventual consistency, of the following fundamental properties or qualities:

- High availability and replication

- Scalability
- Sharding
- High performance
- Schemaless
- Not based on relational model and/or SQL (Elmasri and Navathe 2016, 885; Coronel and Morris 2016, 54)

High availability corresponds with the CAP Theorem and also with the concept of "eventual consistency". This property is connected to certain form of replication, since replication shall reassure high availability of the data stored in databases. Therefore, data is transparently replicated in at least two nodes, to ensure the data is still available even if one node failed (Elmasri and Navathe 2016, 885). Thus replication improves data availability and it can also enhance the read performance (Elmasri and Navathe 2016, 885).

Scalability of the NoSQL databases shall be secured by the means of horizontal scalability. This is being "employed while the system is operational, so techniques for distributing the existing data among new nodes without interrupting system operation are necessary" (Elmasri and Navathe 2016, 885). This horizontal scalability is generally applied in case of a growing data volume by adding new nodes for the data storage, thus enlarging the distributed database system itself (Elmasri and Navathe 2016, 885).

Sharding is also referred to as "horizontal partitioning" (Harrison 2015, 41). Since the databases may contain millions of records and those might be desired to be accessed by a high number of users, this load of accesses of the database file records is to be horizontally partitioned, sharded or distributed to multiple nodes (Elmasri and Navathe 2016, 886). This concept enhances the data availability.

The NoSQL databases are often used in the solutions where it is highly desired to search for and eventually to find certain data records among millions of them. Therefore, a high-performance data access is required. This can be achieved using one of the following two techniques: "hashing or range partitioning on object keys" (Elmasri and Navathe 2016, 885). In this case the majority of object accesses is to be provided by the key value provision and not by query conditions (Elmasri and Navathe 2016, 886). The former uses a hash function "h(K)" which is applied to the key "K" and thus the object location with key "K" is determined by the value of the function "h(K)" (Elmasri and Navathe 2016, 886). The latter – range partitioning – the object position is determined by a range of key values, for instance, "location "i" would hold the objects whose key values "K" are in the range "$Ki_{min} < K < Ki_{max}$" (Elmasri and Navathe 2016, 886).

The NoSQL databases deals with semi-structured data, and therefore, contrary to the SQL-based relational databases these non-relational databases may not require any schema. Shall there be no schema, the data constraints have to be programmed directly in the application programmes accessing the data items (Elmasri and Navathe 2016, 886). The semi-structured data which populates the databases can be described in the languages such as JSON and XML.

The NoSQL databases try to address the huge flow of millions of records, therefore it is said they provide an answer to the “big data” issue. In correspondence with this the NoSQL databases are not based on relational model. On the contrary they are based on “schemaless” key-value data model (Coronel and Morris 2016, 54).

The lines above provided basic information on the common characteristic of the NoSQL databases. All in all, we can add the advantages and disadvantages of these databases outlined by Coronel and Morrison:

Advantages: high scalability and availability; low-cost commodity hardware usage; Big Data support; storage efficiency improved by the key-value model.

Disadvantages: required complex programming; relationship support possible only via programming; without transaction integrity support; only “eventually consistent” model possible (Coronel and Morris 2016, 58).

After the substantial qualities bear and coined by the so called “NoSQL movement” were introduced it is time to look closer to the three fundamental NoSQL “representatives” – column, document and key-value databases.

2.3.1 Key-Value Databases

The key-value database is the first one to be introduced in this thesis among the NoSQL databases, since it was the first NoSQL database created. More concretely, a database named as “Dynamo” developed and applied by *Amazon* in 2007 is considered to be the pioneer among the key-value databases (Harrison 2015, 45). As its name invokes a key-value database stores the data in the form of key-value pairs collection, whereas the key is an identifier of the value. (Coronel and Morris 2016, 663) The key, the identifier of the value is also used to provide the location of the value, of the stored data. This value can dispose of various formats and it can store structured, semi-structured and unstructured data, which can be in the form of text, image, XML or JSON document, whereas the formats mentioned do not represent an exhaustive list. (Elmasri and Navathe 2016, 896) Due to absence of foreign keys one cannot identify relationships among keys, what enhances speed and scalability of the key-value databases.(Coronel and Morris 2016, 663) The content of these databases is organized in the logical grouping named as “buckets” which, for better imagination, can be presented in a table-like form. (Coronel and Morris 2016, 663) The operations are based on combination of a certain bucket and a key, what implies the keys stored in certain bucket are of a unique character, whereas there might be more keys with identical name across the buckets of a database. The operations used in case of the key-value databases are the following: get; store; delete.

(Coronel and Morris 2016, 663) The following picture is to provide an example of the data storage/organization in a bucket within a key-value database:

Bucket = Player	
Key	**Value**
10039	„FName Richard LName Lacouvee NTeam Kings JNumber 54"
10047	„FName William LName Mannieu NTeam Jackets JNumber 19"
10057	„FName James LName Oshea NTeam Penguins JNumber 87"
10071	„FName Frank LName Parise NTeam Capitals JNumber 12"

Figure 7: An example of a key-value database elaborated based on the book written by Coronel and Morris (Coronel and Morris 2016, 664)

2.3.2 Document Databases

Document databases represent non-relational databases that store data as structured documents, for instance in BSON, JSON and XML formats (Harrison 2015, 53). These documents are part of bigger logical groupings named "collections" (Elmasri and Navathe 2016, 890). Contrary to the pure key-value databases the document databases might not be entirely schema-less. This means that some documents of a database can dispose of certain tags – named portions of a document (Coronel and Morris 2016, 664), but not all documents of a particular database are required to dispose of the same tags, of the same structure, since the documents are also characterized as "self-describing data" (Harrison 2015, 53; Elmasri and Navathe 2016, 890). The key difference from the "conventional" key-value databases is that the document databases endeavour to understand the data written in a document (Coronel and Morris 2016, 664). This is possible thank to the mentioned tags, which allows us to write a query which would retrieve data fulfilling certain condition, regarding the last example we could write a query which would exclusively retrieve the players of the team named "Jackets". Similar to the key-value databases, the document databases do not support relationships identification among the document keys (Coronel and Morris 2016, 664). The following picture is to represent the previously displayed bucket in a document database tagged format:

Collection = Player	
Key	**Document**
10039	(Fname: "Richard", Lname: "Lacouvee", Nteam: "Kings", Jnumber: "54")
10047	(Fname: "William", Lname: "Mannieu", Nteam: "Jackets", Jnumber: „19")
10057	(Fname: "James", Lname: "Oshea", Nteam: "Penguins", Jnumber: "87")
10071	(Fname: "Frank ", Lname: "Parise ", Nteam: "Capitals ", Jnumber: "12 ")

Figure 8: An example of a document-based database elaborated based on the book written by Coronel and Morris (Coronel and Morris 2016, 664)

2.3.3 Column Databases

Column database is embodiment of a fact that the “euro-centric” or “western” view of organizing the data in rows might not be the best solution for all the situations. Therefore, column databases are similar to the relational databases. However the is one significant difference: the relational databases store data in blocks by rows, whereas the column databases store data in blocks by columns (Harrison 2015, 53). To provide a better understanding of the column-centric storage the table provided as an example of a document-based database will be presented in the column-centric mode:

Figure 9: An example of a column-based database elaborated based on the book written by Coronel and Morris (Coronel and Morris 2016, 665)

The figure above illustrates the fact that the data on a single player are „distributed“ across several blocks, in this case across five blocks, whereas the data on the last name of the players are stored in one single block. Therefore, the column-based databases are appropriate for data-intensive reporting systems and data warehouses (Coronel and Morris 2016, 664). The column-centric databases are also to be used in the concept of the so called “column family database”. These databases “organizes data in key-value ”pairs with keys mapped to a set of columns in the value component” while a column represents a key-value pair (Coronel and Morris 2016, 666). The key is represented by the name of the column and the value is the data stored in the column. Bigger logical groupings of a columns creates the so called “super column” (Coronel and Morris 2016, 666). An example of a column family database is provided below:

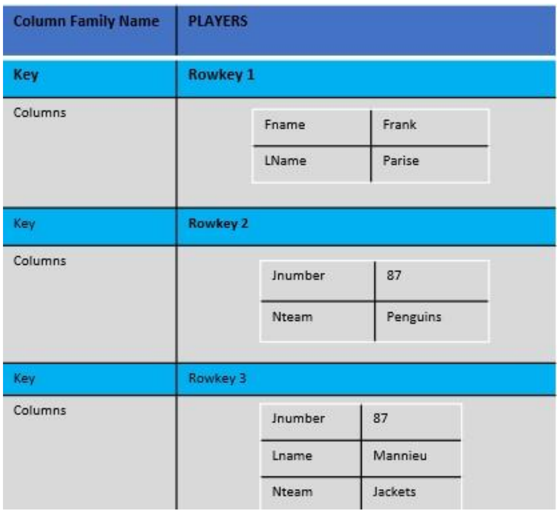


Figure 10: An example of a key-value database elaborated based on the book written by Coronel and Morris (Coronel and Morris 2016, 667)

2.3.4 Concrete NoSQL databases

Similar to the open-source scena of the relational databases, the NoSQL schema, although being considerably young, offers a wide palette of the concrete database systems. According to a high number of resources the following open-source NoSQL databases have been considered as the most popular in the course of the last two preceding years: “MongoDB”, “Apache Cassandra”, “Apache HBase”. Therefore a brief description of each will be provided.

2.3.4.1 MongoDB

MongoDB belongs to the group of the document-based NoSQL databases since it stores documents in BSON format. Atomicity and consistency of the documents are ensured by the two-phase commit method (Elmasri and Navathe 2016, 894). This database was written in C++ and it shall master huge data loads by replication (a variation of a master-slave approach) and horizontal scaling combined with a flexible data model (Perkins et al. 2018, 133; Elmasri and Navathe 2016, 894). The open-source community edition provide also support for ad-hoc queries, secondary indexing and real-time aggregations (MongoDB n.d.).

2.3.4.2 Apache Cassandra

Apache Cassandra represents a column-based NoSQL database which is claimed to offer a high linear scalability and availability even “under high stress” (The Apache Software Foundation 2022a). Cassandra disposes of a masterless architecture, what means there is no master node, so that each node of a database is able to provide the same functionality, thus contributing to its robustness or fault-tolerance and resilience (The Apache Software Foundation 2022a). It supports automatic synchronous or asynchronous data replication and in terms of the CAP theorem it is of a “AP” (available and partition-tolerant) character (The Apache Software Foundation 2022a). Whereas Cassandra belong to the NoSQL group, it support its own query language – CQL Cassandra Query Language designed for communication with the Cassandra database (The Apache Software Foundation 2022b).

2.3.4.3 Apache HBase

Frequently referred to as “HBase”, “Hadoop Database” or “Bigtable” is an open-source database written in Java is characterized as column-oriented. HBase is or can be implemented on the top of the HDFS Hadoop Distributed File System (Harrison 2015, 115). HBase supports linear and modular scalability, strictly consistent writes and reads, convenient base classes for backing Hadoop MapReduce jobs with Apache HBase tables, real-time queries and “easy to use Java API for client access” (The Apache Software Foundation 2022c). It also disposes of built-in versioning and compression capabilities (Perkins et al. 2018, 89), that make it able to keep several versions of a data item accompanied by respective timestamps (Elmasri and Navathe 2016, 901).

2.3.4.4 Comparison: MongoDB, Apache Cassandra, Apache HBase

After a brief introduction to the three concrete, popular NoSQL database systems, the following table is to outline and to compare the main characteristics of them.

