SQL versus NOSQL performance in Big Data analytics

Feltrin, Bruno

University of Zagreb

Faculty of Economics and Business

Master’s Degree in Managerial Informatics

SQL VERSUS NOSQL PERFORMANCE IN BIG DATA ANALYTICS

Master thesis

Bruno Feltrin

Zagreb, September 2022

University of Zagreb

Faculty of Economics and Business

Master's Degree in Managerial Informatics

UČINAK SQL-A U ODNOSU NA NOSQL U ANALITICI VELIKIH PODATAKA

SQL VERSUS NOSQL PERFORMANCE IN BIG DATA ANALYTICS

Master Thesis

Bruno Feltrin, 0116155253

Academic year: 2021/2022

Mentor: Professor Katarina Ćurko, Ph.D.

Subject: Data Management

Zagreb, September 2022

Summary

The paper investigates the main characteristics of SQL and NoSQL, and its comparison in the area of Big Data Analytics. Big Data leading internet companies were the first to use NoSQL databases to get around the problems with relational database management system. The paper aims to investigate on how relational database management systems aren't always the best choice because they can't keep up with the expanding amount of unstructured data. As the amount of data that needs to be processed grows at an exponential rate, NoSQL is a dynamic and cloud-friendly way to process unstructured data easily. IT professionals often argue about whether SQL or NoSQL is better, but NoSQL is becoming the new favorite of the big data movement as business data management is constantly growing. Through the paper there will be a more closer description of the SQL and NoSQL movement, with both the concepts and techniques, and also what is the future of SQL and NoSQL. Data analytics lifecycle will be discused as well as the main big data analysis techniques. Big data analytics process contains a wide variety of information, and has the potential to provide value for businesses. The complexity of big data makes it impossible to manage using the same methods and technologies that have traditionally been used. The paper examines SQL and NoSQL challenges and solutions in the view of Big Data, with the discussion of their advantages and disadvantages. In addition, there is also a discussion in the case study, where is a comparison of most popular SQL and NoSQL databases, which are MySQL and MongoDB

Table of Contents

1. INTRODUCTION

1.1. Topic and Goals of the Thesis

In the aspects of Data Management there have been two major revolutions that happened recently, and those are namely Big Data Analytics and NoSQL databases. Those two have developed with different purposes through some time, but their independent development are still complementing each other. The convergence of Big Data Analytics and NoSQL databases would greatly benefit businesses in the terms of making real-time decisions using large amounts of extremelly complicated data sets, that can be both structured or unstructured type. There have been many softwares solutions on the market that have emerged to support Big Data Analytics, as so did NoSQL database packages, however many of them lack independent benchmarking and comparative evaluations. Goal of the thesis is to provide an understanding in the context of both SQL and NoSQL, how they perform in Big Data analytics. The thesis will go to an in-depth study of comparing the four primary NoSQL data models that have developed through time. Also the aim is to analyze what are the challenges and solutions of SQL in terms of Big Data Analytics, and to have an understanding of advantages and disadvantages of both SQL and NoSQL in Big Data

1.2. Explanation of methodology

For the main purpose of writing this paper, there was a need to decide an overall approach and the main method for researching and collection of data, in order to present the paper clearly to the reader. Firstly, there was a detailed examination of available literature, which was collected both online and offline in a library provided by the faculty. To get a bigger picture of SQL and NoSQL and how they function, there will be a case study which will compare MySQL and MongoDB, and how they deal with Big Data

1.3. Structure of the Thesis

The thesis is structured in a way that firstly it will gives an overview of SQL and relational databases. Thesis will describe how the analysis is performed in SQL, and what will SQL look like in the near future, when talking about handling Big Data. After the description of SQL, there will be a closer look into the NoSQL movement, and how NoSQL functions when analyzing data. There will be an in-depth study of the four main NoSQL data models including document databases, key value store databases and graph databases. After the clear description of SQL and NoSQL, there will be a closer look on Big Data Analytics, what is it’s life cycle, and four of the main analytical methods used when analyzing data. Fifth part will be an SQL and NoSQL comparison where will be a further analysis of SQL and NoSQL on Big data and their challenges and solutions, as well as advantages and disadvantages. Also to summarize the SQL and NoSQL comparison, there will be a case study involving MySQL and MongoDB, where will be a closer look on how they perform when analyzing Big Data

2. AN OVERVIEW OF SQL

2.1. The Database Essentials

There are several approaches to take when attempting to define or explain what a database is, but one of the definitions is a structured system which can save data or information users can quickly and effectively access when the need comes.[1] There is more to a database than just a collection of tables. Multiple tiers of additional structures aid in maintaining the data's integrity. The schema of a database offers an general arrangement to all data tables. The domain of a table column defines the categories of data that can be kept in that column. A user can also put a set of restrictions to a database table to prohibit anybody from putting erroneous data in the table. The schema or conceptual picture of a whole database represents its structure.

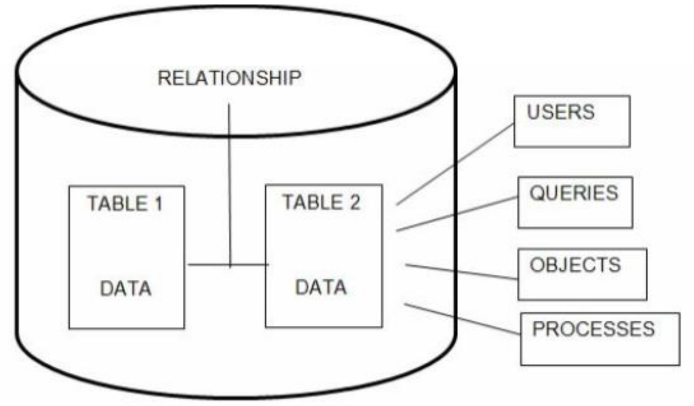
In relation to databases, this structure is frequently referred to as the "whole logical view.". As metadata, schema is a component of the database. The data storage metadata, which explains the database's layout, is kept in the tables identical to those used to store ordinary data. Also, in a database, attribute of some of the relation, may take a limited number of particular values, and a whole collection of that kind of value is the named atribute domain. Contraints are another important part of relational databases, though sometimes they are overlooked. Constraints can be defined as the rules that determine what values the table attributes can assume.[2]

The first person who formulated relational database model was Dr. Codd in 1970, after that relational databases will nearly entirely supplanted older database types. This is partly due to the fact that changing the structure of a relational database does not need changing or modifying programs relying on the previous structures. Considering the addition that there is one or many additional columns in the database table. It is not needed to modify any pre made programs that process this table only if there is a modification of some of the columns that the particular application system uses. In the following figure there will be a representation on how the different parts of a relational database are linked to each other.

[1] Alvaro, F. "SQL: Easy SQL Programming & Database Management For Beginners." Your Step-By-Step Guide To Learning The SQL Database, Amazon, eizdanje (2018)

[2] Taylor G. A., SQL for Dummies-9th Edition, John Walley & Sons Inc, Hoboken; New Jersey, (2019)

Figure 1. Relational Database Connection



Source: Alvaro, F. "SQL: Easy SQL Programming & Database Management For Beginners." Your Step-By-Step Guide To Learning The SQL Database, Amazon, eizdanje (2018)

This figure shows when rows and columns make up a table and are used to store information. When a query is entered by the user into a relational database, the user can make different kinds of queries, and because the tables are related user can easily have a data retrieval process.

When talking about Database Management Systems or DBMS is an essential piece of software that is made out of collection of programs which can design, manage, and process databases as well as any applications that are linked with them[3]. The user can also construct a structure where he can carry out operations from the important data which database can store in a manner that is highly efficient as a result of this. The conventional user, who is liable for data retrieval and modification, and the administrator, who is accountable for the ongoing upkeep of the database's structural integrity, are the two primary categories of users who operate with DBMS.

The following are a DBMS's main characteristics:[4]

- Enables the development of new databases and their corresponding data formats.
- Permits data update and inquiry by using a relevant programming language
- Enables the long-term storage of enormous volumes of data
- Allows database recovery in the event of a malfunction, mistake, or malicious abuse
- Limits simultaneous access to data by numerous individuals

[3] Silva Y. N., Almeid I., Queiroz M., SQL: From Traditional Databases to Big Data, Arizona State University, (2016)

[4] Alvaro, F. "SQL: Easy SQL Programming & Database Management For Beginners." Your Step-By-Step Guide To Learning The SQL Database, Amazon, eizdanje (2018)

2.2.SQL Basics

Whenever it comes to communicating with RDBMS, SQL is a versatile computer language which may be implemented in a number of various ways. This software is distinguished from other programming applications by a number of unique characteristics that set it apart from other programs. SQL is a language that, first and foremost, does not follow any procedures. The majority of computer programs, such as C, C++, and Java, find solutions to problems by carrying out a series of steps that are collectively referred to as a process. In this instance, a particular operation is carried out immediately after the last one that was completed in order to achieve the specified objective.

Depending on what the programmer has defined, the flow of operation may either be a sequential one that goes in a linear order or one that goes in a loop. This does not apply to SQL in any way. The user will be only required to indicate the output that he wants when using this program. The user will not be required to explain how he wishes to produce the output. Learning the syntax of SQL is similar to understanding the structure of the English language. The so called commanding language, which is built from restricted sum of statements, is responsible for performing the three fundamental data tasks of defining, manipulating, and controlling data. In addition, reserved terms are words in the SQL programming language that can only be used in certain ways. Therefore, the user is unable to use the terms in the names for some variables, tables, or columns in database; or in any other capacity that is not consistent with how they were meant to be used. Before beginning to program in SQL, users need to have a solid understanding of its fundamental command categories in order to successfully carry out various number of functions. Database construction, manipulating objects, data populating as well as updating, deletion of the data, query submitting, access control, and database administration are typical but not exclusive examples of these tasks.

- **ACID Properties**

RDBMSs maintain transaction control via the use of a method known as ACID, which stands for atomic, consistent, independent, and durable.[5] RDBMSs make advantage of such qualities to guarantee the reliability of transactions. Each of the linked attributes will now be defined in the following.

[5] McCreary D. & Kelly A., Making Sense of NoSQL; A guide for managers and the rest of us, Manning Publications, New York, (2014)

- **Atomicity**

This concept is referred regarded in the scientific community as atomicity, which originates from the Greek word for "dividable." The failure modes of a system must be taken into consideration if it claims to have atomic transactions. These failure types include disk crashes, network failures, hardware problems, and simple software mistakes. Even on a single CPU, testing atomic transactions is regarded as a challenging process.

- **Consistency**

A transaction that is consistent will not break the integrity restrictions that have been set on the data by the rules of the database. If a database goes into an unlawful state, the procedure will be halted and modifications will be rolled back to their prior, legal state if consistency is maintained. This assures that the database will remain in a legal state.

- **Isolation**

Isolation is the idea that each portion of a transaction takes place in a bubble, without any awareness of the other parts of the transaction taking place.

- **Durability**

The property of being durable guarantees that any changes to the database or transactions that are properly committed will continue to exist permanently, even in the event that the system crashes. Because of this, there is no risk of the data contained within the database becoming corrupted.

- **SQL Command Types**

Users need a clear understanding of the SQL language's fundamental command categories prior to getting started with SQL programming.. Database administration covers but is not restricted to the following tasks: creating databases, manipulating objects, populating databases, updating databases, removing databases, submitting queries, regulating access to databases, and administering databases. Key categories are as follows:

- Data Definition Language (DDL)

With the help of Data Definition Language, also known as DDL for short, it is possible to generate new basic elements for a relational database, alter or restructure existing ones, or even remove existing ones. DDL is concerned with the structure and pays no attention to the data that is stored within the elements. Tables, schemas, views, and other things are all examples of these fundamental elements or data objects. Spite of the fact that it does not possess its own separate physical presence, a view is nevertheless percieved to be a virtualized table due to the fact that its description is the only place it can be found within the metadata.. On the other hand, the data for the view is derived out of the table from where a user can create the view. The most common commands of DDL are; CREATE, ALTER and DROP

- Data Manipulation Language (DML)

Simply referred to as DML, Data Manipulation Language is largely made up of SQL commands that are used to manage data maintenance functions. This indicates that users have the ability to manipulate the data that is contained within the objects that make up the relational database. Users will be able to enter data, change it, delete it, or retrieve it with the help of the command statements, which are written in standard English sentences. The most frequently used Data Manipulation Language statements are INSERT, UPDATE and DELETE.

- Data Query Language (DQL)

In the world of the structured query language (SQL), the primary focus of users of relational databases is on data selection. Data Query Language, also known as DQL for short, is comprised of commands that perform data selection. Users extracted results will be presented in a format that is well-organized and easy to read if a user uses the SELECT statement. This statement can be supported by other clauses or options. The users get to have the choice of submitting a query to the database via a distinct application interface or via a simple command line.

- Data Control Language (DCL)

The Data Control Language, also known as DCL for short, is made up of commands that, when executed, allow programmers to manage how users access data within the database. In addition, the user privileges are controlled in such a way that the database is guarded against any accidental or intentional misuse. The focus of DCL is on transactions, which record all SQL statements used to operate on a database and save the results of those statements in a log file. These are typical command statements used in DCL, which are GRANT and REVOKE.

- **Types of Data**

The following is a list of the general data types that have already been predefined by the SQL language, which are then furtherly classified into subtypes:

- **Numeric -** The value that is described by the numeric data type must be some form of a number, that can be represented with either an exact or only an approximation value. Both of these ways of expressing the value are acceptable.

From the Exact Numeric data type, there are five additional subtypes, including INTEGER, SMALLINT, BIGINT, NUMERIC (p, s), and DECIMAL (p, s). INTEGER data type only includes whole numbers, and they can be either positive or negative. There is neither a decimal nor a fractional component included in it. The SMALLINT data type is employed in place of integers to conserve space in storage; however, its accuracy cannot be higher compared to that of an integer. The use of this is limited to situations where integers cannot be used. When it comes to programming computers, precision refers to the maximum possible sum of significant digits that a number can have. BIGINT data type is the opposite of the previous SMALLINT data type, in which the minimal accuracy is equal to or higher than, that of the integer data type. NUMERIC (p, s) is a data type where a fractional component is also included and it gives an indication of the magnitude and accuracy of the value. The scale is the number of digits or spaces that are reserved in a fractional portion of the data, and it is displayed to the right of the decimal point in the data. Similar to the NUMERIC data type, the DECIMAL data type includes a fractional component in its representation, in which the value precision and scale can both be specified by the user. Nevertheless, this data type enables a higher degree of precision.

The approximate Numeric type has three additional subtypes, and those are REAL (s), DOUBLE PRECISION (p, s), and FLOAT (p, s). REAL (s) is a single precision floating point value, which means that the decimal value can "float" to various locations within the actual number. This indicates that the decimal value of this data type has an unbounded degree of precision and a scale with lengths that can vary. DOUBLE PRECISION (p, s) consists of a floating-point number with double precision, and its capacity is twice as large as that of the REAL data type. This type of data is useful in situations where more accurate numerical representation is required, such as in the majority of scientific fields of study. FLOAT can be described as a data type which gives the user the ability to specify the accuracy while also giving the computer the ability to automatically decide between single and double precision based on the user's preference.

- **String -** The string data type is regarded as one of the most widely utilized data types and is also responsible for the storage of alphanumeric information.[6]

CHARACTER or also specifed as CHAR data type may alternatively be referred to as a constant character or a string with a definite length. This indicates that the length of each string that is saved in that specific column is the same, which is denoted by the letter 'n' which stands for the number of characters or the maximum length that may be allotted for the specified field. CHARACTER VARYING (n) or VARCHAR (n) is a data type where data items are of varying lengths or are not constant, but a user does not want SQL to replace the leftover spaces with blanks. Therefore, the exact quantity of characters typed by users will be stored in the database, which will then further reduce the size of the storage space that is necessary. The CHARACTER LARGE OBJECT or CLOB data type was initially presented in SQL in the year 1999. Its purpose is to store unicode character-based data that is too extensive to be preserved as a CHARACTER type, such as large documents.

