

**Czech University of Life Sciences Prague**

**Faculty of Economics and Management**

**Department of Information Engineering**



**Bachelor Thesis**

**Database Design for a Promotional Item Shop**

**Abdul Azim Khaliqyar**

**© 2020 CULS Prague**



# CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

## BACHELOR THESIS ASSIGNMENT

Abdul Azim Khaliqyar

Systems Engineering and Informatics  
Informatics

Thesis title

**Database Design for a Promotional Item Shop**

---

### Objectives of thesis

The main aim of the thesis is to create a database system for small business.

To fulfill this aim, following partial objectives are set:

- to analyze and gather business requirements
- to create a data model and implement it in DBMS
- to implement a data-driven application to facilitate access to data for users
- to assess if the resulting solution fulfills the need of the business

The thesis will provide a real example of how a database could help even a small business with more effective usage of data.

### Methodology

Methodology of this thesis is based on analysis and study of various information sources related to database systems, data modelling and software development. Based on the synthesis of the gained knowledge the traditional top-down approach was chosen. Regular interviews with the customer are set as the main method for the first objective. The second objective is covered by creating conceptual, logical and physical data models in selected CASE tool and consequently implementing the database in DBMS. A small application prototype will be created to facilitate data access for regular users and to test if the business requirements have been properly captured. Eventually the customer will be interviewed to evaluate the resulting database system.

**The proposed extent of the thesis**

40-50 pages

**Keywords**

Database, data modelling, ERD, database design, RDBMS

---

**Recommended information sources**

CONNOLLY, Thomas M., Carolyn E. BEGG a Jennifer WIDOM. Database systems: a practical approach to design, implementation, and management. 5th ed. London: Addison-Wesley, c2010. ISBN 978-0-32-152306-8.

GARCIA-MOLINA, Hector, Jeffrey D. ULLMAN a Jennifer WIDOM. Database systems: the complete book. 2nd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2009. ISBN 978-0-13-187325-4.

ULLMAN, Jeffrey D. a Jennifer WIDOM. A first course in database systems: a practical guide to secure computing. 3rd ed. Upper Saddle River, NJ: Pearson/Prentice Hall, c2008. ISBN 978-0-13-600637-4.

---

**Expected date of thesis defence**

2019/20 WS – FEM (February 2020)

**The Bachelor Thesis Supervisor**

Ing. Petr Hanzlík, Ph.D.

**Supervising department**

Department of Information Engineering

Electronic approval: 25. 11. 2019

**Ing. Martin Pelikán, Ph.D.**

Head of department

Electronic approval: 25. 11. 2019

**Ing. Martin Pelikán, Ph.D.**

Dean

Prague on 23. 03. 2020

---

### **Declaration**

I declare that I have worked on my bachelor thesis titled "Database Design for a Promotional Item Shop" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the bachelor thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 23<sup>rd</sup> March 2020

---

## **Acknowledgement**

I would like to express my gratitude to Ing. Petr Hanzlík, Ph.D. for his valuable advices and supervision of this bachelor's thesis. I would like to thank my entire family and especially my girlfriend, and all who supported me during the work on this thesis.

# Database Design for a Promotional Item Shop

## Abstract

This bachelor's thesis deals with a database design for a small shop selling promotional items. The main objective of the thesis is to create a complex database solution. Analysing the business, creating data models, creating application and testing, are considered as partial objectives of the thesis. For the whole process of database design, the traditional top-down approach is followed. The main benefit of this thesis is not only to show how a database solution can help even a small business with more effective usage of data, but also to provide a manual, explaining the theory first with putting the knowledge into practice later, which could be followed when starting the creation of any database.

**Keywords:** Database, database design, data modelling, conceptual data model, physical data model, ERD, DBMS, SQL

# Návrh databáze pro obchod s reklamními předměty

## Abstrakt

Tato bakalářská práce se zabývá návrhem databáze pro malý podnik prodávající reklamní předměty. Hlavním cílem práce je vytvoření komplexního databázového řešení. Analýza byznysu, vytvoření datových modelů, vytvoření aplikace a testování, jsou stanoveny cíli dílčími. Během celého procesu návrhu databáze je následována tradiční metodika shora-dolů. Hlavním přínosem práce není pouze ukázka toho, jak může databázové řešení pomoci i malému podniku s efektivnějším využitím dat, ale i poskytnutí příručky, nejprve vysvětlující teorii s její následnou aplikací v praxi, která by mohla být následována při návrhu jakékoliv databáze.

**Klíčová slova:** Databáze, návrh databáze, datové modelování, konceptuální datový model, fyzický datový model, ERD, DBMS, SQL



# Table of content

<b>1</b>	<b>Introduction</b>	<b>13</b>
<b>2</b>	<b>Objectives and Methodology</b>	<b>15</b>
2.1	Objectives	15
2.2	Methodology	15
<b>3</b>	<b>Theoretical Part</b>	<b>16</b>
3.1	Database Design Process	16
3.1.1	Database Design Phases	17
3.2	Business Analysis	17
3.2.1	Interview	18
3.2.2	Business Model	19
3.2.3	Business Requirements	19
3.3	Data Models	19
3.3.1	Conceptual Data Model	20
3.3.1.1	Identification of Entities	20
3.3.1.2	Identification of Relationships	21
3.3.1.3	Control of Model for Redundancy	21
3.3.1.4	Questions from Business and User Requirements	23
3.3.1.5	Review of Conceptual Data Model with User	23
3.3.2	Physical Data Model	23
3.3.2.1	Inheritance Relationships	23
3.3.2.2	Integrity Constraints	23
3.3.2.3	Required Data	24
3.3.2.4	Generation of Primary Key	24
3.3.2.5	User Access Right	25
3.3.2.6	Review of Physical Data Model with User	25
3.3.3	Implementation in Database Management System	25
3.4	Application	25
<b>4</b>	<b>Practical Part</b>	<b>27</b>
4.1	Business Analysis	27
4.1.1	Interview	27
4.1.2	Business Model	28
4.1.3	Business Requirements	28
4.2	Data Models	29
4.2.1	Conceptual Data Model	29
4.2.1.1	Identification of Entities	29
4.2.1.2	Identification of Relationships	38

4.2.1.3	Control of Model for Redundancy.....	43
4.2.1.4	Questions from Business and User Requirements.....	44
4.2.1.5	Review of Conceptual Data Model with User.....	45
4.2.1.6	Conceptual Data Model Diagram.....	46
4.2.2	Physical Data Model.....	46
4.2.2.1	Inheritance Relationships.....	47
4.2.2.2	Integrity Constraints.....	47
4.2.2.3	Required Data.....	49
4.2.2.4	Generation of Primary Key.....	49
4.2.2.5	User Access Right.....	50
4.2.2.6	Review of Physical Data Model with User.....	50
4.2.2.7	Physical Data Model Diagram.....	50
4.2.3	Implementation in Database Management System.....	51
4.3	Application.....	52
<b>5</b>	<b>Results and Discussion.....</b>	<b>54</b>
<b>6</b>	<b>Conclusion.....</b>	<b>56</b>
<b>7</b>	<b>References.....</b>	<b>57</b>

## List of pictures

Picture 1 Conceptual Diagram.....	46
Picture 2 Physical Diagram.....	51
Picture 3 Application Form for Employee.....	53
Picture 4 Application Form for Customer .....	53

## List of tables

Table 1 Company.....	30
Table 2 Person .....	30
Table 3 Customer.....	31
Table 4 Order .....	31
Table 5 Invoice .....	32
Table 6 Complaint .....	32
Table 7 Purchase.....	33
Table 8 Item.....	34
Table 9 Item Type.....	35
Table 10 Adjustment.....	35
Table 11 Adjustment Type .....	36
Table 12 Employee .....	37
Table 13 Branch.....	38
Table 14 Customer - Purchase .....	38
Table 15 Customer - Order .....	39
Table 16 Order - Invoice.....	39
Table 17 Order - Item .....	39
Table 18 Invoice - Complaint .....	40
Table 19 Invoice - Employee.....	40
Table 20 Branch – Employee.....	41
Table 21 Employee - Adjustment .....	41
Table 22 Item - Purchase .....	41
Table 23 Item - Supplier .....	42
Table 24 Item - Item Type .....	42
Table 25 Item - Adjustment .....	43
Table 26 Adjustment - Adjustment Type .....	43
Table 27 User Role and Access Right .....	50

## List of abbreviations

Abbreviation	Definition
DB	Database
DBA	Database Administrator
DBMS	Database Management System
DDL	Data Definition Language
ERD	Entity-Relationship Diagram
IS	Information System
IT	Information Technology
RDD	Relational Database Design
SQL	Structured Query Language

# 1 Introduction

Since man started to establish a more organized and structured manner to conduct business and trading activities, data has played a pivotal role in the process. Through business activities data is produced, observed, measured, and archived, in attempt to construct a database that will act as guideline in many aspects of a business environment such as decision-making.

