Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Systems Engineering (FEM)

Diploma Thesis

Compensation Benchmarking Database

Harold Miranda

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

MSc. Harold Miranda, MBA

Systems Engineering and Informatics
Informatics

Thesis title

Compensation Benchmarking Database

Objectives of thesis

The objective of the Compensation Benchmarking Database is to implement a database application that will produce salary market data (salary, bonus and total compensation) for job functions based on the provided data.

Methodology

This thesis will consist of two parts. First one will include the theoretical overview of tools, methods and approaches used in this thesis. The second part will speak about the database tool which will include:

- To gather employee's data from a selected industry or in its defect to model employee's data based on market references.

- To clean and to process data in order to get standardized data

- To create a relational data database or an object-oriented database

- To create a front-end tool to input data and to publish a survey report (tentatively)

The proposed extent of the thesis

80 – 140 pages

Keywords

Salary Survey, Salary Database, Database Application

Recommended information sources

Date, C J. An Introduction to Database Systems. Pearson Education, Inc. 2004. Eighth Edition

Levine David, Stephan David, Krehbiel Timothy, Berenson Mark. Statistics for Managers Using MicrosoftExcel. Prentice Hall. 2008. Fifth Edition

Expected date of thesis defence

2020/21 SS – FEM

The Diploma Thesis Supervisor

doc. Ing. Vojtěch Merunka, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 12. 03. 2021

Declaration

I declare that I have worked on my diploma thesis titled "Compensation Benchmarking Database" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break any copyrights.

In Prague on 30.03.2021 ______________________

Acknowledgement

I would like to thank assoc. prof. Vojtěch Merunka, Ph.D. and my parents for their advice and support during my work on this thesis.

Compensation Benchmarking Database

Abstract

The work presented in this project thesis employed the knowledge acquired in Relational Databases to develop a Compensation Benchmarking Database to produce salary, bonus and total compensation for companies in the Financial Services industry. More precisely, I focused on the United States Corporate Banking segment.
To accomplish my goal, I reviewed subjects related to Compensation, Statistics and Databases to use some fundamental knowledge to develop the database. This database serves as a tool to store and query salary compensation data. During the process, I used my own data work to model the compensation dataset for several companies. Additionally, I covered the tasks related to cleaning, processing and creating and querying the database.

Keywords: Compensation, Corporate Banking, Database Application, Financial Services, Relational Database, Salary Database, Salary Survey, Statistics, Structured Query Language, SQL.

Kompenzační srovnávací databáze

Abstrakt

Tato diplomová práce využívá znalosti získané v oblasti návrhu relačních databází k vývoji databáze kompenzačních výkonnostních testů na tvorbu platů, bonusů a celkových kompenzací pro společnosti v odvětví finančních služeb. Přesněji je zaměřena na segment korporátního bankovnictví v USA. Návrh databáze začal studiem předmětů souvisejících s kompenzacemi, statistikami a bázemi dat, které byly využity jako východisko k vývoji databázové aplikace. Tato aplikace slouží jako nástroj k ukládání podkladů a dotazování na údaje o odměňování. Tvorba využívá vlastní způsob práce s daty k modelování datového souboru kompenzací pro několik společností. V práci jsou dále úkoly související s čištěním, zpracováním a vytvářením a dotazováním na bázi dat.

Klíčová slova: Kompenzace, Firemní Bankovnictví, Databázové Aplikace, Finanční Služby, Relační Databáze, Databáze Platů, Průzkum Platů, Statistika, Jazyk Strukturovaných Dotazů, SQL.

Table of content

List of figures

List of tables

List of abbreviations

DB — Database

DBMS — Database management system

DDL — Data Definition Language

DML — Data Manipulation Language

TComp — Total Compensation

MCT — Measures of Central Tendency

M — Mean

Mdn — Median

MO — Mode

Q — Quartile

SQL — Structured Query Language

1NF — First Normal Form

2NF — Second Normal Form

3NF — Third Normal Form

4NF — Fourth Normal Form

BCNF — Boyce Codd Normal Form

DDL — Data Definition Language

DDM — Data Manipulation Language

TCL — Data Control Language

DCL – Data Control Language

1 Introduction

In the business world, successful companies are not just companies having better resources and technology, but also companies that know how to get and use the right information to make decisions.

In this project thesis, I worked with a relational database and structured query language (SQL) to store, manipulate and query modelled compensation data on salary, bonus, and total compensation for employees in the United States Corporate Banking segment.

This project thesis is an example of how a business need can be satisfied by using technology. As a result, the database becomes a good source for querying and answering questions from Human Resources managers interested in attracting, retaining, motivating or letting go prospects or current employees.

2 Objectives and Methodology

2.1 Objectives

This project thesis objective is to create a relational database application to produce salary market data such as salary, bonus, and total compensation for functions based on the provided data. To achieve my objective, I used MySQL Workbench to model the database, store data, and create a set of queries attached to this thesis. In the end, the database serves to summarize market data in the form of measures of central tendency and variation. This type of database intends to answer requirements from a Human Resources or a Compensation manager who wishes to benchmark his/her company's current salary levels against the market's salary levels.

2.2 Methodology

To achieve the proposed objective there are some steps to be followed:

1. Theoretical overview of methods and approaches used in this thesis, which include:
   a. Database, Relational Database and SQL introduction.
   b. Information on compensation elements that could be included and reported in the database.
   c. Statistical definitions on measures of central tendency.

2. Interconnected to the prior section, the second part will cover the practical phase about the database development and the corresponding reporting:

   a. Gathering and/or modelling employee's data from the financial industry (based on market references).
   b. Data cleaning and processing to standardize data.
   c. Creating a relational database.
   d. Querying the database.

3 Literature Review

3.1 Theoretical Background: Databases

3.1.1 Introduction

Before even covering databases (DB), it is appropriate to define the term data. Data itself can be confusing if it is kept untouched and unorganized. Everybody has some data in their hands, as a data producer or as a data receptor. But what is data? According to the popular Merriam-Webster online dictionary, data is information based in facts ultimately employed to reason, discuss or calculate.

Given this introduction, the data principal repository is the database itself and the larger body to manage databases is the Database Management System (DBMS). Based on the explanation offered by Connolly & Begg in "Database Systems. A Practical Approach to Design, Implementation and Management" 2015 on page 52, I can define a database and DBMS in my terms. The database is a shared grouping of logically related data designed to meet an organization information needs and DBMS is the software in charge of managing and controlling access to a database.

In the business world, companies to continue with daily operations need to deal with data. It is a fact that businesses acquire, store, process, and manipulate large amounts of data; without much doubt, data should be handled efficiently.

Regarding the usefulness of data, Coronel & Steven, in their book "Database Systems: Design, Implementation, and Management" 2019, on page 3, convey the idea that data can flow to different parts of the company and become a tool that helps management make educated decisions when handled appropriately. Therefore, the need to properly manage data gives space for the birth of a database. As it is known now, databases are there hidden in every informational system.

For some people, the development of databases was an immense leap in the field of computers. It has contributed to modifying how companies operate the management of information. (Connolly, 2015). But its use goes beyond the business spheres; it has been applied to by any organizations inclined to manage its data systematically.

Thanks to advances in technology such as hardware capacity, communications, the internet itself, etc., managing databases has become much more relevant than in the past. We could say that it is unimaginable thinking about operating any process without having a solid grasp of the underlying data feed.

3.1.2 A Brief Step in History

As to give some perspective about databases development, it is relevant to mention that around 1960, Charles Bachman worked and designed what was known as an Integrated Database System, which is considered the first Database Management System (DBMS).

At the same time, developments in computers processing speed demanded standardization. It was getting more complex to deal with the existence of various database systems that were already in the market.

Charles Bachman's entrepreneurialism took him to found the company Database Task Group. Later, he stepped into the invention of COBOL, the Common Business Oriented Language. This language created by Bachman's company became to be known as CODASYL. This new standardized language was not seen as an easy-to-manage solution as it needed a greater level of expertise to learn it and operate it. Technically, the language used internally a linked data set technique, which created an arrangement that led to extensive networks. Some of the difficulties with operating linked list and large networks paved the way for other approaches characterized for simplicity.

While at IBM, one of its employees, Edgar Codd unsatisfied with CODASYL and its company's IMS system started to work on research to device the creation of better database systems. Around 1970, Codd came up with one of his notable papers, a Relational Model of Data for Large Shared Data Banks.

Against IBM's business strategy and as a result of the work of Michael Stonebraker and Eugene Wong, IBM was in the direction to create the Structured Query Language (SQL) in 1974. (Foote, 2017).

3.1.3 What is a Database?

As defined previously, the database is a data repository used concurrently by many areas inside an organization with the advantage that it is not a separated body; it is in connection with the main advantage of having integrated data with limited duplication.

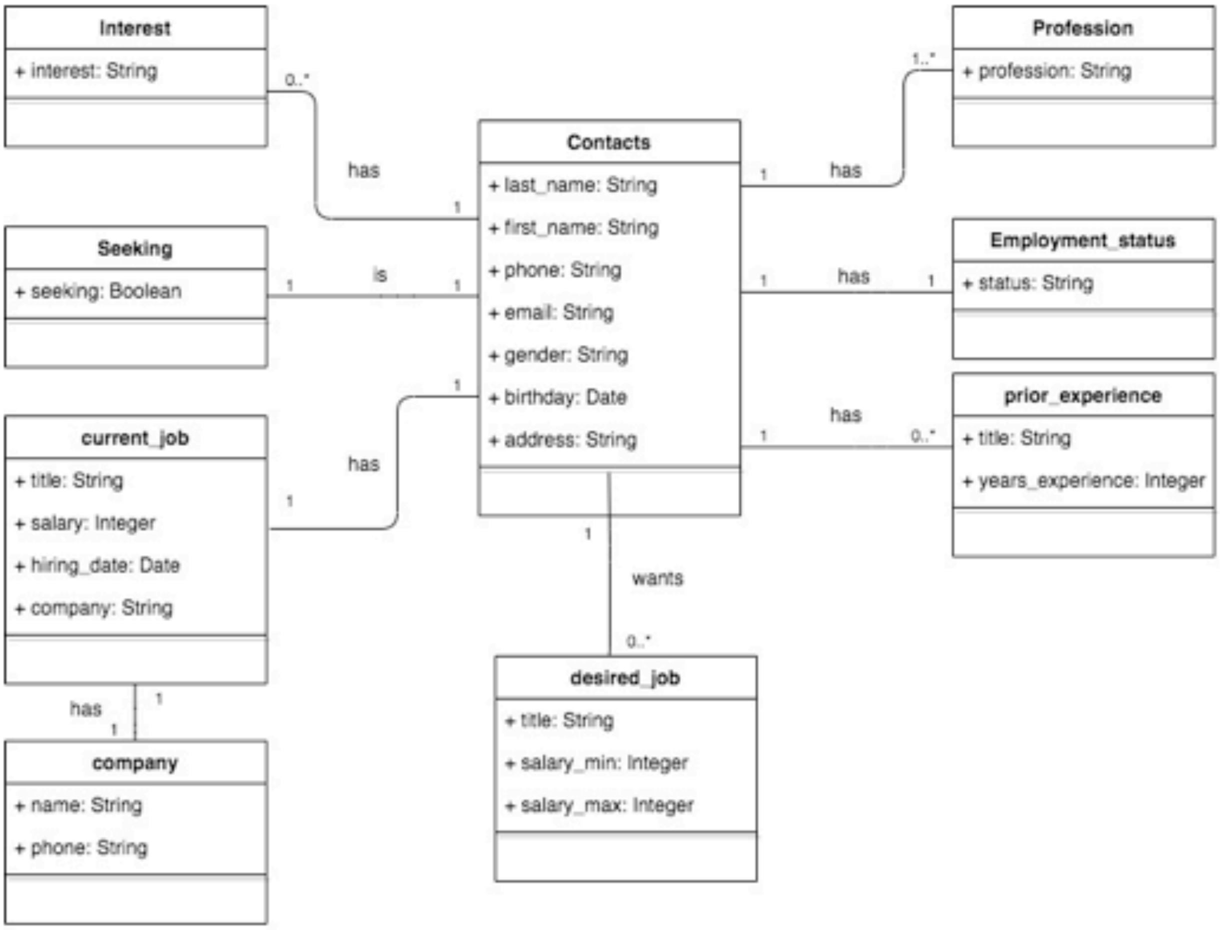
With integration comes another salient aspect that makes databases extremely useful, databases are a shared body and hold specific detail that describes the data, which is also known as a data dictionary.

One of the advantages of using databases is data abstraction; just as in object-oriented programming, it allows separate internal and external definitions. This separation maintains the data structure distant from application programs.

When developing databases, some other concepts deserve an explanation. These concepts are an entity, attribute and relationship.

The entity represents a unique object in the organization. An attribute is a characteristic that describes an object's feature which will be added as one of the records in the database. A relationship is a link among entities. The following figure show these elements.

Figure 1: Objet and Relational Database Example



Source: Miranda, 2020.

3.1.4 What is a Database Management System (DBMS)?

As previously defined, the Database Management System is the software in charge of managing, including the definition, creation, maintaining and controlling access to a database.

More explicitly, the DBMS permits a user to set the database. This action is carried out with the use of Data Definition Language (DDL). This language gives access to setting data types, structures and constraints regarding the data intended to be in storage in the database.

Another important aspect of DBMS is that allows for data manipulation, which is understood as inserting, updating, deleting and retrieving data from the database; this is done with the use of a Data Manipulation Language (DML).

This language takes advantage of the existence of data to perform the above-referred actions, commonly called querying. Querying should be performed by a Structured Query Language (SQL). At the same time, SQL grants a secure system to stop unauthorized users from accessing a database. It also provides an integrity system to ensure consistency regarding the stored data, a concurrency control system to share access to the database, a recovery control system to restore the database after being hit by a failure and a catalogue to save the data descriptions in the database. As a reference, some popular SQL tools include MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, etc.

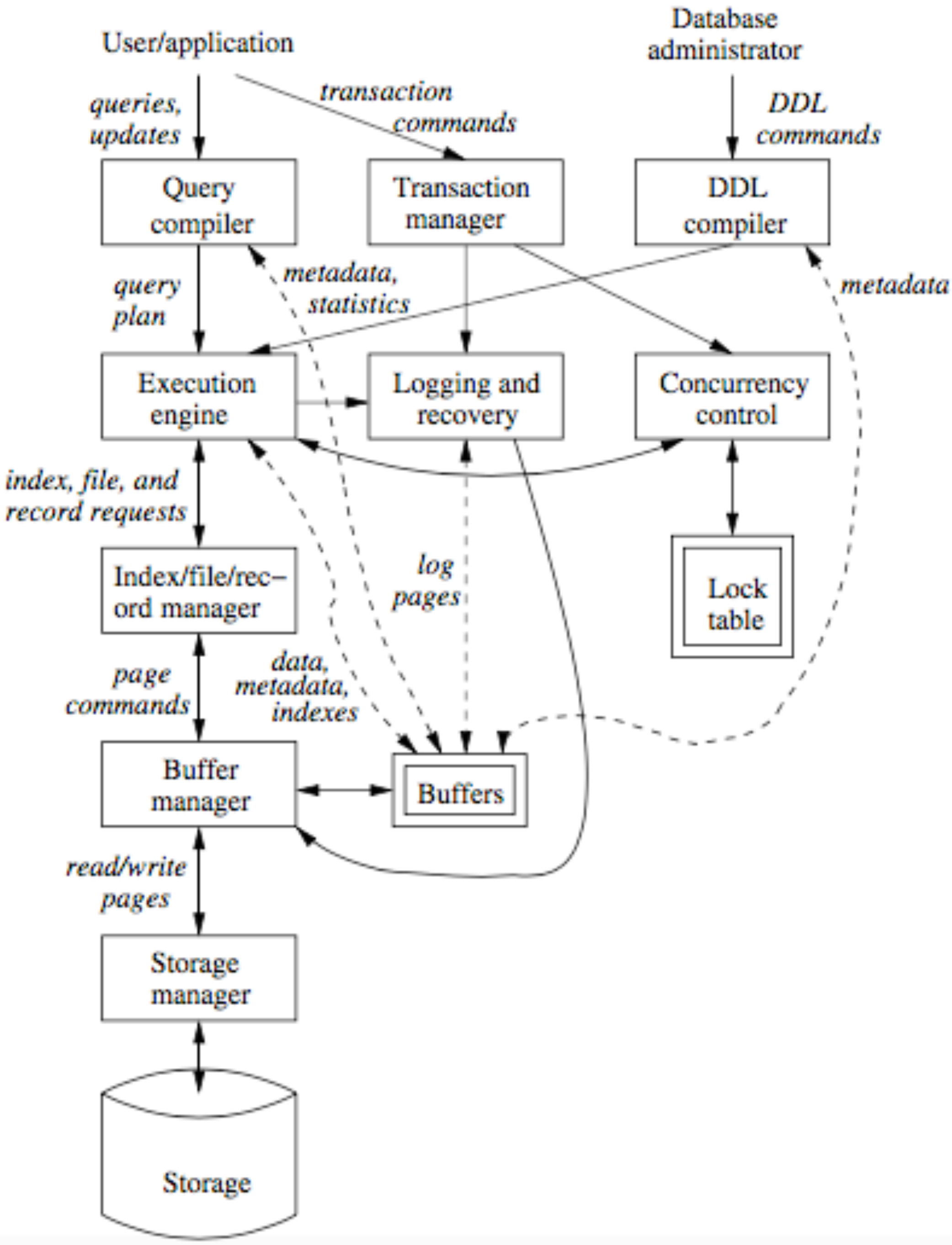
3.1.5 Database Application Programs

Database Application Programs are what users operate to create and maintain a database and generate the desired information. Independently on how the applications are created, these applications serve as the mean for interaction between users and the database.

3.1.6 Database Management System Components

As seen in the chart below, cited from Ullman & Widom, “A First Course in Database Systems”, 2014, page 6. Five components are part of any Database Management System: hardware, software, data, procedure and people.