- **Date and Time –** data type which is managing pieces of information that concern dates and times

DATE data type allows for the recording of a date's year, month, and day values in the specific sequence in which they appear. The value of the year is indicated using four digits, which may be any number that goes from 0001 all the way up to 9999. The format for the date data type is yyyy-mm-dd. The TIME data type is used to store and display different values in an hour-minute-second format, or "HH:MM:SS". DATETIME is used by a user when a specific value includes both date and time information and it is represented in a format of "YYYY-MM-DD HH:MM:SS". TIMESTAMP is similar to the DATETIME data type but the range of values is from “1970-01-01 00:00:01” UTC to “2038-01-19 03:14:07” UTC.[7]

[6] Taylor G. A., SQL for Dummies-9th Edition, John Walley & Sons Inc, Hoboken; New Jersey, (2019)

[7] Alvaro, F. "SQL: Easy SQL Programming & Database Management For Beginners." Your Step-By-Step Guide To Learning The SQL Database, Amazon, eizdanje (2018)

- **Boolean -** This data type includes the values TRUE, FALSE, and NULL, which are used for doing comparisons between sets of data. In order for a given query to provide data, it is necessary for all of the requirements of the defined criteria to be satisfied, which indicates that the value of the Boolean must be TRUE. If data is characterized as not returned, the value then can be FALSE or NULL depending on the circumstances.

2.3. Analysis with SQL

The modern meaning of "data analysis" was made possible by the history of computers, and its development is closely linked to that history. It has been formed by developments in research as well as in commercialization, and the narrative behind it comprises a who's who of various researchers as well as large businesses. The power of computers and the methods of more conventional statistical analysis are combined in data analysis. Data exploration, interpretation of the data, and data transmission are all components of data analysis.[8] The main idea of data analysis is trying to upgrade overall making of decisions, which can refer to either the decision making of humans or, more and more frequently as a consequence of continual automation, the decision making of computers.

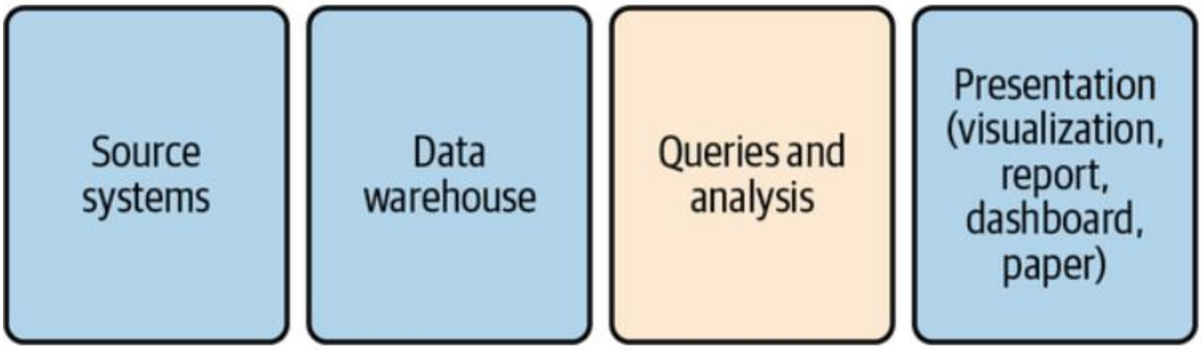
In some circumstances, an analysis is performed on a data file that was acquired to provide a response to a particular issue. This might happen in a scientific environment or during an experiment conducted online. The data that is produced as a consequence of working inside of some organization, such as the sales of goods, as well as the data that is produced for the sake of analytics, such as the monitoring of user engagement on websites and mobile applications, are both subject to analysis. This data can be used for a variety of purposes, such as troubleshooting and planning improvements to the user interface, but it frequently comes in a shape and amount that require the data to be processed before it can provide answers. These applications include planning UI improvements and troubleshooting.

The process of analysis usually begins with a question. This question may concern how many new buyers are recruited, how the pattern of sales is evolving, or why particular users stay around for a lengthy period of time whereas others test product or service and never come back. After the query has been formulated, we evaluate the source of the data, the location of the data

[8] Tanimura, C., SQL for Data Analysis, Advanced Techniques for Transforming Data into Insights, O'Reilly Media, Sebastopol, (2021); [online] https://learning.oreilly.com/library/view/sql-for-data/9781492088776/cover.html

storage, the analysis strategy, and the manner in which the findings will be communicated to the target audience. In the following figure 2, there will be a representation of the steps in the data analysis process

Figure 2. Steps in the Data Analysis Proces



Source: Tanimura, C., SQL for Data Analysis, Advanced Techniques for Transforming Data into Insights, O'Reilly Media, Sebastopol, (2021); [online] https://learning.oreilly.com/library/view/sql-for-data/9781492088776/cover.html

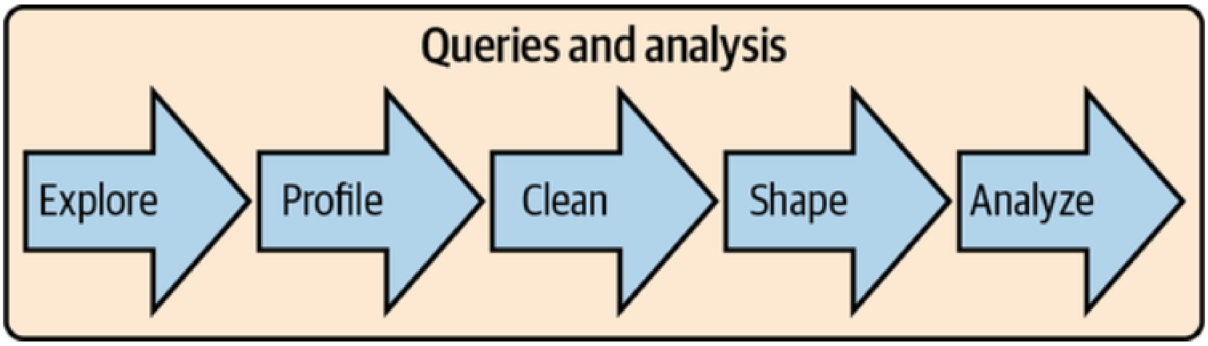
To begin, data are produced by source systems, which is a word that refers to any operation, whether it be performed by a person or a computer, that creates data of some kind of relevance. Data may also be created by humans manually, for example when somebody fills out a form or took notes while visiting the hospital. This kind of data is called "manual data." Data can also be constructed automatically by computers, for example when a software server logs a purchase, or an advertising management tool logs an email getting viewed. Source systems are capable of generating a wide variety of data in a variety of forms. The second stage involves transferring the data and storing it in a repository so that it may be examined later. Data mart, which really is generally a subcategory of a data warehouse or a data warehouse that has a more narrow and specific scope, and data lake, that is term that can represent that either data is kept in a folder storage system or that it is kept inside a database but with no level of data conversion that's also characteristic in data warehouses.[9]

The following phase, after inserting the data into a database, is to run queries and analysis on the data. The academic and technical field of converting raw data into useful insights beneficial for business intelligence operational decisions is generally regarded as consisting of data

[9] Gorelik A., The Enterprise Big Data Lake, Delivering the promise of Big Data and Data Science, O'Reilly Media, USA; Sebastopol, (2019)

extraction and knowledge extraction.[10] During this stage, SQL is utilized in the processes of data exploration, profiling, cleaning, and shaping, as well as doing analysis on it. The overall progression of the procedure will be shown in Figure 3. Learning about the subject, the location in which data was created, and the database tables where it is kept are all necessary steps in the process of exploring the data. Examining the one of a kind values of each record in data collection and the distribution of those records is part of the profiling process. The process of cleaning includes correcting data that is wrong or it is not complete, modifying data by include categories and flags, as well as null values. The act of organizing data into the rows and columns required for the final set is referred to as "shaping." Final stage of the data analysis process, users will evaluate output to look for patterns, conclusions, and insights. In spite of the fact that this procedure is presented in a linear fashion, in practice, it is frequently cyclical

Figure 3 Stages within the queries and analysis step of the analysis workflow.



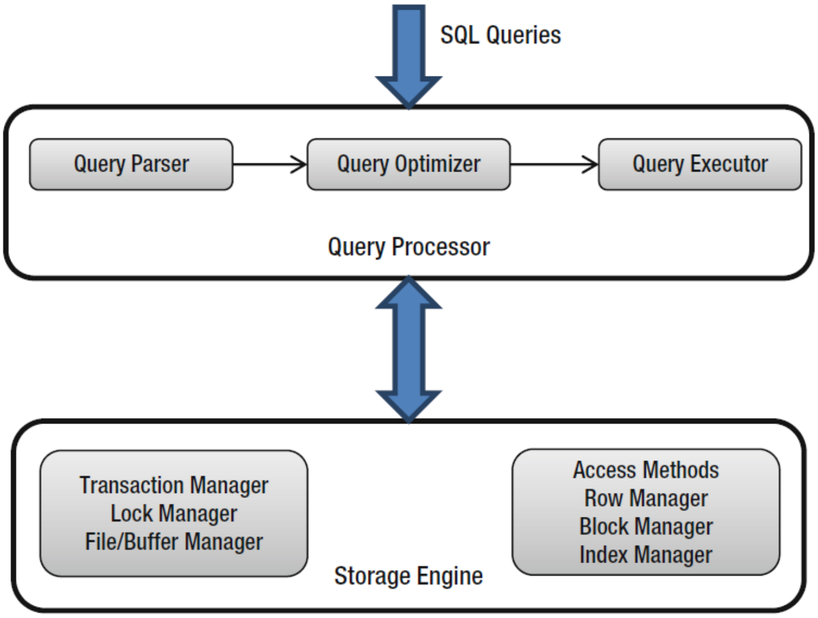
Source: Tanimura, C., SQL for Data Analysis, Advanced Techniques for Transforming Data into Insights, O'Reilly Media, Sebastopol, (2021); [online] https://learning.oreilly.com/library/view/sql-for-data/9781492088776/cover.html

The total operation comes to a close with the presentation of the data in its completed form as the final output. Receiving a file containing SQL code will not be appreciated by businessmen; instead, they anticipate seeing graphs, charts, and insights from data scientists. When it comes to making an effect with analysis, communication is essential, and because of this, there is a need for a mechanism to communicate findings with another individual. At some other times, a user would need to do a statistical analysis that is more advanced than what is available with SQL, or he might wish to input the data into a machine learning (ML) method. Both of these options are viable. To a user's good fortune, the vast majority of reporting and visualization tools are equipped with SQL connectors, which make it possible to import data either from

[10] Fotache M., Catalin S., SQL and Data Analysis. Some Implications for Data Analysis and Higher Education, Procedia Economics and Finance, (2015), pp. 243-251

whole tables or from previously written SQL searches. Statistical tools and programming languages that are often used for machine learning typically come equipped with SQL connections. The moment a SQL search is sent to the engine form the actual database, then a variarity of processes immediately begin operating to respond adequately to the requirements of the query. Both query database engine and the storage database engine are the two primary groups of operations that come into action at this level.

Figure 4. SQL engine architecture for traditional databases



Source: Pal S., SQL on Big Data; Technology Architecture, and Innovation, Apress, Wilmington, Massachusetts, USA, 2016

A logical query plan is created after the search is inspected by the query database engine, its semantics are evaluated by comparing to the catalog store, andthen the evaluation is completed. The query optimizer afterward refines the logical query plan to produce the most efficient action plan while accounting for the expenses of the CPU and network to execute the query. The storage engine will then apply the final query plan to extract the data out of the different data structures, which might be kept in memory or also in storage device, depending on some situation. To retrieve and update the data in accordance with the requirements of the query, many processes, including the Buffer Manager or Lock Manager amongst others, are put to work within the storage engine. The fact that this architecture must transport data between backplanes and I/O channels is the most significant disadvantage of it. When huge data sets are to be queried, this does not scale and does not perform well. This is especially true when the queries include complicated joins that need numerous stages of processing. When enormous amounts of data are read from disk, sent across the network, and loaded into memory for processing by the Database Management System, there is a significant loss of efficiency

2.4. SQL Today and Tomorrow

The rapidly expanding fields of data warehousing and business intelligence are one of the most significant factors that are now influencing the development of relational database technology and the SQL language. Utilizing the gathered data to generate knowledge and insights that may be used for decision making is the primary goal of data warehousing. The concept of "respecting one's data as a valuable asset" is common across the data warehousing industry's marketing materials. The activity known as "data mining" is doing an in-depth investigation of both historical data and trend data in order to locate "nuggets" of useful knowledge. Relational databases that are powered by SQL are an essential component of the technology that underpins data warehousing solutions.

The "decision support" systems were the driving force behind the adoption of relational technology. The majority of relational database providers moved their attention to competing for new transaction processing applications as the popularity of these applications expanded. With the introduction of data warehousing, the focus has returned to what was originally known as "decision support," along different terms and technologies that are far more powerful than those that were available 15 years ago.

The introduction of a trend that is as important is the rising popularity of object-oriented technologies which has been the only major development in recent years that has posed a substantial threat to the supremacy of SQL as well as relational database management systems. The core technologies for contemporary software design now include object-oriented computer languages like C++ and Java, object-oriented design tools, and object oriented connections.[11]

[11] Groff J. R. & Weinberg P. N., SQL: The Complete Reference, Osborne/McGraw-Hill, California, (1999)

3. THE NOSQL MOVEMENT

The category of non relational data management systems known as NoSQL is becoming increasingly widespread. For the purpose of performing operations on the data, these are the systems where databases are generally non constructed mainly on tables and do not employ SQL. Distributed, databases with no relations known as NoSQL systems are made for parallel processing data and massive data collection on a great amount commodity servers.[12] NoSQL systems came into being at the same time as large internet businesses like Google, Amazon, and Facebook did. These businesses had to manage with enormous amounts of data using traditional relational database managament systems, which were unable to keep up with the need for data storage and management. These systems overcome the performance limitations imposed by conventional RDBMS by utilizing NoSQL-style characteristics including column data storage and distributed structures, or by making use of technology like memory computation.

3.1. Concepts and Techniques of NoSQL

This chapter describes certain basic concepts, techniques, and characteristics that are common across NoSQL datastores and are not specific to just one category of nonrelational databases or a particular NoSQL store. Rather, these concepts, techniques, and patterns are shared by many NoSQL datastores.

- **The CAP Theorem**

At the ACM's PODC1 symposium in the year 2000, Eric Brewer gave a keynote presentation titled "Towards Robust Distributed Systems."[13] During his presentation, he developed a CAP theorem, that now is hugely empraced by both the NoSQL community and huge online firms like Amazon.. Consistency, Availability, and Partition Tolerance are the three components that make up the CAP acronym.

[12] Moniruzzaman, A. B., & Hossain, S. A. NoSQL database: New era of databases for big data analytics - Classification, characteristics and comparison. International Journal of Database Theory and Application, 6(4), (2013), pp.1-14

[13] Strauch, Christof and Walter Kriha. "NoSQL databases." Lecture Notes, Stuttgart Media University 20, no. 24 (2011)

- **Consistency**

A system is said to be consistent if and when it can be shown to be in a consistent condition following the completion of some operation. It is standard practice to consider a distributed system to be consistent if all viewers are able to view the changes made by a given writer in a specific shared data source following an update operation executed by that writer.

- **Availability**

The term "availability," and more specifically "high availability," refers to the fact that a system has been developed and installed in such a way that it is able to continuously operate, or in other words, it is able to permit read and write operations even in the event that some nodes in the cluster crash or even some software components that can be unavailable because of improvements.