Data are individual, atomic pieces of information that when read for what they are, provide little or no meaning, therefore the need arises to provide a reliable context to that data in order to essentially transform it into information, which will later be used to create knowledge. Knowledge is important for making precise and meaningful decisions about current or future events. To provide that, data has gone through appropriate processes. Therefore, it is most productive for any organization to have their data organized, structured, and stored safely in a way that can be accessed quickly, manipulated, and analyzed it further for making precise decisions in order to boost productivity.

As every day technological advances shape our everyday life and many aspects of it, so has the data field of study been affected by said changes as well, making it easier today than ever before to control the flow of information. Storing data has become easy, quick, effective and easily referenced by means of creating enterprise level databases, easily accessible for all departments of an enterprise providing a more direct and accurate data flow. Nowadays, such a process has become an irreplaceable necessity that provides various advantages for any business no matter how big or small.

As in every other aspect of the financial sector, if the move does not lead to higher productivity and increased revenue, it will most likely not be implemented. Some of the benefits of maintaining a reliable database include but are not limited to allowing businesses to be more flexible, quicker decision-making, adjusting market positions and direct department communication.

Any database system has the ability to connect partial data into access points of information in a way that queries can be retrieved instantly and this connectivity in any organization and quick access to information can impact overall productivity and positive output. Furthermore, other aspects can be improved upon by the uses of databases, such as customer service, in the means of analyzing customer data to provide a more personally

tailored experience, improving the likelihood that a customer will return. This approach also provides valuable insight into marketing aspects especially during the last decade where personal advertising is becoming the golden standard in advertising.

This thesis will describe the importance of database usage and how said database can help even small businesses improve their productivity and increase their revenue, through its use. The theoretical part focuses on the analysis of the business and the process of the database design based on the gathered requirements of the business analysis and the practical part focuses on the step by step implementation of the gathered information from the business requirements analyses through all phases of the database design process. After the completion of each phase, the need arises for the steps to be reviewed by the users in order to ensure that the implemented design is in accordance with the specific business needs.

## **2 Objectives and Methodology**

### **2.1 Objectives**

The main aim of this thesis is to create a database system for a small business. To fulfil this aim, following partial objectives are set:

- to analyse and gather business requirements
- to create a data model and implement it in DBMS
- to implement a data-driven application to facilitate access to data for users
- to assess if the resulting solution fulfils the need of the business

The thesis will provide a real example of how a database could help even a small business with more effective usage of data.

### **2.2 Methodology**

Methodology of this thesis is based on analysis and study of various information sources related to database systems, data modelling and software development. Based on the synthesis of gained knowledge, the traditional top-down approach was chosen. Regular interviews with the customer are set as the main method for the first objective. The second objective is covered by creating conceptual, logical and physical data models in selected CASE tool and consequently implementing the database in DBMS. A small application prototype will be created to facilitate data access for regular users and to test if the business requirements have been properly captured. Eventually the customer will be interviewed to evaluate the resulting database system.

## **3 Theoretical Part**

### **3.1 Database Design Process**

The designing process plays the key role for a successful creation of any database, because a poor design can cause lots of troubles afterwards – such as not fulfilling organization needs, serving in a user-unfriendly way, making updates or future changes much harder or even causing a failure. Usually, it is expensive and time-consuming to change the design after its creation, therefore it is important to focus on this activity and create a good database design at first place. To create one, from business perspective it is necessary to have for instance a good understanding of the main objectives of the organization, good knowledge of the business, analytical thinking or deep understanding of user requirements. From IT point of view, it is important to have the knowledge of database design principles, hardware and software utilization and limitation, and database implementation. Creating the connection between business area and IT area is another crucial step towards success, but unfortunately the list of all these prerequisites has not come to an end yet. In another way, it simply shows the complexity of the database design process and that is the reason why many methodologies have been created in order to support the whole process and help all involved parties. One might assume that all these methodologies are based on similar concepts, but each methodology can support different approaches depending for instance on the organization interest. It also needs to be mentioned that there is not only one right approach for a database design process.

For the purpose of this bachelor thesis, the traditional top-down approach was chosen. The reason that makes the top-down approach very common is because it covers the important and key entities of the business first, which makes it easier to understand and to expand the model afterwards. On the other hand, the bottom-up approach starts with details, such as attributes, that are then formed into entities. This approach can work well for small and simple businesses with small number of attributes. However, the difficulties raise directly, proportional to the increase of the number of the attributes.



### **3.1.1 Database Design Phases**

Each database design process consists of four main database design phases. These phases are followed within the theoretical part of this thesis, where the theory for each phase is explained, and within the practical part, where the knowledge is put into practice for the business case of the thesis. Four phases mentioned above are following:

- Creating business analysis
- Creating data models
- Implementing database application
- Testing

Creating a business analysis is the first phase, which is there in order to create a wide knowledge base that will come from the analysis of the business and from collecting all business requirements and that will represent the first pillar of the future database as all outputs of this phase will serve as the key inputs for all other phases afterwards.

Data models that are created in the second phase are tied up together with the first phase as some parts of the analysis that were not covered in the first phase may be brought up at this time. That is one of the reasons why there is the necessity to review all the outputs of this phase (conceptual and physical diagram) with the future users.

Third phase represents the real implementation of a database application that will be used by the users. User interface is created to support user's needs to view, insert, update or delete the data, simply to mediate the interaction between them and the database.

Final phase of the database design process is usually the user acceptance testing. Users evaluate the whole process of the database design, provide their feedback and decides whether the solution corresponds to their expectations.

## **3.2 Business Analysis**

Analysing the business is usually the first step of a database design process. It helps to identify the enterprise requirements, needs and objectives. In this stage it is necessary to identify which information is required by the enterprise to be included in information system. Understanding how the business operates, which data needs to be part of the information system and studying the business model to figure out which data is important for the business and for the users.

### 3.2.1 Interview

“Interviewing is the most commonly used and normally the most useful factfinding technique. We can interview to collect information from individuals face-to-face. There can be several objectives to using interviewing, such as finding out facts, verifying facts, clarifying facts, generating enthusiasm, getting the end-user involved, identifying requirements, and gathering ideas and opinions. For the database design structured interview is required that has specific set of questions to ask the interviewers.” (Connolly & Begg, 2005)

The process of analysing the business starts with interviewing the owner of the business, users and key members of the organization. Set of questions needs to be asked during the interview such as:

- What is the mission of the organization?
- Which data the organization considers as valuable?
- Which data must be part of the information systems?
- What is their expectation from the information system?
- What is the organization view about database that will help to solve their problem?
- Does the company see the need for a database system?

Set of questions needs to be asked from users such as:

- What is the difficulty that users face providing services?
- What kind of data users interact with during their typical day?
- Do users have problem looking back for data?
- How user make the reports to the manager and how often?
- Where user store data and what kind of a data and for how long they keep that data?

An interview plan needs to be set – such as how often to meet for the interview and who to meet for the interview. Another step is to decide whether to use documentation, and if yes, then what kind of documentation should be used? Since the main objective for the interview is to understand the business requirements and business model itself, it is necessary to avoid the issue of misunderstanding and miscommunication of the information from all sides – IT, owner, and users.

### **3.2.2 Business Model**

Shio Kuma Singh (Singh, 2009) describes business model as a representation of objects and events from a real world that have associations between each other. A simple definition for business model can be the answer for what is the story behind the goal of the company to make money. It can be a procedure that company takes to deliver its services that fulfil the customers' needs and satisfy them. The business model differs depending on the organization's objective. For example, the business model for Microsoft can be defined as selling software, for a logistic company it could be transporting goods to customers. These key processes are performed thanks to many supporting processes within the organization.

### **3.2.3 Business Requirements**

Business requirements are usually stored as questions from the business side. Important questions which they want to find the answers for. Some of the examples may be:

- What product is sold the most?
- What is the average amount of an order?
- Who is our most regular customer?

However, some requirements can also include the need of having a specific user interface, specific security setting or archival demands for instance.

## **3.3 Data Models**

To create a data model, it is essential to understand the business model, entities, relationship and attributes. A data model is a simplified model of real data. For example, a constructional company before beginning an actual construction may need to create a map or draw a picture of a house and must show to the house owner how it could look like before construction, because reconstructing might be very expensive, and drawing might help to avoid that. Creating model is cheap, easy and possible to redesign until the model represents the real meaning of the data or objects.