Aspect	**MongoDB**	**Apache Cassandra**	**Apache HBase**
Provider	MongoDB, Inc	The Apache Software Foundation	The Apache Software Foundation
License	Open source; possible commercial “extensions”	Open source	Open source
Type	Document-based	Column-Based	Column-based
Language(s)	C++	Java	Java
Compatible operation systems	Linux, MacOS, Open, Oracle Solaris, Windows	FreeBSD, Linux, MacOS, Windows	Linux, Unix, Windows

Support for APIs	Actionscript, C, C#, C++, Clojure, ColdFusion, D, Dart, Delphi, Erlang, Go, Groovy, Haskell, Java, JavaScript, Lisp, Lua, MatLab, Perl, PHP, Powershell, Prolog, Python, R, Ruby, Rust, Scala, Smalltalk, Swift	C#, C++, Clojure, Erlang, Go, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, Scala	C, C#, C++, Groovy, Java, PHP, Python, Scala
SQL compliance	Read-only SQL queries via the MongoDB Connector for BI	SQL-like SELECT, DML, and DDL statements in CQL language	No
Replication	Multi-source; source-replica	Selectable replication factor	Multi-source; source-replica
ACID compliance	Yes	No	Yes
CAP Theorem	CP	AP	CP
Distributed System Consistency	Eventual and Immediate Consistency	Eventual and Immediate Consistency	Immediate Consistency
Data Storage	Disk	Disk	Hadoop
Queries	Map/Reduce	Map/Reduce	Map/Reduce
Concurrency Control	Locks	MVVC Multi Version Concurrency Control	Locks
Protocol	TCP/IP	TCP/IP	HTTP/REST
Popularity	Stack overflow: 170 509 questions till 11.01.2023 Google: 17 800 000 results in 2022	Stack overflow: 500 questions till 11.01.2023 Google: 1 420 000 results in 2022	Stack overflow: 500 questions till 11.01.2023 Google:71 300 results in 2022
Data Model	Collection-Document	Keyspace-Column Family	Regions-Column Family

Fault tolerance	No single point of failure with sharding approach as we can configure multiple mongo s instances. Single point of failure in master slave approach.	No single point of failure with peer to peer architecture	Single point of failure in master slave approach. Can be overcome by failover clustering
Write performance	Fast write-performance if data in RAM; very fast write-performance with allowance for data loss	Very fast write-performance thank to peer to peer architecture and Cassandra data model	Slower write-performance if synchronous writes are applied, better write-performance in case of asynchronous writes
Read performance	Dependent on consistency level	Dependent on consistency level and replication	Very fast
Durability	Achieved by WAL (Write Ahead Log)	Achieved by Commit Log	Achieved by WAL (Write Ahead Log)
Indexes	B-Tree Indexes	Hash Indexes	LSM-Tree Indexes
Aggregate Functions	Yes	No	Yes
Relative Comparison			
Availability	D	A	D
Consistency	A	A	C
Durability	B	B	B
Maintainability	C	C	D
Read Performance	A	D	D
Reliability	A	B	B
Robustness	C	B	E
Scalability	D	A	A
Write Performance	D	A	B

The table is focussed on the latest versions of the compared entities and it was composed based on the following resources:

(Klein n.d.; DB-Engines 2022b; Padhy et al. 2011; Manoj 2014; Lourenço 2015)

The “Relative Comparison” part of the table is based or derived from the article written by Joao R. Lourenco et al. The authors defined the subjected quality criteria as follows:

Availability – mirroring the downtime

Consistency – its grading based on two criteria: ACID-semantics consistency level and possible level of consistency fine-tuning

Durability – based on the use of single or multi version concurrency control schemes

Maintainability – measured according to the setup and use complexity; the accessibility of the tool which can interact with the database system; the accessibility of relevant expert literature on the particular database system

Read & Write Performance – measured by the results of experimental studies and the fine-tuning possibilities

Reliability – the ability of a database system to operate without any failures, also based on synchronous propagation modes

Robustness – the measures gained by literature review related to the database systems robustness in case of crashes and attacks

Scalability – this was evaluated depending on the database elasticity, its performance increase due to horizontal scaling and the complexity of the online scalability (Lourenço et al. 2015, 19-20)

The final marks or points were presented in the table above, were transformed from the original version, where they were displayed as various geometric formations, into letters A-D.

3. CRM&BI SYSTEMS

This main goal of this thesis is to identify the most suitable database and to design an appropriate database for the future CRM/BI application. Since an additional or a marginal aim of this thesis is also to design a dashboard design this chapter is to briefly provide information on the mentioned entities – CRM customer relationship management, BI business intelligence and dashboards. The following lines will not include any detail comparison of the aforementioned entities´ types, since this would not be of a high relevance for the purpose of this thesis. Nevertheless, the most fundamental classification of CRMs (including a brief information, based on which the differences among the types presented can be derived) and dashboards will be addressed.

3.1 Customer Relationship Management

The CRM abbreviation hides the well-known scientific and business term “customer relationship management” which is fundamental in the current economic environment. However, it is not fundamental to discuss when the birth of CRM happened, since in a certain form, it could be dated back to the beginning of the world, but on the contrary its meaning is more important. As usual even this term do not dispose of any unequivocal definitions, therefore this thesis will outline two definitions of CRM. By several referred to as “The Godfather of CRM”, Paul Greenberg came up with the following definition: “CRM is a philosophy and a business strategy supported by a system and a technology designed to improve human interactions in a business environment” (Greenberg 2009, 30). Perhaps a more focused and economically business-oriented understanding of CRM was provides by V. Kamar and W. Reinartz, who defined CRM as “the strategic process f selecting customers that a firm can most profitably serve and shaping interactions between a company and these customers. The ultimate goal is to optimize the current and future value of customers for the company” (Kumar and Reinartz 2018, 5).

Similar to the definitional ambiguity, the same state affairs can be found in the matter of the CRM systems classification, or the types of the CRM systems. Generally we can define three types of the CRM systems, or we can divide the CRM system as a whole into the four following subtypes: analytical; collaborative; operational, strategic (Al-Homery et al. 2019, 131-134; Kumar and Reinartz 2018, 5).

The main role of analytical CRM is to provide a basis and/or a support in the decision making process based on the customers data. The type of CRM does not come into interaction with the customers directly, it is rather focused on the “back-office” support by outlining patterns of the customers behaviour, for instance thank to data mining.

Various techniques and/or methods are executed in relation to the gained and stored customer data that enables the executives to make “knowledge-driven strategic business decisions” (Kumar and Reinartz 2018, 136).

The primary focus of the operational CRM is to record the instant information on the customers and thus to automate or at least to contribute to the automation of the business processes in relation to customer. The fact that automation of the customer-oriented processes are at the core of the operational CRM is embodied by its three fundamental parts: salesforce automation; marketing automation; service automation (Hayley 2016, 31). The service provided by operational CRM lead or at least shall lead to more efficient business processes, to costs reduction and also to enhanced analytical CRM, if the latter is also put into effect by a business entity.

Whereas the first two CRM types are tightly linked to the field of data science and/or informatics as such, strategic CRM is of a substantially different character, and could rather be connected to the field of architecture and/or even philosophy. Only one single substance remains unchanged – the customer, seen as “the source of wealth generation”((Kumar and Reinartz 2018, 34) is in the centre of attraction. Strategic CRM can be regarded as a business strategy, whereby CRM is implemented and “examined” on a company-wide level ((Kumar and Reinartz 2018, 34). Therefore, strategic CRM refers to a vision how a concrete CRM system for a concrete company shall be built up. This is the reason, why strategic CRM is not inevitably related to any concrete technology or process (Kumar and Reinartz 2018, 35). On the other side it consists of the following components: strategic process; interactions; customers, current and future value of the customer (Kumar and Reinartz 2018, 35).

The last CRM type, collaborative CRM, can be defined as a type of internal communication tool of a business organization. This statement was derived from its application, since this enables the individual teams of a business organization to access the customer data and thus to take appropriate actions (Hayley 2016, 31). The idea of collaborative CRM systems may have also resulted from the uprise of a collaborative value chain, which considers vendors and their suppliers as partners, what “supersedes the idea of mere integration among CRM, supply chain management (SCM), and enterprise resource planning (ERP) applications (Greenberg 2009, 254). On the other hand collaborative CRM “exploits” collaborative services and infrastructure to enhance the interaction between a business organization and the customers. (Al-Homery et al. 2019, 131-13; (Kumar and Reinartz 2018, 5). Thus it can be claimed it enables or supports channel and interaction management of a business entity.

3.2 Business Intelligence

The term “business intelligence” has become increasingly fashionable in the course of the decade. However, the origin of this term, together with science and technology behind dates back to the last 1950s. Similar to the relational databases, even in this case there is a strong connection to the company “IBM”, since in 1958 a scientist employed by “IBM”, and more importantly a person who gave a birth to the hashing algorithm, Hans Peter Luhn used a selective dissemination of information technique whereby he presented the following concept resembling the modern definition of business intelligence: “Business is a collection of activities carried on for whatever purpose, be it science, technology, commerce, industry, law, government, defense, et cetera. The communication facility serving the conduct of a business (in the broad sense) may be referred to as an intelligence system. The notion of intelligence is also defined here, in a more general sense, as “the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal” (Luhn 1958). The similarity is evident when one compares it to the modern definition of business intelligence elaborated by Howard Dresner, former analyst at the company “Gartner Group”, often entitled as father of business intelligence systems, defined it as follows: “a broad category of software and solutions for gathering, consolidating, analysing and providing access to data in a way that lets enterprise users make better business decisions” (Gibson et al. 2004). More recent definitions of business intelligence could be also added, but they represent more-or-less resemblances of the two definitions already mentioned, and thank to some of them business intelligence can be regarded as an extended or even integrated part of a CRM system, therefore a comparison of individual BI methods and/or systems would be out of the scope of this thesis.

However certain methods and technologies used in the field of business intelligence will be outline in the following lines. According t what was already written, it is obvious that the existence of business intelligence stems from existence of certain data, that shall undergo an analytical process. In this regard three fundamental types of analysis are strongly related to business intelligence: predictive – providing possible future outcomes based on the gained data; descriptive – endeavours to identify links and trends based on the gained data; prescriptive – providing a “panacea” for a business organization in certain situations (Stedman and Burns 2020). However, one shall not exclude application of other different types of analysis primarily used in the field of advanced analysis. By the means of the aforestated analytical approaches in combination with other methods/techniques the following styles or categories of business intelligence applications have been elaborated:

- Data Mining and Advanced Analysis
- Visual and OLAP analysis
- Enterprise Reporting
- Dashboards and Scorecards

- Mobile Apps and Alerts (Nedelcu 2013, 13)

As one could notice, dashboards were also mentioned among the fundamental styles of the business intelligence systems, and they will also be mentioned in the "conclusive" lines of the first part of this thesis.

3.3 Dashboards

Dashboards attract visual sensation of a human and therefore it is not surprising that this term itself is automatically associated with certain visual presentation. Dashboards have become increasingly popular since the beginning of this millennium, whereby data play a key role. In the words of an expert on the topic of dashboards and dashboard design, Stephen Few, a dashboard can be characterized as "a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance" (Few 2006, 26). Dashboard might be considered as a unique fusion of our visual sense and technology, since thank to visual data presentation we "harness the incredible power of our visual system to spot relationships and trends" (Wexler et al. 2017, 3). Besides the visual effect, a high-quality dashboard are of a considerable time-saving character. To provide a broader information on the reliable architectonic rules of a visually high-quality dashboard would be a suitable topic at least for a one separate bachelor thesis. Therefore we will just point out, that a quality dashboard shall be designed in accordance with the following six Gestalt principles of visual perception: proximity, closure, similarity, continuity, enclosure, connection (Few 2006, 74).

Similar to databases, we can categorize dashboards according to a high number of criteria. However, due to the range and content character limitation the dashboard categories based on a role of dashboard will be presented. In accordance with Stephen Few, the dashboards are divided into the following three groups based on their role (in sense of purpose): strategic; analytical; operational (Few 2006, 30). Not only the names of the dashboard categories correspond with those of the CRM systems, but we can say that they dispose of the same character.

4. The Information on the Agency

The fourth part of this bachelor thesis is to outline the information on the agency´s structure, whereby mainly relevant structures and relevant potential users will be introduced, and of course the needs regarding the application database and dashboard will not be left behind. These factors, the agency´s structure, people (whether users or potential administrators), purposes of the future application and the needs of the agency regarding our final artifact(s) – database and dashboard design of the application shall navigate us to create a relevant form of a weighted scoring model. Based on this model the most appropriate database type for this specific case will be chosen. Considering the fact, that SARIO, the Slovak Investment and Trade Development Agency represents a state entity, certain more concrete information might be of a sensitive character, and therefore this kind of information might be presented in a slightly modified form. However, this situation and the character of the modifications will not have any negative effects on the quality of the final artifact that shall stem from this thesis.