Figure 2: Database Management System Components



Source: Ullman, 2014.

Depending on the organization's size and database, hardware can range from just a personal computer to a computer network. Additionally, it could be set to serve the backend and frontend portion of the Database Management System.

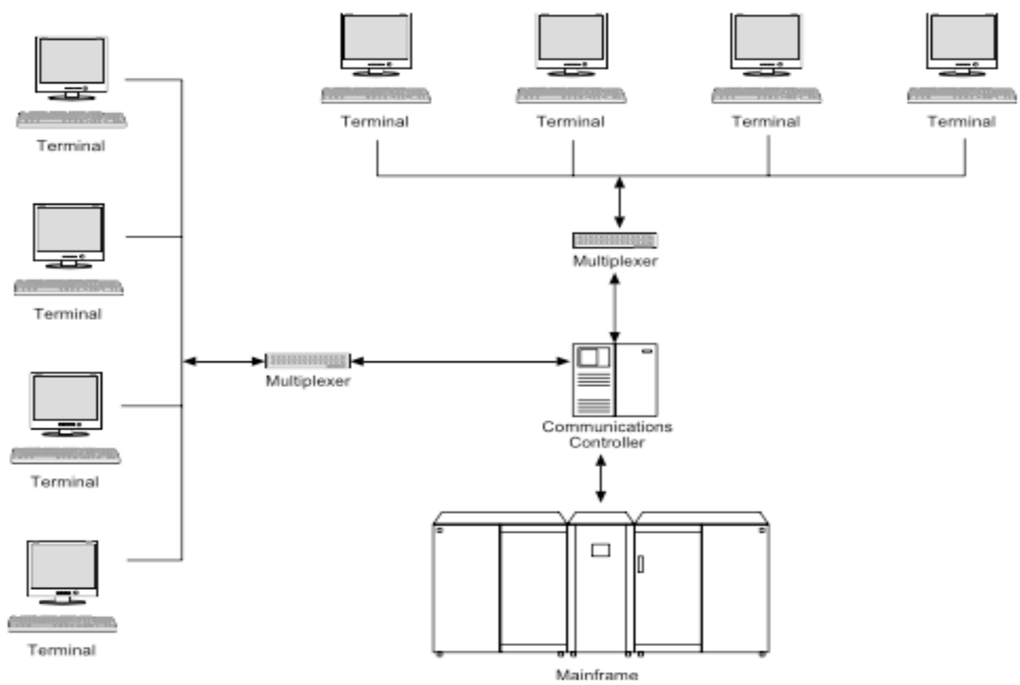
The software covers not only the Database Management System but also the application programs. Some of these programs are written in languages such as C, C++, C#, Java, etc. (Coronel, 2019).

Data, as previously covered, it is the main ingredient of any Database Management System. Without its presence, there is no meaning for any database. As we will see in the practical section, data is organized in tables and each table has attributes that describe the data by adding the data values.

Procedures are the instructions and rules set to govern the design and use of the database. Some procedure examples are log on, use of facilities or programs, etc.

People are everybody who interacts with the Database Management System, more than the user. People also involves data and database administrators, database designers, application developers.

Figure 3: Hardware Representation of Database Architecture



Source: Harrington, 2009.

3.1.7 Data Models

Even before thinking about designing any database, a stop on the concepts related to data is necessary. Practically, everybody working with data ranging from designers, programmers and data users have different perspectives about the same data. This situation unequivocally will lead to a database that differs from the intended database concept.

The prevention of such an unwanted event is accomplished by laying out a good communication plan. The plan needs to have procedures to gather precise requirements, data descriptions, and data uses. All of it can be achieved by using data modelling techniques, which indeed helps to clarify communication among parts by employing abstraction to define entities and relationships. Therefore, data modelling should be the first step into creating a database, this model ends up being just a simple graphical representation of the real world. The graphical representation constitutes a set of data structures and the corresponding characteristics, relations, constraints and transformations.

Coronel & Steven, in "Database Systems: Design, Implementation, and Management", 2019, on page 35, advise that when the data modelling process is correct, the data model will serve as the blueprint to include all the user requirements.

3.1.7.1 Data Attributes, Relationships and Constraints

Attributes are the characteristics innate to a specific entity, while relationships are the vehicle to describe how entities are associated. In this thesis, this could be shown with the following example. A survey participant/company (entity) has a name or an ID (attributes), and this participant has employees (entity) who work, in our case solely, for the company.

The above relationships come in different forms, as is shown below:

In this project thesis, a one-to-many relationship could be presented as the relationship between an employer and its employees. These characteristics for the type of industry used to develop de database, employees usually work under full-time contracts for one employee and do not have a second job.

A many-to-many relationship is present in cases where from one table many of its elements have a corresponding element in a second table and vice versa.

One-to-one relationship, our data collected could show that employees could have only one job function to perform.

Constraints exist in databases to make sure that data is maintaining its integrity. For example, an employee cannot have more than one identifier; he must have a unique identifier.

3.1.7.2 Business Rules

Business rules are set to ensure actions within a company are carried out as planned or how they occur in the real world. These business rules, to be effective, have to be easy to understand and propagated in a way that becomes common knowledge inside the company. As an example, a salary base can be assigned to multiple employees, but one employee can have only one salary as a base salary.

3.1.7.3 Brief Insight on the Evolution of Data Models

The following section will provide brief definitions and a succinct chart on some data models, it also contains a sequential perspective on the evolution of the data models. As a reminder, this thesis will work with a relational database, but it is fundamental to have some insight on these models (Hierarchical, Network, Relational, Entity Relationship, Object-Oriented, Extended Relational and NoSQL).

3.1.7.3.1 Hierarchical Model

Ullman & Widom offer a succinct definition of the hierarchical model in their book “A First Course in Database Systems” 2014, on page 16, they call it “a tree-oriented model” (Ullman, 2014). It mainly operated at the physical level, which became difficult for developers to write code at a high level.

3.1.7.3.2 Network Model

This is a graph-oriented model and just like the hierarchical model operated at the physical-level model. In truth, both the hierarchical model and today's semistructured models allow full graph structures and do not limit to trees. (Ullman, 2014).

3.1.7.3.3 Relational Model

As expressed by E.F. Codd in his book "The Relational Model for Database Management", 1990, on page 6, "It is a collection of relations of various degrees. All queries and manipulations are upon relations, and all of them generate relations as results." (Codd, 1990)

3.1.7.3.4 Entity-Relationship Model

The Entity-Relationship is an approach that goes from top to bottom to design databases, that is the high-level introduction offered by Connolly & Begg, in "Database Systems. A Practical Approach to Design, Implementation and Management", 2015, on page 405. They also indicate that this model begins by identifying entities and relationships between the data that must be part of the model, then adds details regarding the entities and relationships called attributes and any constraints on the entities, relationships, and attributes.

3.1.7.3.5 Object-Oriented Model

Coronel & Steven claim in "Database Systems: Design, Implementation, and Management", 2019, on page 47, that the object-oriented model came to fruition to solve issues of the real-world. The market demanded the existence of a data model that could reflect the real world. Under this model, an object is a structure holding data and relationships. Most programming languages define objects as abstraction used to represent the attributes of a unique real-world entity.

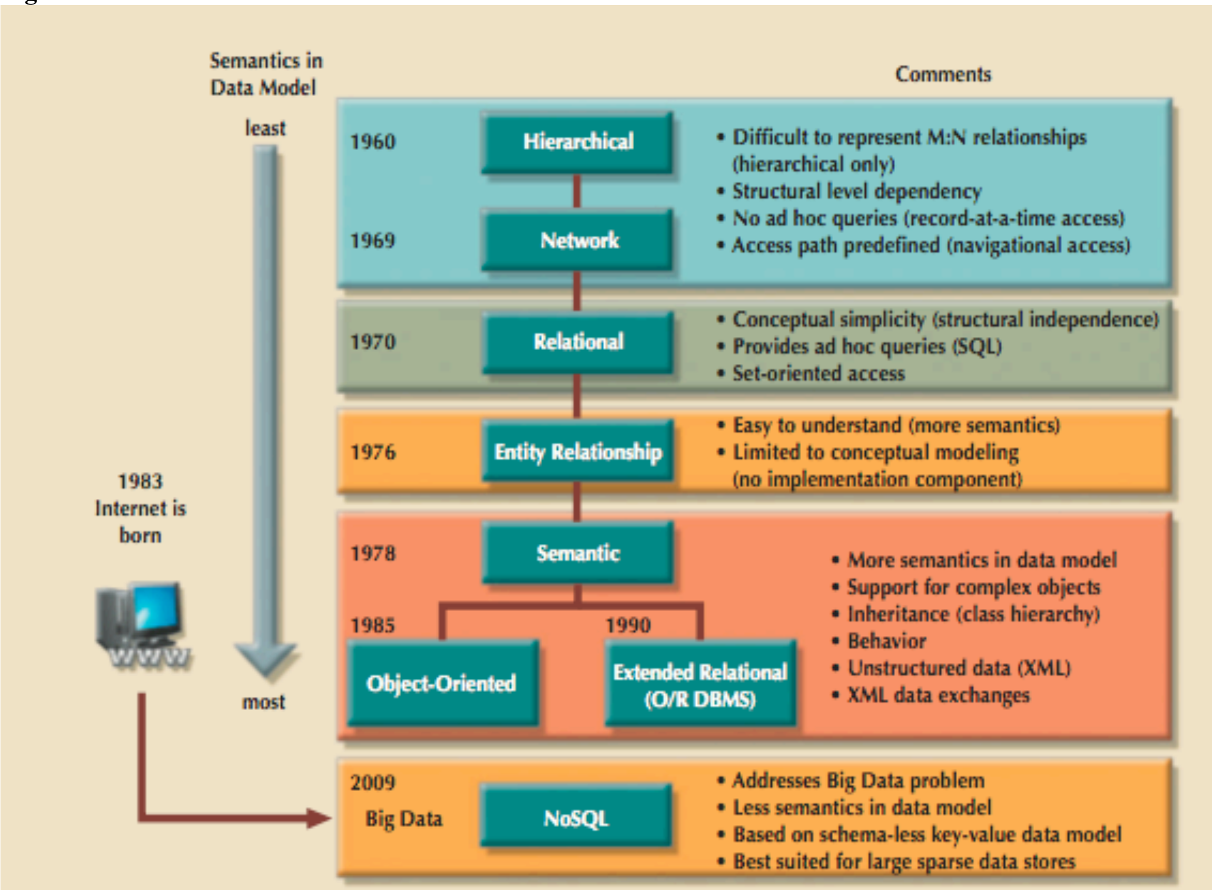
3.1.7.3.6 Extended Relational Model

This model could qualify as an addition to the relational model, it basically takes many of the features from the object-oriented model and adds them to the relational model.

3.1.7.3.7 NoSQL Model

The NoSQL model is the most recent type of database management systems. The main feature is that its structure and operation are not related to the well-known relational database model. On top of the aforementioned, it does an excellent job at managing structured and unstructured data in more efficient ways.

Figure 4: Data Model Evolution



Source: Coronel, 2019.

3.1.7.4 Abstraction: The Conceptual and Internal Model

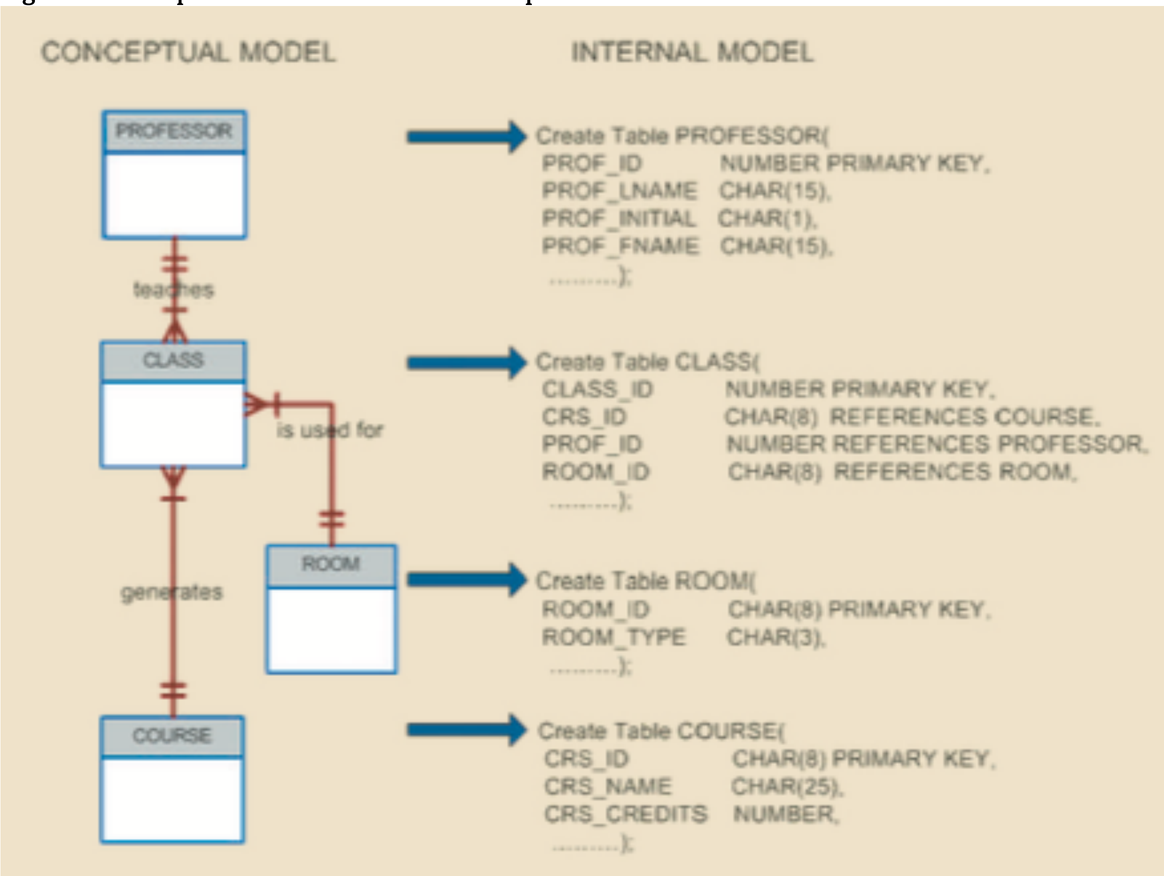
Even though there are more existent models such as the external and physical models, the work in this thesis will rely on the conceptual and internal model because I consider it more applicable for the size and scope of my work.

The conceptual model gives a single snapshot of the entire organization and the database. It assembles all entities, relationships, processes and constraints, as a result, it gives a clear view of all the existent objects.

The internal model goes deeper. It does it by linking the conceptual model into the Database Management System. In other words, it takes every single entity with its characteristics and constraints as transforms it into the database language.

The following figure offers a visual explanation of the two models, the conceptual and the internal models that will be implemented for the compensation benchmarking database.

Figure 5: Conceptual and Internal Model example



Source: Connolly, 2015.

3.1.8 Introduction to Relational Database Model

As this topic is the main theoretical foundation for this project thesis, it was imperative to include Mr. E.F. Codd's own definition when needed. In his book "The Relational Model for Database Management", 1990, on page 6, he defines it as a "collection of relations of various degrees where all queries and manipulations are upon relations and all of them generate relations as results". (Codd, 1990)

Oracle’s website offers its own definition, “A relational database is a type of database intended to store and grant access to data elements that are related to one another. Relational databases have as their foundation the relational model, which is considered as an intuitive, direct mean to represent data in tables

Additionally, a more technical description of the relational models has been offered by citing three main aspects a structural aspect, and integrity aspect and a manipulative aspect. (Date, 2004). The first aspect, structural, mentions that databases are seen as tables only. The second aspect, integrity, refer to tables obeying constraints. Finally, the third aspect is concerned with the possibility to use operators to get out of tables.

One of the keys to the success of relational database models is their practicality. It allowed its creators to pay more attention to aspects related to the data’s logical representation and relationships instead of worrying about storage-related problems.

In its basic concept, a table in a relational database has structural and data independence. It allows the users to think of having independent tables storing related records. Regarding this project thesis, it is more appropriate to work with a model that turns to be simple enough to perform the design, store and query data.

3.1.8.1 Tables

As we keep referring to the term table, it is valuable to provide more information on this term. Ullman & Widom offer a concise definition of a table in his book “A First Course in Database Systems”, 2014, page 14; they call it a “two-dimension table” to represent data. (Ullman, 2014)

However, more than being perceived only as a two-dimensional body composed of rows and columns, each row constitute a single occurrence or record, each column is a distinctive attribute and has a defined range of values, every intersection of a row and column represent a value, every value inserted in a column has to be subject of one unique format, row and column order is not relevant and each one of the tables must have a unique identifier assigned to every row or record.

3.1.8.2 Keys

Alan Beaulieu offers a practical approach in his textbook “Learning SQL”, 2009, on page 5. Alan indicates that tables need some unique identifiers in the form of keys because they guarantee row searchability. Additionally, help to set relationships among tables and to ensure data integrity. Keys are generated by using one or more attributes that comprise other attributes. When a key is not chosen, as a default, database servers have a mechanism to generate unique sets of numbers to use as the primary key within a table.

Relational database models use different types of keys such as composite key, superkey, candidate key, foreign key and secondary key. A composite key is a composition of more than one attribute from a table, a superkey is one selected to identify any of the rows form a table, a candidate key is a superkey without repeated attributes and a secondary key is the one that is used to retrieve data.

3.1.8.3 Integrity Rules

Integrity rules are divided into two, entity and referential integrity. Both help to guarantee a high standard of database design. Entity integrity has the purpose of granting every row a unique identity and ensuring that every foreign key could reference a primary key value. Referential integrity’s goal is better observed when in some instances, attributes may not have a value, but it does not allow to have an invalid value inside an attribute.