For the database design process data modelling is used to represent a data model from the enterprise entities and requirements. To ensure the enterprise data is understood accurately by all sides such as enterprise, users, and designers, a data model must be created first,

before any physical implementation of the data. Therefore, in business requirement section the enterprise requirements were analysed first, and a data model is now to be created.

A CASE tool which will be used for creating stated data models needs to be chosen. Current market offers quite a few software tools which could be used. One of the things that should be considered here is whether the tool includes conceptual and physical data models and whether a database can be generated afterwards, for instance.

### 3.3.1 Conceptual Data Model

“The conceptual database design phase begins with the creation of a conceptual data model of the enterprise that is entirely independent of implementation details such as the target DBMS, application programs, programming languages, hardware platform, performance issues, or any other physical considerations.” (Connolly & Begg, 2005)

In order to create a conceptual data model from the business requirements it is necessary to build at least one conceptual data model which includes: identifying entity types, the relationship types among entities, and primary keys, if needed alternative keys. Following chapters are devoted to the process of conceptual modelling in details. Before that, each of these terms that are used above is defined shortly as follows:

- **Entity:** “An object, thing, person or unit that we can store data about it and contain attribute which can define the existing of the entity.” (Connolly & Begg, 2005)
- **Primary key:** “Unique key or identification for an entity used in relational database to distinguish each record uniquely on the database table.” (Connolly & Begg, 2005)
- **Integrity constraints:** “Set of rules for maintaining the quality of information and ensuring the data integrity is not violated during the data insertion, updating, deleting and other processes.” (Connolly & Begg, 2005)

#### 3.3.1.1 Identification of Entities

The starting step in creating conceptual data model is to identify the most important objects, that users interact and work with the most. These objects are the entity types for this model. Entities can be identified from user requirements, it can be done by identifying the most common phrases and nouns that are mentioned in the user requirement (for

example, customer name, customer address, customer roles, supplier name, supplier address, item name, item serial number, item cost). Knowing this information make it clear that one of the entities will be the customer because all this information such as (customer name, customer address, customer roles) can be defined under one object or entity called customer and another entity will be item by grouping item name, item serial number, and item cost with an entity called item. Another step that can be used to identify the entities is to look for major objects, such as people, places or the company concept of interest. For this company, both methods were used to identify entities, although nearly all entities were identified using the first method. However, both methods were used to ensure that all important entities were observed, or entities that are at least adequate for the system requirements.

### **3.3.1.2 Identification of Relationships**

After identifying the entities, the next step is to identify relationships among the entities. In previous section, when identifying entities, the method was to look for phrases and nouns that are more important for the user from the user requirements. Similar method applies to identify relationships that is required among entities. With slightly differences, that relationships are described using verbs such as Customer creates an Order, Supplier supplies an Item, Employee manages the Adjustment etc. Another step to describe the relationship is to draw the Entity-Relationship Diagrams (ERD). Following chapters are devoted to the explanation of the relationships among the entities in details. After completing this step, the ER diagram will be presented to show the summary of the entity's identification and their associated relationships.

### **3.3.1.3 Control of Model for Redundancy**

After identifying entities and relationships, the proposed model should be checked against redundancy. This check is done in three steps, as described below.

#### **Re-Examination of one-to-one Relationships**

According to Connolly and Begg (2005), in this step the model is checked if during entities identification two entities were identified that represent the same objects in the enterprise. For example, if two entities such as one called customer and other one called buyer were

identified as two different entities, they will not be seen like that anymore because buyer is just a synonym for customer. In such case both entities should be combined as one entity. If the primary keys are different, one primary can be chosen, and another can be used as alternative key.

### **Removal of Redundant Relationships**

“A relationship is redundant if the same information can be obtained via other relationships.” (Connolly & Begg, 2005) There is no rule that if there are more relationships among entities one must be removed because those relationships may represent different associations between entities.

### **Consideration of Time Dimension**

In this step Connolly and Begg (2005) emphasize the importance of the relationships over time. The meaning of the relationship and importantly of the model should be considered not only for present but also for the future during the conceptual database design phase. For example, if some relationship seems to be redundant for now, it might not be redundant in the future. Such relationship must not be removed and omitted.

### **Validation of Conceptual Data Model Against User Transaction**

The current conceptual data model represents data requirements of the business. In this stage the model is checked if it supports required user transactions. According to Connolly and Begg (2005), there are two approach to ensure the conceptual data model support user required transactions: 1. Describing the transaction; 2. Using transaction pathways. For the purpose of this thesis the first approach is chosen to validate required user transactions. Required user transactions relate to inserting, delating, updating, and querying. Basically, actions that are performed by user with the database system. Checking insertion, editing, and deleting is insignificant compared to checking necessary and required querying. There can be two kinds of inquiries required by user – one that can serve user practical information needed for their day to day work, second one that is needed for the future planning and development of the enterprise.

#### **3.3.1.4 Questions from Business and User Requirements**

Data model must be according to the user and business requirements. Thus, collected questions and requirements must be stored as part of the documentation which then serve as a key reference to ensure that the design follows user requirements.

#### **3.3.1.5 Review of Conceptual Data Model with User**

The last step of the conceptual database design is to review the model with users to get their feedback whether the created model reflects the data requirements of the enterprise. If the model is approved as a true representation of the data requirements of the enterprise, the next steps, such as logical or physical data model, can be taken. Otherwise the model must be changed and may require repeating some of the previous steps.

### **3.3.2 Physical Data Model**

The same CASE tool that was used for conceptual data model should be used for physical data model in order to keep the consistency across the models and in order to perform any changes or updates easily when needed.

#### **3.3.2.1 Inheritance Relationships**

Inheritance relationships between entities need to be solved before proceeding further to the creation of physical model. To solve this issue, it requires to look back on the conceptual model. A decision whether generate only the parent entity/superclass, subclasses, or combination of both classes needs to be made.

#### **3.3.2.2 Integrity Constraints**

Integrity constraints are a way that needs to be followed in order to avoid inaccurate and unwanted data to be inserted in the database. It is a way to provide accuracy and integrity for the data in the database. Connolly and Begg (2005) describe integrity constraints as ways of protecting the database from becoming incomplete, inconsistent, or inaccurate. One of the aspects of logical and physical design is integrity constraint. In this phase following integrity constraints are examined: entity integrity, attributes domain constraints, required data, and referential integrity.

## Entity Integrity

Entity constraint ensures the value of primary key cannot be null. Each entity must have an identifier that describes each row in the database table uniquely, it is done by identifying primary key for each entity.

## Attributes Domain Constraints

Attributes domain constraints are the set of valid values or range of values that an attribute can hold. A domain can have varied data types such as integer, string, character, time, date, etc.

## Referential Integrity

In simple definition, referential integrity defines the relationship between two entity types. A foreign key in the child table refers to the primary key in the parent table. Referential integrity ensures data accuracy with linking the data of two or more tables.

### 3.3.2.3 Required Data

The value for some attributes cannot be null, such attributes are limited with a constraint "NOT NULL".

### 3.3.2.4 Generation of Primary Key

A primary key can be created by SQL CREATE or ALTER commands, but there are two options to create either business key as primary key or generate a surrogate key as primary key. Each key has some advantages and disadvantages explained shortly as follows:

- **Business key:** A real representor of real entity and objects, for instance item serial number, employee birth certificate etc. It is easy to understand the meaning of it; it doesn't need to be interpreted. The downside of business key is slower index result because business key is usually alphanumeric, for instance, CD12011, cdb1245 etc. It is also possible it can be changed over time and this can cause difficulty to change the key across the tables.



- **Surrogate key:** Usually a small integer number, which require less memory space, faster and better performance, does not change over time, etc. The downside of it is, that it is meaningless integer number.

### **3.3.2.5 User Access Right**

For any organization in real live, users are assigned with specific role(s) within the organization. The same concept applies when users interact with the information system (IS). Assigning access right to users based on the user's role can be beneficial not only for better access to the information but also for security reason.

### **3.3.2.6 Review of Physical Data Model with User**

Just how the conceptual data model was validated against user required transactions in previous section, the physical data model is validated as well with similar method, to check if the model supports user required transactions.

### **3.3.3 Implementation in Database Management System**

Once the physical data model is reviewed and considered as correct and final, the database can be generated. It needs to be decided what DBMS and tool to use for the real implementation of the database – this option is selected in the CASE tool so that the generated SQL file will correspond to chosen DBMS. Once the implementation is done, it is the right time to find out whether all questions collected in business analysis phase find their answers – this can be done via querying the data using SQL.