4.1 The Agency and Its Purposes and Structure

As it was already mentioned in the first lines of this bachelor thesis, the abbreviation "SARIO" stands for the Slovak Investment and Trade Development Agency. This state agency was established in the year of 2001 and since the it has been working under the auspices of the Ministry of Economy of the Slovak Republic (SARIO – Slovenská agentúra pre rozvoj investícií a obchodu 2023a). The main purposes of the agency were already stipulated in its name. However, it is beneficial to outline them more concretely since they may be related to the final artifact of this thesis. According to the official documents of the agency its main aim is to enhance the economic environment of the Slovak Republic, and thus it endeavours to boost the export activities of the entrepreneurs based in Slovakia, and simultaneously the agency´s employees shall contribute to the inflow of the foreign direct or indirect investments to the country. Those purposes shall be fulfilled by certain activities developed and implemented by the agency employees that are "functioning" in the following agency´s departments: Department of Investment; Department of Foreign Trade; Department of Marketing, Department of Regional Development; Department of Human Resources; Economic Department; National Project Department – Support of Internationalisation of the Small and Medium Enterprises (SARIO – Slovenská agentúra pre rozvoj investícií a obchodu 2022). This enumeration represents the structure of the Slovak Investment and Trade Development Agency that was an employer of 88 people in 2021(SARIO – Slovenská agentúra pre rozvoj investícií a obchodu 2022, 30). The departmental names indicate that not every single department might be directly involved in the main purposes of the agency or in the external communication, communication towards public, national or foreign enterprises and foreign state entities. This assumption is correct and since the future application shall be designed to provide an aid to the employees (and also to their clients) active and responsible in the core export support activities a more detailed insight into those respective activities will be provided in the following lines.

4.2 National Project Department and Information Requirements

The aforementioned actions heading towards increasingly more positive trade balance of the Slovak Republic and thus towards creation of more plausible economic environment go to the core competencies of the National Project Department – Support of Internationalisation of the Small and Medium Enterprises, further abbreviated as "NPDSI". Therefore, the following lines will be devoted exclusively to this department, since its employees embody the potential users of the final artifact of this thesis.

The activities of the NPDSI are concentrated on the support of export efforts of the small and medium enterprises based on the territory of the Slovak Republic. Of course, not all the companies based in Slovakia are automatically eligible to be provided the so-called "non-financial" aid, but they have to fulfil certain criteria to be authorized to take part in the activities of the NPDSI. However those criteria are not in the centre of attention when outlining the needs of the NPDSI employees, therefore they will not be discussed more concretely. On the contrary, the export-support-oriented activities of the NPDSI are of a more decisive character, and they can be outlined and enumerated as follows:

- Organization of the national stand at international trade fairs/exhibitions
- Organization of business missions
- Organization of business fora and individual B2B meetings
- Specialized consulting services regarding actual business opportunities
- Organizations of training courses and best-practices business seminars
- Supply-chains development meetings (SARIO – Slovenská agentúra pre rozvoj investícií a obchodu 2023b)
- Information services regarding business-oriented facts and practices

The aforementioned activities have been implemented since 2017, when the project was launched. Since then hundreds of events have been organized which has been of a beneficial character for thousands, of both, Slovak and foreign entrepreneurs. We can also mention more concrete figures regarding the year of 2021. According to the annual report of the agency, the NPDSI was responsible for the organization of 14 national stands at international exhibitions, 16 business missions on 4 world continents and 34 training and best-practices seminars (SARIO – Slovenská agentúra pre rozvoj investícií a obchodu 2022, 19). The concrete figures for the “remaining” types of services were not provided. Over 470 Slovak small and medium enterprises could benefit from their participation on the events created by the NPDSI. This figure might not resemble any extremely high value, however the so-called “administrative burden” can be considerably heightened by every single record, in this case by every single company. This “burden” stems from compulsory registration, screening, other legally inevitable actions related to the company´s authorization and participation. On the other side the legally or formally compulsory information would not be sufficient to provide more tailor-made services for the representatives of the authorized small and medium enterprises, what would generally decrease the niveau of the provided services and of course of their precepted quality what might negatively affect the experience of the entrepreneurs. In order to eliminate or at least to mitigate this risk, additional information on and from the companies is highly welcomed. Under the “additional information”, one may understand the following: short company description; business sector; desired export destinations; export capacities; products and services to be exported; desired business partners; desired sectors of the business partners; production volumes; information on possible advertisement campaigns; information on specific form of export development activities; information on the company representative(s) / participant(s), agent(s); desired export activities; desired export-support events; information on the business progress achieved during the events; .

The aforementioned terms represent an enumeration of information that could be collected when an entrepreneur makes an application for certain event organized by the NPDSI. Of course, this list is not of exhaustive character and it would satisfy the needs of the data desired only partially. The information on the companies and other business entities taking part in the activities of the NPSI represents only one side of a coin. The other one is represented by the “internal information”. In this regard we refer to the information on the activities and on the competent employees. In this case, the data on the activities could be the following ones: name of the event; date of the event; type of the event; event venue; sector of the event; internal person in charge of the event; event partner; contractor; space size financial information on the event; companies interested in the event; companies applied for the event; companies taking part in the event; event objective; event anticipation; events results. Only certain aspects relevant for the NPDSI activities were mentioned in the aforementioned list and they are of a more general character. This “more-general” character of the information stipulated is caused by the uniqueness of the NPDSI activities types which is “translated” into the fact that each activity type is assumed to require – at least – slightly different (type) of information on itself and also on the respective companies and employees. However, the information stored in a database shall not exclusively refer to the organized activities, but the information related to the respective employee shall be stored too.

One of the purposes of the future application shall be to provide an overview of the information on the respective employees, members of the NPDSI team. This information shall be accessible for the NPDSI team members and also for the registered entrepreneurs and their representative, however in some restricted mode, the concrete form of which is not a subject of this thesis. On the other side we can introduce the information or data on the employees that shall be stored in the application database: employee first name; employee second name; employee phone number; employee email; employee education; languages spoken by employee; countries and/or territories managed by employee; events organized by the employee; employee evaluation. This kind of information shall be of a valuable character for all “players” of the game – for the representatives of the companies who could make a better anticipation of the aid/service provided by a concrete employee and who could mark the performance of the employee(s); for the agency management, who could immediately get an overview on the employee´s professional qualities (education, languages), on the employee´s “utilisation” (organized events, information on the participants´ success) and satisfaction rate (employee evaluation); and last but not least for the employees, who could immediately gain access to the feedbacks from the business entities, so that they can keep on enhancing their services and approach to made them more tailor-made for the needs of the companies.

4.3 Database Type Requirements and Weighted Scoring Model

Regarding the information stated in the lines above and following the interviews and/or discussions with the competent employees of the NPDSI we have arrived at certain statements regarding the requirements/needs on the application database (and also dashboard) which are to be introduced in the following lines. The first requirement is of a non-technical character and it is related to the financial angle of the view. The database solution shall be **as financially effective and efficient as possible**, in other words the best possible option would be the one which did not represent any financial costs. Since all the database management systems introduced in this bachelor thesis dispose of an open source version, the requirement of financial effectiveness and/or efficiency does not have to be included in the final weighted scoring model.

The second decisive requirement can be characterized as a difficulty level of the database set-up, its maintainability and support availability. To simplify our terminology, the three aforementioned matters will be hidden under the term “**maintainability**”. Certain solutions may require highly-skilled and experienced IT-workforce for both of the processes – set-up and maintenance, whereas some might be developed by more accessible IT-specialists. This might be mirrored in the lower financial and time costs. Accessibility and or/availability of support services whether by the product licence holders or by contributions of other users in various fora may be decisive in the application and/or database lifecycle, especially if any unforeseen and undesired situations shall be taken into account. Another advantageous aspects stemming from this requirement would be better readiness and availability of the database solution. Therefore, a substantial emphasis is to be put on this requirement.

Regarding the figures mentioned in the above lines which were dedicated to the specific data on the agency, mainly throughout the year of 2021, it can be clearly estimated that the database will not be heavily overloaded by the data – the transactions will not be performed on a permanent basis. On the other side **the data and the information stored in the database shall be permanently accessible** in order to provide the wished information or overview. Thus, the criterion "**data storage accessibility**" is to be added to the weighted scoring model.

It was already stipulated that the data flow into the database will not be permanent. On the contrary the data flowing into the database might be of an important character, since based on the information stored in the data and based on its rank, the agency services and the non-financial aid might be allocated. Therefore, it is desirable to make a decision in favour of a database which makes its mark by a fair fast performance, mainly in terms of **read performance**. This should also ensures that the data will be displayed more-or-less immediately. Considering the information gained from the agency´s representatives it seems justifiable to make a remark, or at least to assume, that this database shall not face any extraordinarily complicated queries. The same was already mentioned on the size of data that is intended to be stored. Therefore, **scalability** might be included in the criteria, but it shall not play a decisive role.

The expression" **security**" cannot be omitted when dealing with the information systems and especially with database. The security of the data and information stored in the database shall be of a high level in order to prevent from any unauthorized database access and leaks.

The data to be stored in a database shall be stored in an easy-to-read manner (ideally for both – computer and human) what would consequently speed up the decision process. Therefore, the forms of **structured data** seems to be more desirable than the unstructured form of data.

According to the requirements of the relevant agency´s department and in accordance to the intended purpose of the application – an orientation towards customer management and business intelligence, the ideal database solution shall comply with both – **OLAP**, online analytical processing and **OLTP**, online transaction processing. The transactions might be presented by a new registration of a business entity to certain event of the agency, whereas the analytical processing-orientation of the database might be used or applied when deriving specific information on certain interests of the entrepreneurial plenipotentiaries, what might assist the subjective agency´s department to enhance its professional performance and to provide more tailored services. In accordance with the aforementioned, the database shall represent an ideal option or intersection fulfilling the requirements for both of the aforementioned processing types.

The lines above were to present the conditions or the requirements of the agency representatives, mainly of the potential users of the CRM&BI application the data of which shall be stored in or final artifact – in the database. Following the goal, to choose the most ideal database type for this concrete case and bearing in mind concrete importance of each particular requirement for the relevant NPDSI representatives the following weighted scoring model, accompanied by the respective numeric values, was elaborated:

Weighted Scoring Model Aiming at Choosing the most Appropriate Database Type						
Criteria | Databases ↓ →	**Weight**	MySQL	PostgreSQL	MongoDB	Apache Cassandra	Apache HBase
Maintainability	**20%**	6	4	2	2	1
Data Storage Accessibility	**15%**	1	0	1	1	1
Read Performance	**15%**	4	3	6	5	5
Scalability	**5%**	4	5	6	6	6
Security	**15%**	4	5	3	2	2
Structured Data Appropriateness	**15%**	6	6	1	1	1
OLAP+OLTP Orientation	**20%**	6	5	3	2	2
Summary	**-**	**4, 85**	**4, 15**	**2, 95**	**2, 45**	**2, 25**

As already mentioned the weighted scoring model is a result of the requirements consulted with the relevant NPDSI representatives, which indicated the importance or relevance of the particular requirements. Based on the consultations the final weights were assigned to the respective criteria. Afterwards, the database managements systems, the SQL and NoSQL "plenipotentiaries" selected for this bachelor thesis were evaluated on a scale from 1 (the least appropriate/conforming) to 6 (the most appropriate/conforming), based on their appropriateness/conformance or performance related to the particular criterion. Regarding the scale “1-6” an exception was made in case of the criterion named as “Data Storage Accessibility”, since there were only two possible options available, the “scale score” equals either 1 or 0. The last row of the table is devoted to the summary of the weighted score attained by the particular database managements systems. These results are of a decisive character when choosing the most appropriate database type for the future CRM&BI application of the agency´s department. In correspondence to the design science research premises introduced in the final chapter of this thesis the score was assigned based on the comparison of the relevant database types and thus based on the theoretical knowledge derived from a knowledge base of the author of this thesis. This knowledge base, regarding the database types and database design was to a large extent created in the course of the thesis “composition” and is to be found in the bearing theoretical chapters of this thesis. The theoretical knowledge was then “supplemented” and influenced by the facts, information on and from the agency´s representatives who presented their perceptions on the future CRM&BI application, its purposes and demands placed on the database. Combination of the aforestated factors resulted in a final decision to adopt “MySQL” as a database management system for the future CRM&BI application.