- **Partition Tolerance**

Capacity of a system to advance functioning normally despite the existence of one or more partitions in a network is referred to as "partition tolerance." These appear when two or more "islands" from the network nodes come into existence and are unable to interact to each other, either temporarily or permanently. A further interpretation of partition tolerance is that it is the capacity from the program to deal with the ongoing extension and subtraction of nodes.

- **ACID vs BASE**

According to the CAP-theorem, a decision can only be made between two different possibilities when considering consistency, availability, and partition tolerance. When it comes to an ever-increasing number of applications and use-cases, it can said that availability and partition tolerance are quickly becoming extra essential than rigid consistency which includes online programs, particularly at big and super-large scales. This is especially true for web applications. Because of this, applications need to beceome dependable, and that usually requires availability and redundancy. Because it is difficult to generate these qualities with ACID properties, methods such as BASE are used instead. Basicaly available, soft state and eventual consistency are the components from which the acronym BASE is formed.

- **Basically available**

The provision of basic availability makes it possible for systems to be intermittently inconsistent, which makes managing transactions easier. The information and service capacity is "essentially available" inside BASE systems.

- **Soft State**

In order to reduce the overall number of resources used, the soft state accepts that some extent of incorrectness is partially permissible and that data could modify whilst it is being used.

- **Eventual Consistency**

In the end, after all of the service logic has been carried out, the system will be left in a consistent state. This is what is meant by the term "eventual consistency."

Because BASE systems don't have to create code to deal with locking and unlocking resources, they are often simpler and quicker than other types of computer systems. Their job is to keep the process going forward and address any problems with the components at a later stage. BASE systems are well suited for use in online shops, where the primary objective is to facilitate the process of making an order and filling up a shopping basket.

3.2. NoSQL Database Type

Many different forms of storage are available, and the content of NoSQL databases may be described in few of them. In the next sections, there is going to be clarification of the following forms of storage which are, Document Store databases, Key Value Store databases, Column oriented databases and Graph databases

- **Key Value Store Databases**

The architecture of this kind of data store is centered, as its name suggests, on the idea of storing information with identifiers recognized as keys.[14] Key-value store is a straightforward database that, at the time it is given an uncomplicated text string which is known as the key, outputs an unboundedly big binary object (BLOB) of information. The query language is absent from key-value stores; instead, they provide a mechanism for adding and removing key-value pairs into and from a database. There are numerous benefits when using key value store databases.

[14] Sulivan D., NoSQL For Mere Mortals, Pearson, Oregon, (2015)

By shifting attention away from architectural design and toward limiting services through: precise service levels, precise service monitoring and notification, scalability and reliability, flexibility, and reduced operational costs, their generalization and clarity actually save developers both money and time.[15]

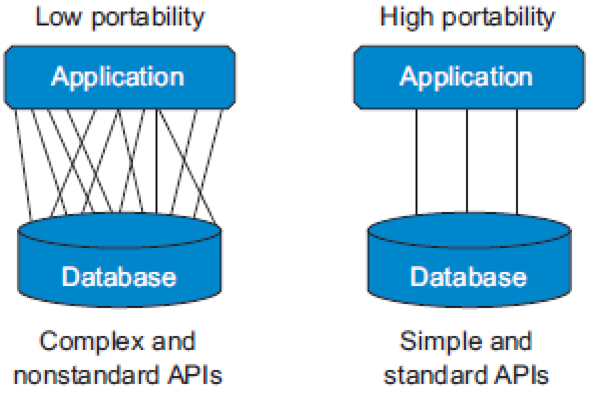
In the precision service levels users are better able to concentrate on conditions like establishing accurate service levels for data services when a user has a data service interface that is straightforward and is used by various apps. The Application Programming Interface (API) is unaffected by a service level; all that it does is provide exact specifications for how fast or reliably the service will run under a variety of load scenarios. Specifying service levels is a good first step, but a user should also consider making an investment in technologies that will help them monitor those levels. Users who define the amount of reads per second a provider performs and enter an incorrect value for the variable will cause problems, it is possible that the user would notice a delay during times of high demand. Users may identify potential bottlenecks in the system that may need further resource modifications by utilizing a simple API to generate thorough data comparing the predicted and actual load levels. When the number of reads as well as writes exceeds a certain limit within a predetermined period of time, automatic notification systems are also able to send out email messages as a notice trigger. For instance, users could dispatch an email if the amount of readings per minute surpasses 70% of some specified value during of a estimated period of one hour. This message might be triggered when the number of reads per second reaches a certain threshold. If maintaining a satisfactory level of service for customers was of the utmost importance, the email users sent them may include a connection to the monitoring systems and connections for adding additional data centers.

In terms of scalability and reliability, the resulting systems may have greater scalability and dependability when the database interface is kept as simple as possible. This indicates that it is able to customize any solution to the specifications that are wanted. Keeping an interface simple helps data modelers of all experience levels construct systems that make effective use of the capability available. The Programmers's sole obligation is to learn how to put this ability to work for the company by finding solutions to business challenges. A user will be able to put his whole attention on stress testing an loading as well as examinating service levels thanks to the straightforward user interface. One of the issues faced by managers of information systems

[15] Vaish, G., Getting Started With NoSQL-Your guide to the world of technology and NoSQL, Packt Publishing, Birmingham, (2013)

is the ongoing search for methods that will allow them to reduce the operating expenses of installing systems. A lot of companies can experience issues, and in terms of minimalistic costs, it is very improbable that one provider would solve those kind of problems. In a perfect world, the administrators of information systems would ask their database providers for bids on data services on an annual basis. In the realm of conventional relational databases, this is not possible since the cost of moving programs from one system to another is too high when weighed against the potential cost savings of storing data on the system of a different vendor. The more complex and nonstandard they are, the less portable they are likely to be, and the more challenging it will be to move them to the lowest cost operator.

Figure 5. Portability in Key Value Stores



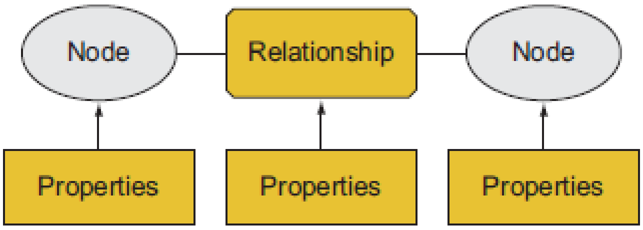
Source: McCreary D. & Kelly A., Making Sense of NoSQL; A guide for managers and the rest of us, Manning Publications, New York, (2014)

The complexity of the database interface has a direct effect on the portability of any program. The system on the left is not very portable since it has a lot of different interfaces that are difficult to understand in between the application and database. Transferring the application among two databases may be a difficult operation that requires a significant amount of testing. In comparison, the high-portability application on the right makes use of just a few number of standardized interfaces. These interfaces include put, get, and delete, and the program is capable of being easily converted to a new database at a reduced testing cost.

- **Graph Store Databases**

Among the most essential analytical abstractions in computer programming is the graph, which is widely used to symbolize relationships and hierarchy, particularly once the complexity of the hierarchy is larger than two.[16] Systems that demand evaluating connections among items or travel each node in graphs in a certain way are good candidates for making use of graph stores. Graph store is largely designed to store graph nodes and connections in an efficient manner, and they also provide the ability to query the stored graphs. Any corporate issue involving intricate interactions among objects can be helped by graph databases. It can be said that graph systems and also social networking, are really quick in analyzing network frameworks and also recognizing trends in some examples of such issues. Graph databases are useful for solving such problems. When talking about forming a graph, graph stores are widely portrated as type of systems which can keep track of some collections of interactions. The key value store consists of two data fields, which are referred to as the key and the value. A graph store, on the other hand, is comprised of three different data fields: nodes, relationships, and properties. As a result of the node-relationship-node structure that they possess. In addition, there are numerous varieties, where triple store is also known as the graph store.

Figure 6. Representation of a Triple Store



Source: McCreary D. & Kelly A., Making Sense of NoSQL; A guide for managers and the rest of us, Manning Publications, New York, (2014)

Users are able to do simple searches on graph stores, which will display the nodes that are immediately near, and also complex searches within the database which can explore the network. Despite the fact that graph stores are constructed on top of the straightforward and multipurpose node-relationship structure, graph stores have a unique vocabulary that may be confusing and inconsistent depending on how they are put to use.

[16] Mitreva, E. & Kaloyanova, K., NoSQL Solutions to Handle Big Data, Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”, 5, James Bourchier Blvd., 1164 Sofia, (2013)

Graph Store can be used to effectively solve some business problems by using link analyisis, rules and inference, and integrating linked data. Link analysis is a tool that can be used in situations when users want to run queries and look for trends and correlations, such as in social networking, or in emails. If users wish to execute queries on complicated structures like class libraries, taxonomies, or rule-based systems, you will need to employ rules and inference. Real-time integration and the creation of mashups may be accomplished without the need to store any data when linked data integration is combined with massive volumes of open linked data. In addition to this, graph stores come in handy when it comes to bringing data together and looking for trends and patters in the extensive collections of text documents. Process of the determining which entities inside a document are the most significant is known as "entity extraction." Nouns, such as persons, dates, locations, and things, are examples of entities that may be found in a document. When all of the important entities have been located, the next step is to put those results through some sophisticated search capabilities. In additional to this, graph stores come in handy when the data needs to be put back together and also looking for continuous pattern in numerous documents. The process of determining which entities inside a document are the most significant is known as entity extraction. Nouns, such as persons, dates, locations, and things, are examples of entities that may be found in a document. When all of the important entities have been located, the next step is to put those results through some sophisticated search capabilities.

- **Column-oriented databases**

Business intelligence and analytics have roots where the stored information is processed by columns, and not by rows. In these fields, it is possible to build high-performance applications by utilizing column stores that operate within a shared-nothing massively parallel processing architecture. Because of their scalability and ability to handle enormous amounts of data, column family systems are vital components of the NoSQL data architecture pattern set. In addition to this, it is well known that they have tight ties with a large number of MapReduce systems. A system that uses numerous computers with large datasets, and caries out procesing parareraly is called MapReduce.[17] For the sake of data lookup, row and column IDs are used as general-purpose keys by column family stores. Due to the fact that they do not have the elements that are often found in traditional databases, they are more frequently referred to as

[17] McCreary D. & Kelly A., Making Sense of NoSQL; A guide for managers and the rest of us, Manning Publications, New York, (2014)

data stores rather than databases. Typed columns, secondary indexes, triggers, and query languages are some features that are missing from these databases.

The first document that Google has published on the Bigtable has had a significant impact on almost all column family stores. Although the manner in which each system implements its Bigtable-like interface is different, HBase, Hypertable, and Cassandra are systems that feature such interface. It is important to highlight that the phrase "column family" should not be confused with a column store. At the same way that a row-store database maintains the data for each row together, a column store database saves all of the information that is included inside a column in memory storage. Many OLAP systems make use of column stores since their power is on the ability to do fast column aggregate calculation. Column-store systems include things like MonetDB, SybaseIQ, and Vertica, amongst others. An SQL interface is provided by column store databases so users may access their data. When talking about a column family technique, it's a very versatile approach to storing data, because it uses a name of a column and ID from the row when new data is added to a system. In addition to providing you with the advantages of higher scalability and availability, this method also spare users time and reduces the amount of inconvenience users experience. As users go over each of these advantages, they can take some time to consider the information that a company already gathers and determine whether or not a column family store might be useful to them to achieve an advantage in the sector where they operate.

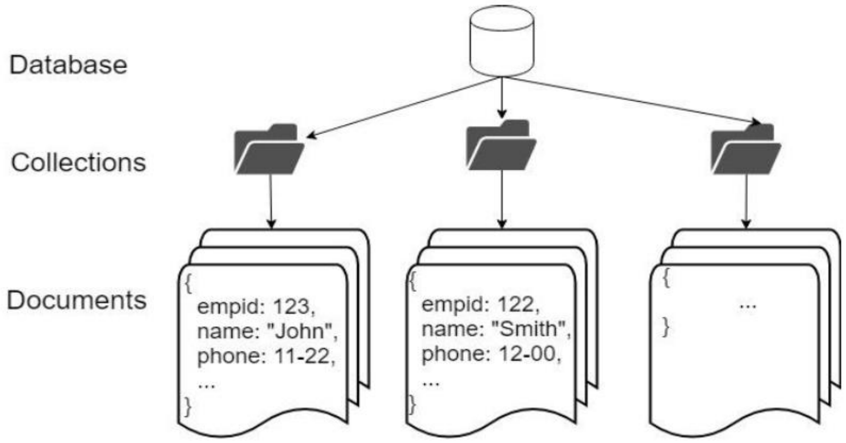
Column family systems often scale effectively on distributed systems since they do not rely on joins to function properly. Even though it is possible to begin production on one computer, when entering some data in systems, there is a flexible approach to have data stored in different nodes that are located in distinct locations when talking about production family colums. This is done to ensure that high availability is maintained.

- **Document Store Databases**

The discussion of NoSQL data structures would not be complete without a discussion of the document store, which is the most popular part of the NoSQL movement because it is the most generic, adaptable, powerful, and popular section of the NoSQL movement. When a key is sent to a key-value or Bigtable store, the store responds by returning a value that is connected with the key. Both key-value store and the exact values that are stored in Bigtable don't have a formal structure, therefore they can't be indexed or searched. On the other hand, keys in document store are functioning differently, where keys are just basic ID-s that are not accesed and used.

Users are nonetheless able to get practically any object by just doing a search of any kind of values in the storage where the document is located. When a new document is added to a document store, the contents of each document are automatically indexed. This is one of the benefits of utilizing a document store. Everything can be looked up, despite the fact that the indexes are rather vast.. This suggests that if a user is aware of any characteristic of a document, he will be able to easily recognize any and all papers that possess that property. This is because the documents share a common property. Node values in the structure of a tree can be accessed with the help of a document paths, which are kinds of keys, document stores are able to determine not only whether or not the search item you are looking for is present in the document, but also the precise placement of the search item inside the document.

Figure 7. The organization of data in document store databases



Source: Deari, R., Zenuni, X., Ajdari, J., Ismaili F. & Raufi, B., Analysis and Comparison of Document-Based Databases with SQL Relational Databases: MongoDB vs MySQL, Proceedings of the International Conference on Information Technologies, Bulgaria, (2018)

Predetermined schema is not required in the document database, where data is saved in the view of documents that are often in the format of JSON.[18] The values may either be of fundamental kinds such as integers or strings, or they can be expressed as structures. The keys are always written as strings, just as objects or can also be arrays. In addition, documents are known for carrying a variety of data structures, including embedded arrays, nested documents, and pointers. This not only makes accessing the data easier but also, in many instances, removes the need to do costly joins. An application programming interface (API) for a document storage

[18] Deari, R., Zenuni, X., Ajdari, J., Ismaili F. & Raufi, B., Analysis and Comparison of Document-Based Databases with SQL Relational Databases: MongoDB vs MySQL, Proceedings of the International Conference on Information Technologies, Bulgaria, (2018)

may keep its simplicity in spite of the complexity of a document's structure, and it can make it easy to choose a document or a portion of a document. The full document may hold a value part of the key value store, whereas a document store is capable of extracting subcategories from a big number of documents in an effective manner without individually putting each document into storage. A whole document could be stored in the values section of the key value store. A user doesn't have to load a complete book into RAM in order to show a single paragraph from it, all the user need is the paragraph he wishes to display. The vast majority of document storage options organize documents into collections. These collections have the appearance of a directory structure, such as one you might find on a UNIX or Windows disk. The management of big document repositories may be simplified with the help of document collections in a number of different ways.