3.1.8.4 Relational Algebra

Relational algebra is the key to data manipulation. In the practical part of this thesis, I would demonstrate that even though we are referencing the term relational algebra, there is no need to use any mathematical or algebraic formulas, thanks to the existence of the underlying querying language. Two main aspects of relational algebra are the terms relation and relation variable.

3.1.8.4.1 Relation and Relational Variables

According to C.J. Date's well-known textbook "An Introduction to Database Systems", 2004, a relation is no more than another name for a table. The term was introduced by E.F Codd while working on the relational database principles, and a relational variable is a container holding data.

3.1.8.4.2 Relational Set Operators

Figure 6: Relational Set Operators

OPERATION	NOTATION	FUNCTION
Selection	$\sigma_{predicate}(R)$	Produces a relation that contains only those tuples of R that satisfy the specified *predicate*.
Projection	$\Pi_{a_1, \ldots, a_n}(R)$	Produces a relation that contains a vertical subset of R, extracting the values of specified attributes and eliminating duplicates.
Union	$R \cup S$	Produces a relation that contains all the tuples of R, or S, or both R and S, duplicate tuples being eliminated. R and S must be union-compatible.
Set difference	$R - S$	Produces a relation that contains all the tuples in R that are not in S. R and S must be union-compatible.
Intersection	$R \cap S$	Produces a relation that contains all the tuples in both R and S. R and S must be union-compatible.
Cartesian product	$R \times S$	Produces a relation that is the concatenation of every tuple of relation R with every tuple of relation S.
Theta join	$R \bowtie_F S$	Produces a relation that contains tuples satisfying the predicate *F* from the Cartesian product of R and S.
Equijoin	$R \bowtie_F S$	Produces a relation that contains tuples satisfying the predicate *F* (which contains only equality comparisons) from the Cartesian product of R and S.
Natural join	$R \bowtie S$	An Equijoin of the two relations R and S over all common attributes *x*. One occurrence of each common attribute is eliminated.
(Left) Outer join	$R ⟕ S$	A join in which tuples from R that do not have matching values in the common attributes of S are also included in the result relation.
Semijoin	$R \triangleright_F S$	Produces a relation that contains the tuples of R that participate in the join of R with S satisfying the predicate *F*.
Division	$R \div S$	Produces a relation that consists of the set of tuples from R defined over the attributes C that match the combination of **every** tuple in S, where C is the set of attributes that are in R but not in S.
Aggregate	$_{AL}(R)$	Applies the aggregate function list, AL, to the relation R to define a relation over the aggregate list. AL contains one or more (<aggregate_function>, <attribute>) pairs.
Grouping	$_{GA}\ _{AL}(R)$	Groups the tuples of relation R by the grouping attributes, GA, and then applies the aggregate function list AL to define a new relation. AL contains one or more (<aggregate_function>, <attribute>) pairs. The resulting relation contains the grouping attributes, GA, along with the results of each of the aggregate functions.

Source: Connolly Thomas, 2015.

The following is one of the recommendations Connolly & Begg present in their publication “Database Systems. A Practical Approach to Design, Implementation and Management”, the use of the operators has a main result, the creation of other relations or tables. Some of the operators are explained concisely in the prior chart.

3.1.8.5 Data Dictionary and System Catalog

A data dictionary is a metadata container. It behaves like a table, it can be queried via a select statement to view the current data structures in the database; the content has all the attribute's names and characteristics for every table in the database. (Beaulieu, 2009).

Just like the data dictionary, the system catalogue is a system data dictionary able to describe the database objects. As an example, within the documentation it produces, it is possible to find the data related to table names and other aspects related to its creation.

3.1.8.6 Data Redundancy

One of the benefits of using a relational database is controlling data redundancy. This control is achieved with the use of foreign keys. One situation to keep in mind is that data redundancy is not eliminated. In some cases, foreign key values could be duplicated several times.

3.1.8.7 Indexes

As explained by Alan Beaulieu in “Learning SQL”, 2009, on page 1, just like an individual would use an index in a book to find some information, databases use indexes to reach some rows in a relational table. More precisely, indexes are particular tables that maintain order with a clear goal of allowing for efficient searches by keeping certain columns to locate rows in a specific table.

Conceptually, an index consists of an index key and a set of pointers having each key pointing at the precise location of the data that has been identified by the key. As a caveat, a table could have as many indexes as possible, but every single index can be associated with one table only.

3.1.8.8 Relational Database Rules

In 1985, Dr. E. F. Codd published a set of twelve rules to define a relational database system, with the caveat that these rules may not be supported by databases. Coronel & Steven offer a description of Codd's rules in their book "Database Systems Design, Implementation and Management", 2019 and the following is my interpretation of it.

1st Rule: Information
All data have to be presented as column values in each row within a table.

2nd Rule: Guaranteed access
All data should be accessible without any ambiguity. This can be obtained with the use of a combination of the table name, primary key, and column name.

3rd Rule: Systematic treatment of nulls
Any field in a table should be allowed to stay empty, this fact involves the support of a null value.

4th Rule: Dynamic online catalogue based on the relational model
A relational database must provide access to its structure by employing the same tools that are used to access the data.

5th Rule: Comprehensive data sublanguage
The relational database should be set to support a variety of languages. But, it will have to support a declarative language, data and view definition, data manipulation, integrity constraints and database transaction control.

6th Rule: View updating

Data can be presented to the user as views and each view should support the same functionality available to any table.

7th Rule: High-level insert, update and delete

Insert, update and delete operations should be allowed for any set.

8th Rule: Physical data independence

User's access should not be affected by any changes made to the hardware or storage methods.

9th Rule: Logical data independence

The applications used by the user to view data should not be affected by physical access methods or storage structure changes.

10th Rule: Integrity independence

A database language should allow constraints on any input to guarantee database integrity.

11th Rule: Distribution independence

End users and application programs should not be affected by the location whether or not the database is distributed or locally based.

12th Rule: Nonsubversion rule

Users should not be able to modify the database structure. In other words, there should not be a chance to ignore database integrity rules.

3.1.9 Database Normalization

An explanation provided by Elmasri & Navathe, in "Database Systems", 2011, page 18, has helped me to conclude that the purpose of database normalization focuses on controlling or reducing data redundancies to avoid unwelcomed data anomalies in processes such as insertion, deletion and updating. From a more simplistic perspective, normalization could be seen as a filtering process to achieve improved database design.

The normalization process is accomplished in a set of stages. These stages are the first normal form (1NF), second normal form (2NF) and third normal form (3NF). Practically, these forms could be what is really needed for business database design. However, for theoretical purposes, a brief definition of the Boyce-Codd Normal Form (BCNF) and fourth normal form (4NF) will be included. It is important to bear in mind that reaching the last stages does not mean reaching an optimal design, the result could be more complexity in the relation interactions.

The other side of the coin when it comes to normalization is the use of the entity-relationship model, this model assists especially when designing new databases that need to incorporate the user's requirements.

3.1.9.1 Conversion to First Normal Form (1NF)

The first normal form is a process where repeating groups are discarded. This process presents all the data in a tabular format, where each cell has a single value, there are no repeating groups, the primary key has been identified as well as all the corresponding dependencies.

3.1.9.2 Conversion to Second Normal Form (2NF)

The conversion to the second normal initiates after the first normal form has been finalized and a composite primary key has been assigned. The second normal form creates new tables to eliminate partial dependencies, reassigns the corresponding dependent attributes and then place them in a new table.

3.1.9.3 Conversion to Third Normal Form (3NF)

In order to continue with the third normal form, it is necessary to have reached the second normal form. The step to get to the third normal form can be reduced to making new tables to eliminate transitive dependencies for every transitive dependency and reassigning the corresponding dependent attributes.

3.1.9.4 The Boyce-Codd Normal Form

The Boyce-Codd normal form (BCNF) is reached when every determinant in the table is identified as a candidate key, but a table does not have more than one candidate key. In this situation where there is only one candidate key, there is a point of equivalence between the third normal form and the Boyce-Codd normal form

3.1.9.5 Fourth Normal Form (4NF)

The fourth normal form has a goal the elimination of all the problems caused by multivariate dependency and it can be done when the component of the multivalued dependency is assigned to new tables.

3.1.10 Structured Query Language (SQL)

To manage all the work related to the creation, storage, manipulation and reporting of the data included in a database, SQL is the right tool to perform all these activities. One of the reason is that the work of relational databases is based on tables and SQL is a tool specialized in producing tables as a result of the querying exercises.

This language even though is simple can be too extensive to cover, for that reason, only the main tasks related to schema statements, data statements, and transaction statements will be outlined in the following sections under different areas.

But before getting into more technicalities, it is fundamental to speak about data. SQL handles different types of data.

3.1.10.1 Data Types

The types of data allowed in SQL are characters, text, numeric and temporal data, all of them with some capacity ranges. Characters/varchar can store single and multibyte with a capacity of 64KB. Text could store up to 4.2 million bytes in its different forms (tinytext, text, mediumtext and longtext). Numeric data depending on the classification (tinyint, smallint, mediumint, int, bigint) could store from 2^7 to 2^63, both positive and negatives. Within the numeric types, decimal numbers are also allowed. At last, temporal data is stored under the required date and time formats.

3.1.10.2 Data Manipulation Language (DML)

DML incorporates commands to insert, update, delete, and retrieve data within the database tables.

SELECT is a clause that chooses columns from a table.

FROM gives direction to select in reference to choosing columns from a specific table.

WHERE allows filtering the selected information.

Additional Query Capabilities:

ORDER BY, this clause sorts the resulting data.

JOIN (INNER, LEFT, RIGHT), in many cases it will be necessary to the normalized data and these are the command groups that will permit to extract data from various tables in the same query.

SUBQUERIES, these are queries that have been introduced as part of one query to obtain some specific data that is required by the main query to proceed.

GROUP BY, this clause helps to gather all the related data and presented in block sections giving the results a view that is organized by a certain selected attribute.

AGGREGATE functions (COUNT, SUM, AVG, MIN, MAX), these functions are characterized for returning just specific values. For example, the average height of a group of people is just one number.

3.1.10.3 Data Definition Language (DDL)

DDL provides commands to create tables, indexes, views and to grant access rights such as:

CREATE TABLE — creates a new table.

CREATE VIEW— creates a view table.

DROP TABLE— removes a table.

DROP VIEW — removes a view table.

ALTER TABLE — alters the structure of a table.

3.1.10.4 Transaction Control Language (TCL)

TCL has commands related to the processing of transactions such as:

COMMIT Statement -- commit all changes for the ongoing transaction.

ROLLBACK Statement -- roll back all changes for the ongoing transaction.

3.1.10.5 Data control language (DCL)

The data control language (DCL) employed to control access to data objects. For example, giving permission to only view a certain table, and giving another user access to make changes to the same table.

GRANT —grants access privileges to users.

REVOKE — revokes granted access privileges.

3.1.10.6 Schema Statements

Schema statements are used to define the data structures as part of a database. The keywords are as follows:

3.1.10.7 Data Statements

These are the statements that allow the manipulation of the data structures through the use of queries.

3.1.10.8 Transaction Statements

Even though these statements will not be used in this project thesis, it is a basic part of SQL to leave it ignored. Transaction statements are used to begin, end and rollback transactions.

3.2 Theoretical Background: Compensation

3.2.1 Introduction to Compensation

Compensation is one of the elements companies, most specifically Human Resources departments, use to motivate employee's behavior. The expectation is that compensation measures will translate into the achievement of companies' business goals.

As George Milkovich et. al hints between lines all over his book on compensation "Compensation", 2014, if compensation is a well-managed tool, it can certainly play a positive role in companies successfully executing their strategies through their employees. (Milkovich, 2014)

3.2.2 Compensation Definition

Compensation is understood as the total cash and non-cash payments employees receive in exchange for the work performed in a company. Depending on the type of company, salary compensation could be one of the most representative expenses, more than just a basic salary, it could include other wages and benefits. In more detail, total compensation may incorporate base pay, commissions, overtime, bonus, merit pay, insurance, vacation, retirement, stock options and non-cash benefits. (Kappel, 2018).

3.2.3 Compensation Perspectives

It is important, as in any communication exchange, to define what compensation means for the various incumbents to correctly address any actions that may concern to each of them.

3.2.3.1 Society

In our society, salary compensation has been seen as a way to measure justice. Blau & Khan's reported many of these issues in "The Gender Pay Gap: Have Women Gone as Far as They Can?", it is common to hear nowadays about pay differences between genders, races, ages, and the efforts of lawmakers to diminish any of these pay differences.

Additional to pay, other benefits such as health insurance can be considered as a privilege in some nations. While in other places of the world, it is a right that is not part of any compensation negotiation. (The Henry J. Kaiser Foundation, 2012).

Across nations, compensation is a fundamental part when it is examined as a cause for job losses. For example, members of society worry about the tradeoffs of pay differential among countries, particularly in areas or industries where some technical skills or means of production are not replicated with ease. The following chart shows a quantitative difference for some countries around the globe.

Table 1: Hourly Compensation Costs

	Hourly Compensation Costs	
	in U.S. dollars 1997 (1)	in U.S. dollars 2012
Norway	25.83	63.36
Australia	18.93	47.68
Germany	29.16	45.79
United States	23.04	35.67
Japan	21.99	35.34
Italy	19.76	34.18
United Kingdom	19.31	31.23
Spain	13.95	26.83
Greece	11.61	19.41
Argentina	7.55	18.87
Portugal	6.46	12.1
Czech Republic	3.25	11.95
Slovakia	2.84	11.3
Brazil	7.07	11.2
Taiwan	7.04	9.46
Hungary	3.05	8.95
Poland	3.15	8.25
Mexico	3.47	6.36
Philippines	1.28	2.1

Source: U.S. Bureau of Labor Statistics, International Labor Comparisons, August 2013.

Within the same society, some segments believe that compensation increases lead, in some way, to the increase of inflation, while others under the argument of social justice fight for the right to a fair pay increase.

3.2.3.2 Stockholders

As owners of companies mainly interested in profiting from their business success, they are also interested in the amount of money and the types of compensation employees are getting paid. The amount of payment is determined by internal and external comparison analysis, which up to an extent is more of an internal exercise performed by Human Resources staff.

How employees, particularly top executives, receive their salary compensation is a matter of concern for the stockholder. The explanation is because top executives are responsible for leading companies, but also because salaries are a point of discussions in the media and society in general. In the case of top executives, they usually receive stock options, which could be traded in the stock market and enlarge their compensation benefits.

Table 2: Highest-paid C.E.O.s

Highest-paid C.E.O.s				
Name	**Company**	**Compensation**	**Revenue**	**Rank**
Elon Musk	Tesla	$2.3 billions	$21 billions	21
David Zaslav	Discovery	$129 millions	$11 billions	11
Nikesh Arora	Palo Alto Networks	$125 millions	$2 billions	187
Safra Catz	Oracle	$108 millions	$40 billions	50
Mark Hurd	Oracle	$108 millions	$40 billions	50
John Legere	T-Mobile	$67 millions	$43 billions	45

Source: Eavis, Peter, New York Times, 2019.

An important aspect to keep present, stockholders have the power to decide on the executive's compensation. They do it by using shareholder proposals, vote and clawback provisions to ensure executives act in the best interest of the company.

3.2.3.3 Managers

For managers, compensation is an element they manage the cost structure in order to have a direct impact on the bottom line. They also use it as a mean to guide an intended behavior with the intention of improving employee's and company's performance.

Under normal conditions, the compensation expense varies from industry to industry. For industries such as Technology or Finance where employee's skills are specialized, this expense line could represent an expense bordering 50% of the cost structure, while in other industries where employee's level of specialization is lower, this percentage could drop between 15% to 20%.

Even if compensation varies from industry to industry, nowadays, there is a modern management attitude to see salary compensation beyond the type of expenses that needs to be minimized.

There is an inclination to believe that employees will be more likely to stick with the company and perform at their best because the reward is tangible in the form of better pay. This thought is communicated in "Pay Matters. The Art and Science of Employee Compensation.", 2020, page 234, written by David Weaver, "The best thing you can do for your employees is to link their pay to their performance". (Weaver, 2020)

3.2.3.4 Employees

To maximize their opportunities in life, employees invest in education and dedicate their time and energy to the companies where they work. In exchange, they expect and receive an agreed amount of pay in the form of compensation.

As employees dedicate most of their time to work for a company, their salary becomes the biggest or the sole source of income that provides them with the needed financial security. It would be fair to say that employees see compensation as the return or a form of reward for their efforts and achievements. (Lawler, 1971)

3.2.4 Compensation Forms

For the effects of this thesis, when I refer to compensation, I will be focusing on cash incentives such as base salary, bonus, and total compensation. However, for informational purposes, it would be appropriate to name other types of benefits that are used as part of their compensation package.

Table 3: Total Return Elements

Total Returns		
Total Compensation		Relational Returns
Cash Compensation	*Benefits*	
Base Salary	Income Protection	Recognition
Merit Cost of Living	Work/Life Balance	Employment Security
Short-Term Incentives	Allowances	Challenging Work
Long-Term Incentives		Learning Opportunities

Source: Milkovich, 2014.

3.2.4.1 Base Salary

Base salary is the cash compensation received in cash for the work an employee performs. It is intended to represent the value of the work/skills and ignores differences that are related to individual employees.

3.2.4.2 Merit Cost of Living

Merit cost of living represents adjustments made to base salary in different periods. These adjustments are usually made on a comparison basis to what other employers are currently paying for the same type of jobs.

3.2.4.3 Incentives

This compensation scheme links increases in pay to performance. The main differences with merit adjustments are that incentives do not increase the base wage neither the size of the incentive payment opportunity is known beforehand.

3.2.4.4 Long-Term Incentives

Long-term incentives are designed to focus employee actions on several year results. Typically, these incentives are present as a form of stock ownership or options to buy stocks at a predetermined price, which leads to a financial gain as long as the stock price goes up.