4.4 Dashboard Design Requirements

Dashboard design of the application represents the second and complementary final artefact of this thesis. The respective subsection already mentioned the three types of dashboard based on its role – strategic, analytical and operational. Strategic dashboard might be named as “executive dashboard” applied mainly for the purposes of the higher-management level. Its main goal is to present an overview in order to keep an eye on the development of the business or organization activities what may lead to detection of new trends, opportunities and risks (Few 2006, 31). The second type – analytical dashboard – may be characterized as an upgrade of the strategic dashboard, mainly in terms of detail of the information provided. Moreover, the way in which the data are to be displayed shall demonstrate certain relationship among them, and additionally, more concrete information shall be available for the user by the means of incorporated interactive features (Few 2006, 32). The third entity of this classification, operational dashboard would include immediate real-time information (Few 2006, 32). This type of database would be of a higher relevance for technical processes, or for banking, financial operations. Therefore, after consultation with representative of the NPDSI it was decided that, regarding the dashboard type, an intersection of the strategic and analytical approaches shall be applied.

The concrete resemblance of a dashboard design will be dependent on the user´s role. The application will be used by various users with different rights. However this rights and user roles are not in the focus of this thesis. Therefore, it is essential to note that the final artifact – the database dashboard elaborated and presented in this thesis will correspond with a dashboard designed for the leading manager of the NPDSI. For that reason, it is obvious that the dashboard design will not contain any in-depth and real-time information that would stem from in real-time executed transactions, but rather data that would be updated on a daily basis. Based on the consultations with the representatives of the NPDSI the following information shall be included in the designed dashboard: budgetary information on the budget already spent; budgetary information on particular activities; number of organized events; number of events in progress; companies sectors; the most desired future events; number of companies registered; number of newly registered companies in the course of the recent 3 months; export destinations defined in the last 6 months; calendar indicating basic information on the activities of the NPDSI employees. All of the information shall be displayed clearly, with the best possible adherence to the so-called gestalt-principles, and also utilizing the qualities of human brain by application of the “preattentive” attributes (Few 2006, 67). Ideal application of the preattentive attributes of form and position shall assist the recipient of the information to process the “message” of the information more time-efficiently.

5. Database and Dashboard Design

5.1 Database Conceptual E-R Model

The Entity-Relationship model represents one of the possibilities that are applied to display the database entities. However, as its name indicates, besides the database entities as such, their relations create an inevitable part of these models. Generally, we can claim that the E-R model embodies an illustration of a real world, the objects or concepts of which and their intertwined net of connections is displayed based on the real or possible events that may occur in certain real-life situations. These real and possible events can be suggested in cooperation with the main stakeholder of a project and can be transformed or translated into the form of "use-cases". Use-cases can be presented by simple text lines, or they can be visualized as use-case-diagrams. The first option as selected for the purposes of a database design of our final artifact, what implies "use-cases" or a problem description presented in simple text lines. On the other side these text-lines will be accompanied by a more-or-less self-explaining E-R model of the subjected database.

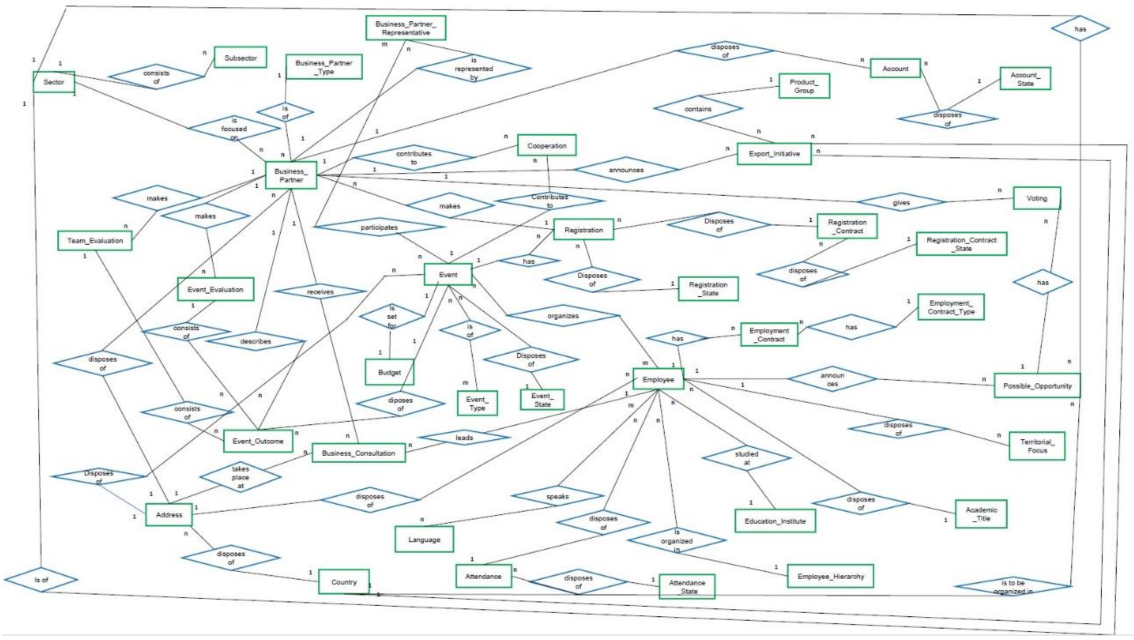


Figure 11: Conceptual Entity-Relationship Model of the Database

According to the American National Standards Institute the entity-relationship model belongs to the category of conceptual models which shall mirror business requirements, thus representing an interconnection between the customers, or potential users and developers of the information system (Juba et al. 2016, 22). Moreover, it is important to note that the business requirements included in the E-R model shall not be described to their last detail. The E-R model consists of entities – embodying the real-world objects; attributes – standing for certain qualities of the defined objects; relationships – illustrating association between E-R model entities (Coronel and Morris 2016, 124). It is crucial to note, that the entities displayed in the E-R models do not refer to a single entity occurrence, but on the contrary each E-R model entity is an entity set (Coronel and Morris 2016, 118). Entity occurrence might be defined as a single row or a record of attribute values assigned to particular entity. A particular entity instance can be unambiguously identified using entity identifier named as primary key. The primary key is composed of at least one (and usually one) or more attributes throughout which a unique entity occurrence is to be defined. This attribute, the primary key, is usually displayed with an underscore. In our case this approach is to be applied in the relational model of the database, since our form of E-R model will not include any attributes in order to preserve at least moderate clarity level of the E-R model, which is caused by a considerably higher number of entities and their associations. The E-R model dedicated to the future CRM&BI application can be considered as an evidence that even a relatively simpler application demands a relatively complex database, as it is to observe in the depicted model above in this thesis. This nature of database was brought into effect also than to the "participations". Using database jargon, a participation is referred to the entities and to the relationship between those entities. The entities taking part in the relationship are known as participants (Coronel and Morris 2016, 124). Taking a glance at out E-R model we can focus on the entities "BUSINESS_PARTNER" and BUSINESS_PARTNER_REPRESENTATIVE". The relationship or association between these entities was named as "is represented by", since a every single "BUSINESS_PARTNER" (in our case a company or a state entity/organization) is represented by at least one person – "BUSINESS_PARTNER_REPRESENTATIVE". The bidirectionality of this relationship is demonstrated by the fact that, switching to active voice, each "BUSINESS_PARTNER_REPRESENTATIVE" represents exactly one "BUSINESS_PARTNER". This quality of an entity relationship is named as "connectivity", in the described case the connectivity is of a value "1:n" also expressed as "1:m" or "one to many". Connectivity classifies the relationships between the entities. Other common and basic connectivities are as follows: "1:1" and "M:N" meaning "many to many". The connectivity value might be accompanied by the value of cardinality. Cardinality might be seen as an extension of connectivity since it shows maximum and minimum values of entity occurrences with one entity occurrence in the specific relation. Cardinality of relationship is not part of the above displayed conceptual E-R model, however we can imagine a situation where the number of the business partner representatives would be limited to the maximum value of 5 per business partner, and on the other side each business partner representative would be allowed to represent up to 2 companies. In this case the cardinality assigned to the entity "BUSINESS_PARTNER" would equal "(1,2)" and the cardinality devoted to the entity "BUSINESS_PARTNER_REPRESENTATIVE" would refer to "(1,5)". Another quality associated with the relationships is a degree of relationship. Generally, there are the following four types of the relationship degree: unary, binary, ternary (Coronel and Morris 2016, 134). The value of relationship degree indicates how many entities are associated with certain relationship (Coronel and Morris 2016, 134). The first degree, the unary one occurs within one entity. In case of our model this situation could have been found within the entity called “Employee”, since there is an anticipation that the employees are organized in certain hierarchical structure, in other words that some employees are supervised, while the other employees are supervising. This possibility was captured and therefore, the E-R model contains entity called “EMPLOYEE_HIERARCHY”. This entity disposes of a binary relationship with entity “Employee”, since there is an association between them named as “is organized in”. As expected, the ternary relationship degree disposes of three entities – participants. Regarding the requirements that are to be presented in the lines below, the elaborated E-R model could have been enriched by the following ternary relationship: a business partner can take participation in several events, where it is represented by a representative. The so far “invisible” entity, further in the relational model named as “Participation” stemmed from the “M:N” connectivity, and therefore it is referred to as a composite entity, that will be defined in the lines dedicated to the relational model. However, both, the conceptual and relational models were built differently from this assumption, and of course, the business partners were not stripped of the possibility to take part in various events.

The whole conceptual E-R model was elaborated based on the requirements proposed by the NPDSI representatives and also on the business cases that were derived from these requirements. For that reason let us represent several business cases – accompanied by a brief model description/explanation – that are to be found in the form of functionalities in the future application, and that are to be mirrored in the elaborated database model. As already mentioned the use cases will not be presented using use-case diagrams, but the text-line-presentation was chosen to be a sufficient form. The core of the NPDSI activity concentrated in organization of various events proposed to the entrepreneurs; business consultations provided to the entrepreneurs; cooperation development with entrepreneurial and state organizations. Therefore, when elaborating the conceptual E-R model, the main emphasis was put on the following entities: “EMPLOYEE”, “EVENT”, “BUSINESS_PARTNER”. Let us start with the right section of the conceptual E-R model. This is populated mainly by relationships and entities associated with the entity named “Employee”. The employees are organized in certain “Employee hierarchy”, which consists of three level, therefore the relationship “is organized in” was created. Every employee of the NPDSI disposes of an academic title (whereas only the highest one is taken into consideration), that was achieved by the employee at certain education institute. Each of the employees speaks, besides Slovak, at least one foreign language and every employee lives at one particular address. In order to become an employee, they had to negotiate and sign a certain type of employment contract. As usual, based on the employment contract each employee is supposed to commit to their employer what include their individual presence at workplace. This presence or attendance might be hindered by many circumstances which may result in absence, and therefore a different type of attendance shall be at disposal to choose from. The employee´s commitment is focused on certain area(s) of specification defined as territorial focus. An employee can announce a possible opportunity for the entrepreneurs, which is associated with certain country and industrial/business branch grouped under "Sector" entity which consist of certain subsectors. Similarly, an employee leads a meeting or a consultation to/with entrepreneurs or state organization that takes place at certain address, and therefore in certain country. Arriving at the core activity, at least one employee may organizes at least one event of a different type. For every event a certain budget is to be assigned; every event is in a certain state and every event disposes of a one type. In order to be represented by its representative each of the willing business partners are obliged to make a registration for that particular event. Every registration is of certain state, further it may dispose of a registration contract that is again assigned a particular state. Since the entity named "BUSINESS_PARTNER" shall cover both, entrepreneurial and state bodies, an entity named "Cooperation" was inevitable to be created. This situation was caused by the possible and often appearing use-case: (Other) State organization cooperates at organization of an event. For that reason, each cooperation is to be associated with a particular business partner and simultaneously with a specific event. Each event possesses of certain outcome that is to be elaborated by a business partner. Each event outcome disposes of two evaluations – event and team which is assigned to the outcome by a relevant business partner. Each business partner is represented by at least one representative. This business partner representative may participate at several events. Each business partner defines its main focus by a sector. Similar to employees, each business partner defines its "headquarters". A business partner may make account(s) that will be assigned certain state. Following the aim of the NPDSI to provide a more tailor-made services each business partner shall dispose of the opportunity to dispose of in favour of possible opportunity announced by an employee, which might be considered, from the business partner´s view angle, as a rather passive activity. On the other side the use case defined as "a business partner announces export initiative of a certain product group directed to a defined country" provides the entrepreneurs with a active participation in the process of the event organization, which in case of the E-R model gave birth to the entities "PRODUCT_GROUP" and "EXPORT_INITIATIVE".