3.3. Consistency Models

A strong desire to break free from the constraints imposed by strict ACID consistency is one of the primary forces driving the shift toward nonrelational database systems. It is a commonly held belief that the current generation of nonrelational databases can only give a shaky or, at most, an eventual consistency, and that the consistency mechanisms that lay under the surface are too simple. This misconception illustrates a basic lack of familiarity with nonrelational database management systems. Systems that are nonrelational are able to provide consistency, which also includes rigid consistency, although they only ensure consistency at the level of a single object. When there is not an access to the rigid and foreseeable constraints that ACID transactions provide, users are forced to rely on more complicated architectural designs in order to maintain a level of consistency that is considered acceptable. When bearing in mind consistency, numerous methods have arisen, when there was no model of ACID transactions. Those including ideas that are generally well known, such as eventual consistency, as well as numerous models that may be tuned to achieve a desired level of consistency.

Even while many non-relational databases do not enable tight multi-object consistency, regarding the objects that these databases store, they are considered to be perfectly consistent. Systems like as Cassandra enable a more straightforward and lightweight transaction paradigm, despite the fact that ACID transactions are not there. The consistency models used in a database have a significant impact on both its concurrency which is defined as the capacity for many users to have an available access to the identical data at the same time, and its availability. In order to establish whether or not a database can fulfill the requirements of an application, it is vital to have a solid understanding of how the database system in question handles consistency. The phrase "consistency" is applied in the database world in a variety of different ways, and it might indicate any of the following things depending on who is asked.[19]

- Consistency with other users

If there are two users that simultaneously query the database, will both of them see the same data? Non-relational databases are often more lenient in their approach, in contrast to traditional relational systems, which would normally make an effort to guarantee that they do.

- Consistency within a single session

Within the limits of a single database session, there is a question whether does the data preserve some degree of the logical consistency that was originally intended. For example, if there is a change to a row and then read it again, would a user notice the changes that were made after the update

- Consistency within a single request

Does each individual request deliver data that makes sense to the system as a whole? For instance, when users read all of the rows in a relational database, a user is typically guaranteed to see the state that the table was in at a particular point in time. This is because relational tables are designed to store data in a state that changes over time After a user started running the query, any changes that were made to the table after that point would not be included.

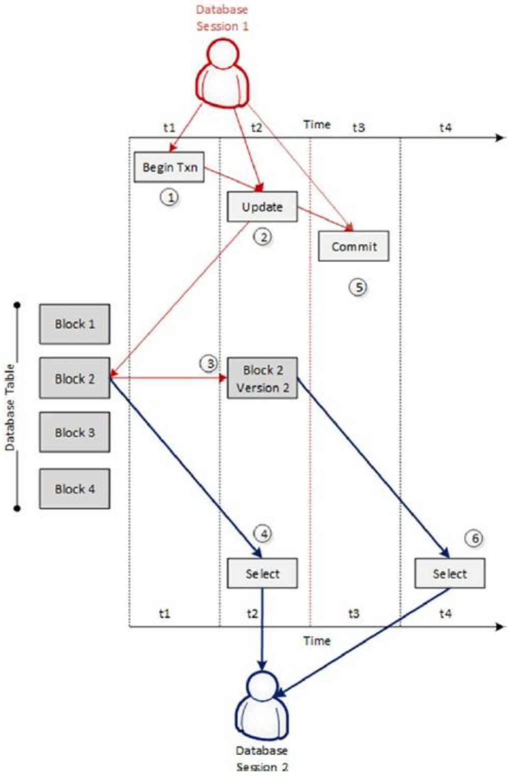
- Consistency with reality

Here the question is does the data match the reality that the database is attempting to reflect, or does it diverge from it. For instance, it is not sufficient for a financial transaction to merely be consistent at the conclusion of the transaction; it must also accurately reflect the real account balances at the time of the transaction. In most cases, accepting consistency at the price of accuracy is unacceptable behavior.

[19] Harrison G.,Next Generation Databases, MySQL, NewSQL and Big Data; What every professional needs to know about the future of databases in a world of NoSQL and Big Data, Apress, New York

In the MVCC model, multiple copies of data are tagged with timestamps or change identifiers that allow the database to construct a snapshot of the database at a given point in time.[20] In this way, MVCC provides for transaction isolation and consistency while maximizing concurrency. For instance, in MVCC, if a database table is modified between the time a session begins reading the table and the time the session concludes, the database will use older versions of the table's data to guarantee that the session views a consistent version of the table. This is done to prevent inconsistencies in the data that is presented to the user during the session.

Figure 8. Illustration of Multi-version concurrency control (MVCC)



Source: Harrison G.,Next Generation Databases, MySQL, NewSQL and Big Data; What every professional needs to know about the future of databases in a world of NoSQL and Big Data, Apress, New York

The MVCC model is seen in figure number 7. At the point in time t1, a transaction is started by a database session. The data which is held in database table gets modified at time t2, that session leads a current data which is then generated. The same database is then searched in a session from the second database in same time, but because of the reason that the transaction from the first database is not finished, then the session from the second database is viewing data out of

[20] Strauch, Christof and Walter Kriha. "NoSQL databases." Lecture Notes, Stuttgart Media University 20, no. 24 (2011)

earlier session. The session from the second database can begin reading, at the time when the first database has finished it's processing of the transaction

- **MongoDB consistency**

A MongoDB database with a single server deployment offers single-document consistency by default and in the most basic configuration. It is not probable for other sessions to write or also even read to a MongoDB document while it is being changed because the document is locked. The implementation of MongoDB replica sets, can be done by letting a read finish on other servers which have the out of date data, that way it is probable to create an eventual consistency. This allows something to be configured that is more like eventual consistency. In MongoDB, the employment of locks is the mechanism by which consistency for individual documents is ensured. Locks are used to ensure that the data a reader sees is always accurate, as well as to prevent two writers from concurrently attempting to modify a document. In addition, locks ensure that a document cannot be modified by more than one writer at the same time.

Because MongoDB does not use an MVCC system, it is impossible for readers to access a document that is currently being modification while it is being read. Throughout its existence, MongoDB has implemented many changes to the granularity of its locks. Earlier versions of MongoDB, up to the second version of the program, were using a so called one global lock to serialize every writing activity. This lock could prevent simultaneous users who could write or read documents on existing server, by accessing documents for the lenght of any write.

3.4. Databases of the Future

To this point, In the future, database systems may come together under a single "unified model," resolving the current divide. This will occur after a period of time in which database technologies have been separating. Extrapolating from what we already know about technology is a fascinating and helpful activity that is sometimes the only method of prediction that is available. On the other hand, if we learn anything from history, it's that the progression of technology does not necessarily follow the same path. Emerging technologies have the potential to cause discontinuities that cannot be extended and are not always possible to foresee entirely. There is a possibility that a game-changing new database technology is on the horizon; however, it is also possible that the significant advancements in the area of technology of databases which took place from the past ten years, are representing the maximum amount of change that we are able to immediately incorporate. Having said that, there are a few developments in computer technology that go beyond database design and that may have a significant impact on databases in the future. These trends are discussed further below.

- **Storage Technologies**

It can be said that ever since from the beginning of digital databases, there is an everlasting battle among how fast data is transmited to the database, and the overall storage space from the database. The slowest transaction of the data and worst latency are associated with storage mediums (magnetic disk and tape) that provide the largest cost savings when it comes to storing vast volumes of data. On the other hand, the costs of memory and solid-state drives (SSDs) are the highest per unit of storage space while having the minimum latencies and the best throughput. While HANA can be described as a system which aims to lower the overall costs associated with offering high throughput or low latency, Hadoop is specialized in optimizing costs within the terabyte of data. If, on the other hand, a technology were to emerge that could simultaneously give acceptable economies for big storage while also reducing latency, then we may witness a transition in database design relatively immediately. This kind of global storage would offer access speed that is equivalent to RAM in addition to the durability, persistence, and storage economy benefits offered by disk.

The vast majority of those who work in technology are assuming that the technology that can be considered disruptive and will appear on the market, is still years and years ahead. However, given the massive and ongoing investments, it seems likely that we will eventually develop a storage medium that is persistent, fast, and economical and that can satisfy the requirements of all database workloads. When this occurs, many of the database designs that are in use today will have lost a significant portion of the reason that led to their creation in the first place. If persistent storage, often known as disk, could move at the same speed as memory, for instance, the gap between Spark and Hadoop would shrink significantly. Memristors and phase-change memory are only two examples of the key new storage technologies that are on the horizon. However, it does not seem that any of these new technologies will in the near future be capable of realizing the goals of global memory.

- **Blockchain**

The consequences of cryptocurrencies are not within the focus of this discussion but nonetheless, the blockchain idea may have significant effects for database systems. Blockchain technology eliminates the need for a reliable third party, which is traditionally required to act as a mediator for every financial transaction. Instead of having a central database that keeps records of transactions and verifies every participant, through the blockchain and its network, it is allowed identify and make transactions which are authenticated by a unique consensus. This signifies prior to any transactions can be concluded it must first be verified with authenticating several nodes, which is made by the help of the key which is publicly owned. A cryptocurrency named Bitcoin is also based on blockchain technology and is open to the public, but there is also a possibility of "invitation only" blockchains which could not be seen from the open public, but only with the permission key. It might be argued that blockchains constitute a new kind of shared distributed database, regardless of whether they are public or private. This kind of technology received a wide range of opinions, and it goes from naming it the greatest advancement from the era of the early internet appearance to calling it just a database in a camouflage.[21]

The data which is situated in the blockchain, much like the data situated in systems based on the Dynamo architecture, is spread redundantly over a big amount of servers. In addition to this, the blockchain mark big fundamental departure from how permissions are traditionally handled inside databases. In a database management system that is already in place, the person who can say that has an overall control over the same data that is situated in the database, owns that same database. On the other hand, the data's original inventor is the one who retains ownership in a blockchain-based system. Facebook is a first social network and it also needs to manage a database, even though that particular system is designed for permiting users a modification of their own public posts, Facebook is having a complete insight and overall control of each of the users online data. That is being done on Facebook even so if it is designed to only give permission to individual users to do that. If they so want, the personnel there may delete posting from the users, and also censor them, or edit the same particular posts. However, this is just an option if they so choose. You would maintain full ownership of your postings inside a database that was powered by blockchain technology, and it would be impossible for any other entity to make any changes to them.

Applications built on blockchain technology have the potential to cause widespread disruption across a diverse variety of social and economic processes. Blockchain technology may one day be used to handle transactions including money, property, the administration of global

[21] Senthil, N., Govindarajan, C., Saraf, A., Sethi, M. & Jayachandran, P., Blockchain Meets Database: Design and Implementation of a Blockchain Relational Database, VLDB Endowment, Vol. 12, No. 11, ISSN 2150-8097. (2019)

identification, such as passports and birth certificates, or health data, amongst a varierity of other types of transactional data. The investing industry is by no means isolated from the impact of the blockchain technology, that technology will spread onto whole financial as well as the nonfinancial organizations.[22] It's possible that the databases that now keep records of these kinds of transactions may become irrelevant in the future. It is quite improbable that database technology will get revolutionize by the technology of blockchain in the near future since the majority of the databases and their owner are likely wishing to keep an overall control of their data in existing databases. On the other hand, it does not seem improbable that database systems would include authentication and authorisation methods based on blockchain technology for certain application situations. In addition, the development of formal database systems that are constructed on top of a blockchain infrastructure seems to be imminent.

- **Quantum Computing**

Quantum physics may trace its roots back more than a century to the discovery that energy can be broken down into tiny units, known as quanta. The majority of quantum physics' mind-boggling notions had been adequately stated by the time the 1930s rolled around. In the well-known experiment known as the "twin-slit experiment," individual photons of light seem to travel through several slits simultaneously provided that they are not being seen. The photons superimpose themselves over a number of different states. The photons will only exist in a single state if it is attempted to measure their journey, which will force them to combine. When photons are entangled, the state of one photon may be irrevocably linked with the state of a particle that is otherwise unconnected. This phenomenon, which Albert Einstein referred to as "spooky action at a distance," may occur when photons are entangled. While it is becoming more plausible that quantum computers will be able to break conventional private and public key encryption systems, quantum key transmission already offers a tamper-proof mechanism in the transfer of certificates across lengths of a several tens if not hundreds of miles. In the area of quantum computing, overall operations are including different types of join operations, cross-product and the composition of SQL operation. That would be significant for any type of database system, because that implementation could then allow using a selection of project join queries.[23]

[22] Isson J. P., Unstructured Data Analytics; How to Improve Customer Acquisition, Customer Retention, and Fraud Detection and Preventio, John Willey & Sons, Canada, (2018)

[23] Joczik, S. & Kiss, A., Quantum Computation and Its Effects in Database Systems, Faculty of Informatics, Eotvos Lorand University, 1117 Budapest, Hungary, (2017)

In the event that the theoretical promise of quantum computing is realized, it would have a significant influence on all facets of computing, including databases. In addition to this, there are additional database specific suggestions for quantum computing, those are quantum transactions, quantum search and a quantum querry language.[24]

In quantum transactions has been claimed that data stored in a database may be maintained in a "quantum state," which would effectively reflect several alternative outcomes. This idea was inspired by the notion of superimposition. When "seen," the many states resolve themselves into a single conclusion. For instance, seat assignments in an aircraft may be modeled as the total of all potential seat configurations. Once the final seat assignments were established, the model would "collapse," leaving only the seats that were actually assigned. For instance, seat assignments in an aircraft may be modeled as the total of all potential seat configurations. Once the final seat assignments were established, the model would "collapse," leaving only the seats that were actually assigned. This strategy makes use of quantum principles but does not need an infrastructure for quantum computing, even though a quantum computer may possibly allow similar operations to be carried out on a much larger scale. Also, In comparison to a conventional database, the search performance of a quantum computer could be able to be improved. A quantum computer would be able to discover matching rows for a complicated non-indexed search phrase and complete a full-table scan more quickly than a traditional computer. When conventional disk access is the limiting issue, it is doubtful that the improvement will be decisive; but, with in-memory databases, It is plausible that in the future, a so called quantum query in particular database could be an innovation which will be achievable and also practical. It is necessary to have a database that is from the aspect of quantum computing, really enabled and is able of carrying transactions by qubit standard, as opposed to Boolean bit standard to permanently save data generated by a quantum computer. In order to perform operations on this kind of databases, an invention of an innovative language would be necessary. This language would describe quantum operations rather than the relational predicates that are used by SQL. Quantum Query Language is the name given to one such language that has been suggested and defined by the acronym QQL.

[24] Harrison G.,Next Generation Databases, MySQL, NewSQL and Big Data; What every professional needs to know about the future of databases in a world of NoSQL and Big Data, Apress, New York

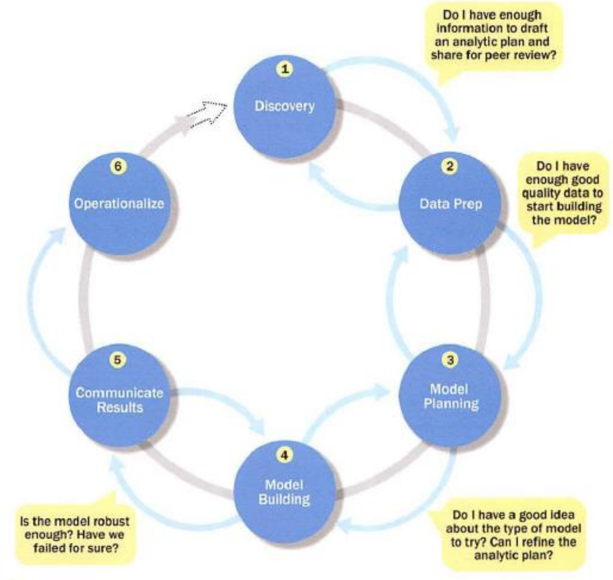
4. BIG DATA ANALYTICS

Big Data has been through a phase of evolution, and so did analytics. That was made feasible because of a number of significant developments in technical innovation. Because of the reason of recent developments in data processing and data storage, programmers have acquired massive sets of data, which were before unavailable to them, and were beyond the scope of capacities to utilize. Improved networks are available to link various sets of information to each other for the purpose of analysis, also programs like Hadoop, MapReduce and Big Table enable users to segment the analysis of the data.[25]

4.1. Data Analytics Life Cycle

A summary of the lifeycle from analytics of the data is shown in figure 8. This summary consists of a cycle that is broken down into six distinct stages. The cyclical movement between phases is represented by circular arrows, and it continues until the members of the team have enough knowledge to go to the next step. The callouts provide examples of questions that one should inquire in order to determine whether or not each member of the team has sufficient knowledge and has achieved enough advance to go on to the following stage of the process.