## **3.4 Application**

Main benefit of the application for the future users is to increase the database security level, for instance to limit access to sensitive data. Another benefit is to provide end user secure access to the database as all users of the organization may not be skilled enough to interact with the database directly. Considering mentioned reasons, a graphical user interface application is required for users of the organization to work with shared database across the company.

“It should be recognized that the interface may be one of the most important components of the system. If it is easy to learn, simple to use, straightforward and forgiving, the users

will be inclined to make good use of what information is presented. On the other hand, if the interface has none of these characteristics, the system will undoubtedly cause problems.” (Connolly & Begg, 2005)

## **4 Practical Part**

The practical part focuses on the implementation of gained knowledge from the theoretical part. This part will demonstrate the principles described in the previous parts of the thesis on a case study focusing on design of a database for a promotional item shop.

### **4.1 Business Analysis**

For this bachelor thesis the business analysis is set in following way:

- Regular meeting with the owner of the promotional item shop twice a month.
- Regular meeting with users of the organization once per week.

The organization's main objective was analysed and discussed in detail, from owner's point of view as well as from the user's point of view. The business was analysed and all important data was covered and it was decided which data will be part of the information systems. Some data was missing in the first meeting and after reviewing the data suggestion upon covering the missing data was provided in the second meeting. For example, the information about customer and product was covered. After reviewing all data from this meeting covering the information about customer service was suggested to be part of the information systems too. Because storing the customer service data into information system can possibly help to have for instance statistical data about returned products, complained products etc. They can find out what was the most common reason for complained products, if the complaints happened before the expiration dates or within the product warranty. Knowing this information can help the organization to decide whether to produce that product more or increase the quality or to stop selling it.

#### **4.1.1 Interview**

The summary of each meeting and main important points was written and signed by all sides as part of the documentation. The written documentation was reviewed again on the next meeting in case if some problem occurred on the understanding the documents or expanding it. For understanding the business requirement in most significant way, the drawing and prototyping was used which helped everyone to have a clear image from the work in the same way.

### **4.1.2 Business Model**

The company sells promotional items to the customer. sold items are either purchased directly or bought through creating order. Typically, the items that are purchased directly are the items that customer do not wish to customize it, or the items has been already customized and the items that customers created order are items that requested to be customized. Thus, it is done through creating order because it requires recording details of which items and how to be customized. Ordered items are delivered to the customer once the customization is applied

The company sells promotional items to the customer, items are categorized as ready for purchasing and ready to be customized. For customizable items, company receives an order by a customer and then the customization/adjustment takes place as well as creating the order. Once both things are done, the order is delivered to the customer. Company buys ready products from its supplier and has many branches across the country. Each branch is run and managed by assigned manager and each branch has employees who work under mentioned branch manager. Customers can decide what adjustment they want to be applied on the items they choose – the items can be decorated, engraved or there can be even something printed on them for instance. Each adjustment is led by a certain employee who keeps an eye on the employee who performs the adjustment and makes sure everything goes smoothly. The leading employee is there to keep a track of all the adjustment that are created and takes the responsibility for each of them to his/her own shoulders.

### **4.1.3 Business Requirements**

Business requirements coming from the user interviews are stored as questions. User wants to know answers for these questions:

- What item is sold the most?
- What adjustment types take the most time and for what item type?
- What customer had the highest invoice amount?
- What are the most complained item types? And for what reason?
- How many invoiced items were adjusted? What is the share between invoiced items that were adjusted and invoiced items without adjustment?
- How much time it takes for each product to be adjusted?

## **4.2 Data Models**

For creating the data models, PowerDesigner CASE tool was chosen. It includes required data models and there is even the option to generate a database once the data models are finalized.

### **4.2.1 Conceptual Data Model**

As stated in the theoretical part of the thesis, the first step of conceptual data modelling is to start identifying the entities for the promotional item shop. More details are described in the next chapter.

#### **4.2.1.1 Identification of Entities**

For the staff, client, and users of the promotional item shop following entities were identified:

- Customer (Company or Individual(person))
- Purchase
- Supplier
- Item
- Item type
- Adjustment type (customization type)
- Adjustment (customization)
- Order
- Invoice
- Complaint
- Employee
- Branch

#### **Customer**

One of the most important entities of the conceptual data model is the entity customer, as the company mainly operate with customer to deliver goods and services. The enterprise sees the entity customer in two distinguishing ways, customer as individual (person) who orders or purchases and other type of customer as a company which also buys or purchases

the items, in other words they both fall under one super class customer. Using generalization concept, from two subclasses one super class customer is created. The entity customer is a super class and the class company and class individual (person) are the subclasses of it. The super class customer contains attributes that are same for both subclasses, while the subclasses contain only the attributes that are distinguishing from their parent class. Attributes of the subclass company: this class contains data about the customers who represent a company not individuals. Attributes of this class are Company Identification Number (CIN) and Tax Identification Number (TIN). Details of the attributes are shown in the **Table 1** below. N column in the table stands for nullability.

**Table 1 Company**

Attribute	Description	Data Type	N
CIN	The company identification number	INTEGER	No
TIN	Company tax identification number	VARCHAR (15)	No

Attributes of the subclass person (individual): The owner of the business is interested in recording date of birth of the customers so that they can use this information to send special offer to their customers. First name, surname, gender, and title of the customer are recorded as well. Details of the attributes are shown in the **Table 2** below.

**Table 2 Person**

Attribute	Description	Data Type	N
First Name	First name of all individuals	VARCHAR (50)	No
Surname	Last name of individuals (person)	VARCHAR (50)	No
Gender	Male/Female or unknown (M, F, U)	VARCHAR (1)	Yes
Title	Title of individual	VARCHAR (20)	Yes
Date of Birth	Customers date of birth if they agree to share	DATE	Yes

Attributes of the superclass customer are contacts details of the customers such as email, phone, address, etc. The primary key for this entity and all other entities will be generated. This kind of primary key is called surrogate key or artificial key which is a sequence of

unique integer number generated by SQL. The primary key for this entity is noted as Customer ID. Details of the attributes in the **Table 3** below.

**Table 3 Customer**

Attribute	Description	Data Type	N
Customer ID	It is generated and it is primary key	INTEGER	No
Telephone Number	Customer contact number in the international form (country code + number)	NUMBER (14)	Yes
Email	Customers email address	VARCHAR (50)	Yes
Country	Customer country	VARCHAR (50)	Yes
City	Customer city	VARCHAR (50)	Yes
Street	Customer address of current resident	VARCHAR (50)	Yes
Postal Code	Customer postal code of current resident	NUMBER (5)	Yes

## Order

Entity order is the second most important when it comes to the user interaction with the entity. The entity Order stores information about the date when customer make the order by phone call or at the store, because the user wants to know how long it takes for the processing of an order until that order is created, designed, or adjusted for delivery. The entity also stores the customer notes to the order, discount amount, and other ones. Details are shown in **Table 4**.

**Table 4 Order**

Attribute	Description	Data Type	N
Order ID	It is generated and it is primary key	INTEGER	No
Order Number	For each order a unique number is assigned	NUMBER (14)	No
Order Date	When the customer made the order	DATE	No

Discount	The discount percentage or special offer	DECIMAL (3,2)	Yes
Notes	Notes and comments of the customer	VARCHAR (255)	Yes
Statuses	Finished, pending, cancelled	VARCHAR (20)	No

## Invoice

The entity invoice is important for billing, it helps to have clear documentation of receiving and delivering goods. This entity is dependent on previous entity order, without order the invoice cannot be created which is called Identification Dependency in Relational Database Design (RDD). The entity stores information about delivery date, invoice date, amount due, etc. Details of the attributes are shown in the **Table 5** below.

**Table 5 Invoice**

Attribute	Description	Data Type	N
Invoice ID	It is generated and it is primary key	INTEGER	No
Invoice Number	Invoice number	NUMBER (10)	No
Invoice Date	When the bill is printed	DATE	No
Delivery Date	When the order is delivered	DATE	Yes
Amount Due	Total paid amount for the order	DECIMAL (20,4)	No

## Complaint

The entity complaint is dependent on previous entity invoice. Entity contains attributes such as complaint date, reason, etc. The attribute Type defines the complaint type whether the complaint is about repairing or returning. Based on the organization policy the customized order cannot be returned but can be repaired. Details are shown in **Table 6**.

**Table 6 Complaint**

Attribute	Description	Data Type	N
Complaint ID	It is generated and it is primary key	INTEGER	No
Complaint Date	Date when customer complained	DATE	No



Reason	Purpose of the Complaint	VARCHAR (30)	No
Notes	Notes and comments from customers	VARCHAR (100)	Yes
Type	1.Return (for none customize items only) 2. Repair	VARCHAR (6)	No
Status	Confirmed, in process, rejected	VARCHAR (20)	No

## Purchase

The entity purchase stores information only about the items that are ready to be sold. Items that are already designed and produced and don't need to be customized. Only items below certain price limit (e.g. 500 CZK), can be directly purchased without making an order. This entity holds attributes such as Purchase Id, Date, and Return Date. In attributes Return Date if it is filled, it means that the item was returned and if it is blank/null it means the item was only purchased. Details of the attributes are shown in the table below, **Table 7**.