Analysis of the requirements accompanied by the elaboration and definition of the use-cases resulted in the conceptual entity-relationship model of 36 entities and 48 relationships. The displayed model is a product of the second iteration and was constructed in conformation with the so-called "Chen notation" (Coronel and Morris 2016, 46). This model was further used to elaborate a relational and logical models that will be provided in the following section.

5.2 Database Relational and Logical Model

This section provides a translation of the preceding conceptual entity-relationship model into the so-called relational and logical model. In comparison to the previous E-R model these model will include the specific defined attributes of the entities, and thus also primary and secondary keys assigned to the entities. However the presence of the attributes and of the keys does not represent any exhaustive list. On the contrary, during the elaboration process certain amplifications, enhancements and adoptions to so-far unexpected state of affairs might have occurred. Nevertheless, these cardinal changes will be commented in the coming lines.

One could register in the picture dedicated to the conceptual E-R model the presence of certain "M:N" or in other words "many to many" relationships. The presence of this kind of relationships is a source of redundancies. However, the relational model, or relational databases as such are to be designed and put into effect in order to get rid of unnecessary redundancies. Although this may not be possible or it is not being applied in every single case, we have endeavoured to "fight against" the redundancies in the "redesign" of the E-R- model to the relational one. The final version of the relational model consisting of 39 tables connected by 52 relationships is depictured in the next figure and is commented on below the elaborated figure.

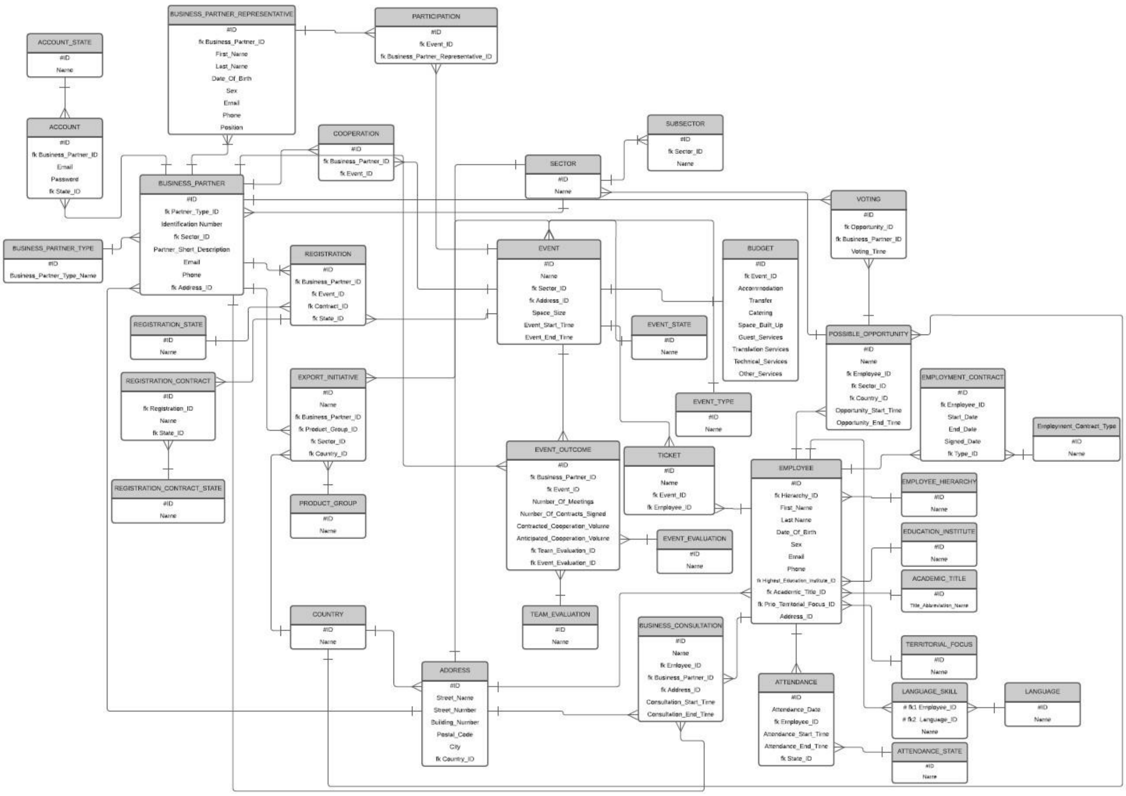


Figure 12: Relational Database Model

The entities are connected by the relationships the connectivity of which was represented by number “1” and letter “n” in the conceptual E-R model. Instead of this representation, the relational model represents the connectivity between the relations by different ending art of the lines connecting the relations. Each relation is “equipped” with its unique identifier named “primary key” marked by “#” before the corresponding attribute. In order to establish a logical connection between the tables the presence of the foreign keys is to be find in most of the tables. This is to be identified by the abbreviation “fk” place in front of the corresponding attribute. The attribute assigned the position of a foreign key refers to the same attribute in other table, where this attribute is assigned the position of the unique identifier – primary key. The primary keys for each relation was created artificially, therefore the “ID”, an abbreviation for “identifier” represents was assigned the role of a unique identifier. The unique identifier of the relation named “LAGUAGE_SKILL” is of an exceptional nature that is to be explained in the lines below.

As aforementioned, the relational model is to prevent from the “M:N” associations occurred in the conceptual E-R model. This is possible thank to creation of additional entity also named as composite entity (Coronel and Morris, 2016, 139). Applying the composite entities the relational model is stripped of the “M:N” relationships that are substituted by the new “1:N” relationships. Regarding our model, this was effectuated in the following “M:N” relationship cases:

“BUSINESS_PARTNER_REPRESENTATIVE” participates in “EVENT”

“EMPLOYEE” organizes “EVENT”

“EMPLOYEE” speaks “LANGUAGE”

All of this relationships are bidirectionally of “M:N” character. This issue of the first relationship between the entities “BUSINESS_PARTNER_REPREENTATIVE” and “EVENT” was solved by the creation of the composite entity named “Participation”. This entity decomposed the former relationship into the following two “1:n” relationships:

“BUSINESS_PARTNER_REPRESENTATIVE” (1) can dispose of (n) “Participation”

“EVENT”(1) can dispose of (n) “PARTICIPATION”

This means that each participation record is regarded to be a unique occurrence assigned with a unique event and unique business partner representative, the values of which are the foreign keys of the composite relation “PARTICIPATION”.

The second “M:N” relationships was decomposed similarly:

“EMPLOYEE”(1) is assigned with (n) “TICKET”

“TICKET”(n) aims at organization of (1) “EVENT”

In other words, each employee can be assigned with more tickets. However, each ticket is used for the purpose of organization of exactly one single event. This creates the opportunity to assign the event organization to “n” employees.

The third relationships was decomposed using a slightly different approach when it comes to the primary key assignment. Although the redesigned relationship might be of the logically same appearance:

“EMPLOYEE”(1) disposes of (n) “LANGUAGE_SKILL”

“LANGUAGE”(1) is a quality of (n) “LANGUAGE_SKILL”

In other word, each employee may disposes of several language skills, and each language skill can be regarded as a composition of a language and of a certain niveau of the language skill which is to be defined in the attribute “Named” in the relation named “LANGUAGE_SKILL”. In order to built a design that would be in correspondence with the relational theory it was also needed to assign a unique attribute to the language skills. In this case the primary key of the entity “LANGUAGE_SKILL” is composed of the primary keys of its “parent” entities – “Employee_ID” and “Language_ID”.

Furthermore, the entities that would be of a rather redundant character was deleted or better expressed incorporated into another entity. This "incorporative" approach was applied in case of the entities presented in the conceptual E-R model named as "TEAM_EVALUATION" and "EVENT_EVALUATION". Since both entities are part of the entity "EVENT_OUTCOME", with which both of the disposed of a "1:1" relationship it was technically and logically possible to incorporate both of them to the entity "EVENT_OUTCOME". On the other side, it would be also possible to leave the entities "TEAM_EVALUATION" and "EVENT_EVALUATION" their "independent" character. However, this incorporation may contribute to the higher transaction speed of the database management system (Coronel and Morris 2016, 149). One could register that an identical situation of "one to one" relationship is to be found between the entities "EVENT" and "BUDGET". The resemblance of this relationship was not change in the course of the transformation of the conceptual to the relational model, what means that the entity "BUDGET" was preserved. This decision was based on the fact that, in the future the entity "BUDGET" might require to be populated by additional attributes such as "Version" and "Type" in order to distinguish between possible several states of affair regarding the budget, and there was no unequivocal expectation considering the information requirement put on the "BUDGET" entity.

The aforestated information requirements gave birth to the concrete attributes assigned to the entities. The presented relational model, including its entities and attributes, is mirrored by the logical design of the whole relational schema that is to be proposed below:

ACADEMIC_TITLE (<u>ID</u>, TITLE_ABBREVIATION_NAME)
PK: ID

ACCOUNT (<u>ID</u>, BUSINESS_PARTNER_ID, EMAIL, PASSWORD, STATE_ID)
PK: ID
FK1: fk-ACCOUNT.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID
FK2: fk-ACCOUNT.STATE_ID ◊ pk-ACCOUNT_STATE.ID)

ACCOUNT_STATE (<u>ID</u>, NAME)
PK: ID

ADDRESS (<u>ID</u>, STREET_NAME, STREET_NUMBER, BUILDING_NUMBER, POSTAL_CODE, CITY, COUNTRY_ID)
PK: ID
FK1: fk-ADDRESS.COUNTRY_ID ◊ pk-COUNTRY.ID)

ATTENDANCE (<u>ID</u>, ATTENDANCE_DATE, EMPLOYEE_ID, ATTENDANCE_START_TIME, ATTENDANCE_END_TIME, STATE_ID)
PK: ID
FK1: fk-ATTENDANCE.EMPLOYEE_ID ◊ pk-EMPLOYEE.ID
FK2: fk-ATTENDANCE.STATE_ID ◊ pk-ATTENDANCE_STATE.ID

ATTENDANCE_STATE (<u>ID</u>, NAME)
PK: ID

BUDGET (<u>ID</u>, EVENT_ID, ACCOMMODATION, TRANSFER, CATERING, SPACE_RENT, SPACE_BUILT_UP, GUEST_SERVICES, TRANSLATION_SERVICES, TECHNICAL_SERVICES, OTHER_SERVICES)
PK: ID
FK: fk-Budget.Event_ID ◊ pk-Event.ID

BUSINESS_CONSULTATION (<u>ID</u>, NAME, EMPLOYEE_ID, BUSINESS_PARTNER_ID, ADDRESS_ID, CONSULTATION_START_TIME, CONSULTTION_END_TIME)

PK:ID
FK1: fk-BUSINESS_CONSULTATION.EMPLOYEE_ID ◊ pk-EMPLOYEE.ID
FK2: fk-BUSINESS_CONSULTATION.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID
FK3: fk-BUSINESS_CONSULTATION.ADDRESS_ID ◊ pk-ADDRESS.ID