Figure 9.Data Analytics Lifecycle


Do I have enough information to draft an analytic plan and share for peer review?
1
Discovery
2
Data Prep
Do I have enough good quality data to start building the model?
6
Operationalize
3
Model Planning
5
Communicate Results
4
Model Building
Is the model robust enough? Have we failed for sure?
Do I have a good idea about the type of model to try? Can I refine the analytic plan?

Source: Dietrich D., Heller B., Yang B., Data Science & Big Data Analytics; Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, Indianapolis (2015)

[25] Marr, B., Big Data-Using smart Big Data Analytics and metrics to make better decisions and improve performance, John Willey & Sons Ltd, Southern Gate, (2015)

- **Phase 1: Discovery**

During the beggining stages of the project, the team will gain an understanding of the business domain. This will involve historical aspects, such as establishing whether or not the company or a business has previously attempted projects that are comparable to the ones they will be working on. The group conducts an analysis of the resources that including people, technology, time, and data—that are at its disposal in order to support the project. In the following stages there is an important activity that takes place during this phase, and that is reframing the issues that particular organization is facing, and analyzing it. Other important activities include the formulation of beggining hypotheses of testing and extracting the data to obtain information.

- **Phase 2: Data Preparation**

A precondition for moving on to the second phase is the establishment of analytics, where data scientists will be capable working with the data they were assigned to, and carry out analysis throughout the remaining phases. In order for the data to be inputted, data scientists need to execute an operation known as ELT, which is shortened for extracting, loading and transforming data. In order for the team to make use of the data and conduct analysis on it, the ELT process requires that the data is transformed. During this phase, the team is responsible for getting themselves completely acquainted with the data as well as taking the necessary procedures to apply some conditions to the data.

- **Phase 3: Model Planning**

Third step, which is called "model planning," data scientists decide which strategies, procedures, and workflows it will use in the future phase, "model building," to construct the model. The team investigates the data to get a better understanding of the connections that exist between the variables, and then they choose important variables and the models that are the best fit for the data.

- **Phase 4: Model building**

During the fourth phase of the project, data scientists create sets of data because of the purpose for production, testing and training data. Additionally, in the duration of the following phase, scientists are responsible for constructing and executing all the models that were planned in the previous step. Data scientists are analyzing currently available tools and if they are enough for running the models, or whether or not they are in need for better equipment for running the models and processes

- **Phase 5: Communicate results**

Data scientists will decide if the observed and analyzed data was a triumph or not which will be catagorized on the criterium that was set in the first stage during Phase 5 of the project. This decision will be made in consultation with important stakeholders. The group has to choose the most important results, determine how much those discoveries are worth to the company, and then compose a narrative to both explain and communicate those findings to the relevant stakeholders.

- **Phase 6: Operationalize**

During phase 6. all the important documents, reports and pieces of code which are bounded for the analysis are brought together by the whole team. In order to see how good all the models function when they are used in the real world production, the team can test a project

4.2. Cluster Analysis

The basic process of "looking" into the data is done in cluster analysis, which is also recognized as exploratory data analysis and is used as a spark of innovation that provides alternative and suggestive models for data.[26] Clustering is a technique that is often used in the process of exploratory data analysis. There are never any predictions made while grouping data. Instead, clustering algorithms search for similarities between items based on the properties of the objects and then group together the objects that have those similarities into clusters. Marketing, economics, and many departments of research are just a few of the fields that make use of clustering methods. The k-means clustering approach is one of the most used ones.

- **K-means**

K clusters and its objects dependent on the closeness in the middle of a k group, are found by K means, which is a method in analysis. In K means, it can be said that its center is described in terms of arithmetic average or mean, of every individual cluster vectors and it's attributes.[27]

[26] Eds. C. H. Chen, L. F. Pau & P. S. P., Wang, Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Company, (1998), pp. 3-32

[27] Dietrich D., Heller B., Yang B., Data Science & Big Data Analytics; Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, Indianapolis (2015)

The step of clustering is often taken before moving on to the classification stage. After the clusters have been located, it is possible to assign labels to each cluster in order to categorize each group according to the features that they share. Clustering is typically used as an exploratory method to unearth previously unknown structures within the data. This might be done as an introduction to more targeted research or decision-making procedures. Image processing, medical diagnosis, and consumer categorization are a few examples of specialized uses of the k-means algorithm. Videos are a way of showing the rising amounts of unstructured data that is captured, and there are many more. A k-means analysis may be used to determine the locations of objects in a movie by looking at each frame individually. The aim is to figure out, for each individual frame, which pixel pairs are the most similar to one another. Brightness, color, and placement within the frame, as indicated by x and y coordinates, are examples of the characteristics that may be associated with individual pixels. When looking at surveillance video footage, for instance, consecutive frames are analyzed to see whether or not there have been any shifts in the clusters. These newly discovered clusters can suggest that illegal individuals gained access to a facility. Patient characteristics including age, height, weight, characteristics, may be used to detect naturally occurring clusters.

These clusters might be used to identify people to target for particular preventative actions or for participation in research trials. The practice of clustering, in its broadest sense, may be helpful in the study of biology, namely in the realm of human genetics and in the categorization of plants and animals. K-means is used by marketing and sales departments in order to improve their ability to identify clients who display comparable habits and spending habits. For instance, a wireless service provider may take into consideration a client's monthly cost, the amount of text messages sent and received, the volume of data used, the minutes spent at different times of the day, and the number of years the consumer has been a subscriber.

Clustering has also different categories, to be precise there are three of them, and they are Unsupervised, semisupervised, and supervised clustering:[28]

- Unsupervised clustering

Given a similarity/dissimilarity measure, the goal of unsupervised clustering is to increase the degree of similarity that exists inside each cluster while decreasing the degree of similarity that exists across clusters. It does this by using a particular objective function, such as one that seeks

[28] Dean J., Big Data, Data Mining and Machine Learning; Value Creation for Business Leaders and Practitioners, John Wiley & Sons, New Jersey, (2014)

to find clusters that are as close together as possible within each class. It uses a data collection that does not include a target variable in any way. In the field of segmentation, the unsupervised clustering approaches that are used the most often include K-means and hierarchical clustering.

- Semisupervised clustering

In order to improve the clustering results, semi-supervised clustering utilizes a wide range of directing and adapting domain information in addition to the measure of similarity which is also included in unsupervised clustering. Such domain information may be used to create pairing constraints between some of the observations or target variables for those observations. These constraints can be either mandatory or optional. This guiding information is used either to adjust the similarity measure or to change the search for clusters in order to bias the search. The search is biased in either direction depending on which method is employed. In addition to the goals that are set for unsupervised clustering, the goals that are set for semisupervised clustering include achieving a high level of consistency between the clusters and the information that is used for guiding and altering them.

- Supervised clustering

Identifying clusters that are having large probability densities with particular classes is the objective of supervised clustering. These clusters are referred to as class-uniform clusters. This clustering is used in situations in which there is a targeted variable as well as a training set that contains the variables to cluster.

4.3. Association Rules

Association rules is an approach that is used to find intriguing correlations that are hidden inside a huge dataset and are frequently utilized for the mining of transactions in databases. This method is descriptive rather than predictive. The revealed relationships are capable of being represented in the end as either rules or frequent item sets. Following are usual questions that can be answered with the help of association rules[29]

- What merchandise customers usually buy together?
- From the observed customers. which correspond to some person, what merchandise do they purchase?

[29] Dietrich D., Heller B., Yang B., Data Science & Big Data Analytics; Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, Indianapolis (2015)

- People that have bought the product, what are the other products which they usually view or tend to purchasing?

When association rules are applied to a big amount of versitile transactions, as an example of this are multiple bundles of invoices, where each transaction contains one or more items, they search through the products being bought to evaluate which products are commonly purchased collectively and then assemble a document with a list where all patterns are described of connsumers buying habits. In other words, the consumer's shopping patterns can be concluded from the associations between the items. The purpose of the method known as association rule mining or for short ARM, which is regarded as a foundation of data-mining, is identifying connections among various factors in databases which are known to be very big.[30] The purpose of association rules is to aid in the discovery of relevant relationships that exist between the objects. The relationship exists far too often for it to be coincidental, and it is significant from a financial standpoint, but the significance of this may or may not be immediately obvious. The nature of some companies environment also kind of algorithm that is going to be applied for discovering both have a role in determining which relationships are interesting to the business itself.

The method of association rules places a focus on apriori as its primary algorithm. It is one of the earliest algorithms developed, and it is also one of the most important ones, for producing association rules. It was a pioneer in the application of assistance in the process of pruning the item sets and managing the development of the item sets that are named candidates. In order to develop longer frequent item sets, sets of shorter candidate items can be joined with one another and pruned. Because of this approach, it is not necessary for the algorithm to enumerate each and every potential item set. This is due to the fact that the number of all possible itemsets has the potential to grow at an exponential pace.

4.4. Regression Analysis

Today, regression analysis has become an important factor influencing strategic choices from companies. It has several beneficial features, one of which is the simplicity of interpreting final results.[31] In general, main idea from regression analysis is influencing when a certain

[30] Pandey, G., Cawla, S., Poon, S., Arunasalam, B. & Davis, J. G., Association Rules Network: Definition and Applications, Wiley Periodicals, Minneapolis, (2009), pp. 260–279

[31] Dean J., Big Data, Data Mining and Machine Learning; Value Creation for Business Leaders and Practitioners, John Wiley & Sons, New Jersey, (2014)

amount of determinating factors influences the final result of the other variable. This is accomplished by examining the relationship that exists between the two variables. The term "dependent variable" refers to the fact that the value of the final result from the dependent variable is determined by different values of another variable. Those extra variables are also reffered as variables of input, which are independent.

- **Linear Regression**

Linear regression can be used in describing or measuring relationships among factors which are being investigated.[32] One of the most important presumptions is that the connection between a variable of input and the variable of output is linear. Although this may seem to be a limiting assumption, it is often necessary to correctly modify the variables of either the input or the result in order to produce a linear connection between the variables of the changed input and the modified outcome. A probabilistic model that takes into consideration the randomness that may play a role in any given result is referred to as a linear regression model. Using the values of the input parameters and the specified input values, linear regression model predicts the result of the outcome variable. However, some uncertainty may exist in forecasting any given outcome due to the fact that the inputed determinants are previously given. A wide variety of settings, including those involving business, government, and other domains, often make use of linear regression, and in the following are examples of some of the most typical practical uses of linear regression.

Estimating residential house pricing as a purpose of living space may be accomplished with the use of a simple linear regression analysis. A model of this kind assists in establishing or evaluating the list price of a property that is currently available on the market. A more predicting model might be increased by adding more input variables. The demand for products and services may be forecasted using linear regression models, which can be used by both businesses and governments. It is possible to construct comparable models in order to forecast retail sales, trips to emergency rooms, and ambulance deployments. Using linear regression model can also be used in medical purpose. The length of time spent receiving a single radiation treatment, the number of times a patient receives radiation treatments, and patient characteristics like age and weight might all be considered input factors.

[32] Kumari, Khushbu & Yadav, Suniti. Linear regression analysis study. Journal of the Practice of Cardiovascular Sciences. 4. 33. (2018), 10.4103/jpcs.jpcs_8_18

- **Logistic Regression**

Logistic Regression, or LR for short, is a method used by statisticians and researchers for the purpose of analyzing and classifying data sets that include binary and proportional responses. It is widely acknowledged as a statistic tool which can be used in data mining, and is considered very important approach currently available. The ability of Linear regression to naturally generate possibilities and expand to situations involving several classes is among the most significant benefits associated with this method. An additional benefit is also considered to be the majority of techniques employed in logistic model analysis adhere to the exact fundamentals as those in linear regression.[33] For instance, a logistic regression model can be constructed to predict whether or not a person would buy a new car over the following 12 months.

This model may take into account a variety of factors. The training set could include input variables for things such as how old a person is, his salary, and sex, in addition to number from years their current car has been on the road. In addition to this, the outcome variable of whether or not the individual acquired a new vehicle during the previous year would be included in the training set. The logistic regression model is used in a wide range of contexts, both in the public and private sectors. In the marketing sector, data scientists could calculate the odds that a wireless client would switch carriers, often known as churning, depending on various different factors which can give value to the marketing sector.

Using this information, with the power of analysis user could target the customers who have a high risk of leaving by providing them with relevant offers. Logistic regression could also be used in the field of finance where data scientists could determine the likelihood of an application defaulting on the loan by analyzing the applicant's credit history in combination with the specifics of the loan. On the basis of the forecast, the loan application may be accepted, it might be rejected, or the conditions might be changed.

4.5. Classification

Classification is yet another essential learning technique that's also utilized in various systems that are connected to data mining. During the process of classification learning, a classifier is given a collection of examples that have previously been assigned a classification, and using

[33] Maalouf, M., Logistic regression in data analysis: An overview. International Journal of Data Analysis Techniques and Strategies. 3. 281-299. 10.1504/IJDATS.2011.041335, (2011)

these examples, the classifier learns how to assign instances that it has not yet seen.[34] To put it another way, the major responsibility that classifiers are tasked with is the process of appointing class labels to new observations.

- **Decision Trees**

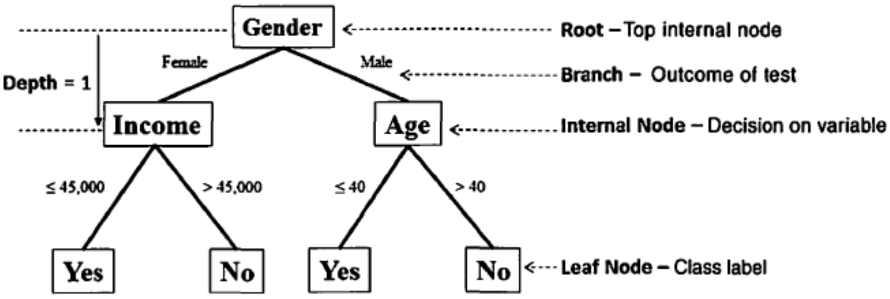
Decision trees are among the classification techniques. A decision tree has a straightforward cyclical architecture for defining a sequence in classification in which an instance can be given to particular disjoint group of the classes after being defined from the collection of attributes.[35] The main objective of decision trees is to forecast output of the variable Y based on given input X. An "input variable" is what is the name of each individual member of the set. One strategy for making the prediction is to build a decision tree containing test points and branches. At each of the checkpoints, a choice has to be made on which branch to choose in order to go down the tree. After some time, one arrives at the conclusion, at which point one is able to make a forecast. A point from the decision tree is being tested from a particular variable that is inputed, or its characteristic, and every branch from the tree represents a different conclusion that is being considered.