3.2.4.5 Income Protection

In some countries, income protection programs are legally required; These programs mandate employers to pay into a fund that provides income replacement for workers who fall under disability or unemployment.

3.2.4.6 Work-Life Balance

These programs are set to help employees integrate their work & life duties. Some programs include time away from work, services to meet specific needs, and flexible work conditions.

3.2.4.7 Allowances

Allowances are a type of compensation designed to provide for goods or services that are scarce in the place of residence or work.

3.2.4.8 Relational Returns

This type of compensation-related factors falls on the qualitative side and they could include recognition, employment security and opportunities to learn, etc. Additional forms of relational return may incorporate satisfaction from new challenges, teaming, etc.

3.2.5 Pay Models

One of the options companies use to design a pay model is through the use of salary surveys to determine how competitive the current pay is benchmarking against competitors. When companies use a benchmarking tool such as a survey, they will decide where to position themselves. The options are to pay above the market, at the market, or under the market; all of them have some benefits and consequences. Therefore, the right strategy could result in a mix of the three positioning points to attract, retain, motivate or even let go, candidates or employees.

3.2.6 Survey Design

After reviewing compensation terminology at a high level and considering that the focus of this thesis is to create a compensation benchmarking database, it is appropriate to explore how a survey is designed. The survey is the tool that will feed the database with the required data to produce the benchmarking information.

There are four main aspects to consider when designing a survey.

- Who will be in charge of managing the survey?
- How many companies should participate in the survey?
- What job functions should be benchmarked?
- What is the information we should collect for benchmarking?

The following chart is a representation of one of the types of surveys found in the market. This survey covers some functional areas (Accounting, Controlling, Financial Management, Audit and Tax), only specific jobs and monthly salary.

Figure 7: Hays Survey Sample – Base Salary Only

SALARY GUIDE
ACCOUNTANCY & FINANCE

Basic btto monthly salaries in CZK for full time roles within ACCOUNTANCY & FINANCE sector:

ACCOUNTING	MIN	MAX	TYPICAL
Invoicing	30 000	35 000	33 000
Junior Accountant	35 000	40 000	37 000
Financial Accountat	40 000	50 000	45 000
Independent / Senior Accountant	50 000	60 000	55 000
Chief Accountant	60 000	80 000	70 000
Junior Payroll Accountant	30 000	40 000	35 000
Payroll Accountant	40 000	60 000	50 000

CONTROLLING	MIN	MAX	TYPICAL
Junior Controller	40 000	50 000	45 000
Senior Controller	60 000	80 000	70 000
Financial Analyst (mid senior)	55 000	70 000	65 000
Business Analyst (mid senior)	50 000	70 000	60 000

FINANCIAL MANAGEMENT	MIN	MAX	TYPICAL
Finance Manager junior	80 000	110 000	90 000
Finance Manager senior	100 000	150 000	120 000
Finance Director	150 000	250 000	180 000

AUDIT	MIN	MAX	TYPICAL
Internal Auditor	60 000	100 000	80 000
Internal Audit Manager	100 000	150 000	120 000
External Auditor / 0-2 years exp.	38 000	55 000	45 000
External Auditor / 3 years exp.	50 000	70 000	60 000
External Auditor / 4+ years exp.	60 000	120 000	80 000
External Audit Manager	80 000	150 000	110 000

TAX	MIN	MAX	TYPICAL
Tax Assistant	35 000	45 000	40 000
Tax Consultant / 3 years exp.	60 000	80 000	70 000
Tax Manager	90 000	150 000	130 000

Source: Hays Salary Guide 2019 - Czech Labor Market Trends.

3.2.6.1 Survey Management

There are two ways on how to organize a compensation benchmarking survey. One is to have a leading or a group of leading companies sponsor a compensation survey, the other option is to bring participants into a compensation survey performed by a consulting company. In any of these cases, each participating company should delegate a compensation manager or an assistant to represent the company and serve as a focal point to supply consultants with compensation data and answer any inquires.

The best solution to guarantee the survey will be objective is to choose a third-party consultant as a mean to protect data and to ensure neutrality and objectivity. A third-party consultant is an organization in charge of creating the compensation survey, maintaining the compensation benchmarking database and sharing with all participants the survey results. The list of market consultants is extensive, it could include company names as Hays, Mercer and Towers.

3.2.6.2 Survey Participants

As in the prior section, selecting and inviting participants will depend on who is in charge of organizing the survey. If the survey is organized by a company or a group of companies, they will offer guidelines about who should be on the list to participate. The main reason is to gather data from companies considered as a company within the industry, direct competitors, or possible competitors. All of these considerations help to perform a more useful benchmarking.

On the other side, if the survey is organized by a third-party consultant as part of the normal business activities, the consultant will focus on getting a large number of participants within an industry. The consultant's idea is to attract relevant competitors to generate a general survey and also to have enough data to create some specialized on-demand surveys for interested clients.

3.2.6.3 Surveyed Jobs

Ultimately a survey is one more business project which has scope, time, and budget constraints. It is obvious to see that the number of jobs maintains some correlation with these constraints. Surveys could cover an entire company structure staring from top to low ranked employees or cover some specialized or targeted jobs. The last word will depend on the organizers and market interest.

3.2.6.4 Survey Data

The data to be collected, on the quantitative side, ranges from base salary, bonus, commissions, allowances, total compensation (TComp) and any relevant compensation category. Additional to these quantitative components, surveys can also collect qualitative data related to Human Resources policies. These are the type of policies that have a direct impact on employees.
As explained with the prior considerations for survey design, all of them depend on the survey organizers, participants and market interests.

3.3 Theoretical Background: Statistics

Keeping in mind this thesis employs some basic statistical concepts related to measures of central tendency and variation, I should define Statistics. Levin et. al defines in “Statistics For Managers Using Microsoft Excel”, 2008, page 2, “Statistics is the branch of mathematics that transforms data into useful information for decision-makers”. (Levine, 2008)

Measures of Central Tendency (MCT) definitions will be introduced in the following sections, I will bring up terms such as mean, median, mode and quartiles along with Variation definitions such as range and interquartile range. These measures could be used to model the data entered in the compensation benchmarking database.

3.3.1 Measures of Central Tendency

Most of the data arrays present different tendency around a central point, this is known as measures of central tendency and they include the mean, median, mode and quartiles.

3.3.1.1 The Mean

As defined by Willard "Statistical Methods", 2020, page 51, the mean (M) is "the sum total of all of the scores in a distribution divided by the total number of scores." (Willard, 2020)

M = SUM(X) / n
X = Each element of the array.
N = Existent element count in the array.

3.3.1.2 The Median

The second measure of central tendency under consideration is the median (Mdn), which is the middle point in a distribution. (Willard, 2020).

The median calculation depends on the number of elements in the array. If the number of elements is odd, the array has to be organized in descending order to then apply the next formula:

Mdn = (N + 1) / 2
N = Existent element count in the array.

The median calculation for an array of even numbers has a different set of steps. First, after placing numbers in ascending order, the array has to be divided into two sets, then the middle terms have to be added and divided by two:

Mdn = (Mid1 + Mid2) / 2
Mid1 and Mid2 = Middle number for the split array.

3.3.1.3 The Mode

The mode (MO) is the number with the highest count in an array. To determine the mode, arrange the values in descending order and identify the numbers repeating the most.

3.3.1.4 Quartiles

Quartiles is the method used to divide an array or distribution into equal parts. (Hahs-Vaughn Debbie L., 2020). Quartiles' function is to divide an array into four equal parts. The first quartile, Q1, divides the smallest 25.0% of the values from the other 75.0% that are larger. The second quartile, Q2, is the median, 50.0% of the values are smaller than the median and 50.0% are larger. The third quartile, Q3, separates the smallest 75.0% of the values from the largest 25.0%. (Levine, 2008).

3.3.2 Variation

Variation is also known as dispersion, this measure can be represented in multiple forms, but for the effects of this thesis, only the range could be considered for any calculations.

3.3.2.1 The Range

The range is calculated with a simple difference; the following formula is enough to describe its calculation.

Range = X largest – X smallest

3.3.2.2 The Interquartile Range

The interquartile range (also called midspread) is the difference between the third and first quartiles (Q) in a set of data.

Interquartile Range = Q3 – Q1

Q3 = Third quartile

Q1 = First quartile

3.3.3 Extreme Values

To analyze data sets, it is necessary to standardize the data, this task will allow for analysis that includes relevant data. The excluded data is what is commonly known as outliers. These outliers could be detected and eliminated by simple observation, particularly when they are obvious, in other cases when simple observation may be difficult z-cores can be employed to reject data point beyond plus or minus three z-scores.

$Z = (X - M) / S$

Z = Z-score

M = Mean

S = Standard deviation

4 Practical Part

4.1 Background

As I stated from the start, my goal is to employ the acquired knowledge in Relational databases to develop a Compensation Benchmarking Database to produce salary, bonus and total compensation for companies in the Financial Services industry. More precisely, I will focus on the United States Corporate Banking segment.

There are multiple companies in the market currently providing compensation consulting services to financial institutions interested in obtaining salary compensation data. These companies use the data to benchmark their salary compensation against the salary compensation offered by a selected group of financial institutions considered as competitors.

4.1.1 Business Scenarios

The business relationship between the consulting company and the financial institutions occur under to business settings.

The first business setting and the most common occurs when the consulting company, as part of its business operations, starts its annual compensation benchmarking survey and contacts financial institutions as participants only. As a result, the survey has two main actors, the consulting company and the survey participants.

The second setting occurs when financial institutions are interested in sponsoring a compensation benchmarking survey and hires the consulting company to lead a compensation survey. Under this scheme, the survey has three main actors, the consulting company, the sponsors and the participants.

The consulting company's role is to ensure that it has an adequate database system for gathering, processing (cleaning, standardizing, flagging), storing, querying and publishing.

The following sections will focus on three main areas. The first two areas will cover the business engagement and the process used to prepare the data for the database. Meanwhile, the third area will cover the database developments activities, excluding anything related to publishing results in any reporting format used in the market to provide survey results to participants or sponsors.

4.2 The Chosen Path: Business Scenario

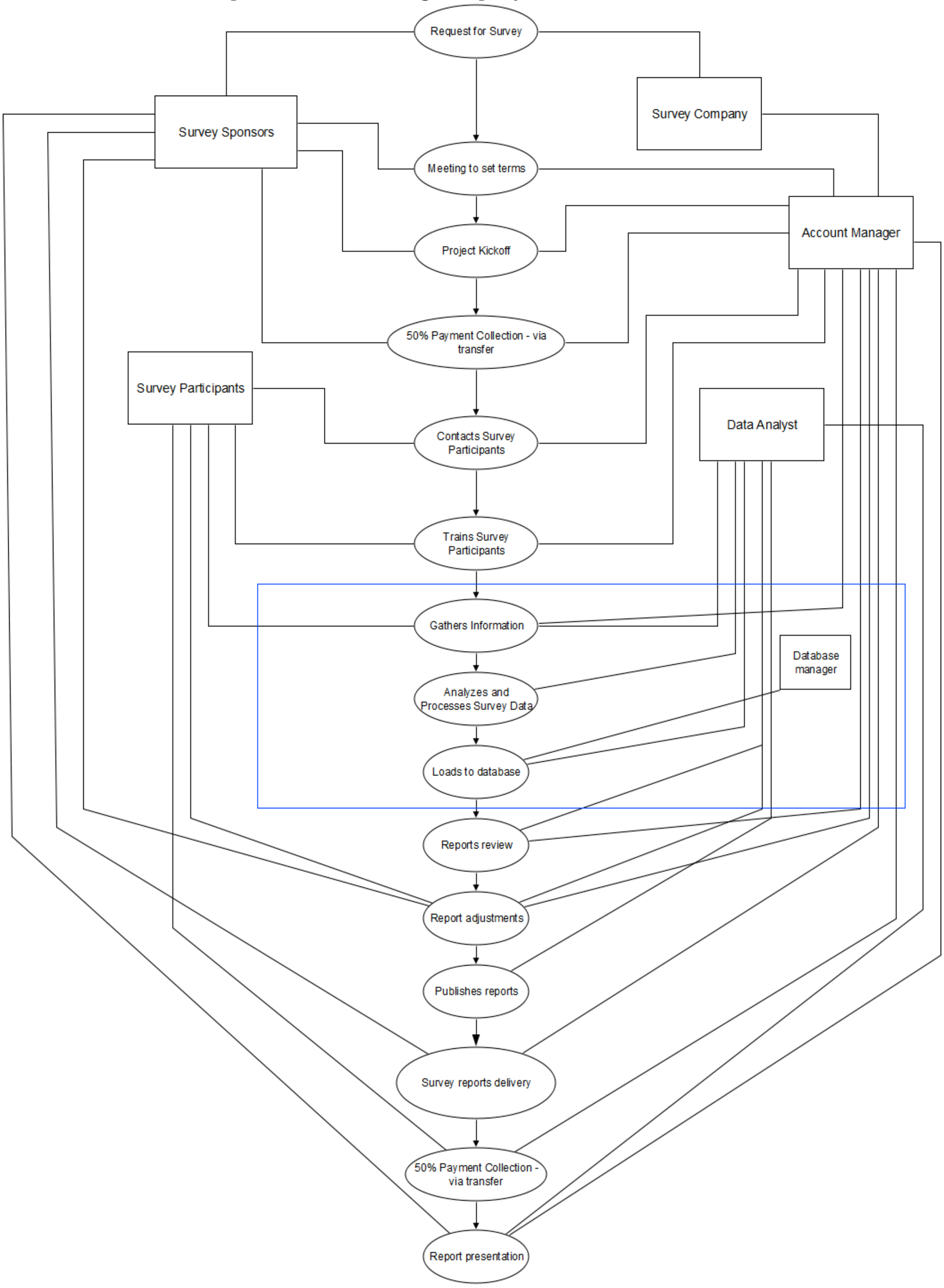
The scenario corresponds to the second business setting described in the background section. This setting arises when financial institutions are interested in sponsoring a compensation benchmarking survey and hires the consulting company to lead a compensation survey. Under this scheme, the survey has three main actors, the consulting company, the sponsors and the participants.

The business scenario depicted in the chart seems to be performed during a short period, but in reality, the whole process lasts between six to eight months. The information-gathering process takes the longest because data providers (sponsors and participants) have to work with their internal databases to pull data and perform sanity checks. The data section will cover more details about this process.

Three companies integrate the sponsor group; this group will pay the consulting company to lead the survey, and a group of participants supplying their employee's compensation data.

To offer more details at a high level about this business setting, I will add the following chart from an information engineering perspective. Here, we can see all the interactions and processes which are part of developing a compensation salary benchmarking survey.

Figure 8: Interview with a Compensation Consulting Company



Source: Miranda, 2021.

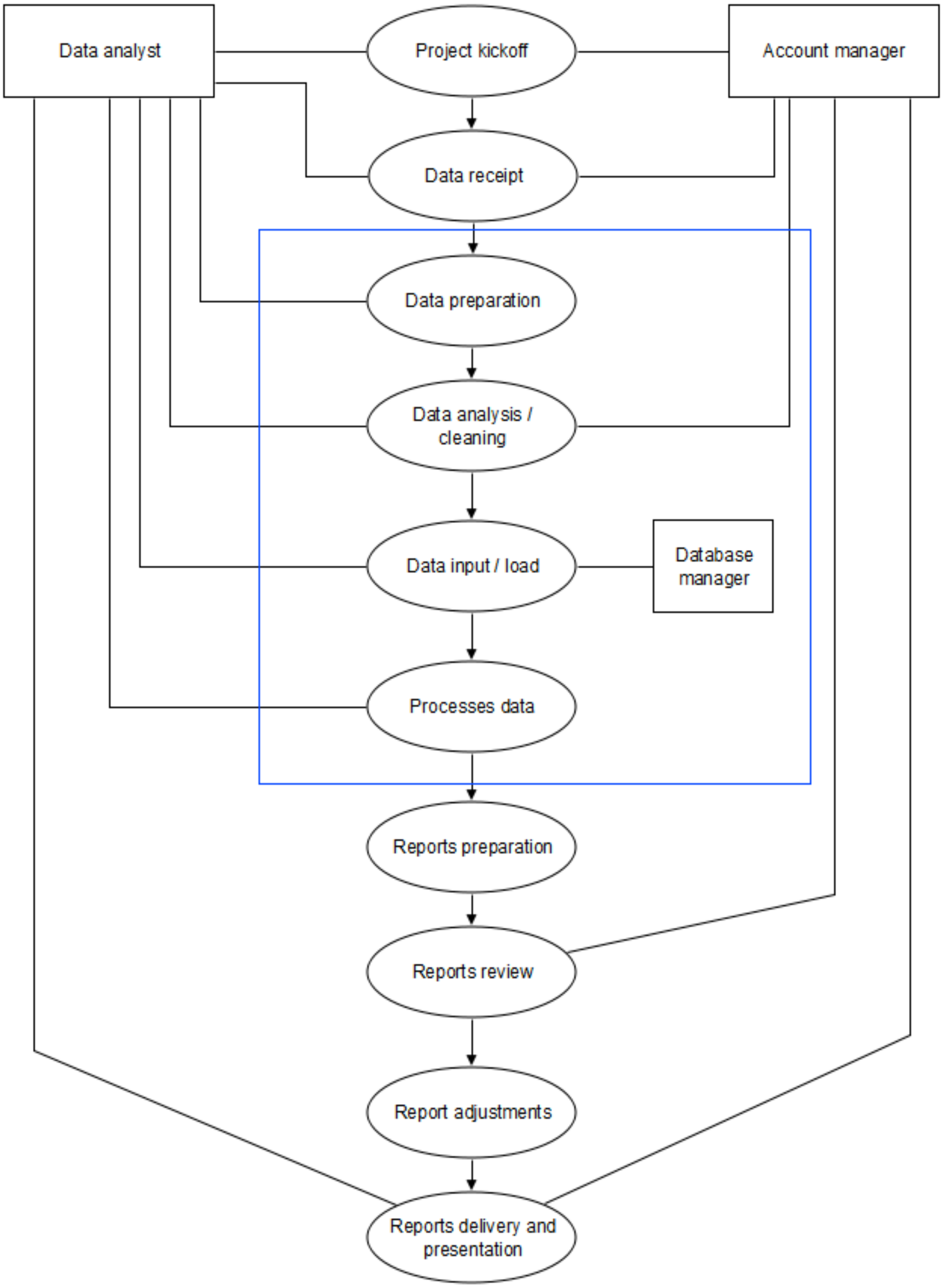
The entire chart covers the relationships between the consulting company (the company developing the compensation survey), the sponsors (three financial institutions paying for the development of the compensation survey) and a group of participants (other financial institutions providing their employee's compensation data).