**Table 7 Purchase**

Attribute	Description	Data Type	N
Purchase ID	It is generated and it is primary key	INTEGER	No
Receipt Number	Receipt number for each purchase	NUMBER (10)	No
Purchase Date	Date when customer made the purchase	DATE	No
Return Date	Date when customer returned the item	DATE	Yes

## Item

Entity item is a single unique unit. Items are sold as single units not by set. For example, each wallpaper is unique unit and is identified by its assigned serial number and a customer can buy for instance 10 wallpapers and each wallpaper will be recorded individually as a unit, not as a set of wallpapers. This unique item can be purchased with the condition as mentioned before or can be ordered. The entity Item contain only one attribute Serial Number, accepts Item ID which is a unique key, as mentioned before for all entities the surrogate key is assigned by SQL. Details for these two attributes are shown in the table below, **Table 8**.

**Table 8 Item**

Attribute	Description	Data Type	N
Item ID	It is generated and it is primary key	INTEGER	No
Item SN	Item serial number, used to identify each unit of an inventories	VARCHAR (30)	No

**Supplier**

Supplier supplies goods and products, some products which is customised in the shop by company employee and other products which is ready product for sell. Attributes of this entity consist of supplier contacts details. Details of the attributes are shown in the table below, Table 9.

Attribute	Description	Data Type	N
Supplier ID	It is generated and it is primary key	INTEGER	No
Telephone Number	Supplier contact number in international form (country code + number)	NUMBER (14)	No
Email	Email address of supplier	VARCHAR (50)	No
Country	Supplier country	VARCHAR (50)	No
City	Supplier city	VARCHAR (50)	No
Street	Street address of the supplier	VARCHAR (50)	No
Postal code	Postal code	Number (5)	No
Notes	Any notes regarding goods and products	Varchar (100)	Yes
Rate	Rating supplier 1 is worse and 5 is best	Integer	Yes

**Item Type**

Item type entity keeps track of organized datils of the items. Such as item name, selling price ( $\pm$  adjustment cost), buying price/cost, warranty, and customizability. Each item has different variety when it comes to customizability, some can be customized and sold or sold without customizing, depending on the customer order and some other item or final

product with no possibility to be customized. For instance, an item smart phone cover is in the shop, a customer makes an order to engrave the name of his/her favourite football league, logo of his/her company, famous quotes and so on. Or no customizable item such as already painted poster, wallpaper, bedroom light, 3D lamp/sleep lightning, digital clock because these items are either already customized or cannot be customized. Detail of the attributes is shown in the table below, **Table 9**.

**Table 9 Item Type**

Attribute	Description	Data Type	N
ItemType ID	It is generated and it is primary key	INTEGER	No
Name	Name of the item type (pen, cup, etc.)	VARCHAR (50)	No
Price	Selling price ( $\pm$ ) adjustment cost	DECIMAL (20,4)	No
Cost	Final cost of the product from the supplier	DECIMAL (20,4)	No
Warranty	1. Item with warranty, usually one or two years (1, 2) 2. Item without warranty (0)	INTEGER	Yes
Customizability	1. Customizable 2. Not customizable (yes, no)	VARCHAR (3)	No
Returnable	Yes or no	VARCHAR (3)	Yes

### **Adjustment (Customization)**

This entity is dependent in the entity Item. The entity stores information about how long it takes for an item to be adjusted. Each adjustment is defined as a unique unit the same as item which is defined as a unique unit. Details of the attributes are shown in **Table 10**.

**Table 10 Adjustment**

Attribute Name	Description	Data Type	N
Adjustment ID	It is generated and it is primary key	INTEGER	No
Start Date Time	Starting date and time of the adjustment	DATETIME	No

End Date Time	Ending date and time of the adjustment	DATETIME	Yes
---------------	--	----------	-----

### Adjustment Type (Customization Type)

The most important attribute type of this entity is the attribute adjustment type name, it keeps track of the data about what type of an adjustment is applied on each item like if it is engraved, painted, or printed. The entity also keeps information about the difficulty level of each adjustment like whether if it is easy, average, or tough. Depending on the adjustment difficulty level the adjustment cost will be counted, and this cost will be added to the item price. For the adjustment type level-one usually the cost will be more less the same and known while for the adjustment type level-two-three, the cost can be accurately counted only after the adjustment type is compiled. However, for all adjustment type the cost will be added to the item price. For the business model printing refer to level-one (easy) engraving refer to level-two (average) and painting refer to the level-three (tough). Detail of the attributes is shown in the table below, **Table 11**.

**Table 11 Adjustment Type**

Attribute	Description	Data Type	N
Adjustment Type ID	It is generated and it is primary key	INTEGER	No
Name	Type of an adjustment (engraved, painted, printed)	VARCHAR (20)	No
Difficulty	1. Easy 2. Average 3. Tough	VARCHAR (10)	No
Price	Price for an adjustment	DECIMAL (20,4)	No

### Employee

Almost for all business, employee is playing the key role in the organization structure, success, and performance and that makes this entity one of the key parts of the information system. Due to these facts keeping information about employee is essential for the organization. Entity employee keeps record of employee basic data as well as employee job role, contract type, contact details, starting date of the job, and so on. Each employee depending on their job role can create an item, manage the branch, design an item, customize an item, deliver the item or sell the item, etc. Recording some data is required

and some are optional such as employee title is optional, but employee job role is required because job role define the responsibility. Detail of the attributes are in **Table 12** below.

**Table 12 Employee**

Attribute	Description	Data Type	N
Employee ID	It is generated and it is primary key	INTEGER	No
First Name	Name of the employee	VARCHAR (50)	No
Surname	Last name of employee	VARCHAR (50)	No
Title	Employee title	VARCHAR (10)	Yes
Role	Job and responsibility of an employee	VARCHAR (50)	No
Contract Start Date	When the customer joining the company	DATE	No
Contract Type	Permanent contract, part time contract	VARCHAR (20)	No
Salary	Employee salary	DECIMAL (20,4)	No
Telephone Number	Employee contact number in the international form	NUMBER (14)	Yes
Email	Employee email address	VARCHAR (50)	No
Country	Employee country	VARCHAR (50)	No
City	Employee city	VARCHAR (50)	No
Street Address	Employee address of current resident	VARCHAR (50)	Yes
Postal Code	Employee postal code of current resident	NUMBER (5)	Yes
Date of Birth	Employee date of birth	DATE	No
Gender	Male, Female, Unknown (M, F, U)	VARCHAR (1)	No

## Branch

Each branch of the organization manages the employee which falls under the branch area of operating. Based on branch location employee of each branch operates under the assigned branch management system. Details of the attributes are shown in **Table 13**.

**Table 13 Branch**

Attribute	Description	Data Type	N
Branch ID	It is generated and it is primary key	INTEGER	No
Name	Name of the branch	VARCHAR (20)	No
Country	Country where the branch is located	VARCHAR (20)	No
City	City where the branch is located	VARCHAR (20)	No
Postal Code	The postal code of the branch	VARCHAR (5)	No

#### 4.2.1.2 Identification of Relationships

In the following chapters the relationships are identified among entities in details such as the cardinality (C) of the relationships and participation constraints (P). Cardinality is described as the relationship of one table to another table in database. Such as if it is one-to-one, many-to-many, or one-to-many. Participation constraints describe the existing dependency of one entity related to another entity. For instance, a customer can make purchases, but a purchase cannot exist without a customer. There are two types of participation called partial and total partial. From the given example above the relationship from customer side it is partial and from purchase it is total partial.

#### Customer - Purchase

There is one relationship between entity customer and entity purchase denoted by a word Requires. A customer requires many purchases however a customer doesn't have to requires any purchases. A purchase cannot be required without a customer, in another word a purchase is required by one and only one customer. Details of the attributes are shown in **Table 14** below.

**Table 14 Customer - Purchase**

Relationship Name	Entity	Relationship	P/C	Entity
Require	Customer	Requires	0...N	Purchase
	Purchase	Is required	1...1	Customer

## Customer – Order

A customer can create many orders however a customer doesn't need to create any order. In the database there can be customer with zero order but for an order there must be always one and only one customer. An order cannot exist without a customer. Details of the relationships are shown below, **Table 15**.