Decision trees can be frequently utilized in the applications related to mining of the data for the purpose of categorization. This is because decision trees are both flexible and simple to visualize. It is possible for a decision tree to take either categorical or continuous values as input. The structure of a decision tree consists of test points, also known as nodes, and branches, each of which is used to represent a different aspect of a choice. A leaf node is a node that does not branch out into any other branches. The leaf nodes are responsible for returning the class labels and also in certain implementation of the same.

[34] Dietrich D., Heller B., Yang B., Data Science & Big Data Analytics; Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, Indianapolis (2015)

[35] Quinlan, J. R., Generating production rules from decision trees. In ijcai, Vol. 87, (1987, August), pp. 304-307.

Figure 10. Decision tree exapmle



Source: Dietrich D., Heller B., Yang B., Data Science & Big Data Analytics; Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, Indianapolis (2015)

Figure 10. is an illustration of one way to use a decision tree to forecast whether or not buyers would purchase a certain product. The conclusion of a choice is referred to as a branch, and branches may be represented in the form of a line that connects two nodes. When making a decision based on a numerical value, it is common practice to position a "greater than" branch which is situated on the right side, while the "less than" branch is positioned on the opposite side which is left. It's possible that some branch will must have an "equal to" characteristic, but that will depend on the type of the variable. The decision or test points are located at the internal nodes. Every internal node has a corresponding input variable or property that it talks about. The root is the name given to the top node inside the tree. Figure 10. depicts a binary tree, sometimes known as a decision tree, because each internal node on the tree has no more than two branches. A split is the name given to the branching off of a node in a tree. The minimal number of steps necessary to get from the root to a certain node is referred to as the node's depth. The nodes Income and Age in Figure 10 each have a depth of one, whereas the four nodes that are located at the very end of the tree and every one of them are having two depths. The tips of the tree's most distal branches are where leaf nodes are. They are representative of the class names that have been decided in the past. The route from the upper side of a tree towards the leaf involves the number of some particular decision that must be taken at different internal nodes along the way.

In figure 10. the root node is splitting into two branches to form a gender. The records that make up the depth 1 internal nodes are split between the right branch, which includes all of those records where the variable Gender is set to Male, and the left branch, which contains all of those recordings where the variable Gender is set to Female. It may be said that each internal node performs the function of the tree's root, and the optimal test for each node is determined in isolation from the results of the tests performed on the other internal nodes. The internal node on the left-hand side divides based on a question about the Income variable to produce leaf nodes at depth 2, whereas the internal node on the right-hand side divides based on a question about the Age variable. Figure 10. depicts a decision tree, which reveals that individuals who are categorised as people who would purchase the product are ladies with an income of less than or equal to $45,000 and men who are 40 years old or younger. When climbing this tree, neither a person's age nor their money is taken into consideration, and the same goes for females.

5. SQL AND NOSQL COMPARISON IN BIG DATA

5.1. SQL on Big Data

Businesses must learn how to make the most of the business intelligence tools and applications they already have in order to overcome the challenges brought on by large data quantities, rising data variety, and rising information demands that is the reason most of the organizations are adopting big data tools at an increasing rate.. Existing corporate solutions for transactional, operational, and analytical workloads have a difficult time delivering results. These systems suffer from poor response times, a lack of agility, and an inability to manage new data types and unpredictable workload patterns. There has been a growth of SQL–on–big data solutions developed in recent years in response to the difficulties that have been described. These technologies are helping businesses begin the process of moving their data to big data platforms. The "SQL on big data" trend has quickly developed into its mature state. Historically, the primary focus of big data tools and technology has been on developing solutions in the analytic arena, ranging from basic business intelligence to sophisticated analytics. Big data systems have only been used sparingly in operational and transactional systems. Building operational systems on top of big data technologies appears to be achiaveble today, thanks to recent improvements made to SQL engines on Hadoop, such as the addition of support for transactional semantics in Hive 0.13 and later versions, as well as the debut of open software goods like Trafodion and suppliers like Splice Machines.

As an example, since its release, MySQL has been the relational database of choice for the vast majority of businesses. Because there is a big quantity of data which is being simultaneously created and gathered by companies, there is a need for adoption of big data tools which will help in analyzing all of the data. Big data analytics could be probable with adopting and integrating MySQL and Hadoop system. That is because Hadoop can be utilized for storing big amounts of data, and using clusters from Hadoop.[36] A system like Hadoop can be considered as the most popular option because of its strong computing and enormous parallel processing capabilities. Together, MySQL and Hadoop provide real-time analysis, allowing Hadoop to store information and run at the same time as MySQL to deliver final outcomes in the real time, which was previously not feasible.

[36] Challawala, J. Lakhatariya, C. Mehta, K. Patel, MySQL 8 for Big Data; Effective data processing with MySQL 8, Hadoop, NoSQL APIs, and other Big Data tools, Pact Publishing, Mumbai, (2017)

A solution for SQL on big data may have numerous objectives, one of which is to perform the same kinds of operations as a traditional RDBMS, either from the point of view of processing particular transactions via online and analytical queries. There are some typical goals that SQL in terms of big data sets solution includes. One of the ideas is an architecture that is both distributed and scalable. The aim is trying to enabling SQL to scaling out storage and processing over the number of computers, and by that way achieve distributed architecture. Prior to emergence of structured query language on large data, there were very few distributed systems for storage and computing, and those that did exist were quite costly. Also, an affordable alternative to pricey analytical databases and tools. It should be possible to do low-latency, scalable analytics on huge data volumes while keeping costs down. Existing relational database management system (RDBMS) processors are vertically scaled devices that experience scaling and overal performing barriers when a particular amount of data capacity has been achieved. The options were to invest in highly expensive Massively Parallel Processing machines with cutting-edge designs and solutions, or to employ highly scalable analytic databases that were effective and centered on columnar design and compression. Both options were available.

In the field of large data sets that needs to be stored or analyzed, another objective from SQL is trying to achieve that several users could execute commands simultaneously on huge data sets. Hadoop has never been very excellent at managing several users at the same time, whether for a needed analysis or for just loading, transforming and extracting workloads. When it comes to these kinds of workloads, the allocation and scheduling of resources has traditionally been a bottleneck. The majority of engines in SQL that are a must have for large data sets have always had the goal of delivering little latency for searches which were ad hoc in SQL and run on enormous data as their primary focus. When the velocity and variety components of big data are handled by SQL queries, this becomes much more complicated. The "native" format of the data is preserved when it is written to HDFS using the schema on demand technique in Hadoop. The main storage solution utilized throughout Hadoop is called Hadoop Distributed File System. As the name suggests, HDFS is a distributed file system that allows secure and fast operations by transferring numerous copies of data blocks over a network of compute nodes and providing high bandwidth accessibility to application data[37]. It makes a SQL in large data sets products stand out from the competition and makes it possible for current business inteligence tools to use SQL to interact with all these semi-structured data.

[37] Pothuganti, A. 'Big Data Analytics: Hadoop-Map Reduce & NoSQL Databases', International Journal of Computer Science and Information Technologies, 6(1), (2015), pp. 522-527

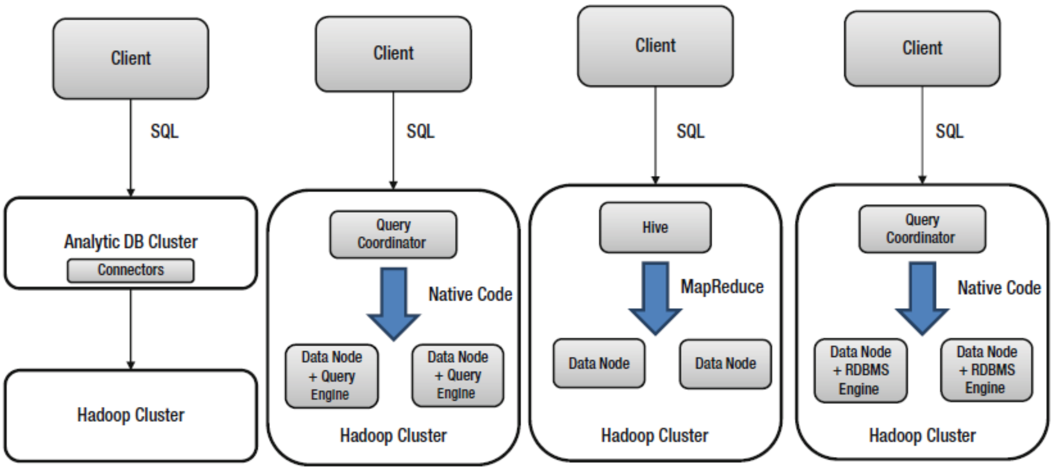
There is a surge of effort and genuine enthusiasm in the field of developing SQL for large data sets and Hadoop. Numerous technologies, whether by its SQL community or by for-profit corporations, have been developed to make SQL accessible through all the big data platforms. This is a highly competitive landscape, and every tool and vendor is attempting to compete on at least one of the following dimensions: low latency, SQL feature set, etc. Every one of these products and tools on the todays free market has undergone some sort of innovation, either in the form of an entirely novel strategy for resolving SQL and its problems with the big data or in the form of a retrofitting of some of the more established ideas from the relational databases area into the realm of large data sets for storage and computation.

5.2. SQL: Challenges and Solutions on Big Data

Performance- and scalability-wise, relational databases can be limited to a certain extent of data collecting. These challenges may be conquered using strategies such as manual sharding and the segmentation of data; but, these approaches, too, have their own unique challenges to face. One of the most difficult aspects of working with distributed systems is figuring out how to make distributed joins work efficiently across a group of computers while keeping latency to a minimum. The resolution of this issue is complicated by the inherent difficulties associated with the transport of bits over the network link at high speed and throughput. Because doing so necessitates executing SQL queries on data streams, where the latency demands are very strict and computations necessitate retaining the state from prior computations, it is extremely difficult to create scalable SQL engines that are effective across all workloadsScalable SQL engine design is now more challenging. When trying to deal with the constantly expanding spectrum of data sizes, a variety of techniques can be utilized to help reduce data and also arrange it in the most efficient way possible. It can be said that there are four main methods for running SQL on large amounts of data.[38]

[38] Pal S., SQL on Big Data; Technology Architecture, and Innovation, Apress, Wilmington, Massachusetts, USA, 2016

Figure 11. Approaches to building SQL on Hadoop engines



Source: Pal S., SQL on Big Data; Technology Architecture, and Innovation, Apress, Wilmington, Massachusetts, USA, 2016

1. One of the approaches is to construct a translating level which takes statement in SQL and converts them into similar MapReduce code that can then be executed on the particular cluster. Apache Hive is the greatest illustration of a batch-oriented SQL on Hadoop tool. In this system, the intermediate processing layer is composed of MapReduce and Apache Tezas. It is used for the purpose of performing complicated processes, such as ETL and production data "pipelines," against enormous data sets. This method is illustrated in figure 11. in the third block.

2. Second approach is that current relational engines, that had previously evolved for over more than 40 years of research, also development and include all of the storage engine and query improvements that have been developed since their beginning. Installing MySQL and PostgreSQL on every data node that makes up the Hadoop cluster would be an excellent example of this, as would also creating a layer inside each of the nodes that permits access to the data which is kept in the underlying distributed file system. The relational database management system engine is co-located with the node of the data, in order to successfully manage data from the Hadoop distributed file system, and then transforms the received data into their own unique data format. This approach is illustrated in figure 11. by the fourth block.

3. Third approach is to construct a new query engine that directly works with the data stored on HDFS and is on the same nodes as the data nodes. This will allow running SQL queries. The query engine employs a search splitter which can then send query fragments to the data handlers, the reason of that is to reach and analyze the raw data. The described approach is illustrated in figure 11. by the second block.

4. The fourth strategy involves utilizing currently operational analytical databases which are set up on a different cluster from the Hadoop and communicate with the data nodes in the Hadoop cluster. Such databases access HDFS data using a dedicated connection, but SQL queries are executed inside the analytical engine itself. Those additional analytical engines might be incorporated so they use the Hive or HCatalog metadata. This allows them to interact easily with the data stored in HDFS. The approach is illustrated in figure 11. in the first block

5.3. NoSQL Database Context with Big Data Analytics

New advancements reported in the existing literature show that in the modern sense, there has been an drastic increase in data volume that is both structured and unstructured data from the wide range of data sources. This phenomenon is known as "Big Data." Accordingly, we are able to claim that structured, semi-structured, and unstructured data obtained from electronic and other sources are present in big data, which sets it apart from other data types. A main issue is utilizing large sets of data, which serve as the source of the main data for effective decisions, in an effective way by applying appropriate data mining methods. The current Big Data challenges are brought on by the following characteristics of business in general:[39]

- High data velocity - refers to the rapid and continual updating of information traffic from various sources and places.
- Data Variety
- Volume of data – refers to big amounts of various datasets
- Complexity information – data arranged across different websites or data hubs.

[39] Ali, W., Usman Shafique, M., Arslan Majeed, M., Raza, A., Comparison between SQL and NoSql Databases and their Relationship with Big Data Analytics, Asian Journal of Research in Computer Science, 4(2): (2019), pp.1-10

With the help of an ACID, large transactional processes do not operate correctly. As a result, the ACID system is demonstrated to be an issue in various distributed programs which is not entirely solvable[40]. Big data analytics refers to the process wherein businesses analyze large data sets that include several types of data. These data sets might come from a variety of sources. Firms are able to analyze vast amounts of data more accurately from the use help of Big Data Analytics. This enables the companies on revealing previously unseen patterns, undiscovered connections, marketing trends, and other information helpful to their businesses. Big data analytics are dependent on massive number of datasets, which need the use of clusters for data storage in order to enable fast and efficient decision-making. Companies are beginning to examine the necessity for the NoSQL movement as a result of the inapplicability of relational databases to clusters and the efficiency concerns that are shown in connection to Big data analytics.

There is no fixed NoSQL schema. It makes use of numerous gateways for storing and processing the sheer volume of user-generated content, personal data, and spatial data produced by cutting-edge apps[41] For this instance, the NoSQL database presents more advantageous choice than the SQL Database does. The NoSQL database's ability to manage horizontal data splitting, dynamic data analysis, and an overall improvement in performance are primarily responsible for this. Major online corporations like Google, LinkedIn, Amazon and Facebook have studied and initiated the progression of NoSQL as a possible answer to their issue with constantly expanding data. These organizations are not able to manage assistance via the use of modern relational databases. In comparison to the dependability and very global nature of systems based on a three tier Internet architecture and cloud computing, the NoSQL database can be seen as the superior option for information technology systems that feature high levels of efficiency and dynamics.

5.4. Advantages and Disadvantages Between SQL and NoSQL

Data can be entered into SQL, searches can be sent, data can be updated and deleted, and the source of access can be managed. SQL is ideal for specific types of jobs, particularly modifying as well as obtaining collections of data. SQL is also excellent for controlling the point of access. Advantages of using SQL as a database language is that it enables users to do operations such

[40] Ahmed R., Khatun A., Ali A., Literature riview on NoSQL Database for Big Data Processing, International Journal of Engeneering & Technology, 7 (2), (2018), pp. 902-906

[41] Abhishek P, Bhavesh N. Gohil, A comparative study of NoSQL databases, International Journal of Advanced Research in Computer Science;5(5), (2014)

as inserting, updating, deleting, or retrieving data using straightforward instructions. Users are granted the ability to do administrative tasks and administer the database thanks to this feature.

SQL is considered to be a very powerful and universal language, which is for some new users easy to learn and understand. It has portable characteristics and multiple data views. It can also be used with any database management system, with any vendor. Its can be defines also as both programming language, as well as an interactive language. Everything considered it is a complete language for a database managing. SQL can also support object based programming and enterprise applications. It can run queries with great speed, and can also intagrate with Java.