BUSINESS_PARTNER (ID, PARTNER_TYPE_ID, IDENTIFICATION_NUMBER, SECTOR_ID, PARTNER_SHORT_DESCRIPTION, PHONE_CONTACT, EMAIL_CONTACT, ADDRESS_ID)
PK: ID
FK1: fk-BUSINESS_PARTNER.PARTNER_TYPE_ID ◊ pk-BUSINESS_PARTNER_TYPE.ID
FK2: fk-BUSINESS_PARTNER.SECTOR_ID ◊ pk-SECTOR.ID
FK3: fk-BUSINESS_PARTNER.ADDRESS_ID ◊ pk-ADDRESS.ID

BUSINESS_PARTNER_REPRESENTATIVE (ID, BUSINESS_PARTNER_ID, FIRST_NAME, LAST_NAME, DATE_OF_BIRTH, SEX, EMAIL, PHONE, POSITION)
PK: ID
FK1: fk-BUSINESS_PARTNER_REPRESENTATIVE.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID

BUSINESS_PARTNER_TYPE (ID, NAME)
PK: ID

COOPERATION (ID, BUSINESS_PARTNER_ID, EVENT_ID)
PK: ID
FK1: fk-COOPERATION.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID
FK2: fk-COOPERATION.EVENT_ID ◊ pk-EVENT.ID

COUNTRY (ID, NAME)
PK: ID

EDUCATION_INSTITUTE (ID, NAME)
PK: ID

EMPLOYEE (ID, HIERARCHY_ID, FIRST_NAME, LAST_NAME, DATE_OF_BIRTH, SEX, EMAIL, PHONE, HIGHEST EDUCATION_INSTITUTE_ID, ACADEMIC_TITLE_ID, PRIO_TERRITORIAL_FOCUS_ID, LANGUAGE_SKILL_ID, ADDRESS_ID)
PK: ID
FK1: fk-EMPLOYEE.HIERARCHY_ID ◊ pk-EMPLOYEE_HIERARCHY.ID
FK2: fk-EMPLOYEE.HIGHEST_EDUCATION_INSTITUTE_ID ◊ pk-EDUCATION_INSTITUTE.ID
FK3: fk-EMPLOYEE.ACADEMIC_TITLE_ID ◊ pk-ACADEMIC_TITLE.ID
FK4: fk-EMPLOYEE.PRIO_TERRITORIAL_FOCUS_ID ◊ pk-TERRITORIAL_FOCUS.ID
FK5: fk-EMPLOYEE.LANGUAGE_SKILL_ID ◊ pk-LANGUAGE_SKILL.ID
FK6: fk-EMPLOYEE.ADDRESS_ID ◊ pk-ADDRESS.ID

EMPLOYEE_HIERARCHY (ID, NAME)
PK: ID

EMPLOYMENT_CONTRACT (ID, EMPLOYEE_ID, STARTE_DATE, END_DATE, SIGNED_DATE, TYPE_ID)
PK: ID
FK1: fk-EMPLOYMENT_CONTRACT.EMPLOYEE_ID ◊ pk-EMPLOYEE.ID
FK2: fk-EMPLOYMENT_CONTRACT.TYPE_ID ◊ pk-EMPLOYMENT_CONTRACT_TYPE.ID

EMPLOYMENT_CONTRACT_TYPE (ID, NAME)
PK: ID

EVENT (ID, NAME, TYPE_ID, SECTOR_ID, ADDRESS_ID, SPACE_SIZE, STATE_ID, EVENT_START_TIME, EVENT_END_TIME)
PK: ID
FK1: fk-EVENT.TYPE_ID ◊ pk-EVENT_TYPE.ID
FK2: fk-EVENT.SECTOR_ID ◊ pk-SECTOR.ID
FK3: fk-EVENT.ADDRESS_ID ◊ pk-ADDRESS.ID
FK5: fk-EVENT.STATE_ID ◊ pk-EVENT_STATE.ID

LANGUAGE (ID, NAME)
PK: ID

LANGUAGE_SKILL (EMPLOYEE_ID, LANGUAGE_ID, NAME)
PK: Employee_ID && Language_ID
FK1: fk-LANGUAGE_SKILL.LANGUAGE_ID ◊ pk-LANGUAGE.ID
FK2: fk-LANGUAGE_SKILL.EMPLOYEE_ID ◊ pk-EMPLOYEE.ID

EVENT_EVALUATION (ID, NAME)
PK: ID

EVENT_OUTCOME (ID, BUSINESS_PARTNER_ID, EVENT_ID, NUMBER_OF_MEETINGS, NUMBER_OF_CONTRACTS_SIGNED, VOLUME_OF_CONTRACTED_COOPERATION, VOLUME_OF_ANTICIPATED_COOPERATION, TEAM_EVALUATION_ID, EVENT_EVALUATION_ID)
PK: ID
FK1: fk-EVENT_OUTCOME.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID
FK2: fk-EVENT_OUTCOME.EVENT_ID ◊ pk-EVENT.ID

FK3: fk-EVENT_OUTCOME.TEAM_EVALUATION ◊ pk-TEAM_EVALUATION.ID
FK4: fk-EVENT_OUTCOME.EVENT_EVALUATION ◊ pk-EVENT_EVALUATION.ID

EVENT_STATE (ID, NAME)
PK: ID

EVENT_TYPE (ID, NAME)
PK: ID

EXPORT_INITIATIVE (ID, NAME, BUSINESS_PARTNER_ID, PRODUCT_GROUP_ID, SECTOR_ID, COUNTRY_ID)
PK: ID
FK1: fk-EXPORT_INITIATIVE.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID
FK2: fk-EXPORT_INITIATIVE.PRODUCT_GROUP_ID ◊ pk-PRODUCT_GROUP.ID
FK3: fk-EXPORT_INITIATIVE.SECTOR_ID ◊ pk-SECTOR.ID
FK4: fk-EXPORT_INITIATIVE.COUNTRY_ID ◊ pk-COUNTRY.ID

PARTICIPATION (ID, EVENT_ID, BUSINESS_PARTNER_REPRESENTATIVE_ID)
PK: ID
FK1: fk-PARTICIPATION.EVENT_ID ◊ pk-EVENT.ID
FK2: fk-PARTICIPATION.BUSINESS_PARTNER_REPRESENTATIVE_ID ◊ pk-BUSINESS_PARTNER_REPRESENTATIVE.ID

POSSIBLE_OPPORTUNITY (ID, NAME, EMPLOYEE_ID, SECTOR_ID, COUNTRY_ID, OPPORTUNITY_START_TIME, OPPORTUNITY_END_TIME)
PK: ID
FK1: fk-POSSIBLE_OPPORTUNITY.EMPLOYEE_ID ◊ pk-EMPLOYEE.ID
FK2: fk-POSSIBLE_OPPRTUNITY.SECTOR_ID ◊ pk-SECTOR.ID
FK3: fk-POSSIBLE_OPPORTUNITY.COUNTRY.ID ◊ pk-COUNTRY.ID

PRODUCT_GROUP (ID, NAME)
PK: ID

REGISTRATION (ID, BUSINESS_PARTNER_ID, EVENT_ID, CONTRACT_ID, STATE_ID)
PK: ID
FK1: fk-REGISTRATION.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID
FK2: fk-REGISTRATION.EVENT_ID ◊ pk-EVENT.ID
FK3: fk-REGISTRATION.CONTRACT_ID ◊ pk-REGISTRATION_CONTRACT.ID
FK4: fk-REGISTRATION.STATE_ID ◊ pk-REGISTRATION_STATE.ID

REGISTRATION_CONTRACT (ID, REGISTRATION_ID, NAME, STATE_ID)
PK: ID
FK1: fk-REGISTRATION_CONTRACT.REGISTRATION_ID ◊ REGISTRATION.ID
FK2: fk-REGISTRATION_CONTRACT.STATE_ID ◊ REGISTRATION_CONTRACT_STATE.ID

REGISTRATION_CONTRACT_STATE (ID, NAME)
PK: ID

REGISTRATION_STATE (ID, NAME)
PK: ID

SECTOR (ID, NAME)
PK: ID

SUBSECTOR (ID, SECTOR_ID, NAME)
PK: ID
FK1: fk-SUBSECTOR.SECTOR_ID ◊ pk-SECTOR.ID

TEAM_EVALUATION (ID, NAME)
PK: ID

TERRITORIAL_FOCUS (ID, NAME)
PK: ID

TICKET (ID, NAME, EVENT_ID, EMPLOYEE_ID)
PK: ID
FK1: fk-TICKET.EVENT_ID ◊ pk-EVENT.ID
FK2: fk-TICKET.EMPLOYEE_ID ◊ pk-EMPLOYEE.ID

VOTING (ID, OPPORTUNITY_ID, BUSINESS_PARTNER_ID, VOTIG_TIME)
PK: ID
FK1: fk-VOTING.OPPORTUNITY_ID ◊ pk-POSSIBLE_OPPORTUNITY.ID
FK2: fk-VOTING.BUSINESS_PARTNER_ID ◊ pk-BUSINESS_PARTNER.ID

5.3 Structured Query Language – Query Samples

The logical model introduced on the previous pages creates a base for the physical design and thus for the implementation of our final artifact – the database for the future application of the National Project Department – Support of Internationalization of the Small and Medium Enterprises that is part of the Slovak Investment and Trade Development Agency. It was decided that it is sufficient to provide the readers just with certain basic examples of the database implementation. Therefore the following queries, corresponding to theoretical basis of this thesis, are to be introduced in the following lines.

SQL Statement used for creation of a database named “AgencyDB”:

```
CREATE DATABASE AgencyDB;
```

SQL Statement used for creation of a table named “Event” with its attributes and foreign keys:

```
CREATE TABLE Event (
ID int NOT NULL AUTO_INCREMENT,
Name varchar(255),
Sector_ID int,
Address_ID int,
Space_Size int,
Event_Start_Time datetime,
Event_End_Time datetime,
PRIMARY KEY (ID)
FOREIGN KEY (Sector_ID) REFERENCES Sector(ID),
FOREIGN KEY (Address_ID) REFERENCES Address(ID),
FOREIGN KEY (Budget_ID) REFERENCES Budget(ID),
);
```

The statement about does not exclusively serve to the table creation. We have also defined the primary key of this relation – “ID”, and also the foreign keys – “Sector_ID”, “Address_ID” and “Budget_ID”. Moreover, the “auto-increment” feature was added to the table, to the attribute “ID”, what means that the attribute “ID” is to be incremented by one if a new record/occurrence is added to the table. Of course, the statement written above counts with the creation of the relations “ADDRESS”, “SECTOR” and “BUDGET”. The attribute “ID” was equipped with a constraint “NOT NULL”, what implies that every single occurrence of this relation has to possess an ID value different from “NULL”. Furthermore, the attributes were assigned with certain database types – “int” for integer values; “varchar” stands for “variable-size string data” in case of the “Name” attribute its maximum length cannot exceed 255 characters; “datetime” for a date and time value represented in the format of “YYYY-MM-DD hh:mm:ss”.

SQL Statement used for a record insertion into the table “Employee”:

```
INSERT INTO Employee (ID, Hierarchy_ID, First_Name, Last_Name, Date-Of_Birth, Sex, Email, Phone, Highest_Education_Institute_ID, Academic_Title_ID, Prio_Territorial_Focus_ID, Address_ID) VALUES (`7`, `1`, `Juraj`, `Senkvicky`, `1990-11-16`, `m`, `juraj.senkvicky@npdsi.sk`, `+421917XXXXXX`, `39`, `2`, `1`, `17`)
```

SQL statement used for retrieving the First_Name, Last_Name, and Date_Of_Birth attributes of the Employee with ID=1:

```
SELECT Employee.First_Name, Employee.Last_Name, Employee_Date_Of_Birth

FROM Employee

WHERE Employee.ID = 7
```

The statement above would provide us with the following output:

First_Name	Last_Name	Date_Of_Birth
Juraj	Senkvicky	1990-11-16

5.4 Dashboard Design

The secondary or complementary artifact and goal of this thesis is to be achieved by a dashboard design. The resulting dashboard design shall represent a dashboard that will be at the disposal of the leading manager of the NPDSI. The business requirements regarding the information resemblance of this dashboard were already mentioned in the corresponding part of this text. Just to emphasize the following information requirements were defined: budgetary information on the budget already spent; budgetary information on particular activities; number of organized events; number of events in progress; companies sectors; the most desired future events; number of companies registered; number of newly registered companies in the course of the recent 3 months; export destinations mentioned in the last 6 months; calendar indicating basic information on the activities of the NPDSI employees. As one can see, the list of information defined in the information requirements does not belong to the shortest ones. Therefore, certain alternative options accompanied with shortcut possibilities were incorporated to the leading manager dashboard. In conclusion it was arrived at the following solution displayed in the next figure.