The chart's mid-section (blue square frame) is where all the activities related to preparing data for any database transpire. It is not identified as part of the database development because a database is not created for every business engagement. This is the area where having a compensation benchmarking database to query data could become an operational solution for the data analyst in charge of answering any post-survey questions from participants or even from any external client interested in getting high-level information about the compensation for a selected market.

The chart below presents a more detailed view of the activities performed inside the consulting company. All these activities will become clear in the Data section of the practical part of this thesis.

The account manager, data analyst and database manager work together at some periods. However, most of the pre-work and post-work burden falls under the responsibility of the Data Analyst. In my working experience, developing a database to manage post-survey requirements could be a quick solution the Data Analyst could use to query data. The database can be created by the same Data Analyst or with cooperation with the Database Manager.

Figure 9: Interview with the Compensation Benchmarking Survey Team



Source: Source: Miranda, 2021.

The prior two sections serve to give an idea about the whole compensation benchmarking survey to anybody in charge of developing a compensation benchmarking database tool for the consulting company.

4.3 Project Analysis

This section's main goal is to connect all the information supplied in the background and context sections and join it with a more organized business analyst perspective.

Before even getting to working with data or the database itself, the consulting company employees collect and aggregate the data to prepare it. One process is to work with survey sponsors and the other one to work with survey participants.

4.3.1 Survey Sponsor Business Diagram

The survey activities become easier to handle with the survey sponsors because it happens in a group session; they are always available because they are funding the project. Additionally, they are one step ahead of any other participant that will be invited to be part of the survey and therefore supply the consulting company with all the required data.

Survey sponsors start early in the process. They have enough time to pull data from their internal database and perform sanity checks to delivered a cleaner version of the compensation data. However, it can be seen in the following figure that there are always situations where the data analyst has to return to them with questions regarding the provided data. The figure assumes there has been a prior business relationship, in the form of a sponsored survey, between the consulting company and the sponsors.

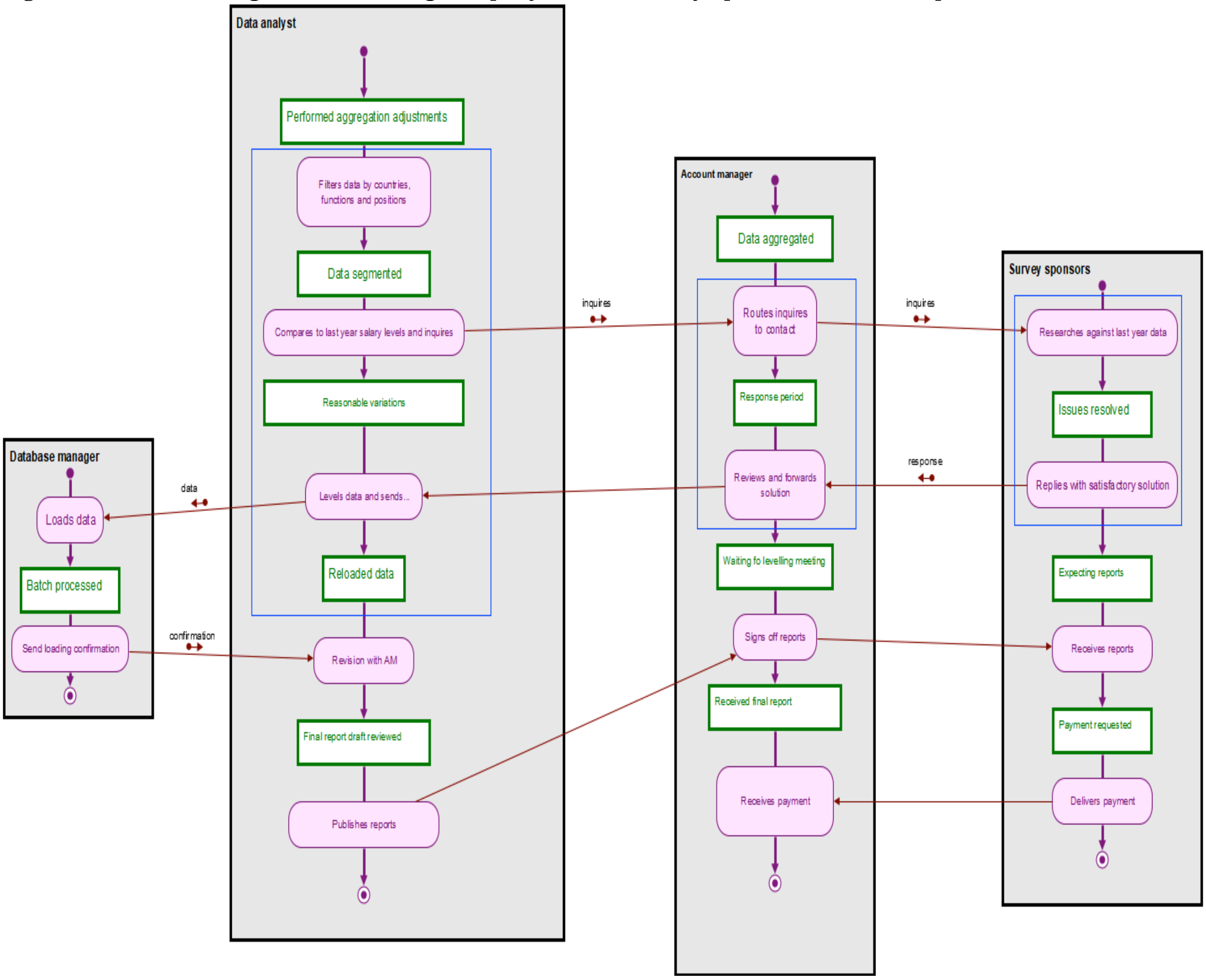
The chart starts from the point where the data analyst has received data for aggregation purposes; some of these steps are:

- Data aggregation adjustments help to exclude any easy-to-identify issues with the data.

- Filtering data by geographies.
- Year-over-year data reviews help to clear any problems caused in the sponsor's internal databases.
- Inquiries about the data are issues that need some research. These issues are usually beyond the data analyst understanding and need to be answered by the sponsor.

The consulting company internal structure uses an Account Manager to route some of the issues or all of the issues; this circumstance varies depending on how the consulting companies choose to manage their business operations.

Figure 10: Business Diagram - Consulting Company and the Survey Sponsors Relationship



Source: Miranda, 2021.

4.3.2 Participants Business Diagram

The complementary portion of the survey is the one related to participants. Here, the Data analyst follows a process that in its essence is similar to the Survey Sponsor process. However, there are some circumstances making it more challenging. Oppossed to the Survey Sponsor's process.

The main difference is that the participant's process is on individual cases. Contrary to the Sponsor's survey process, the consulting company has to deploy a great business development offer to encourage participation.

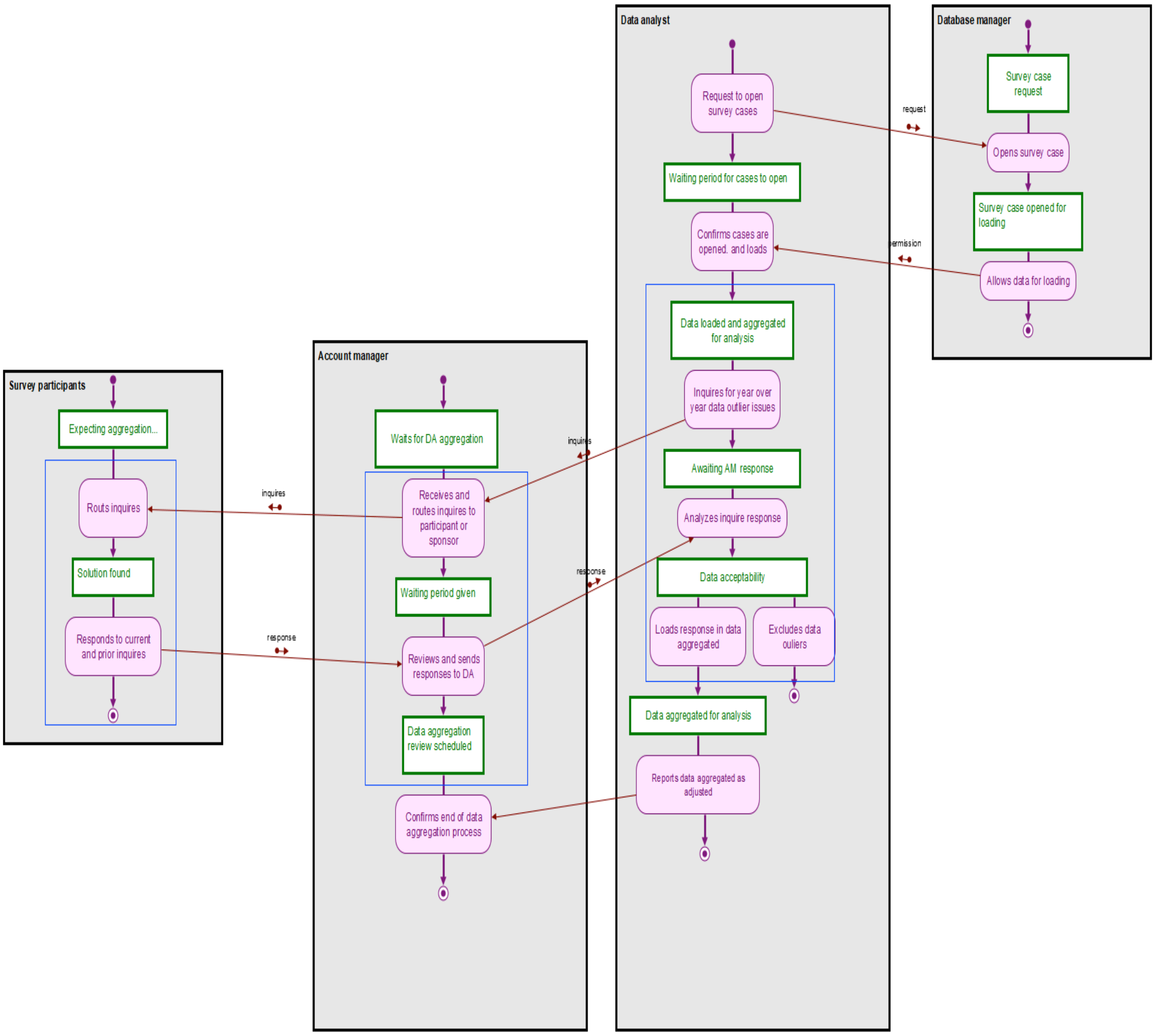
This process starts with setting up group and individual meetings with participants, making compensation survey presentations, agreeing on terms, training personnel to get participants on board.

The following chart includes all the steps the data analyst and other members from the consulting company have to follow for older participants and new participants before using a compensation benchmarking database:

- Data aggregation adjustments serve to exclude any easy-to-identify issues with the data.
- Filtering data by geographies.
- Year-over-year data reviews clear any issues derived from the participant's internal databases.
- Inquiries about the data are issues beyond the data analyst understanding and need to be answered by the participant.

As with the Survey Sponsor process, the consulting company internal structure uses an Account Manager to route issues to participants, but this circumstance varies depending on how the consulting companies decide to manage their business operations.

Figure 11: Business Diagram –Consulting Company and Participants Relationship.



Source: Miranda, 2021.

4.3.3 Business Architecture

The business architecture figure, shown in this section, simplifies the whole concept of a compensation survey that has a component of the compensation benchmarking database.

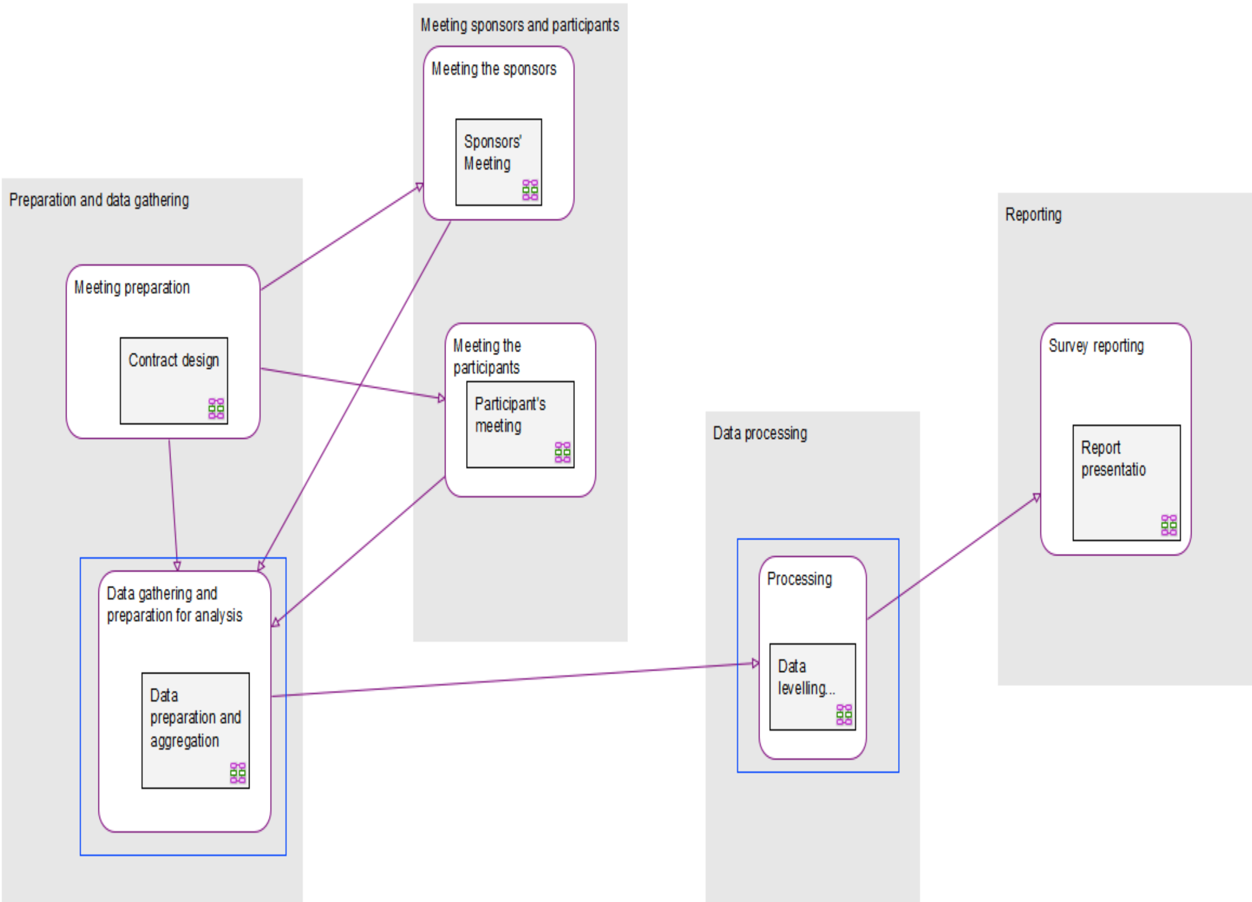
Its main components are:

- Preparation and data gathering
- Meeting Sponsors and participants
- Data Processing

- Reporting (This step is shown in the graph because it is part of a survey process, but it is excluded from the database development as it is a step ahead).

The areas framed in blue (data gathering and processing) are the areas that are interconnected to the work on the compensation benchmarking database. Once the database is created, it will have its operational impact as it could be used as a tool to query data quickly.