In the view od NoSQL, and it's advantages, when the transaction rates and the need for a rapid response rise, the scaling up approach becomes ineffective. This is what is meant by the term "high scalability." On the other hand, the most recent iteration of NoSQL databases are designed and intended to scale out, or also extend horizontally utilizing low-end inexpensive servers. This was done in order to compete with traditional relational databases. NoSQL databases are meant to function primarily with automated repairs, distributed data, and simplified data models. As a result, NoSQL databases have a low manageability and administration level.[42] Additionally, It is common practice for NoSQL databases to be designed with the intention of functioning with a cluster of commodity servers that are relatively inexpensive. This provides the users with the ability to store and analyze a greater quantity of data at a reduced cost. NoSQL databases are able to cope with a wide variety of data formats since their data models are relatively flexible, they don't comply to the RDBMS's strict data models. Therefore, any modifications to the application that need an update to the database schema can be easily implemented.

In addition to the previously mentioned benefits od NoSQL, users also must be aware of a number of challenges before beginning application development on these platforms. When it comes to maturity, the vast majority of NoSQL databases are considered to be still in their pre-production stages, and a number of their most significant characteristics have not yet been implemented. As a result, organizations that are considering using a NoSQL database should conduct a thorough analysis of the product to ensure that their requirements have been fully implemented and are not on a "to-do" list before making a final decision. Users have a responsibility to be aware of the restriction that relates to support. When compared to large

[42] Nayak, A., Poriya, A. & Poojary, D. (2013), „Type of NOSQL Databases and its Comparison with Relational Databases“, International Journal of Applied Information Systems (IJAIS), 5(4) Foundation of Computer Science FCS, New York, USA.

enterprise software companies, the majority of NoSQL databases are open source projects and as a result, there are one or more companies that offer support for these databases. These companies, however, are frequently younger and smaller startups, and they may lack a worldwide reach or enough resources. Due to the fact that NoSQL databases were primarily built to suit the scaling requirements of web-scale applications, the querying capabilities of these databases are restricted. As a result, NoSQL databases feature limited querying capabilities. A simple querying need may demand extensive programming skill from a user. Because NoSQL is a developing field, there is a shortage of people in the developer and administrator communities who are knowledgeable with the technology. Despite the fact that NoSQL is quickly growing to be an important part of the database landscape, users must be conscious of the systems' limitations and advantages in order to decide which NoSQL database platform to use.

5.5. Case Study: MySQL versus MongoDB

Oracle created the popular relational database management system or RDBMS known as MySQL.The program can be referred as a free source available to anyone who wants to use it. similar to other programs base on relational methods, MySQL arranges the data which is entered in the database with rows and columns, maintains relational integrety and enables users to retrieve their data using a a query language which is structured (SQL). People who require access to data from the database should construct a SQL query that will then join numerous tables to produce the desired display of the data that was trying to be viewed. If they do not do this, they will not be able to retrieve particular data they wanted to have. Database designs and database types must be created in preparation, as well as the data which will be kept inside the same database and must fit to the schema that has been chosen. The data may be stored with some degree of safety using this inflexible method, but the flexibility is sacrificed in the process. If the database has to hold data in a different kind or format, a process called schema migration must be performed. This process, which may become difficult and costly as the size of the database increases, is required.

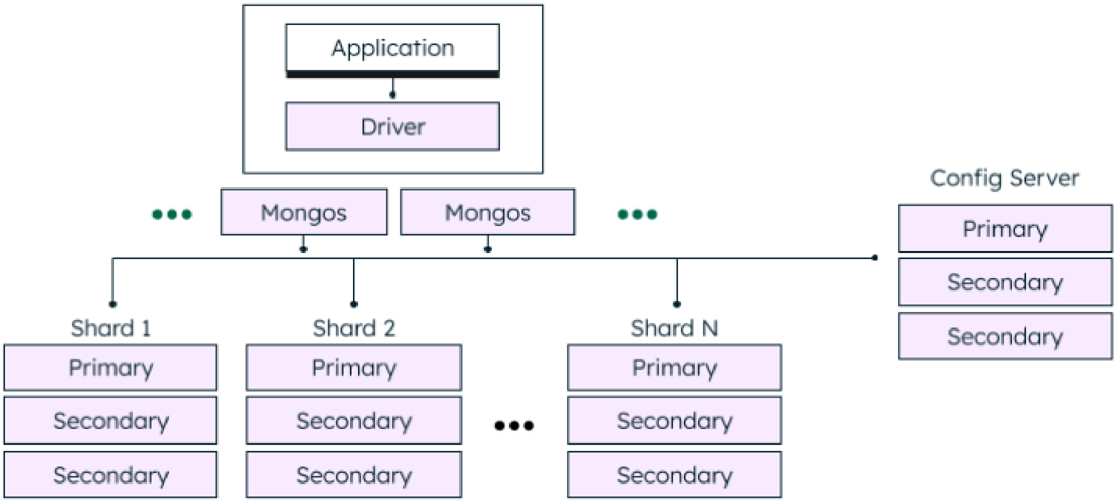
In other case, MongoDB is jus like MySQL an open free source, available to anyone, but its architecture differs from that of conventional databases in some fundamental respects. When it comes to the manner in which it stores data, MongoDB, which is regularly alluded to as a non-relational system or a NoSQL system, takes a fundamentally different approach. Instead of using relational systems' table and row format, it depicts data as a collection of documents that resemble JSON but are actually saved as binary JSON, and BSON. The crucial distinction, however, is that the organization of the key/value pairs in a given collection might vary from document to other document. MongoDB documents are made up of a sequence of key/value pairs of various forms, which includes arrays and nested documents. Due to the fact that documents are self describing, this technique, which is more flexible, is possible.

These two database management systems have fundamental distinctions that are important to note. The choice to choose one over the other is not one that is only based on technical considerations, rather, it is an issue of strategy. MySQL is a well-established relational database technology that provides knowledgeable IT professionals with a database environment that is comfortable to them. A non-relational database system with a long history that provides improved adaptability and horizontal scalability is titled MongoDB. On the other hand, this comes at the sacrifice of certain security features that relational databases provide, such as referential integrity. Developers have a variety of options to choose from, and MongoDB is one of them. Anyone who has ever tried in programming will have no trouble grasping its straightforward and easily comprehensible data storage methodology. Data is stored as collections inside MongoDB, although there is no defined schema. Because it takes such a flexible approach to storing data, it is especially well-suited for application developers who wish to use a database to help the development of their apps but may not have extensive knowledge of databases. This flexibility is a huge benefit when compared to MySQL. In order to get the most out of a relational database, users must first learn the fundamentals of relational database architecture, including normalization, referential integrity, and relational database design.

For members of a team who are going to create programs without needing security capabilities that are given by relational database systems, MongoDB offers a versatile programming interface. This is made possible by MongoDB's ability to store documents with the help of varying schemas, which also includes the ability to store unstructured data sets. This kind of application typically takes the form of a web app without the use of organized schemas. Such an application may easily provide sttructured, semistrusctured and unstructured data, all of which come out of the existing collection in MongoDB. Users who have a lot of expertise with old SQL, and can program in it, and are often in the field of updating programs in the field of relational databases, often choose a system MySQ. For application that require exceedingly complex but also exacting data structures as well as database schemas spread across a vast amount of tables, relational databases could also be the better choice. This is because relational databases can store and organize data in a more organized manner.

A typical example of this kind of system may be a banking application, which requires the enforcement of highly stringent referential integrity and transactional guarantees in order to preserve the precise point-in-time integrity of data. It should be made clear MongoDB also enables ACID features from transactions. This offers a higher degree of freedom in the construction of a transactional data model that is capable of horizontal scaling in an environment that is distributed and hasn't got effect whatsoever on the performance of multi-document transactions. The ease of scalability of the database is one of the main benefits of the MongoDB design. When users create a sharded cluster, a section of the database that is referred to as a shard may also be set up to function as a replica set. A collection of MongoDB servers that duplicate the very same data among them is known as a replica set. In the event of a data loss, this enables data recovery and great availability.

Figure 12 Sharded cluster architecture in MongoDB used in production



Source: MongoDB, Sharded Cluster Components; Available at [online]: https://www.mongodb.com/

In order to distribute data that has been sharded, sharding needs a minimum of two shards. Clusters with a single shard are an option to consider if users anticipate the need to enable sharding in the near future but do not need it at the time of deployment. High availability and scalability may be achieved with the deployment of numerous Mongos routers. The placement of mongos instances on the same hardware that mongod instances are already operating on is a typical pattern that is used for the purpose of achieving high availability at the shard level. The incorporation of MongoDB routers into application-tier infrastructure is another another possibility. There is no limit on the total number of Mongos routers that may be used in a single deployment. However, since mongos routers regularly interface with the config servers, users need to pay careful attention to how well the config servers are performing as users expand the number of routers. If auser begins to see a decline in performance, it is possible that it would be useful to place a limit on the number of mongos routers in the deployment. MongoDB is able to horizontally expand both its read and write speed because to its very flexible methodology, which enables it to accommodate applications of any size.

Scalability possibilities are far more constrained when using a MySQL database management system. Users typically have two options available to them and that is either vertical scalability or the addition of read replicas. In order to scale vertically, more resources must be added to the already present database server, however, the strategy that implies this has a capacity limitation. The read replication process includes adding read only copies of the database to other servers. On the other hand, this is normally restricted to a total of five replicas, and these replicas may only be used for read operations. Due to the fact that it is usual for replicas to lag behind the write master, this may create problems for applications that either do a lot of writing to the database or write and read from the database on a frequent basis. MySQL now has the capability to enable multi-master replication, however the implementation is not as powerful as the capabilities offered by MongoDB.

It is quite difficult to evaluate the performance of two databases that are totally distinct from one another due to the fact that their respective management systems take very different approaches to the process of storing and retrieving data. Using a set of common SQL benchmarks, it is possible to evaluate two SQL databases directly, but comparing them between nonrelational and relational databases which is far harder and susceptible to interpretations. MySQL, for instance, today is designed as a great speed joins through numerous databases as long as those tables have been adequately indexed. MongoDB's documents typically apply to a hierarchical data model and store the majority of the data in a single document, which eliminates the requirement for joins that span through multiple documents. Joins can be performed in MongoDB using the $lookup operation, but their use is less common than it once was because of the way MongoDB documents are typically used. The following syntax may be used on the $lookup stage in order to carry out an equality comparison among a field from the inputted document and a field from the documents that make up the "joined" collection:

```
{
  $lookup:
    {
      from: <collection to join>,
      localField: <field from the input documents>,
      foreignField: <field from the documents of the "from" collection>,
      as: <output array field>
    }
}
```

The "from" field is used to provide the collection from the same database that the join should be performed with The field from the documents that are input to the $lookup stage is defined by LocalField. Using the localField and foreignField from the documents in the from collection, $lookup conducts an equality match. The localField is treated as having a null value by the $lookup for matching purposes if the input document does not contain it.. The field that was extracted from the documents that were part of the from collection is specified by ForeignField. By doing an equality match on the two fields, $lookup checks whether the foreignField and the localField in the input documents are identical. In order to ensure accurate matching, the $lookup function considers a foreignField value to be null if a document in the from collection does not have the field in question. As is used to provide the name of the new array field that should be added to the input documents. The documents that match those in the from collection are stored in the newly created array field. If the input document already has a field with the specified name, then that field will be overwritten.

The same operation would correspond with the following SQL statement:

```
SELECT *, <output array field>
FROM collection
WHERE <output array field> IN (
  SELECT *
  FROM <collection to join>
  WHERE <foreignField> = <collection.localField>
);
```

MongoDB is also designed for write performance, and it includes a particular insertMany() application programming interface for swiftly adding data. This API places a higher priority on speed than transaction safety, while data in MySQL must be entered one row at a time. Following is the syntax of the InsertMany method:

```
db.collection.insertMany(
  [ <document 1> , <document 2>, ... ],
  {
    writeConcern: <document>,
    ordered: <boolean>
  }
)
```

The parameter document the range of documents that may be inserted in the collection. WriteConcern is an optional parameter. A written document that expresses the problem in question. Do not use the write concern that is set as default. If the operation is being conducted inside a transaction, users should not explicitly specify the write concern for the operation. Parameter ordered is also considered as optional, and its type is boolean, which specifies whether should the mongo instance insert data in an ordered or unordered manner. Boolean refers to a system of true or false statements. Documents are placed in an unsorted form and can be reorganized by mongod to improve performance if ordered is set to false. When employing an unordered InsertMany, applications shouldn't rely on the sequence of the inserts. When comparing several of the high-level query characteristics of the two systems, we observe that MongoDB is substantially faster when inserting or updating a large amount of data and MySQL is when selecting a big quantity of records.

In the terms of flexibility, MongoDB can be considered as a much better system. Because of the schemaless design of MongoDB documents, it is incredibly simple to construct and improve applications over time. This eliminates the need to conduct difficult and time-consuming schema conversion processes, as would be required when working with a relational database. There are additional dynamic choices available for modifying the schema of a collection when using MongoDB. These options include the creation of new fields based on an aggregate pipeline as well as the update of nested array fields. This advantage becomes much more significant when the size of databases continues to expand. The db.collection.update() method by definition only refreshes one document. When updating all documents that meet the query criteria user can include the option multi: true, it's sytnax will be shown in the following.

```
db.collection.update(
  <query>,
  <update>,
  {
    upsert: <boolean>,
    multi: <boolean>,
    writeConcern: <document>,
    collation: <document>,
    arrayFilters: [ <filterdocument1>, ... ],
    hint:  <document|string>, // Added in MongoDB 4.2
    let: <document> // Added in MongoDB 5.0
  }
)
```

The parameter of db.collection.update() makes changes to a document or documents that are already part of a collection. The update parameter's value determines which fields in the particular document or documents can be modified by the procedure, or it can completely replace an existing document. Compared to other database management systems, MySQL databases migrate schemas and stored procedures that depend on updated schemas more slowly. Because of MongoDB's adaptable architecture, this is far less of a cause for concern.

In the views of the security, MongoDB makes use of the common role-based access control concept and provides a versatile range of permissions for its users. Users are given a role, and the role determines the rights they have over the activities that may be performed on the database and the datasets. In order to ensure that data is encrypted while it is at rest, it is feasible to write encrypted documents to MongoDB data collections using a master key which is never known to MongoDB. All communication is secured using Transport Layer Security, also known as TLS. Both MySQL and MongoDB have the same encryption technologies, also MySQL and MongoDB use very comparable authentication models. Users can be given roles, but they can also be assigned privileges, which give them rights over certain database actions and against specified datasets.

6. CONCLUSION

For a number of years, SQL databases have been the standard in this business industry. NoSQL database, on the other hand, is the solution to problems like Big Data, as current corporate circumstances have required the need to store and handle enormous datasets for the purpose of doing business analytics. NoSQL can give their users a schemaless data storage and transactions, which allows companies to flexibly add fields to records even without formal necessity of specifying the schema. NoSQL graphs are really an excellent choice for managing data that contains complex patterns of relationships. In addition, through key-value stores simplicity can be achieved. This is because of the growing requirement to manage massive amounts of data and business transactions that are unstructured using platforms such as some of the socializing networks. NoSQL data models offer a variety of alternatives for the storage if the unstructured data, including document-oriented, key-value pair-oriented, column-oriented, and graph-oriented models respectively. NoSQL is, ideally speaking, superior to SQL when it comes to the field of analytics as well as the support of business intelligence. The SQL language has its limitations due to worries about scalability and complexity. Relational databases cannot be scaled without the use of powerful servers, which is both expensive and difficult to manage. The relational database has to be deployed across numerous servers, which is the root cause of the scalability problems that have emerged. Due to the fact that the data must comply to the table structure, the complexity of the SQL server can be a constraining factor. NoSQL deals with the unstructured data type. The NoSQL database does not require fixed table schemas. NoSQL is beneficial in that it is easily scalable; hence it can be used for analytics. Keeping NoSQL servers maintained is also more cost effective because to the low cost of the servers themselves. The use of large amounts of data to the task of providing assistance for decision making in the context of the manufacturing environment reveals a great deal of prospects for businesses.. Analytics can be used to predict the demand for products; hence less time is wasted. The manufacturing process also potentially improves accuracy and the quality of output since they can pinpoint various metrics of production using big data analytics. Using faster response database architectures such as NoSQL allows the management to perform better simulations, affecting the real world decision to make a particular product. NoSQL makes it possible to process information at huge scale in a manner that is both high-performance and agile. It saves unstructured data across a wide variety of processing nodes and servers at the same time. As a direct consequence of this development, the NoSQL distributed database infrastructure have emerged as the most popular choice for the large majority of the biggest data warehouses worldwide.