Figure 13: Dashboard Design

The following lines are to comment of the final dashboard design. The displayed design is a result of the third iteration process. The original list contained nine information requirements and this was not changed. Therefore, at a first sight we can register the presence of the following 7 requirements: general budget information represented by a box named "Budget Spent" expressing the percentage of the budget already spent; number of registered companies represented by the box "Unique Companies" showing the number of unique companies already registered at the NPDSI´s events; number of newly registered companies (in course of the last three months) indicated by the box "New Companies"; number of events in progress shown by the box "Prepared Events"; export destination countries defined by the entrepreneurs in the last 6 months is represented by the box "Export Destinations"; the companies´ sectoral diversification is presented by a graph "The Mot Favourite Sectors"; the rating of the opportunities announced by the employees is to be seen in the graph called "The Most Desired Opportunities" where the votes for the particular favourite opportunities are presented; the basic information on the employees´ activities is represented in a calendar form.

The dashboard is also equipped with the top application bar, where the following icons are present:

"Left arrow" – designed to be used to fulfil the potential "move back" functionality,

"Person" – designed as a shortcut to more detailed information on the employees' activities,

"Company" – designed as a shortcut to more detailed information on the business partners, “My Events” – designed as a shortcut to more detailed information on the events of the leading manager,

“Info” – designed as a shortcut to information on the application manual,

“Home” – designed as a shortcut to go to the “home-dashboard”,

“Today´s date” – designed in a format “DD.MM.”,

“Minimize Icon” – designed to provide the option to minimize the window,

“Miniature Icon” – designed to provide the option to miniature the window,

“Close Icon” – designed to provide the option to close the window.

To continue horizontally in the explanation of the dashboard design, a compressed “menu bar” displayed with the currently chosen menu option is placed under the top bar. The arrow next to the text line “GENERAL OVERVIEW” indicates the presence of the drop-down list of more menu options. Regarding the boxes and graphs, each option – except for the graph displaying the most desired opportunities – is accompanied with the “plus sign”, which indicates an existence of a shortcut providing access to more detailed information on the particular “topic”. Focussing on the graph “The most Favourite Sectors”, the arrow signalizes the option to change the topic of the graph. Considering the calendar, this disposes of several “personalization option” – the arrow in the “Employee” field is designed in order to provide the option to select “n” employees, and the arrow in the “Mode” field serves to choose from the calendar display modes – for instance a concrete day or a concrete month could embody the displaying options. The bars placed under the concrete dates in the calendar indicate the names of the events an employee is currently working on, whereas the bar colours distinguish between the individual employees. The “plus sign” provides the opportunity to load more information on the particular activity.

The colours chosen for the purpose of this dashboard design does not fully correspondent to the identity of the agency´s department. However, this demand was not summed up in the requirements. For that reason, the dark blue colour was chosen as a basis for the dashboard outline and for the headers background. The pure white colour provides a sufficient contrast as a background colour for the figures, and for the same reason the golden colour was applied in the column chart. The remaining colours were used mainly in order to secure the already mentioned efficient contrast to make the text content conveniently readable and thus to contribute to the usability of the future application.

6. Conclusion

This thesis was devoted to the search for the most suitable database type and its consequential design and for the dashboard design. Both of the final artifacts – the database design and the dashboard design are to be at disposal of the future CRM&BI application of the National Project Department – Support of Internationalization of the Small and Medium Enterprises, which is an inevitable part of the SARIO Agency, the Slovak Investment and Trade Development Agency working under auspices of the Ministry of Economy of the Slovak Republic. Excluding this conclusive part of the thesis, its text was divided into the following five bearing chapters: Introduction, Databases, CRM&BI Systems, The Information On The Agency, Database and Dashboard Design. The introductory part provided the readers with the main methodological approaches that should have been applied in this research – comparative analysis, design science research and weighted scoring model. This methodological concepts were chosen to be applied in order to contribute to the search for the most suitable answer to the main research question which was posed as follows: "What is the most appropriate and applicable database type for a CRM/BI application for a state export agency which disposes of a rather unstructured data storage?". According to the hypothesis , MySQL relational database should have been regarded as the most appropriate one for this particular case. Following the aim to answer the main question and to challenge the proposed hypothesis the following four questions were elaborated: What relevant database management systems are currently available; What are the relevant perceivable differences between the relational and non-relational database management systems; What are the relevant perceivable differences among the relational database management systems; What are the relevant perceivable differences among the non-relational database management systems. The answer on the first sub-question of the research was provided in the lines of the second chapter named "Databases", which was filled by the relevant information on the theory of database management systems, and in particular on the relational and the so-called non-relational database management systems. The same chapter provided the readers with answers to the second fractional research question on the differences between the relational and non-relational database systems, which at first sight lie mainly in their "relation" to the relational models, schema, manipulation language, maintainability, suitability for the structured/semi-structured and unstructured data storage and scalability.

Bearing in mind, that one of the fundamental requirements on the database type was the cost-efficiency and/or cost-effectiveness, the database management systems available without any fees were exclusively taken into consideration. Regarding the traditional, relational sphere, the two most popular database management systems – MySQL and PostgreSQL were described and compared. On the opposite spectrum the different types of the “NoSQL” databases were mentioned and described, whereas the three popular representatives, MongoDB, Apache Cassandra and Appache HBase were characterized and compared and thus all of the fractional research questions were answered. The theoretical part of this bachelor thesis was completed by the third chapter named “CRM&BI Systems”, since the designed database and dashboard shall be an inevitable part or at least a starting basis for the creation of the new CMR&BI application. However, regarding the fact that the dashboard design was only of a “complementary” character this this part provided a relatively briefly the essential information on the topic.

The second part of this thesis begins with the chapter named “The Information On The Agency” which provides a description of the Slovak Investment and Trade Development Agency position, structure and its activities, since the future application shall be used by part of this entity. Particularly, an emphasis was put on its concrete department called “National Project Department – Support of Internationalisation of the Small and Medium Enterprises”. Based on the analysis of the information gained from the public source, gained in the course of brainstorming, unstructured interviews with the current employees of the department and also derived from a personal experience – since the thesis author is a former employee of the agency – the requirements or priorities on the suitable database type were defined and weighted. As a result the weighted scoring model was elaborated with the corresponding requirements – maintainability, data storage accessibility, read performance, scalability, security, structured data appropriateness, OLAP + OLTP orientation – that were assigned with the corresponding weight. The knowledge base gained in the course of this research and documented in the theoretical part of this thesis together with the information on the agency and their requirements were decisive by “grading” particular “qualities” – requirements of the concrete database types. The outcome of this weighted scoring model provided us with the answer that MySQL is the most appropriate and applicable database type for a CRM/BI application for a state export agency which disposes of a rather unstructured data storage. It is the current data storage systems that is regarded to be unstructured not the data to be stored, if this was referred to the data, MySQL would not be evaluated as the most appropriate database type. Maintainability, and “universality” in terms of representing an ideal intersection for the application as well as for the transaction processing proved themselves to be of a decisive character. The “structural” orientation of the requirements more-or-less disqualified the non-relational database management systems, since their maintenance is more demanding. It is important to remark that PostgreSQL outperforms MySQL capabilities when it comes to security and scalability. On the other side it is fair to note that the performance of al – presumably – most of the requirements is possible to be tuned up by the right configuration of the software and hardware elements of the concrete system. Therefore the maintainability and the human factor might be of the most decisive character when choosing the right technology or even a database type.

Nevertheless, regarding the presented requirements the MySQL was chosen as the best possible alternative for the future CRM&BI application. Therefore, the fifth bearing part provided a space for the database and dashboard design. Both processes were designed in accordance with the requirements. The database design began with text-presentation of the possible use-cases which was followed by the conceptual entity relationship diagram filled with entities, relationships and their connectivities, which can be considered as a product of the second iteration. The successive step was to elaborate the relational database model. Since certain “M:N” and “1:1” relationships between the several entities were recorder in the final conceptual model, the relational model should have settled these deficiencies. This was accomplished either by constructing the so-called composite entities or by the entities incorporation. Elaboration of the relational model was performed simultaneously with and is mirrored in the text-presentation of the so-called logical model, which outlines the primary and foreign keys of particular relations. The relational model lacks concrete data types that could be assigned to the attributes, what is partially compensated by introduction of the queries samples creating a table with defined data types. The last bearing part of this thesis is dedicated to the secondary artifact – dashboard design of the future application. This contains the information requirements on the dashboard design for the leading manager, the design itself and an explanative description of the elaborated dashboard design.

Bibliography

Book Resources

Cavanillas, José María, Edward Curry, and Wolfgang Wahlster. 2016. New Horizons for a Data-Driven Economy. Cham Springer International Publishing.

Coronel, Carlos, and Steven Morris. 2016. Database Systems Design, Implementation, and Management. 12th ed. Boston, Ma Cengage Learning.

Date, C J. 2004. An Introduction to Database Systems. Boston: Pearson/Addison Wesley.

Date, C J. 2015. SQL and Relational Theory : How to Write Accurate SQL Code. Sebastopol, Ca: O'reilly Media, Inc.

Date, C. J. 2012. Normal Forms and All That Jazz a Database Professional's Guide to the Theory of Database Design. Sebastopol, Ca O'reilly & Associates.

Desai, Bipin C. 1992. An Introduction to Database Systems. St. Paul: West Publ. Co.

Dresch, Aline, Daniel Pacheco Lacerda, and ValleAntônio. 2015. Design Science Research : A Method for Science and Technology Advancement. Cham: Springer.

Elmasri, Ramez, and Shamkant B Navathe. 2016. Fundamentals of Database Systems. Hoboken, New Jersey: Pearson.

Few, Stephen. 2006. Information Dashboard Design: The Effective Visual Communication of Data. Edited by Colleen Wheeler. Sebastopol: O´Reilly Media.

Garcia-Molina, Hector, Jeffrey D Ullman, and Jennifer Widom. 2009. Database Systems. Prentice Hall.

Greenberg, Paul. 2009. CRM at the Speed of Light, Fourth Edition : Social CRM 2.0 Strategies, Tools, and Techniques for Engaging Your Customers /. New York, N.Y.: Mcgraw-Hill Education

Harrison, Guy. 2015. Next Generation Databases. Berkeley, Ca Apress.

Juba, Salahaldin, Achim Vannahme, and Andrey Volkov. 2015. Learning PostgreSQL. Packt Publishing Ltd.

Luc Perkins, Eric Redmond, and Jim R Wilson. 2018. Seven Databases in Seven Weeks : A Guide to Modern Databases and the NoSQL Movement. Raleigh, N.C.: The Pragmatic Bookshelf

Oppel, Andy. (2005) 2012. SQL Bez Předchozích Znalostí. Translated by Jan Gregor and Jakub Mikulaštík. New York/Brno: The McGraw-Hill/Computer Press.

Plato. 2007. The Republic. Translated by Desmond Lee. London: Penguin Books Ltd.

Vineet Kumar, and Werner J Reinartz. 2018. Customer Relationship Management : Concept, Strategy, and Tools. Berlin: Springer, Cop.

Wexler, Steve, Jeffrey Shaffer, and Andy Cotgreave. 2017. The Big Book of Dashboards: Visualizing Our Data Using Real-World Business Scenarios. Hoboken: John Wiley & Sons.

Article Resources

Al-Homery, Hussein A., Hasbullah Asharai, and Azizah Ahmad. 2019. The Core Components and Types of CRM. Pakistan Journal of Humanities and Social Sciences 7 (1): 121–45. https://doi.org/2415-007X

Codd, E. F. 1970. “A Relational Model of Data for Large Shared Data Banks.” Communications of the ACM 13 (6): 377–87. https://doi.org/10.1145/362384.362685.