Figure 12: Business Architecture - Compensation Benchmarking Survey

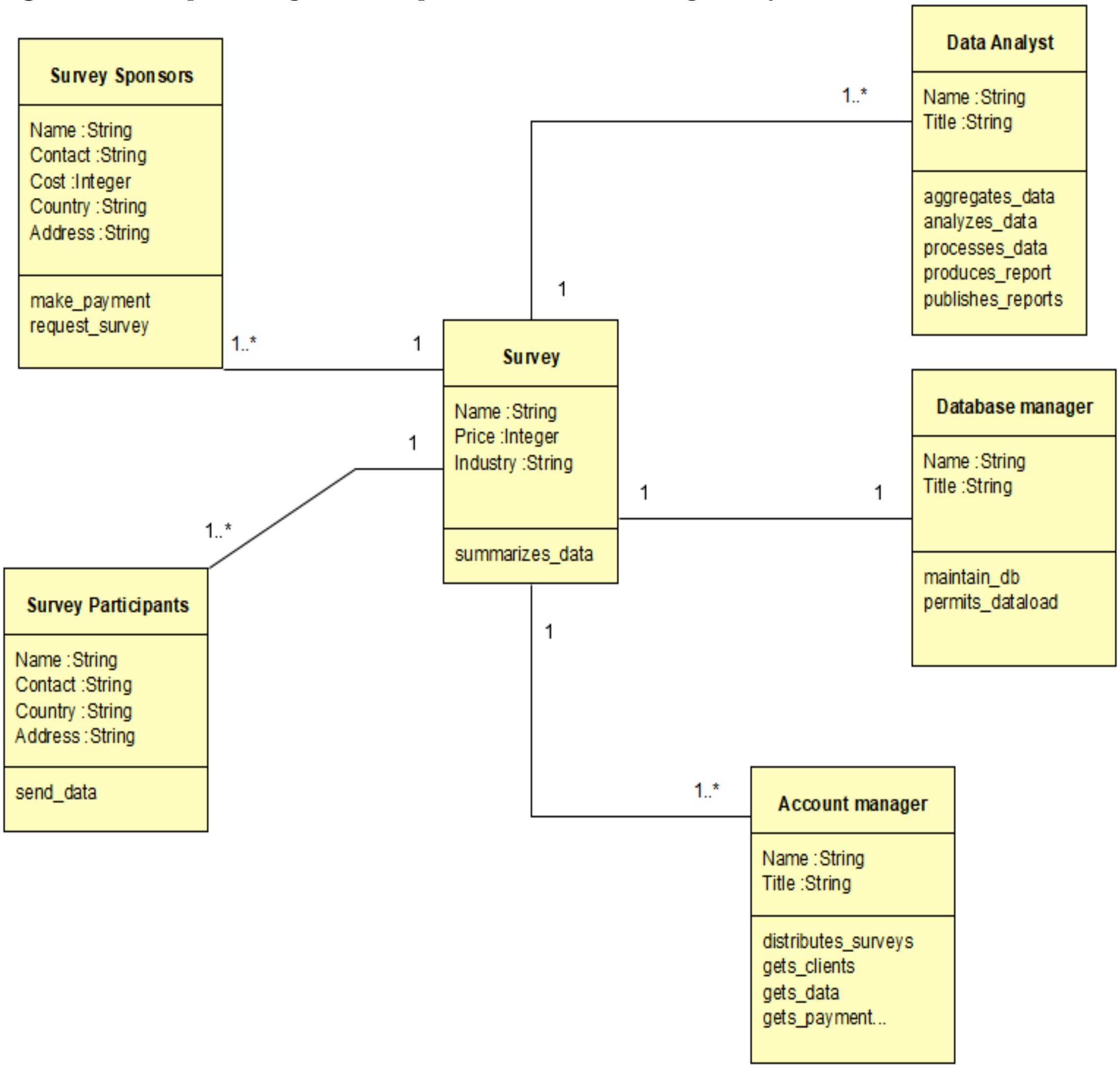


Source: Miranda, 2021.

4.3.4 Conceptual Diagram

Finally, I want to introduce in the project analysis section a conceptual diagram to identify each of the entities that form part of the compensation survey. However, only the information related to survey sponsors, participants and survey are included in the compensation data.

Figure 13: Conceptual Diagram - Compensation Benchmarking Survey



Source: Miranda, 2021.

4.4 Data: What kind of data should be used?

The data corresponds to ten companies (two sponsors and eight selected participants). As mentioned in prior sections, this is the case for a select survey with a targeted number of participants in the Financial Industry, specifically the United States' Corporate banking.

The jobs will include only two Divisional Managers and employees working as Corporate Bankers and exclude any other support functions assisting the corporate bankers. The employees are split into seven levels

Department Head – Head – Level A
Managing Director – MD –Level B
Director – DIR – Level C
Vice President – VP – Level D
Associate – Level E
Junior – Level F
Analyst –Level G

The compensation data includes only:
Base Salary
Annual Bonus – Bonus
Total Compensation

The data line used to represent an employee will be the mean for every job level for a practical reason. The reason is that data points can be too extensive and my purpose is to create a database that uses some referential data instead of working with exact market data. Within the same train of thought, ID numbers are in use and will not select any personal information (names, last names, etc.). In reality, participants do not supply personal information. If there were any information considered private, it is hidden or removed.

4.4.1 Data Collection

The modelled data is the result of increasing or decreasing a personal database between 5% and 15% to model the data. For educational purposes, I will present a data collection format and use a fictitious data submission from a participant.

Table 4: Compensation Data Collection Form – Data Sample

industry	survey	country	company	city	address	contact	phone
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347
FIN	CB	USA	B1	NY	NY	AB	347

unit_code	unit	job_level	title	Emp_ID	salary	bonus	total_compensation
AA	Divisional Management	AA	Head Major Market	1	490	1400	1800
AA	Divisional Management	AB	Vice Chairman	2	395	595	995
R0	Corp Banking	A	Head	10	295	495	795
R0	Corp Banking	B	MD	11	245	245	495
R0	Corp Banking	C	DIR	12	195	145	345
R0	Corp Banking	D	VP	13	145	95	245
R0	Corp Banking	E	Associate	14	105	35	145
R0	Corp Banking	F	Junior	15	85	25	115
R0	Corp Banking	G	Analyst	16	65	5	75

Source: Own work, 2021.

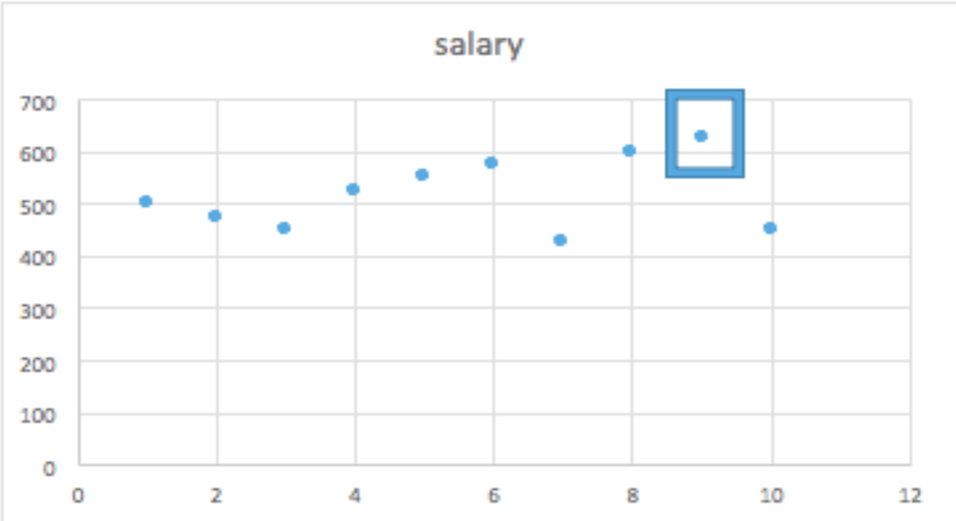
4.4.2 Data Cleaning and Flagging of Outliers

The following charts will illustrate the process used to clean and flag data to normalize it. Flagging means excluding a data point from any calculation, done with a simple observation. Also, as an alternative, it can be done by applying a Z-score and NORM.DIST excel formula as well.

The following three charts are analyzed altogether as the first chart on salary does not reveal any meaningful variation among the data points for the selected sample.

4.4.2.1 Observation Method

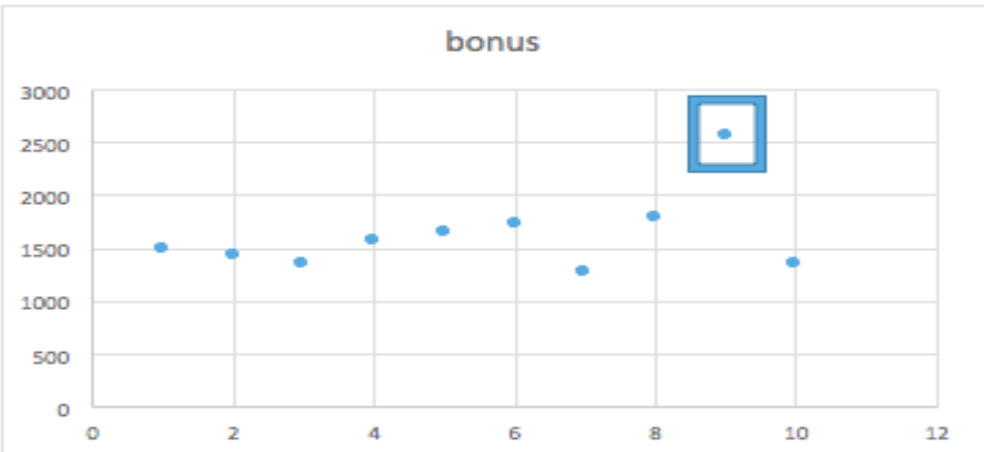
Figure 14: Outlier Flagging – Base Salary



Source: Own work, 2021.

When the data analyst spends more time and looks at every single compensation item, he could pinpoint significant differences. In this case, we observe a top executive data point whose total compensation's main component comes from bonuses, which is opposed to lower-ranked employees whose total compensation originates from a base salary.

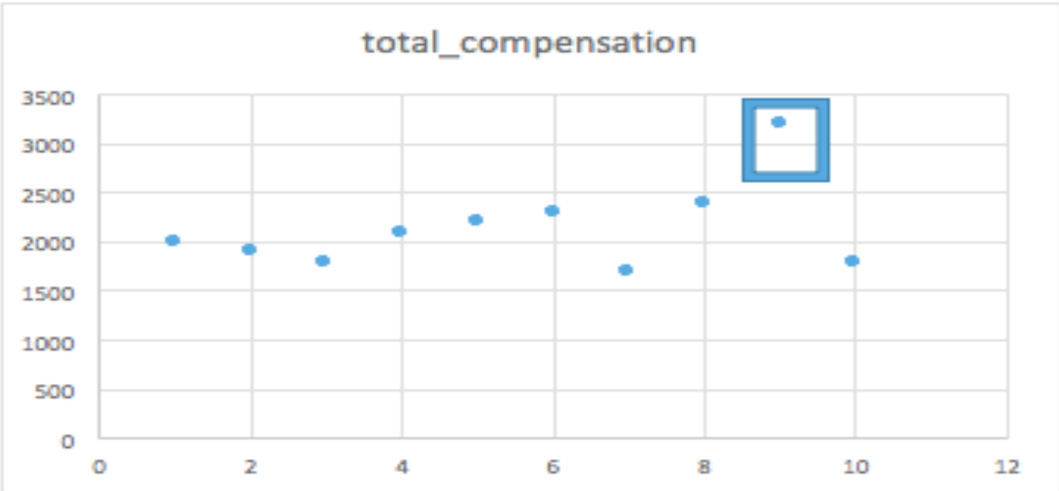
Figure 15: Outlier Flagging - Bonus



Source: Own work, 2021.

Additionally, variable compensation (bonus) has a direct effect on total compensation, shown in the following chart. It seems clear that the out-of-range data point inside the blue frame has an impact on the averages. Additionally, it is easily identified because it would be shown in reports as the maximum. As an alternative, using the median could be a solution to this issue; however, the consulting company could still decide to exclude this data point because it could reveal the employee who receives this total compensation.

Figure 16: Outlier Flagging – Total Compensation



Source: Own work, 2021.

4.4.2.2 Z-score Method

The Z-score is another alternative to identify an outlier. In the chart below, the highlighted section shows that the z-score is higher than the z-scores for the other data points. The same fact is reinforced if the NORM.DIST Excel function is calculated; this data point is bordering the 99% under the curve. Once again, the consulting company's analyst may deem this data point as too far out when compared to the other data points near their corresponding mean value.

Table 5: Flagging – Using the Z-score method

data_points	salary	bonus	total_compensation	salary Z-score	Bonus Z-score	Tcomp Z-score
1	500	1500	2000	-0.25	-0.33	-0.32
2	475	1425	1900	-0.62	-0.52	-0.55
3	450	1350	1800	-0.98	-0.72	-0.78
4	525	1575	2100	0.11	-0.13	-0.09
5	550	1650	2200	0.47	0.07	0.14
6	575	1725	2300	0.84	0.27	0.37
7	425	1275	1700	-1.35	-0.92	-1.01
8	600	1800	2400	1.20	0.47	0.59
9	625	2575	3200	1.56	2.53	2.42
10	450	1350	1800	-0.98	-0.72	-0.78
average	518	1623	2140			
standard deviation	68.77	376.47	437.67			
norm_dist	0.94	0.99	0.99			
norm_inv	625	2575	3200			

Source: Own work, 2021.

4.4.3 Data Preparation

After the cleaning and flagging, the data is ready for upload into the database. The data used in the compensation benchmarking database are the following:

Industry: FIN - Financial Industry

Survey name: CB – Corporate Banking

Country: USA will be the code in this survey

Company: Any code assigned to the company

City: Company's city

Address: Company's address

Contact: HR contact

Telephone number: HR office's telephone number

Unit name: Name of the unit or department

Unit code: code assigned to the unit or department

Job level: The consulting company uses seven levels to describe the company's job structure (From A to G)

Title name: Employee's title name

Employee ID: Identification number assigned by the consulting company

Base salary: Annual Salary (thousands of dollars)

Bonus: Annual bonus compensation (thousands of dollars)

Total compensation: The sum of base salary and bonus (thousands of dollars)

4.5 Database Design

4.5.1 Analysis: Normalization

4.5.1.1 First Normal Form (1NF)

The first normal form includes all the data collected from the sources without any clear segmentation, but it is important to manifest that repeating groups are not present.

Table 6: First Normal Form (1NF)

Data	
industry	FIN
survey	CB
country	USA
year	2020
company	BANK1
city	NY
address	NY
contact	Name
phone	1
unit	Corp Banking
job_level	G
title	Analyst
Emp_ID	16
salary	70
bonus	10
total_compensation	80
exclude_flag	

Source: Own work, 2021.

4.5.1.2 Second Normal Form (2NF)

As the theory suggests, the second normal form creates new tables to eliminate partial dependencies, reassigns the corresponding dependent attributes and then place them in a new table.

Table 7: Second Normal Form (2NF)

Survey Data	
industry	FIN
survey	CB
year	2020
country	USA
Company Data	
company	B1
city	NY
address	NY
contact	Name
phone	1
unit	Corp Banking
fee	200000
Employee Data	
job_level	G
title	Analyst
Emp_ID	16
salary	70
bonus	10
total_compensation	80
exclude_flag	

Source: Own work, 2021.

4.5.1.3 Third Normal Form (3NF)

The step to get to the third normal form can be reduced to making new tables to eliminate transitive dependencies for every transitive dependency and reassigning the corresponding dependent attributes.

Table 8: Third Normal Form (3NF)

Survey Data	
industry	FIN
survey	CB
country	USA
year	2020

Company Data	
company	B1
city	NY
address	NY
contact	Name
phone	1
Unit	
unit	Corp Banking

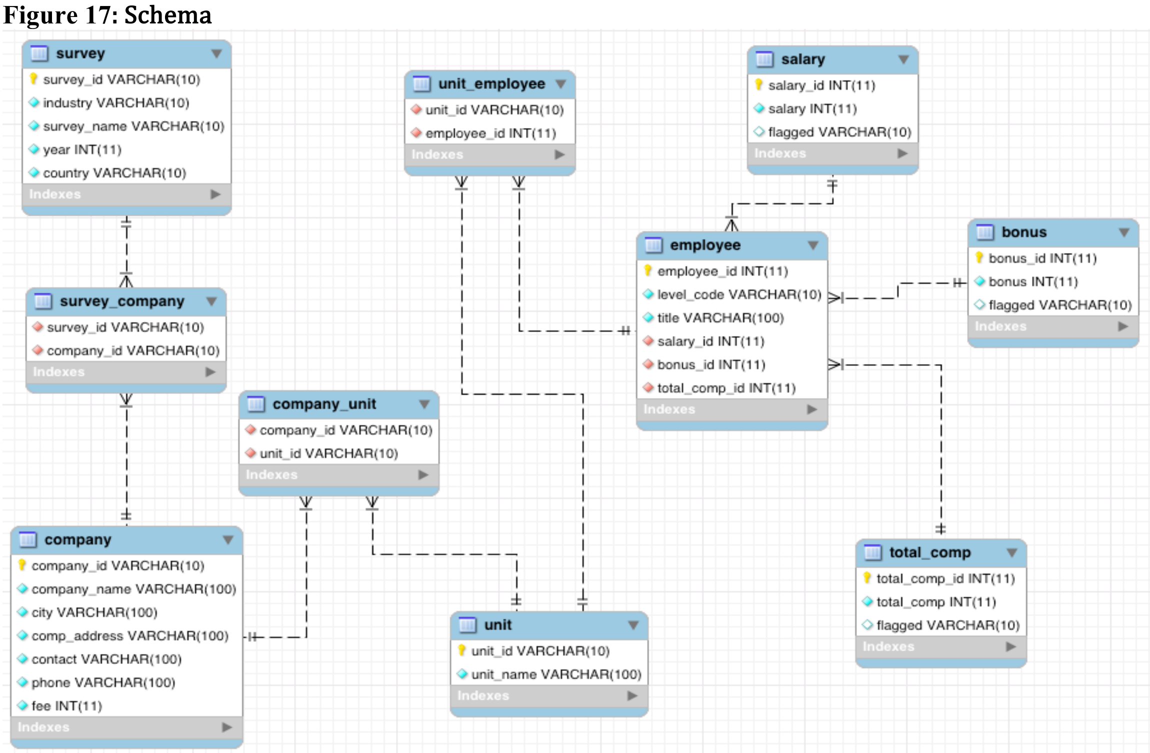
Employee Data	
Emp_ID	16
title	Analyst
job_level	G
Salary	
salary	70
flagged	
Bonus	
bonus	10
flagged	
Total compensation	
total_compensation	80
flagged	

Source: Own work, 2021.

4.5.2 Database Implementation

4.5.2.1 Schema - Compensation Benchmarking Database

Figure 17: Schema



Source: Own work, 2021.

The schema presents an exact picture of how the compensation benchmarking database is structured. In the end, the main blocks are survey, company, unit, employee, salary, bonus and total compensation.