Just as in the case sudy where there was an comparison between MySQL an MongoDB, documents in MongoDB transfer well to current object-oriented programming languages, development can be simplified as a result. When users use MongoDB, the complicated layer of object-relational mapping, which can turn objects written in code into relational tables is removed. Due to the flexibility of MongoDB's data model, the structure of the database may adapt to changing needs as they arise in any company. Because of its rigorous relational structure, MySQL adds more work and complexity to programs, and it makes it more difficult for developers to write code because they have to adapt objects in code to a relational structure. MongoDB also has the capacity to grow both locally and across numerous distributed data centers, which enables it to provide new levels of availability and scalability that were previously impossible to achieve with relational databases like MySQL. MongoDB is able to readily scale without requiring any downtime or modifications to the system, even as the data volume and throughput of the deployments continue to increase. On the other hand, reaching scale with MySQL frequently calls for a large amount of customized engineering work. Internet firms developed NoSQL databases as a means of more effectively managing and analyzing datasets to meet the demand for data management and to deal with the interdependency and complexity of big data that is only increasing. In terms of which SQL or NoSQL is better for an analysis, this depends on a wide variety of circumstances, such as the kind of data that is being analyzed, the quantity of data that is available, and the urgency with which the information is required. Relational databases are ideal for use in applications such as the analysis of user behavior, for instance. If the information can be arranged in a spreadsheet, it would be more suitable for a SQL-style database. This is the reason because SQL database systems are a type of relational database that are useful for performing analysis on data that is laid out in rows and columns. NoSQL databases like MongoDB,function the best for semi-structured data, social media, texts, and geographical data that require a significant amount of text mining or image processing. Analysis of these kinds of databases requires the expertise of a data scientist because it is difficult to execute analytics on semi-structured data without a strong coding foundation.

REFERENCES

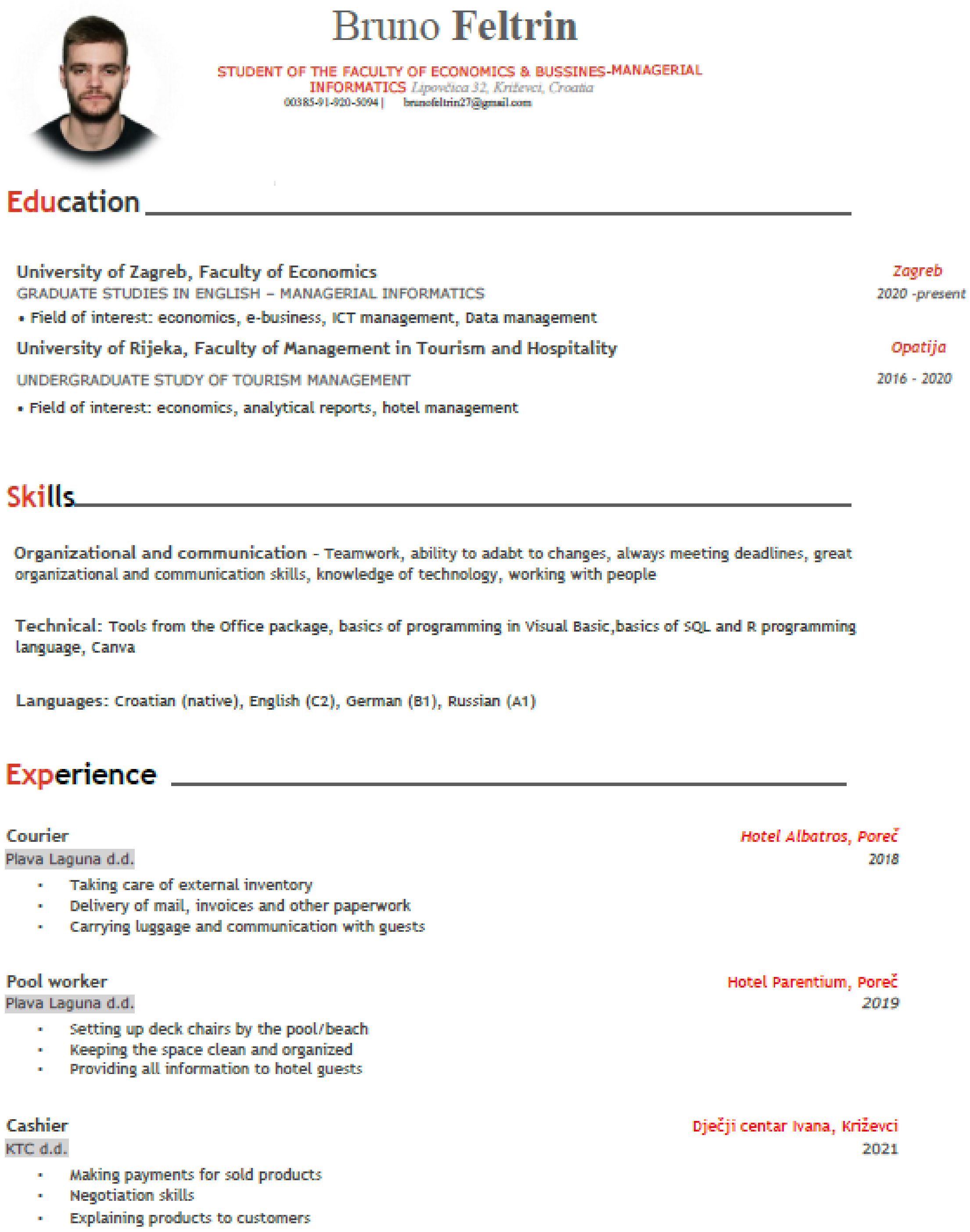
1. Abhishek P, Bhavesh N. Gohil, A comparative study of NoSQL databases, International Journal of Advanced Research in Computer Science;5(5), (2014)
2. Ahmed R., Khatun A., Ali A., Literature riview on NoSQL Database for Big Data Processing, International Journal of Engeneering & Technology, 7 (2), (2018), pp. 902-906
3. Ali, W., Usman Shafique, M., Arslan Majeed, M., Raza, A., Comparison between SQL and NoSql Databases and their Relationship with Big Data Analytics, Asian Journal of Research in Computer Science, 4(2): (2019), pp.1-10
4. Alvaro, F. "SQL: Easy SQL Programming & Database Management For Beginners." Your Step-By-Step Guide To Learning The SQL Database, Amazon, eizdanje (2018)
5. Challawala, J. Lakhatariya, C. Mehta, K. Patel, MySQL 8 for Big Data; Effective data processing with MySQL 8, Hadoop, NoSQL APIs, and other Big Data tools, Pact Publishing, Mumbai, (2017)
6. Dean J., Big Data, Data Mining and Machine Learning; Value Creation for Business Leaders and Practitioners, John Wiley & Sons, New Jersey, (2014)
7. Deari, R., Zenuni, X., Ajdari, J., Ismaili F. & Raufi, B., Analysis and Comparison of Document-Based Databases with SQL Relational Databases: MongoDB vs MySQL, Proceedings of the International Conference on Information Technologies, Bulgaria, (2018)
8. Dietrich D., Heller B., Yang B., Data Science & Big Data Analytics; Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, Indianapolis (2015)
9. Eds. C. H. Chen, L. F. Pau & P. S. P., Wang, Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Company, (1998), pp. 3-32
10. Fotache M., Catalin S., SQL and Data Analysis. Some Implications for Data Analysits and Higher Education, Procedia Economics and Finance, (2015), pp. 243-251
11. Gorelik A., The Enterprise Big Data Lake, Delivering the promise of Big Data and Data Science, O'Reilly Media, USA; Sebastopol, (2019)
12. Groff J. R. & Weinberg P. N., SQL: The Complete Reference, Osborne/McGraw-Hill, California, (1999)
13. Harrison G.,Next Generation Databases, MySQL, NewSQL and Big Data; What every professional needs to know about the future of databases in a world of NoSQL and Big Data, Apress, New York

14. Isson J. P., Unstructured Data Analytics; How to Improve Customer Acquisition, Customer Retention, and Fraud Detection and Preventio, John Willey & Sons, Canada, (2018)
15. Joczik, S. & Kiss, A., Quantum Computation and Its Effects in Database Systems, Faculty of Informatics, Eotvos Lorand University, 1117 Budapest, Hungary, (2017)
16. Kumari, Khushbu & Yadav, Suniti. Linear regression analysis study. Journal of the Practice of Cardiovascular Sciences. 4. 33. (2018), 10.4103/jpcs.jpcs_8_18
17. Maalouf, M., Logistic regression in data analysis: An overview. International Journal of Data Analysis Techniques and Strategies. 3. 281-299. 10.1504/IJDATS.2011.041335, (2011)
18. Marr, B., Big Data-Using smart Big Data Analytics and metrics to make better decisions and improve performance, John Willey & Sons Ltd, Southern Gate, (2015)
19. McCreary D. & Kelly A., Making Sense of NoSQL; A guide for managers and the rest of us, Manning Publications, New York, (2014)
20. Mitreva, E. & Kaloyanova, K., NoSQL Solutions to Handle Big Data, Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”, 5, James Bourchier Blvd., 1164 Sofia, (2013)
21. Moniruzzaman, A. B., & Hossain, S. A. NoSQL database: New era of databases for big data analytics - Classification, characteristics and comparison. International Journal of Database Theory and Application, 6(4), (2013), pp.1-14
22. MongoDB, Sharded Cluster Components; Available at [online]: https://www.mongodb.com/
23. Nayak, A., Poriya, A. & Poojary, D. „Type of NOSQL Databases and its Comparison with Relational Databases‟, International Journal of Applied Information Systems (IJAIS), 5(4) Foundation of Computer Science FCS, New York, (2013)
24. Pal S., SQL on Big Data; Technology Architecture, and Innovation, Apress, Wilmington, Massachusetts, USA, 2016
25. Pandey, G., Cawla, S., Poon, S., Arunasalam, B. & Davis, J. G., Association Rules Network: Definition and Applications, Wiley Periodicals, Minneapolis, (2009), pp. 260–279
26. Pothuganti, A. 'Big Data Analytics: Hadoop-Map Reduce & NoSQL Databases', International Journal of Computer Science and Information Technologies, 6(1), (2015), pp. 522-527

27. Quinlan, J. R., Generating production rules from decision trees. In ijcai, Vol. 87, (1987, August), pp. 304-307.
28. Senthil, N., Govindarajan, C., Saraf, A., Sethi, M. & Jayachandran, P., Blockchain Meets Database: Design and Implementation of a Blockchain Relational Database, VLDB Endowment, Vol. 12, No. 11, ISSN 2150-8097. (2019)
29. Silva Y. N., Almeid I., Queiroz M., SQL: From Traditional Databases to Big Data, Arizona State University, (2016)
30. Strauch, Christof and Walter Kriha. "NoSQL databases." Lecture Notes, Stuttgart Media University 20, no. 24 (2011)
31. Sulivan D., NoSQL For Mere Mortals, Pearson, Oregon, (2015)
32. Tanimura, C., SQL for Data Analysis, Advanced Techniques for Transforming Data into Insights, O'Reilly Media, Sebastopol, (2021); [online] https://learning.oreilly.com/library/view/sql-for-data/9781492088776/cover.html
33. Taylor G. A., SQL for Dummies-9th Edition, John Walley & Sons Inc, Hoboken; New Jersey, (2019)
34. Vaish, G., Getting Started With NoSQL-Your guide to the world of technology and NoSQL, Packt Publishing, Birmingham, (2013)

List of figures

Students's CV



Bruno **Feltrin**

STUDENT OF THE FACULTY OF ECONOMICS & BUSSINES-MANAGERIAL INFORMATICS *Lipovčica 32, Križevci, Croatia*
00385-91-920-5094 | brunofeltrin27@gmail.com

Education

University of Zagreb, Faculty of Economics — *Zagreb*
GRADUATE STUDIES IN ENGLISH – MANAGERIAL INFORMATICS — *2020 -present*

- Field of interest: economics, e-business, ICT management, Data management

University of Rijeka, Faculty of Management in Tourism and Hospitality — *Opatija*
UNDERGRADUATE STUDY OF TOURISM MANAGEMENT — *2016 - 2020*

- Field of interest: economics, analytical reports, hotel management

Skills

Organizational and communication - Teamwork, ability to adabt to changes, always meeting deadlines, great organizational and communication skills, knowledge of technology, working with people

Technical: Tools from the Office package, basics of programming in Visual Basic,basics of SQL and R programming language, Canva

Languages: Croatian (native), English (C2), German (B1), Russian (A1)

Experience

Courier — *Hotel Albatros, Poreč*
Plava Laguna d.d. — *2018*

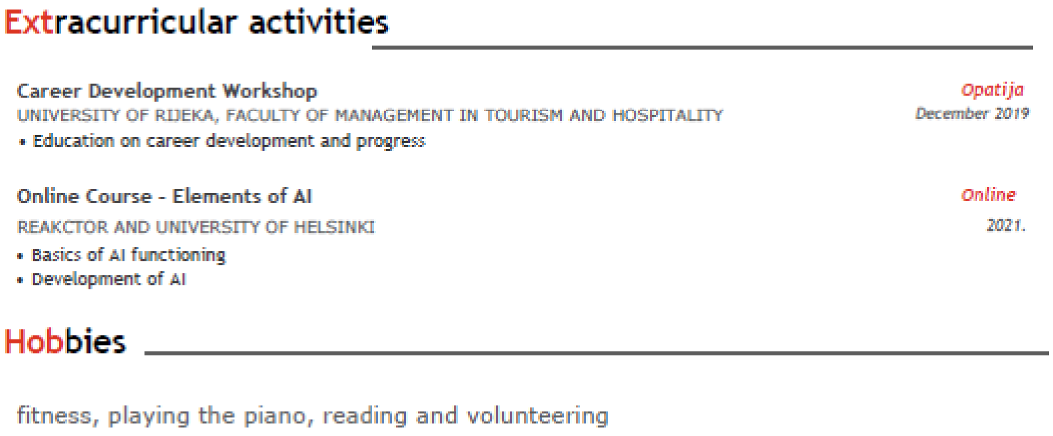
- Taking care of external inventory
- Delivery of mail, invoices and other paperwork
- Carrying luggage and communication with guests

Pool worker — Hotel Parentium, Poreč
Plava Laguna d.d. — *2019*

- Setting up deck chairs by the pool/beach
- Keeping the space clean and organized
- Providing all information to hotel guests

Cashier — Dječji centar Ivana, Križevci
KTC d.d. — 2021

- Making payments for sold products
- Negotiation skills
- Explaining products to customers



Extracurricular activities

Career Development Workshop — *Opatija*

UNIVERSITY OF RIJEKA, FACULTY OF MANAGEMENT IN TOURISM AND HOSPITALITY — *December 2019*

- Education on career development and progress

Online Course - Elements of AI — *Online*

REAKCTOR AND UNIVERSITY OF HELSINKI — *2021.*

- Basics of AI functioning
- Development of AI

Hobbies

fitness, playing the piano, reading and volunteering