**Table 15 Customer - Order**

Relationship Name	Entity	Relationship	P/C	Entity
Create	Customer	Creates	0...N	Order
	Order	Is created	1...1	Customer

## Order – Invoice

An order is changed to an invoice and an invoice is made for an order. There is one-to-one relationship between order and invoice. The entity invoice cannot exist without order however, there can be order without being changed to an invoice. Details of the relationships is shown below, **Table 16**.

**Table 16 Order - Invoice**

Relationship Name	Entity	Relationship	P/K	Entity
Change	Order	Changes to	1...1	Invoice
	Invoice	Is changed from	1...1	Order

## Order – Item

An order includes one or many items, one order must include at least one item. An item belongs to zero or one order. In another word there can be items not assigned to any order. Each item is a unit with a unique serial number assigned to each of them that make them unique. Details of the relationships shown below, **Table 17**.

**Table 17 Order - Item**

Relationship Name	Entity	Relationship	P/K	Entity
Include	Order	Includes	1...N	Item

	Item	Is included	0...1	Order
--	------	-------------	-------	-------

### Invoice – Complaint

A customer with an invoice can assign many complains or zero complain. A complaint is assigned to one and only one invoice. A complaint cannot exist without an invoice. Details in **Table 18**.

**Table 18 Invoice - Complaint**

Relationship Name	Entity	Relationship	P/K	Entity
Assign	Invoice	Has assigned	0...N	Complaint
	Complaint	Is assigned	1...1	Invoice

### Invoice – Employee

An invoice is handled by one and only one employee. An employee can handle many or zero invoice. An invoice cannot exist or cannot be handled without any employee. Details in **Table 19**.

**Table 19 Invoice - Employee**

Relationship Name	Entity	Relationship	P/K	Entity
Handle	Invoice	Is handled by	1...1	Employee
	Employee	Handles	0...N	Invoice

### Branch – Employee

There are two relationships between branch entity and employee entity. One represents the information about what employee manages the branch and the second one gives the information about what employees work at the branch. Each branch can be managed by one and only one employee, and an employee can manage many branches, but does not need to manage any. An employee can work at one and only one branch, and the branch can have work from many employees, but does not need to have assigned any, e. g. when a new branch is being established. Details in **Table 20**.



**Table 20 Branch – Employee**

Relationship Name	Entity	Relationship	P/K	Entity
Manage	Branch	Is managed	1...1	Employee
	Employee	Manages	0...N	Branch
Work	Branch	Has work from	0...N	Employee
	Employee	Works at	1...1	Branch

### **Employee – Adjustment**

There are two relationships also between employee entity and adjustment entity. First one is there to store information about who leads (is responsible for) the adjustment. An employee can lead many adjustments but does not need to lead any. Each adjustment is led by one and only employee. The second relationship provides information about who performed the adjustment. An employee can perform many adjustments but does not need to perform any. Each adjustment is performed only by one employee. Details of the relationships are shown below, **Table 21**.

**Table 21 Employee - Adjustment**

Relationship Name	Entity	Relationship	P/K	Entity
Lead	Employee	Leads	0...N	Adjustment
	Adjustment	Is leaded by	1...1	Employee
Perform	Employee	Performs	0...N	Adjustment
	Adjustment	Is performed by	1...1	Employee

### **Item – Purchase**

An item can be purchased only once but does not need to be purchased at all. A purchase must consist from at least one item but can be consist also from many items. Details of the are shown below **Table 22**.

**Table 22 Item - Purchase**

Relationship Name	Entity	Relationship	P/K	Entity
Consist	Item	Is consisted	0...1	Purchase
	Purchase	Consists of	1...N	Item

**Item – Supplier**

Item can exist without being supplied, for example when the shop has their own items. However, a supplier supplies many items but does not have to supply any item. An item is supplied by one or zero suppliers. Details of the relationships are shown below **Table 23**.

**Table 23 Item - Supplier**

Relationship Name	Entity	Relationship	P/K	Entity
Supply	Item	Is supplied	0...1	Supplier
	Supplier	Supplies	0...N	Item

**Item – Item Type**

Each item is defined by its type. An item is defined by one and only one item type. An item type defines many items or zero item. For instance, smartphone cover is an item type that can record zero or many smartphones covers. The relationships between item and item type is one-to-one only. Details in **Table 24**.

**Table 24 Item - Item Type**

Relationship Name	Entity	Relationship	P/K	Entity
Define	Item	Is defined	1...1	Item Type
	Item Type	Defines	0...N	Item

**Item – Adjustment**

There can be item without being adjusted or customized. However, an item is adjusted zero or many times. An adjustment adjusts one and only one item. An adjustment cannot be made if an item does not exist. Details in **Table 25**.

**Table 25 Item - Adjustment**

Relationship Name	Entity	Relationship	P/K	Entity
Adjust	Item	Is adjusted	0...N	Adjustment
	Adjustment	Adjusts	1...1	Item

### **Adjustment – Adjustment Type**

An adjustment is described by one adjustment type only. An adjustment type describes many adjustments or zero adjustment. Details in **Table 26**.

**Table 26 Adjustment - Adjustment Type**

Relationship Name	Entity	Relationship	P/K	Entity
Describe	Adjustment	Is described	1...1	Adjustment Type
	Adjustment Type	Describes	0...N	Adjustment

### **4.2.1.3 Control of Model for Redundancy**

#### **Re-Examination of one-to-one Relationships**

For the purpose of this thesis there is only one relationship which is one-to-one. Order – Invoice: each one of this entity serves a different purpose, different scenario, and different information thus, cannot be merges to one entity. For example, entity order records data about order date, order status, customer notes to the order etc. All this information is different from the invoice entity. Invoice entity keeps information about delivery date, invoice date and also holds an important relationship with entity complaint, which the relationship cannot be removed and cannot be assigned to order because if an order is not completed, or it is cancelled, customer cannot complain to it or it is cancelled order which is not processed further.

#### **Removal of Redundant Relationships**

In the proposed model all relationships between entities are different thus no relationship is removed from the model.

## Consideration of Time Dimension

For the purpose of this thesis the model and relationships are checked against time dimension. And no such relationship was added or deleted that would change its meaning over time. All relationships are defined carefully and all of them have different meaning.

## Validation of Conceptual Data Model Against User Transaction

The current conceptual data model represents data requirements of the business. In this stage the model is checked if it supports required user transactions. In the following pages the user required transaction related to queries are illustrated only as the complete list would be to large.

### 4.2.1.4 Questions from Business and User Requirements

- What item is sold the most?

Details of all items are held in the item type entity and details of all items which are sold are held in the order and purchase entities. Using the *order includes item* relationship and *purchase consists of item* relationship combined with *item is defined by item type* relationship, users can get the information about which item is sold the most.

- What adjustment types take the most time and for what item type?

Entity adjustment type contains details of the adjustment type and adjustment duration and time is stored in entity adjustment. Details of items are stored an entity item type and entity item. using the *item is defined by item type* relationship and *item is adjusted* relationship combined with *adjustment is described by adjustment type* relationships, user can get the information about adjustment type duration of specific item.

- What customer had the highest invoice amount?

Details of all customer are held in entity customer and details of order and invoice are stored in the entity order and invoice. using the *customer create order* relationship and *order is changed to invoice* relationship user can find out which customer had the highest invoice amount.

- What are the most complained item types? And for what reason?

Entity complaint held detailed information about the complaint, and to get the item type details following entities should be connected: entity invoice, order, item, and item type. using complaint *is assigned to invoice* relationship combined with invoice *is changed from order*, order *includes item*, and item *is defined by item type* relationships user can get the information which item type were complained the most and for what reason.