4.5.2.2 Data Definition Language – MySQL

```
DROP DATABASE IF EXISTS compensation_benchmarking;
CREATE DATABASE compensation_benchmarking;
USE compensation_benchmarking;


CREATE TABLE salary(
  salary_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  salary INT NOT NULL CHECK(salary > 0),
  flagged VARCHAR(10) CHECK(flagged IN('yes', "no"))
);

CREATE TABLE bonus(
  bonus_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  bonus INT NOT NULL CHECK(salary >= 0),
  flagged VARCHAR(10) CHECK(flagged IN('yes', "no"))
);

CREATE TABLE total_comp(
  total_comp_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  total_comp INT NOT NULL CHECK(salary > 0),
  flagged VARCHAR(10) CHECK(flagged IN('yes', "no"))
);
CREATE TABLE employee(
  employee_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  level_code VARCHAR(10) NOT NULL,
  title VARCHAR(100) NOT NULL,
  salary_id INT NOT NULL,
  bonus_id INT NOT NULL,
  total_comp_id INT NOT NULL,
  FOREIGN KEY (salary_id) REFERENCES salary(salary_id),
  FOREIGN KEY (bonus_id) REFERENCES bonus(bonus_id),
  FOREIGN KEY (total_comp_id) REFERENCES total_comp(total_comp_id)
);
CREATE TABLE unit(
  unit_id VARCHAR(10) NOT NULL PRIMARY KEY,
  unit_name VARCHAR(100) NOT NULL
);

CREATE TABLE unit_employee(
  unit_id VARCHAR(10)  NOT NULL,
```

```
  employee_id INT NOT NULL,
  FOREIGN KEY (unit_id) REFERENCES unit(unit_id),
  FOREIGN KEY (employee_id) REFERENCES employee(employee_id)
);

CREATE TABLE company(
  company_id VARCHAR(10) NOT NULL PRIMARY KEY,
  company_name VARCHAR(100) NOT NULL,
  city VARCHAR(100) NOT NULL,
  comp_address VARCHAR(100) NOT NULL,
  contact VARCHAR(100) NOT NULL,
  phone VARCHAR(100) NOT NULL,
  fee INT NOT NULL
);

CREATE TABLE company_unit(
  company_id VARCHAR(10) NOT NULL,
  unit_id VARCHAR(10)  NOT NULL,
  FOREIGN KEY (company_id) REFERENCES company(company_id),
  FOREIGN KEY (unit_id) REFERENCES unit(unit_id)
);

CREATE TABLE survey(
  survey_id VARCHAR(10) NOT NULL PRIMARY KEY,
  industry VARCHAR(10) NOT NULL,
  survey_name VARCHAR(10) NOT NULL,
  year INT NOT NULL,
  country VARCHAR(10) NOT NULL
);

CREATE TABLE survey_company(
  survey_id VARCHAR(10) NOT NULL,
  company_id VARCHAR(10)  NOT NULL,
  FOREIGN KEY (survey_id) REFERENCES survey(survey_id),
  FOREIGN KEY (company_id) REFERENCES company(company_id)
);
```

4.5.2.3 Data Input – MySQL

The following data is a sample similar to the data entered in the compensation benchmarking database (Appendix A). All the compensation is exhibited in dollars of the United States of America.

```sql
USE compensation_benchmarking;

INSERT INTO salary
VALUES
 (1,500000, "yes"), (2,400000, "yes"), (3,300000, "yes"),
 (4,250000, "yes"), (5,200000, "yes"), (6,150000, "yes"),
 (7,110000, "yes"), (8,90000, "yes"), (9,70000, "yes"),
(10,475000, "yes"), (11,380000, "yes"), (12,285000, "yes"),
 (13,237500, "yes"), (14,190000, "yes"), (15,142500, "yes"),
 (16,104500, "yes"), (17,85500, "yes"), (18,66500, "yes"),
 (19,525000, "yes"), (20,420000, "yes"), (21,315000, "yes"),
 (22,262500, "yes"), (23,210000, "yes"), (24,157500, "yes"),
 (25,115500, "yes"), (26,94500, "yes"), (27,73500, "yes");

INSERT INTO bonus
VALUES
(1,1500000, "yes"), (2,600000, "yes"), (3,500000, "yes"),
 (4,250000, "yes"), (5,200000, "yes"), (6,150000, "yes"),
 (7,40000, "yes"), (8,30000, "yes"), (9,10000, "yes"),
(10,1425000, "yes"), (11,570000, "yes"), (12,475000, "yes"),
 (13,237500, "yes"), (14,142500, "yes"), (15,95000, "yes"),
 (16,38000, "yes"), (17,28500, "yes"), (18,9500, "yes"),
 (19,1575000, "yes"), (20,630000, "yes"), (21,525000, "yes"),
 (22,262500, "yes"), (23,157500, "yes"), (24,105000, "yes"),
 (25,42000, "yes"), (26,31500, "yes"), (27,10500, "yes");

INSERT INTO total_comp
VALUES
(1,2000000, "yes"), (2,1000000, "yes"), (3,800000, "yes"),
 (4,500000, "yes"), (5,350000, "yes"), (6,250000, "yes"),
 (7,150000, "yes"), (8,120000, "yes"), (9,80000, "yes"),
(10,1900000, "yes"), (11,950000, "yes"), (12,760000, "yes"),
 (13,475000, "yes"), (14,333000, "yes"), (15,23800, "yes"),
 (16,143000, "yes"), (17,114000, "yes"), (18,76000, "yes"),
 (19,2100000, "yes"), (20,1050000, "yes"), (21,840000, "yes"),
 (22,525000, "yes"), (23,368000, "yes"), (24,263000, "yes"),
 (25,158000, "yes"), (26,126000, "yes"), (27,84000, "yes");

insert into employee
VALUES
(1,"AA","Head Major Market",1,1,1), (2,"AB","Vice Chairman",2,2,2),
 (3,"A","Head",3,3,3), (4,"B","MD",4,4,4), (5,"C","DIR",5,5,5),
 (6,"D","VP",6,6,6), (7,"E","Associate",7,7,7),
(8,"F","Junior",8,8,8), (9,"G","Analyst",9,9,9),
 (10,"AA","Head Major Market",10,10,10), (11,"AB","Vice Chairman",11,11,11),
 (12,"A","Head",12,12,12), (13,"B","MD",13,13,13),
 (14,"C","DIR",14,14,14), (15,"D","VP",15,15,15),
```

```
 (16,"E","Associate",16,16,16), (17,"F","Junior",17,17,17),
 (18,"G","Analyst",18,18,18), (19,"AA","Head Major Market",19,19,19),
 (20,"AB","Vice Chairman",20,20,20), (21,"A","Head",21,21,21),
 (22,"B","MD",22,22,22), (23,"C","DIR",23,23,23),
 (24,"D","VP",24,24,24), (25,"E","Associate",25,25,25),
 (26,"F","Junior",26,26,26), (27,"G","Analyst",27,27,27);

insert into unit
values
(1,"Corporate Banking"), (2,"Corporate Banking"), (3,"Corporate Banking"),
(4,"Corporate Banking"), (5,"Corporate Banking"), (6,"Corporate Banking"),
 (7,"Corporate Banking"), (8,"Corporate Banking"), (9,"Corporate Banking"),
 (10,"Corporate Banking"), (11,"Corporate Banking"), (12,"Corporate Banking"),
(13,"Corporate Banking"), (14,"Corporate Banking"), (15,"Corporate Banking"),
 (16,"Corporate Banking"), (17,"Corporate Banking"), (18,"Corporate Banking"),
(19,"Corporate Banking"), (20,"Corporate Banking"), (21,"Corporate Banking"),
 (22,"Corporate Banking"), (23,"Corporate Banking"), (24,"Corporate Banking"),
 (25,"Corporate Banking"), (26,"Corporate Banking"), (27,"Corporate Banking");

insert into unit_employee
values
(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (8,8), (9,9),
(10,10), (11,11), (12,12), (13,13), (14,14), (15,15), (16,16), (17,17), (18,18),
(19,19), (20,20), (21,21), (22,22), (23,23), (24,24), (25,25), (26,26), (27,27);

insert into company
values
(1, "BANK1", "NYC", "NY", "HM", "347-347-3470", 200000),
(2, "BANK1", "NYC", "NY", "HM", "347-347-3470", 200000),
(3, "BANK1", "NYC", "NY", "HM", "347-347-3470", 200000),
(4, "BANK1", "NYC", "NY", "HM", "347-347-3470", 5000),
(5, "BANK1", "NYC", "NY", "HM", "347-347-3470", 5000),
(6, "BANK1", "NYC", "NY", "HM", "347-347-3470", 5000),
(7, "BANK1", "NYC", "NY", "HM", "347-347-3470", 5000),
(8, "BANK1", "NYC", "NY", "HM", "347-347-3470", 5000),
(9, "BANK1", "NYC", "NY", "HM", "347-347-3470", 5000),
(10, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(11, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(12, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(13, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(14, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(15, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(16, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(17, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(18, "BANK2", "NYC", "NY", "AA", "347-347-3471", 200000),
(19, "BANK3", "NYC", "NY", "AA", "347-347-3471", 200000),
(20, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000),
```

```
(21, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000),
(22, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000),
(23, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000),
(24, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000),
(25, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000),
(26, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000),
(27, "BANK3", "NYC", "NY", "BB", "347-347-3472", 200000);

insert into company_unit
values
(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (8,8), (9,9),
(10,10), (11,11), (12,12), (13,13), (14,14), (15,15), (16,16), (17,17), (18,18),
(19,19), (20,20), (21,21), (22,22), (23,23), (24,24), (25,25), (26,26), (27,27);

insert into survey
values
(1, "FIN", "CB", 2020, "USA"), (2, "FIN", "CB", 2020, "USA"), (3, "FIN", "CB", 2020,
"USA"),
(4, "FIN", "CB", 2020, "USA"), (5, "FIN", "CB", 2020, "USA"), (6, "FIN", "CB", 2020,
"USA"),
(7, "FIN", "CB", 2020, "USA"), (8, "FIN", "CB", 2020, "USA"), (9, "FIN", "CB", 2020,
"USA"),
(10, "FIN", "CB", 2020, "USA"), (11, "FIN", "CB", 2020, "USA"), (12, "FIN", "CB", 2020,
"USA"),
(13, "FIN", "CB", 2020, "USA"), (14, "FIN", "CB", 2020, "USA"), (15, "FIN", "CB", 2020,
"USA"),
(16, "FIN", "CB", 2020, "USA"), (17, "FIN", "CB", 2020, "USA"),(18, "FIN", "CB", 2020,
"USA"),
(19, "FIN", "CB", 2020, "USA"), (20, "FIN", "CB", 2020, "USA"), (21, "FIN", "CB", 2020,
"USA"),
(22, "FIN", "CB", 2020, "USA"), (23, "FIN", "CB", 2020, "USA"), (24, "FIN", "CB", 2020,
"USA"),
(25, "FIN", "CB", 2020, "USA"), (26, "FIN", "CB", 2020, "USA"), (27, "FIN", "CB", 2020,
"USA");

insert into survey_company
values
(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (8,8), (9,9),
(10,10), (11,11), (12,12), (13,13), (14,14), (15,15), (16,16), (17,17), (18,18),
(19,19), (20,20), (21,21), (22,22), (23,23), (24,24), (25,25), (26,26), (27,27);
```

4.5.3 Data Querying - MySQL

The following queries show the compensation benchmarking database capacity and usefulness to create queries to fulfil requests made by clients. Each case has the corresponding query and results; all compensation is exhibited in dollars of the United States of America.

Query case 1: Downloading all employees' salaries

```
-- All employees' salaries
SELECT employee.title, employee.level_code, salary.salary
FROM employee
left join
salary
on employee.salary_id = salary.salary_id
where flagged ="yes";
```

Result Grid Filter Rows: Search Export:

title	level_code	salary
Head Major Market	AA	500000
Vice Chairman	AB	400000
Head	A	300000
MD	B	250000
DIR	C	200000
VP	D	150000
Associate	E	110000
Junior	F	90000
Analyst	G	70000

Result 181

Action Output

	Time	Action	Response	Duration / Fetch Time
1	12:17:18	SELECT employee.title, e...	27 row(s) returned	0.0075 sec / 0.00002...

Query case 2: Downloading all employees' bonus

```
-- All employees' bonus
SELECT employee.title, employee.level_code, bonus.bonus
FROM employee
left join
bonus
on employee.bonus_id = bonus.bonus_id
where flagged ="yes";
```

title	level_code	bonus
Head Major Market	AA	1500000
Vice Chairman	AB	600000
Head	A	500000
MD	B	250000
DIR	C	200000
VP	D	150000
Associate	E	40000
Junior	F	30000
Analyst	G	10000

Result 183

Action Output

	Time	Action	Response	Duration / Fetch Time
1	12:19:21	SELECT employee.title, e...	27 row(s) returned	0.00074 sec / 0.0000...

Query case 3: Downloading all employees' total compensation

```
-- All employees' total compensation
SELECT employee.title, employee.level_code, total_comp.total_comp
FROM employee
left join
total_comp
on employee.total_comp_id = total_comp.total_comp_id
where flagged ="yes";
```

title	level_code	total_comp
Head Major Market	AA	2000000
Vice Chairman	AB	1000000
Head	A	800000
MD	B	500000
DIR	C	350000
VP	D	250000
Associate	E	150000
Junior	F	120000
Analyst	G	80000

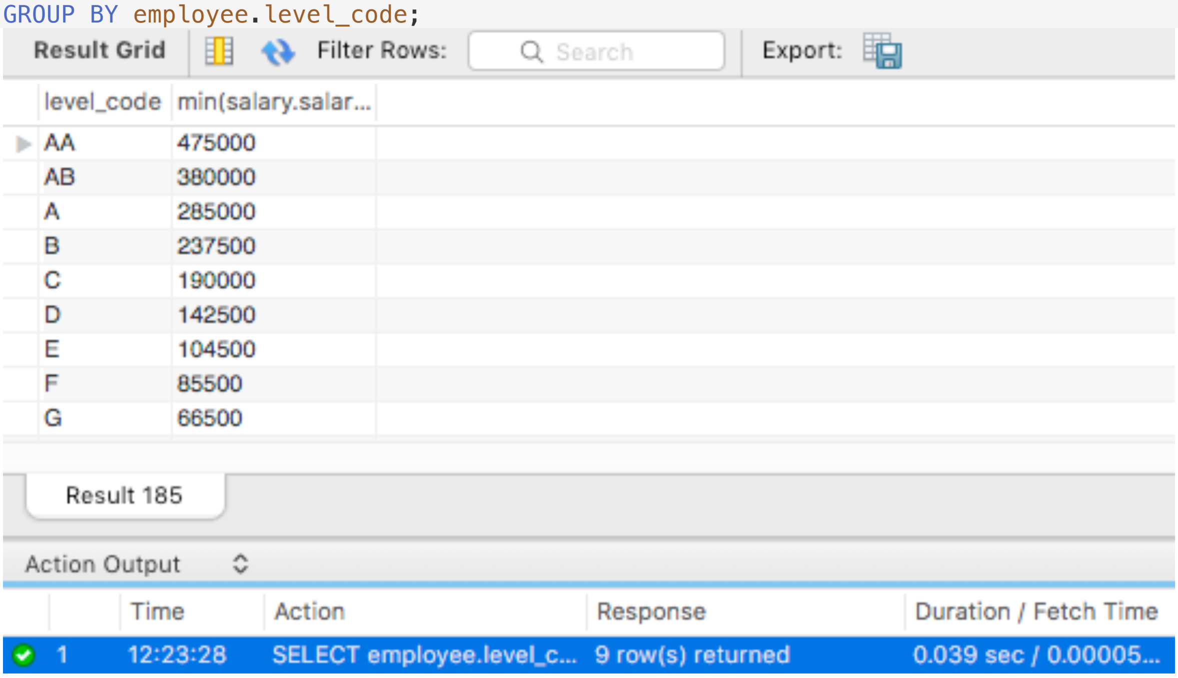
Result 184

Action Output

	Time	Action	Response	Duration / Fetch Time
1	12:21:12	SELECT employee.title, e...	27 row(s) returned	0.022 sec / 0.00003...

Query case 4: Lowest salaries by employee level

```sql
-- Lowest salaries by employee level
SELECT employee.level_code, min(salary.salary)
FROM employee
left join
salary
on employee.salary_id = salary.salary_id
where flagged ="yes"
GROUP BY employee.level_code;
```

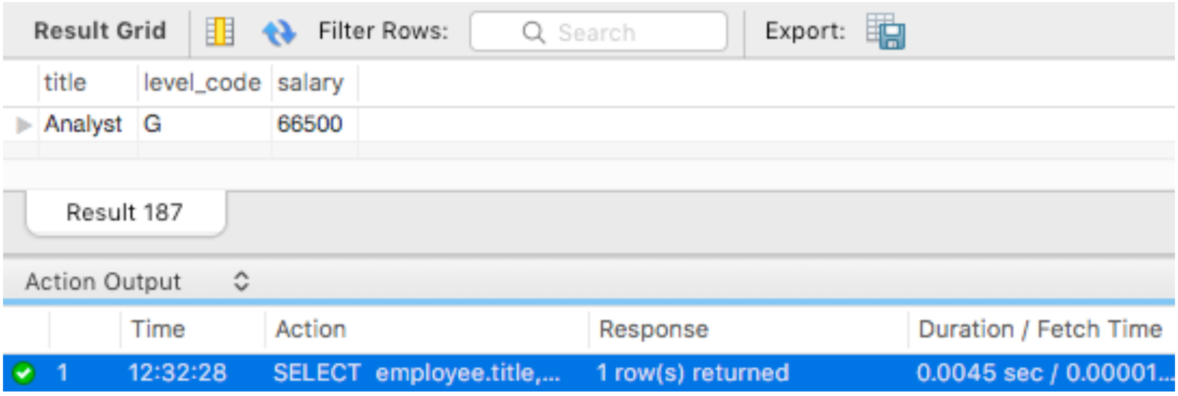


level_code	min(salary.salar...
AA	475000
AB	380000
A	285000
B	237500
C	190000
D	142500
E	104500
F	85500
G	66500

	Time	Action	Response	Duration / Fetch Time
1	12:23:28	SELECT employee.level_c...	9 row(s) returned	0.039 sec / 0.00005...