Gibson, Marcus, David Arnott, and Ilona Jagielska. 2004. Evaluating the Intangible Benefits of Business Intelligence: Review & Research Agenda. In Decision Support in an Uncertain World: Proceedings of the 2004 IFIP International Conference on Decision Support Systems (DSS2004), edited by Rob Meredith, Graeme Shanks, David Arnott, and Sven Carlsson, 295–305. Melbourne: Monash University Publishing.

Hayley, Michael. 2016. A Literature Review on CRM - Definitions, Benefits, Components, and Implementation. Australian Journal of Management and Financial Research 1 (1): 26–34.

Hevner, Alan, Jinsoo Park, Salvatore T. March, and Sudha Ram. 2004. Review of Design Science in Information Systems Research. MIS Quarterly 28 (1): 75–105. https://www.researchgate.net/publication/201168946.

Jain, Sanjay, Singh Gulab and M M Gore. 2003. „Database Design for Nationwide Information System to Govern Country Effectively.

Lourenço, João Ricardo, Bruno Cabral, Paulo Carreiro, Marco Vieira, and Jorge Bernardino. 2015. “Choosing the Right NoSQL Database for the Job: A Quality Attribute Evaluation.” Journal of Big Data 2 (1). https://doi.org/10.1186/s40537-015-0025-0.

Luhn, Hans Peter. 1958. A Business Intelligence System. IBM Journal of Research and Development 2 (4): 314–319. https://doi.org/10.1147/rd.24.0314.

Manoj, V. 2014. Comparative Study of NoSQL Document, Column Store Database and Evaluation of Cassandra. International Journal of Database Management Systems (IJDMS) 6 (4): 11–26. https://doi.org/10.5121/ijdms.2014.6402.

MySQL. 2022. MySQL 8.0 Reference Manual. MySQL. Redwood Shores: Oracle. https://dev.mysql.com/doc/refman/8.0/en/.

Nedelcu, Bogdan. 2013. Business Intelligence Systems. Database Systems Journal 4 (4): 12–20. http://www.dbjournal.ro/archive/14/14_2.pdf.

Padhy, Rabi Prasad, Manas Ranjan Patra, and Suresh Chandra Satapathy. 2011. RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Databases. (IJAEST) INTERNATIONAL JOURNAL of ADVANCED ENGINEERING SCIENCES and TECHNOLOGIES 11 (1): 15–30. https://www.researchgate.net/publication/265062016.

PostgreSQL. 2022. PostgreSQL 15.1 Documentation. PostgreSQL. Berkeley: The PostgreSQL Global Development Group. https://www.postgresql.org/docs/current/index.html.

Sá, Filipe, Pedro Martins, and Maryam Abbasi. 2021. “Portuguese Local E-Government: A Study on the Most Adopted Databases in 2019.” Journal of E-Government Studies and Best Practices, May, 1–12. https://doi.org/10.5171/2021.254977

Singh, Amar J., and Rajesh Chauhan. 2010. “E-Government Databases: A Retrospective Study” Indian Journal of Computer Science and Engineering 1 (2): 66–73. https://doi.org/10.21817/indjcse/2019/v10i4/191004010.

SARIO – Slovenská agentúra pre rozvoj investícií a obchodu. 2022. Výročná správa 2021. Bratislava. SARIO – Slovenská agenúra pre zvoj investícií a obchodu. https://sario.sk/sites/default/files/sario-vyrocna-sprava-2021.pdf.

The British Academy, and The Royal Society. 2017. Data Management and Use: Governance in the 21st Century. The British Academy. London: The British Academy and The Royal Society. https://www.thebritishacademy.ac.uk/documents/105/Data_management_and_use_-_Governance_in_the_21st_century.pdf#:~:text=3%20Data%20management%20and%20use%3A%20Governance%20in%20the,success%20and%20failure%20-%20enhance%20existing%20democratic%20governance.

The Government of The United Kingdom Department for Digital, Culture, Media & Sport. 2020. National Data Strategy. National Data Strategy. London: The Government of The United Kingdom Department for Digital, Culture, Media & Sport. https://www.gov.uk/government/publications/uk-national-data-strategy/national-data-strategy#data-1-1.

Vaishnavi, Vijay, and Bill Kuechler, eds. 2021. Design Science Research in Information Systems, November. http://desrist.org/design-research-in-information-systems/.

Webseite Resources

AWS Amazon Web Services. n.d. What Is PostgreSQL? AWS Amazon Web Services. AWS Amazon Web Services. Accessed December 23, 2022. https://aws.amazon.com/de/rds/postgresql/what-is-postgresql/.

Chen, Brandon. 2021. PostgreSQL vs. MySQL: What You Need to Know. Fivetran. Fivetran. September 2, 2021. https://www.fivetran.com/blog/postgresql-vs-mysql.

Comparative Analysis: Flipped Learning Module. 2021. Boston University Teaching Writing. Trustees of Boston University. 2021. https://www.bu.edu/teaching-writing/resources/comparative-analysis/.

DataStax. 2018. The Evolution of NoSQL. DataStax. DataStax Inc. August 29, 2018. https://www.datastax.com/blog/evolution-nosql.

DB-Engines. 2022a. System Properties Comparison MySQL vs. PostgreSQL. DB-Engines. DB-Engines. Accessed 12, 2022. https://db-engines.com/en/system/MySQL%3BPostgreSQL.

DB-Engines. 2022b. System Properties Comparison Cassandra vs. HBase vs. MongoDB. DB-Engines. DB-Engines. Accessed 12, 2022. https://db-engines.com/en/system/MySQL%3BPostgreSQL.

Devart. n.d. PostgreSQL vs MySQL. Blog.Devart. Devart. Accessed December 26, 2022. https://blog.devart.com/postgresql-vs-mysql.html.

EnterpriseDB. 2019. PostgreSQL vs. MySQL: A 360-Degree Comparison [Syntax, Performance, Scalability and Features]. EDB. EnterpriseDB. December 11, 2019. https://www.enterprisedb.com/blog/postgresql-vs-mysql-360-degree-comparison-syntax-performance-scalability-and-features.

Foote, Keith D. 2021. A Brief History O Database Management. DATAVERSITY. DATAVERSITY. October 21, 2021. https://www.dataversity.net/brief-history-database-management/#.

IBM. n.d. What Is PostgreSQL? IBM. IBM. Accessed December 25, 2022. https://www.ibm.com/topics/postgresql#:~:text=PostgreSQL%2C%20commonly%20pronounced%20%E2%80%9CPost-GRES%2C%E2%80%9D%20is%20an%20open%20source,ibm.com%29%20supports%20both%20non-relational%20and%20relational%20data%20types.

Industry 4.0 and the Fourth Industrial Revolution Explained. n.d. I-SCOOP. i-SCOOP. Accessed September 10, 2022. https://www.i-scoop.eu/industry-4-0/.

Kakarla, Swaathi. 2020. PostgreSQL vs MySQL. Sumo Logic. Sumo Logic. October 1, 2020. https://www.sumologic.com/blog/postgresql-vs-mysql/.

Kinsta. 2022. What Is PostgreSQL? Kinsta. Kinsta Inc. November 18, 2022. https://kinsta.com/knowledgebase/what-is-postgresql/.

Klein, Evan. n.d. Cassandra vs. MongoDB vs. Hbase: A Comparison of NoSQL Databases. Logz.io. Logz.io. Accessed January 5, 2023. https://logz.io/blog/nosql-database-comparison/.

Luzanov, Pavel. 2016. Sravnenie MySQL I PostgreSQL. Losst. Losst. December 20, 2016. https://losst.pro/sravnenie-mysql-i-postgresql.

MongoDB. n.d. Try MongoDB Community Edition. MongoDB. MongoDB. Accessed December 26, 2022. https://www.mongodb.com/try/download/community.

Phaujdar, Amit. 2021. PostgreSQL vs MySQL: 11 Critical Differences. HEVO. Hevo Data. December 1, 2021. https://hevodata.com/learn/postgresql-vs-mysql/.

Ravoof, Salman. 2022. PostgreSQL vs MySQL: Explore Their 12 Critical Differences. Kinsta. Kinsta. November 21, 2022. https://kinsta.com/blog/postgresql-vs-mysql/.

Romanelli, Mark. n.d. The Weighted Scoring Model - Project Management Tools & Techniques. Stakeholdermap. T Murphy. Accessed September 15, 2022. https://www.stakeholdermap.com/project-management/weighted-scoring-model.html.

Satori Cyber. 2021. Guide: Postgre Security: 3 Pillars of PostgreSQL Security. Satoricyber. Satori. June 16, 2021. https://satoricyber.com/postgres-security/3-pillars-of-postgresql-security/#:~:text=PostgreSQL%20Transport-Level%20Security%20%28TLS%29%20TLS%20is%20a%20secure,enable%20secure%20authentication%20for%20clients%2.

SARIO – Slovenská agentúra pre rozvoj investícií a obchodu. 2023a. „Kto sme“. Accessed March 01, 2023. https://sario.sk/sk/o-nas/kto-sme.

SARIO – Slovenská agentúra pre rozvoj investícií a obchodu. 2023b. „Podpora internacionalizácie MSP“. Accessed March 01, 2023. https://sario.sk/sk/np.

Smallcombe, Mark. 2022. PostgreSQL vs MySQL: The Critical Differences. Integrate.io. Integrate. September 23, 2022. https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/.

Stedman, Craig, and Ed Burns. 2020. Business Intelligence (BI). TechTarget: Business Analytics. TechTarget. September 1, 2020. https://www.techtarget.com/searchbusinessanalytics/definition/business-intelligence-BI.

The Apache Software Foundation. 2022a. What Is Apache Cassandra? Cassandra Basics. The Apache Software Foundation. 2022. https://cassandra.apache.org/_/cassandra-basics.html.

The Apache Software Foundation. 2022b. Cassandra Query Language (CQL) v3.4.3. Cassandra Documentation. The Apache Software Foundation. https://cassandra.apache.org/doc/latest/cassandra/cql/cql_singlefile.html

The Apache Software Foundation. 2022c. Welcome to Apache HBase. Hbase.apache. The Apache Software Foundation. December 1, 2022. https://hbase.apache.org/.

United States Geological Survey. n.d. What Are the Differences between Data, a Dataset, and a Database? United States Geological Survey. United States Department of Interior. Accessed December 7, 2022. https://www.usgs.gov/faqs/what-are-differences-between-data-dataset-and-database.

Vinchin. 2022. PostgreSQL vs MySQL: What Are the Differences and When to Use. Vinchin Blog. Vinchin. November 11, 2022. https://www.vinchin.com/en/blog/postgresql-vs-mysql.html.

Weller, Joe. 2021. Everything You Need to Know about Scoring Models in Project Management. Smartsheet. Smartsheet Inc. August 31, 2021. https://www.smartsheet.com/content/project-scoring.

List of Abbreviations Used

ACID	Atomicity, Consistency, Isolation, Durability
ACS	Access Control Security
API	Application Programming Interface
BCNF	Boyce-Codd Normal Form
BI	Business Intelligence
BSON	Binary JavaScript Object Notation
CAP Theorem	Consistency, Availability, Partition Tolerance Theorem
CQL	Cassandra Query Language
CRM	Customer Relationship Management
CRUD	Create, Read, Update, Delete
DB	Database
DBMS	Database Management System(s)
DDL	Data Definition Language
DML	Data Manipulation Language
ERP	Enterprise Resource Planning
E-R	Entity-Relationship
FK	Foreign Key
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
JSON	JavaScript Object Notation
LSM-Tree	Long-Structured Merge-Tree
MyISAM	Indexed Sequential Access Method
NoSQL	Not only Structured Query Language
NPDSI	National Project Department – Support of Internationalization of the Small and Medium Enterprises
OLAP	Online Analytical Processing

OLTP	Online Transaction Processing
OS	Operating System
OSEMN	Obtain-Scrub-Explore-Model-Interpret
PITR	Point-In-Time Recovery
PK	Primary Key
REST	Representational State Transfer
SARIO	Slovak Investment and Trade Development Agency
SCM	Supply Chain Management
SQL	Structured Query Language
TCP	Transmission Control Protocol
UK	The United Kingdom
WAL	Write Ahead Log
XML	Extensible Markup Language