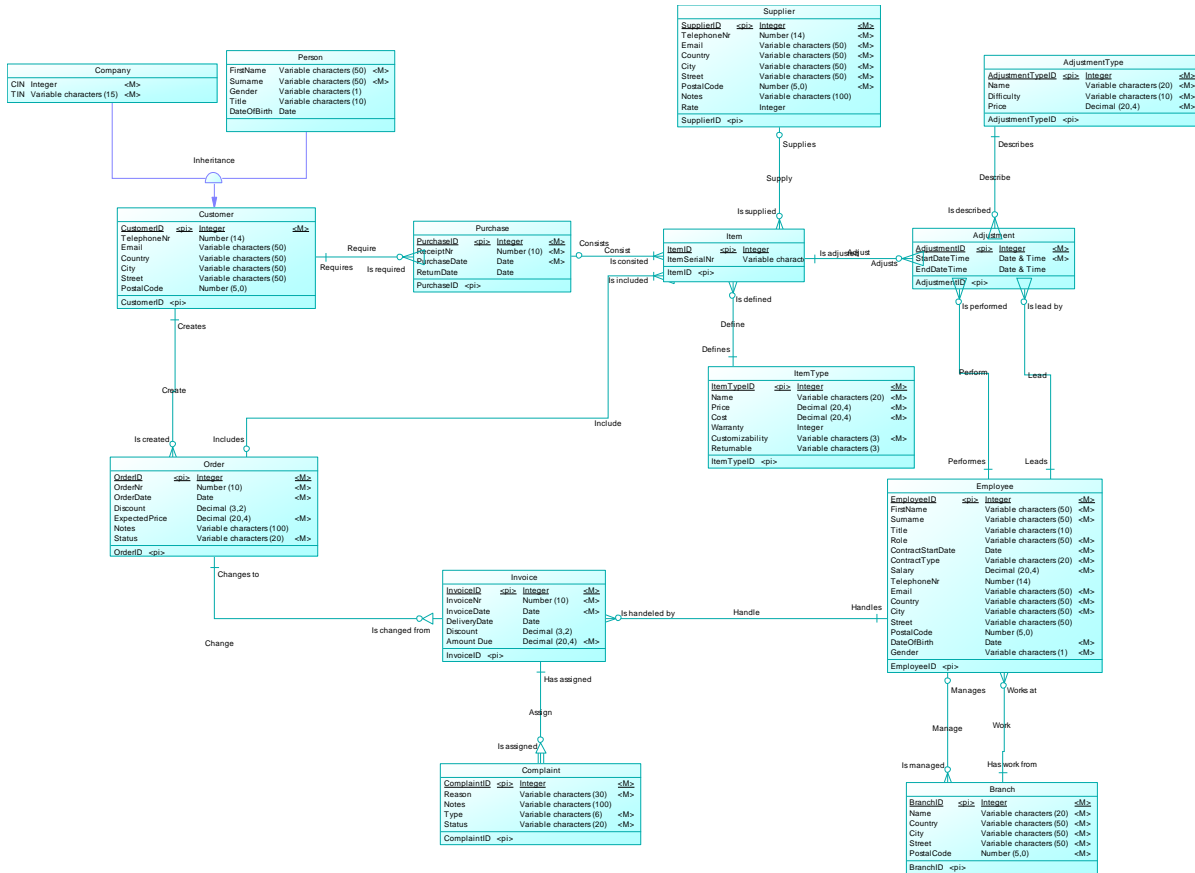
- How many invoiced items were adjusted? What is the share between invoiced items that were adjusted and invoiced items without adjustment?

Details of the adjustment are held in the entity adjustment. And details of invoiced items are in the entity item. Using the *item is adjusted by adjustment* relationships user can retrieve the answer of what is the share between invoiced being adjusted and invoice without being adjusted.

#### **4.2.1.5 Review of Conceptual Data Model with User**

The created model was presented and explained to the users of the promotional item shop with all details. The output of the design model was based on the interview with the users. Users of the company consider that the created model is true representation of reality, including all essential aspects. Therefore, the conceptual design is accepted to be correct and this phase is completed.

### 4.2.1.6 Conceptual Data Model Diagram



Picture 1 Conceptual Diagram

### 4.2.2 Physical Data Model

Physical data model is created using PowerDesigner CASE tool, as mentioned before and as it was used for the conceptual data model. While using this tool, all required steps towards creations of the logical model can be combined in the physical model. So theoretically the logical design phase may seem to be skipped, but in practice it is not, because all necessary aspects which must be followed in logical model, are already part of the physical model. So therefore, following sections describe aspects of physical design. The physical implementation will be performed in Oracle DBMS. As stated in the theoretical part of the thesis, the first step is to decide about the inheritance relationships for the item shop. More details are described in the next chapter.

### 4.2.2.1 Inheritance Relationships

For the conceptual model of this thesis, the superclass customer (parent entity) is generated (because there is no relationship directly from any child class to other entity in the model). Parent entity customer has two child entities such as person and company. These entities are merged into their parent entity customer as attributes.

### 4.2.2.2 Integrity Constraints

#### Entity Integrity

As mentioned before, for the work of this thesis, the primary keys for all entities are generated by using SQL. The term primary key was defined in the theoretical part of the thesis. These constraints can be assigned using DDL (data definition language) commands such as CREATE and ALTER.

There is the nullability shown in bold as an example of the primary key with the constraint (not null) for the entity supplier below.

```
create table Supplier
(
  SupplierID           INTEGER           not null,
  TelephoneNr          NUMBER (14)           not null,
  Email                VARCHAR2 (50)        not null,
  Country              VARCHAR2 (50)        not null,
  City                 VARCHAR2 (50)        not null,
  Street               VARCHAR2 (50)        not null,
  PostalCode           NUMBER (5,0)         not null,
  Notes                VARCHAR2 (100),
  Rate                 INTEGER,
  constraint PK_SUPPLIER primary key (SupplierID)
);
```

#### Attributes Domain Constraints

Attributes domain constraints are the set of valid values or range of values that an attribute can hold. A domain can have varied data type such as integer, string, character, time, date, etc. for example, an employee gender in the table Employee is either M, F, or U, so the domain of attributes gender is a single value of string character M, F, or U. These constraints were identified during the identification of the entities in section above in advance. These constraints can be ensured using the CHECK clause or the CREATE DOMAIN command. Below the domain constraints shown in bold as an example for the entity Employee.

```

create table Employee
(
  EmployeeID          INTEGER          not null,
  BranchID            INTEGER          not null,
  FirstName            VARCHAR2(50)    not null,
  Surname              VARCHAR2(50)    not null,
  Title                VARCHAR2(10),
  Role                 VARCHAR2(50)    not null,
  ContractStartDate   DATE             not null,
  ContractType        VARCHAR2(20)    not null,
  Salary              NUMBER(20,4)    not null,
  TelephoneNr         NUMBER(14),
  Email                VARCHAR2(50)    not null,
  Country              VARCHAR2(50)    not null,
  City                 VARCHAR2(50)    not null,
  Street               VARCHAR2(50),
  PostalCode           NUMBER(5,0),
  DateOfBirth         DATE             not null,
  Gender               VARCHAR2(1)     not null,

  constraint PK_EMPLOYEE primary key (EmployeeID),
  constraint check_Gender check (Gender IN ('M','F','U'));
);

```

## Referential Integrity

For example, table Invoice has a foreign key (EmployeeID) that points to the table employee primary key (EmployeeID), if the foreign key in table invoice contains a value, that value must refer to the table employee (primary key). These constraints can be assigned using the DDL commands CREATE and ALTER. Below the referential integrity is shown as an example for the entity invoice which holds a foreign key that refers to the entity employee primary key (EmployeeID).

```

alter table Invoice
  add constraint FK_INVOICE_HANDLE_EMPLOYEE foreign key (EmployeeID)
  references Employee (EmployeeID)
;

```

An example how to ensure the referential integrity on insert, delete, and update depending on parent table or child (invoice) is illustrated as following:

- **To Insert a record on child table:** E. g. to insert records on invoice, check the value for the foreign key attributes (Employee D) is set to null or to a value corresponding to an existing employee table.
- **To Update a record in child table:** Update is the same as insert.
- **To Delete a record form child table:** In this case of deleting a record in child table, referential integrity will not be affected.
- **To delete/update in case of parent table:** If a record is deleted from a parent table, which exists as a foreign key in the child table, referential integrity is lost. In such case three things can be done, CASCADE, set null, and restrict.



**CASCADE:** On delete from parent table will delete the record from the child, the same happens with an update. An example is shown below for the table Order.

```
alter table Order
add constraint FK_ORDER_CREATE_CUSTOMER foreign key (CustomerID)
references Customer (CustomerID) on delete cascade;
```

**SET NULL:** When the record is deleted from parent table, the foreign key in corresponding child table will be set to null. If that attribute allows null value.