Query case 5: Obtaining lowest salary data point

```sql
-- An employee with the lowest salary
SELECT  employee.title, employee.level_code, salary.salary
FROM employee
left join
salary
on employee.salary_id = salary.salary_id
where flagged ="yes" AND salary.salary = (select min(salary.salary) from salary);
```



Query case 6: Highest compensation for each employee title and level

```
-- Highest compensation for each employee title and level for the entire survey
select  employee.title, employee.level_code, max(salary.salary),
max(bonus.bonus), max(total_comp.total_comp)
from company
left join
company_unit
on company.company_id = company_unit.company_id
left join
unit
on company_unit.unit_id = unit.unit_id
left join
unit_employee
on unit.unit_id = unit_employee.unit_id
 left join
employee
on  unit_employee.employee_id=employee.employee_id
left join
bonus
on employee.bonus_id = bonus.bonus_id
left join
salary
on employee.salary_id = salary.salary_id
left join
total_comp
on employee.total_comp_id = total_comp.total_comp_id
where salary.flagged ="yes" AND employee.level_code in(“AA", "AB", "A", "B", "C",
"D", "E", "F", "G")
GROUP BY employee.level_code, employee.title ;
```

Result Grid | Filter Rows: Search | Export:

title	level_code	max(salary.salary)	max(bonus.bonus)	max(total_comp.total_comp)
Head Major Market	AA	525000	1575000	2100000
Vice Chairman	AB	420000	630000	1050000
Head	A	315000	525000	840000
MD	B	262500	262500	525000
DIR	C	210000	200000	368000
VP	D	157500	150000	263000
Associate	E	115500	42000	158000
Junior	F	94500	31500	126000
Analyst	G	73500	10500	84000

Result 188

Action Output

	Time	Action	Response	Duration / Fetch Time
1	12:34:35	SELECT employee.title, e...	9 row(s) returned	0.00064 sec / 0.000...

Query case 7: Average compensation for each employee title and level

```
-- Average compensation for each employee title and level for the entire survey
select round(avg(salary.salary),0), round(avg(bonus.bonus),0),
round(avg(total_comp.total_comp),0)
from company
left join
company_unit
on company.company_id = company_unit.company_id
left join
unit
on company_unit.unit_id = unit.unit_id
left join
unit_employee
on unit.unit_id = unit_employee.unit_id
 left join
employee
on  unit_employee.employee_id=employee.employee_id
left join
bonus
on employee.bonus_id = bonus.bonus_id
left join
salary
on employee.salary_id = salary.salary_id
left join
total_comp
on employee.total_comp_id = total_comp.total_comp_id
where salary.flagged ="yes" AND employee.level_code in(“AA", "AB", "A", "B", "C",
"D", "E", "F", "G")
```

```sql
GROUP BY employee.level_code, employee.title ;
```

Result Grid | Filter Rows: Search | Export:

title	level_code	round(avg(salary.salary)...	round(avg(bonus.bonus)...	round(avg(total_comp.total_comp...
Head Major Market	AA	500000	1500000	2000000
Vice Chairman	AB	400000	600000	1000000
Head	A	300000	500000	800000
MD	B	250000	250000	500000
DIR	C	200000	166667	350333
VP	D	150000	116667	178933
Associate	E	110000	40000	150333
Junior	F	90000	30000	120000
Analyst	G	70000	10000	80000

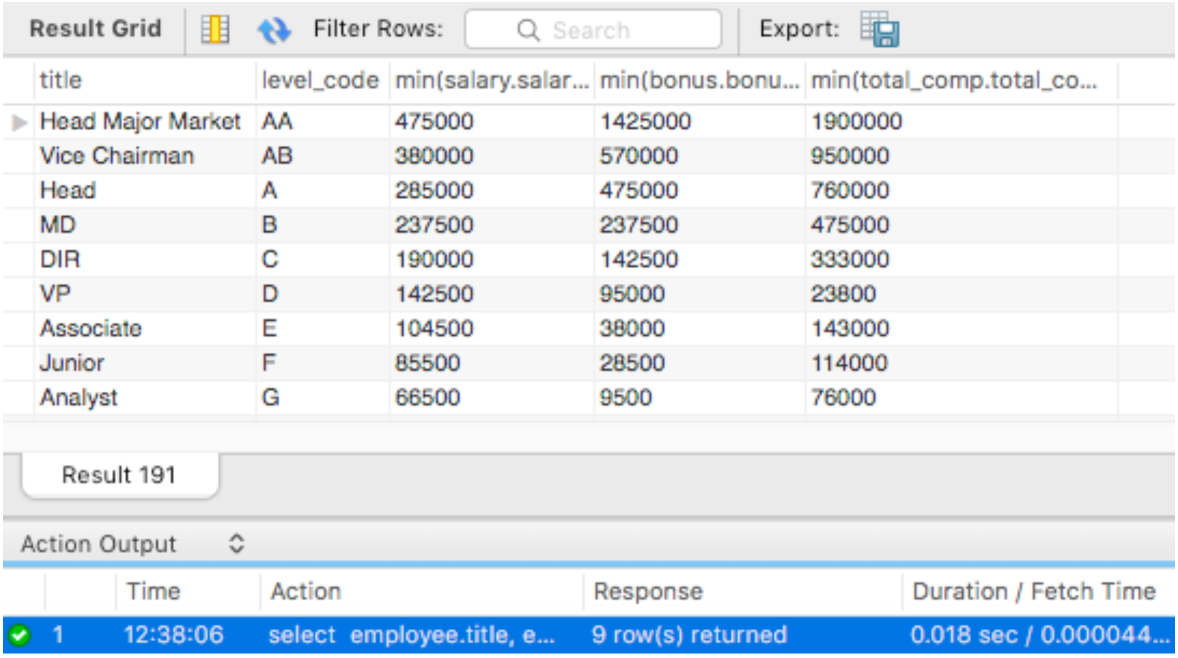
Result 190

Action Output

	Time	Action	Response	Duration / Fetch Time
1	12:36:53	select employee.title, em...	9 row(s) returned	0.027 sec / 0.000076...

Query case 8: Lowest compensation for each employee title and level

```sql
-- Lowest compensation for each employee title and level for the entire survey
select  employee.title, employee.level_code, min(salary.salary),
min(bonus.bonus), min(total_comp.total_comp)
from company
left join
company_unit
on company.company_id = company_unit.company_id
left join
unit
on company_unit.unit_id = unit.unit_id
left join
unit_employee
on unit.unit_id = unit_employee.unit_id
 left join
employee
on  unit_employee.employee_id=employee.employee_id
left join
bonus
on employee.bonus_id = bonus.bonus_id
left join
salary
on employee.salary_id = salary.salary_id
left join
total_comp
on employee.total_comp_id = total_comp.total_comp_id
where salary.flagged ="yes" AND employee.level_code in(“AA", "AB", "A", "B", "C",
"D", "E", "F", "G")
GROUP BY employee.level_code, employee.title ;
```



title	level_code	min(salary.salar...	min(bonus.bonu...	min(total_comp.total_co...
Head Major Market	AA	475000	1425000	1900000
Vice Chairman	AB	380000	570000	950000
Head	A	285000	475000	760000
MD	B	237500	237500	475000
DIR	C	190000	142500	333000
VP	D	142500	95000	23800
Associate	E	104500	38000	143000
Junior	F	85500	28500	114000
Analyst	G	66500	9500	76000

Query case 9: Compensation ranges for each employee title and level

```
-- Compensation Ranges for each employee title and level for the entire survey
select employee.title, employee.level_code, (max(salary.salary)-
min(salary.salary)) as salary_range, (max(bonus.bonus)-min(bonus.bonus)) as
bonus_range, (max(total_comp.total_comp)-min(total_comp.total_comp)) as
totalcomp_range
from company
left join
company_unit
on company.company_id = company_unit.company_id
left join
unit
on company_unit.unit_id = unit.unit_id
left join
unit_employee
on unit.unit_id = unit_employee.unit_id
 left join
employee
on  unit_employee.employee_id=employee.employee_id
left join
bonus
on employee.bonus_id = bonus.bonus_id
left join
salary
on employee.salary_id = salary.salary_id
left join
total_comp
on employee.total_comp_id = total_comp.total_comp_id
```

```
where salary.flagged ="yes" AND employee.level_code in("AA", "AB", "A", "B", "C",
"D", "E", "F", "G")
GROUP BY employee.level_code, employee.title ;
```

Result Grid | Filter Rows: Search | Export:

title	level_code	salary_range	bonus_range	totalcomp_ran...
Head Major Market	AA	50000	150000	200000
Vice Chairman	AB	40000	60000	100000
Head	A	30000	50000	80000
MD	B	25000	25000	50000
DIR	C	20000	57500	35000
VP	D	15000	55000	239200
Associate	E	11000	4000	15000
Junior	F	9000	3000	12000
Analyst	G	7000	1000	8000

Result 192

Action Output

	Time	Action	Response	Duration / Fetch Time
1	12:39:10	select employee.title, e...	9 row(s) returned	0.021 sec / 0.000035...

5 Results and Discussion

It is important to remember this project thesis objective before describing the Compensation Benchmarking Database development and the obtained results. This project thesis objective is to create a relational database application to produce salary market data such as salary, bonus, and total compensation for job functions based on the provided data. Achieving the objective could be appreciated from a consulting company and an employee perspective.

From the business side, the average compensation survey, the reason to create a database, is carried out from the first quarter of the year to the third quarter; it takes approximately eight to nine months. Therefore, any database development does not suffer from strict timing constraints. However, it is relevant to mention that its development should not take longer than a month with testing. This database or a similar one could work for a small to a medium-size consulting firm specialized in benchmarking survey projects to lead and develop its consulting business.

From the data analyst perspective, direct access to a database facilitates data input and access to manipulate the data through custom-made queries. In my experience, consulting companies usually grant access to data pulls to work with Excel with the disadvantage that every single data analysis has to start from zero. Contrary to the recently pointed out scenario, having control over the database through MySQL allows the data analyst to have a ready-to-use set of queries, increasing working efficiency.

5.1 Metrics

Considering that it is more advantageous to work with data collected from a sizeable number of survey participants, I attempted to reproduce the data in the form of a sample. The compensation database includes modelled data for only three compensation components (base salary, bonus and total compensation) for ten data providers.

5.1.1 Database: Coding and Development Period

The compensation database required approximately eighty lines of code to create the database and the corresponding tables. Once again, the tables included a limited number of data fields, but that is the practical side of this tool, it can receive additional data fields with relative ease.

Developing the database takes less than a month. However, the activity absorbing most of the time is related to getting a good understanding of the sort of queries demanded by the consulting firm staff to create reports or by clients interested in custom-made analysis.

5.1.2 Quantitative: Operations and Queries

Based on my personal experience in this consulting business, operationally, a data analyst can reduce his work amount substantially by one or two hours per survey analysis; for example, the presented thesis work only includes one survey in Corporate Banking.

The above activity is possible because the data analyst would be in charge of the database and would not need to pull data or even request access to pull it from a separate database. On top of it, the analyst could have a set of queries ready to perform analysis, there is no need to start from an empty canvas to query and analyse data; a data analyst usually performs analysis with Excel formulas or pivot tables that could result in unmanageable.

Considering data limitations, simple queries designed to pull all the data take up to half of the time (results added to the query section) than the more elaborated queries (e.g. extract the highest-paid employees for every level included in this survey structure).

5.1.3 Qualitative: Consulting Company and Employees

Regardless of any timing differences among queries, the most important and hard to measure aspect is related to making work easier.

The consulting company experiences improvement in day-to-day operations, the company can rebalance the dedicated time from operations to business development. In this type of companies, data analysts need to employ most of their time in client relationship management and business development.

Employees have a direct impact on their performance. While developing the database and also querying the database, I was able to recreate some of the typical queries a data analyst performs, it was evident to me that having control over the data structure reduces the time dedicated to manipulate and analyse it. Additionally, it allows the analyst to commit more time to contribute to the company's business development.

6 Conclusion

This project thesis covered the theoretical parts related to Databases (particularly Relational Database), Compensation (Terminology and Survey) and Basic Statistics (Measures of Central Tendency and Variation) to develop a Compensation Benchmarking Database. I considered this was an opportunity to work with a database tool and data because both are the most vital part of any application.

In retrospect, a crucial project phase was to work with information engineering diagrams to give the database and thesis the required structure. Once I had the proper analysis, I designed the compensation benchmarking database. The resulting database, even though limited by the sample size, was a good exercise. The database allowed me to work with a simplified data model; almost like in an experiment, I worked with a reduced version of the data for this type of surveys in the real world.

With the database tool and some data in hand, I produced basic information, but sufficient to give an analyst or a client insight into the Corporate Banking market. The produced data offers more breadth than what is offered by web tools or free survey sample reports regarding salary, bonus and total compensation.

Regarding future work, it would be critical to have more data from participants to calculate measures of central tendency such as quartiles; at this time, these calculations were not appropriate due to scarce data. With time, data could also allow performing multiple-year variation analysis for each job or among job levels; in practice, some job levels have a direct relationship and may hint about possible compensation changes depending on the market situation. Lastly, due to privacy issues, a frontend tool would not be adequate for the project.

To conclude, from the start, my purpose was to work with a database in a field different from classic commerce or service databases to apply and expand my knowledge. Additionally, I hope that my work could encourage others to explore database applications beyond conventional fields.

7 References

Beaulieu, Alan. 2009. *Learning SQL.* Sebastopol : O'Reilly Media, Inc. , 2009. ISBN: 978-0-596-52083-0.

Blau, Francine & Kahn, Lawrence. 2007. The Gender Pay Gap Have Women Gone as Far as They Can?. The Academy of Management Perspectives 21. 10.5465/AMP.2007.24286161.

Codd, Edgar. 1990. *The Relational Model for Database Management.* New York : Addison-Wesley Publishing Company, Inc., 1990. ISBN: 0-201-14192-2.

Connolly Thomas, Begg Carolyn. 2015. *Database Systems. A Practical Approach to Design, Implementation and Management.* Harlow : Pearson, 2015. ISBN 10: 1-292-06118-9.

Coronel Carlos, Morris Steven. 2019. *Database Systems: Design, Implementation, and Management.* Boston : Cengage, 2019. ISBN: 978-1-337-62790-0.

Date, C.J. 2004. *An Introduction to Database Systems.* Upper Saddle River : Pearson Education Inc., 2004. isbn: 0-321-18956-6.

Eavis, Peter. 2019. New York Times. *It's Never Been Easier to Be a C.E.O., and the Pay Keeps Rising.* New York Times, May 24, 2019, Vol. 2019, May 24.

Elmasri Ramez, Navathe Shamkant. 2011. *Database Systems.* Boston : Pearson, 2011. ISBN 10: 0-136-08620-9.

Foote, Keith. 2017. Dataversity. *Dataversity.* [Online] Dataversity, 03 23, 2017. [Cited: 03 14, 2021.] https://www.dataversity.net/brief-history-database-management/#.

Foundation, The Henry J. Kaiser Family. 2012. Health Care Costs: A Primer. *KFF.* [Online] KFF, 05 01, 2012. [Cited: 03 13, 2021.] https://www.kff.org/health-costs/issue-brief/health-care-costs-a-primer/.

Francine Blau, Lawrence Kahn. 2007. *The Gender Pay Gap: Have Women Gone as Far as They Can?* USA : Academy of Management Perspectives 21, 2007.

Hahs-Vaughn Debbie L., Lomax Richard G. 2020. *An Intrduction to Statistical Concepts.* New York : Routledge, 2020. ISBN: 978-1-315-62435-8 (ebk).

Harrington, Jan. 2009. *Relational Database Design and Implemetation Clearly Explained.* Burlington : Morgan Kaufmann , 2009. ISBN: 978-0-12-374730-3.

Kappel, Mike. 2018. Patriot Software. *Patriot Software.* [Online] Patriot Software, 02 17, 2018. [Cited: 03 13, 2021.] https://www.patriotsoftware.com/blog/payroll/what-is-compensation/#:~:text=Compensation%20is%20the%20total%20cash,they%20do%20for%20your%20business.&text=Compensation%20is%20more%20than%20an,pay%20(hourly%20or%20salary%20wages).

Lawler, Edward E. III. 1971. Pay and Organizational Effectiveness: A Psychological View. New York: McGraw-Hill. ISBN-13: 978-0070367005.

Levine David, Stephan David, Krehbiel Timothy, Berenson Mark. 2008. *Statistics For Managers Using Microsoft Excel.* Upper Saddle River : Pearson, 2008. ISBN0-536-04080X.

Merriam-Webster. 2021. Merriam-Webster. *Merriam-Webster.* [Online] March 13, 2021. [Cited: March 13, 2021.] https://www.merriam-webster.com/dictionary/data.

Milkovich George, Newman Jerry , Gerhart Barry. 2014. *Compensation.* New York : McGraw-Hill Irwin, 2014. ISBN: 978-0-07-802949-3.

Miranda, Harold 2020. "Staff Recruiting Agency". Object and Relational Databases, CULS, student paper.

Miranda, Harold 2021. "Compensation Benchmarking Survey". Information Engineering, CULS, student paper.

Oracle. 2021. Oracle. *https://www.oracle.com.* [Online] 03 13, 2021. https://www.oracle.com/database/what-is-a-relational-database/.

The Henry J. Kaiser Family Foundation, "Health Care Costs: A Primer," May 2012. www. kff.org. Retrieved May 16, 2012.

Ullman Jeffrey, Widom Jennifer. 2014. *A First Course in Database Systems.* Harlow : Pearson Education Ltd, 2014. ISBN 10: 1-292-02582-4.

Weaver, David. 2020. *Pay Matters. The Art and Science of Employee Compensation.* Hillingdon : Lioncrest Publishing, 2020. ISBN: 9781544516677.

Willard, Cherryl Ann. 2020. *Statistical Methods .* New York : Routledge, 2020. ISBN: 978-0-367-20351-1 (hbk).