**RESTRICT:** Nothing will happen, this option is default.

### 4.2.2.3 Required Data

For example, a complaint type must be declared and filled (such as repair or return). The decision whether which attributes can have null value or cannot was assigned to all entities in the sections above when identifying entities noted with later N (shortcut for nullability). These constraints can be assigned using DDL (data definition language) commands, CREATE and ALTER. Below the nullability shown in bold as an example for the entity complaint.

```
create table Complaint
(
  OrderID          INTEGER          not null,
  InvoiceID        INTEGER          not null,
  ComplaintID     INTEGER          not null,
  Reason          VARCHAR2(30)     not null,
  Notes          VARCHAR2(100),
  Type           VARCHAR2(6)       not null,
  Status         VARCHAR2(20)     not null,

  constraint PK_COMPLAINT primary key (OrderID, InvoiceID, ComplaintID)
);
```

### 4.2.2.4 Generation of Primary Key

After comparing the advantages and disadvantages of surrogate keys and business keys, as described in the theoretical part, the surrogate key is chosen as a substitute for the business key. As mentioned before, for the data model of this thesis for all entities the surrogate is generated as a primary key. Surrogate is generated automatically by using trigger and a sequence. An example is shown below for the table order.

```
Create sequence OrderID_SEQ;

Create or replace trigger OrderID_TRG
Before insert on Order
For each row
Begin
Select OrderID_SEQ.nextval into :new.OrderID from dual;
End;
```

#### 4.2.2.5 User Access Right

Employees of the promotional item shop need access to the IS to perform their job, since each employee with same role or different role needs the same or different information on different level of IS, it is suggested to defined the user role first then assign them access right corresponding to their job role. For the user role, three main users are identified for the promotional item shop IS (database administrative, manager, employee). More details information is provided below, **Table 27**.

**Table 27 User Role and Access Right**

User Role	Access Right	Table
Database Administrator (DBA)	All access rights at administrative level	All tables
Manager	Select, Insert, Update, Delete	All tables
Employee	Select, Insert, Update	All tables except table employee

#### 4.2.2.6 Review of Physical Data Model with User

The same way of review as it was used for the conceptual data model, was taken also for the physical model. Users confirmed that presented diagram is correct. Therefore, this modelling was considered as finished.

#### 4.2.2.7 Physical Data Model Diagram

The final physical model is shown below.



- What adjustment types take the most time and for what item type?

```
select
ad.name as adjustment_type_name,
it.name as item_type_name,
a.enddatetime - a.startdatetime as duration
from itemtype it
join item i on it.itemtypeid = i.itemtypeid
join adjustment a on i.itemid = a.itemid
join adjustmenttype ad on ad.adjustmenttypeid = a.adjustmenttypeid
order by a.enddatetime - a.startdatetime desc
```

- What customer had the highest invoice amount?

```
select
c.firstname as first_name,
c.surname,
max( i.amountdue)
from customer c
join order o on c.customerid = o.customerid
join invoice i on o.orderid = i.orderid
group by c.firstname, c.surname
order by max(i.amountdue) desc
```

- What are the most complained item types? And for what reason?

```
Select
co.reason as complaint_reason,
itp.name as item_type_name,
co.invoiceid
from complaint co
left join invoice i on co.invoiceid=i.invoiceid
left join order o on i.orderid=o.orderid
left join item it on o.orderid=it.orderid
left join itemtype itp on itp.itemtypeid=it.itemtypeid
where co.status='Confirmed'
group by co.reason, itp.name, co.invoiceid
order by co.invoiceid
```

- How many invoiced items were adjusted? What is the share between invoiced items that were adjusted and invoiced items without adjustment?

```
select
count(distinct i.itemid) as sold_items,
count(a.itemid) as adjusted_items,
(count(a.itemid) / count(distinct i.itemid)) * 100 as share
from item i
left join adjustment a on i.itemid = a.itemid
join order o on o.orderid = i.orderid and o.status = 'Finished'
```

### 4.3 Application

Testing of the database was successful and future users are provided with graphical user interface application to retrieve, insert, update, and interact with the database smoothly and securely. The application for the database future users is created using Microsoft Access Form. Users can see permitted data, and can apply few necessary functions that are provided such as insert new records, update the records, search, save, etc. For the fields that accepts only specified value the option is listed for the user so that user can insert the correct data from allowed lists, for instance in the field gender the value can be male,

female or unknown so the list of the value is shown for the users to make the work easier and also ensure correct entry of the data. For the other fields if users try to insert wrong values error message that contains the error details will display.

**Picture 3 Application Form for Employee**

**Picture 4 Application Form for Customer**

End users must not see the salary of the employees because it is sensitive data, thus the salary column is not shown as public in the application. For both figures shown above customer id and employee id is not provided in the application as the id is set to be generated automatically. The design of the application can be improved further but the performance satisfied the users.

## 5 Results and Discussion

The created database was reviewed with the user in each phase of the database design process. Thus, ensured that the user requirements were captured in each phase of the database design before continuing to the next phases. Short summary of the phases:

- Creating business analysis
- Creating data models
- Implementing database application

For the first phase the review was done through reviewing the result of each interview with the user. The overall result was processed for the next step of the design. In the second phase, the conceptual and physical data models were explained and reviewed with user. After the implementation of the database in the DBMS, the user questions were tested by creating SQL queries. For the last phase, the application was reviewed with the user. The application was explained to the user and user tested the application.

Since each phase was reviewed subsequently with the users in sections above, it is not needed to be reviewed again one by one in details, but instead in this stage, the whole phases are reviewed with the users as an overall review to see whether the created database fulfils current needs of the enterprise. The overall review was conducted by list of questions from five users of the enterprise to evaluate the proposed database. They evaluated its functionality, consideration of future development, and added their user comments.

Provided questionnaire is illustrated below:

- Does the database fulfil the needs of the promotional item shop? Users answered that the database covers all requirements of the business they need from the database.
- Does the database make the users work easier to look back for the data? Users answered that it seems like their work will become faster and easier with the database. One user commented that creating reports annually, quarterly, or even weekly, will become easier with all the data integrated.
- Having company data integrated in the database system helped company usage of the data efficiently? Users answered that their work with the

company data will become more efficient and the database increased their works accuracy level.

- Does the database effect the customer satisfactions? Users answered that customer are more satisfied after the launch of the database. User commented that the customer orders process and tracking become more accurate.
- Do you like the database performance and what would you suggest for future developments? The users answered that the performance and functionality of database is user-friendly and fulfils their currents requirements. One user graded the application that enables end-user to interact with the database via graphical user interface without writing code.

## 6 Conclusion

This thesis was devoted to a database design for a promotional item shop. The main aim of the thesis was to create a complex database solution and to show how a database can help even a small company to have a quick access to their data, to use their data more effectively and to increase overall performance and productivity. There were four main objectives set for the thesis:

- To analyse and gather business requirements
- To create a data model and implement it in DBMS
- To implement a data-driven application to facilitate access to data for users
- To asses if the resulting solution fulfils the need of the business

For the first objective, the identification and analysis of the business was concluded by interviewing the users and as a result all required information and requirements were collected into a form of business model and business requirements. These outputs then served as a knowledge base for all the phases afterwards. In the second phase both, conceptual and physical model, were created and each one of them was reviewed with the user at the end of its creation. Thus, ensured that the business requirements were maintained through the whole designing process. The output of the phase was implemented in selected DBMS and tested whether even after the implementation, all requirements were reached. The results proofed that the analysis and related data models were correct. In the third phase, an application interface was created for the future users. According to their feedback, the application met their expectations. The final objective was concluded with an overall review of the database design with the users. The result indicated that proposed database will improve and enrich their business in various aspects mentioned before. Based on the overall review and feedback from the users, it can be summarized that the database helped the information flow in the organization, made the access to the data quicker, increased the efficiency of the usage of company data, and significantly impacted overall performance and productivity of the company. Database design was then considered as successfully finished and from user's perspective there was no suggestion for any improvement for the current solution.



## 7 References

- Churcher, C., 2012. *Beginning Database Design: From Novice to Professional*. 2 ed. New York: Apress.
- Churcher, C., 2017. *Beginning SQL Queries*. 2 ed. New York: Apress.
- Connolly, T. M. & Begg, C. E., 2005. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 4th ed. Boston: Addison-Wesley.
- Coronel, C. & Morris, S., 2016. *Database Systems: Design, Implementation, and Management*. 12 ed. Boston: Cengage Learning.
- Elmasri, R. & Navathe, S. B., 2015. *Fundamentals of Database Systems*. 7 ed. London: Pearson Education.
- Faroult, S. & Robson, P., 2006. *The Art of SQL*. 1 ed. Sebastopol: O'Reilly Media.
- Harrington, J. L., 2009. *Relational Database Design and Implementation: Clearly Explained*. 3 ed. Burlington: Morgan Kaufmann.
- Hernandez, M. J., 2013. *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design*. 3 ed. Boston: Addison-Wesley.
- Hogan, R., 2018. *A Practical Guide to Database Design*. 2 ed. Boca Raton: CRC Press.
- Molinaro, A., 2005. *SQL Cookbook*. 1 ed. Sebastopol: O'Reilly Media.
- Nield, T., 2016. *Getting Started with SQL: A Hands-On Approach for Beginners*. 1 ed. Sebastopol: O'Reilly Media.
- Singh, S. K., 2009. *Database Systems: Concepts, Design and Applications*. 3rd ed. New Delhi: Dorling Kindersley.