



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

BACHELOR THESIS

Denis Drobný

**Extracting Information from Database
Modeling Tools**

Department of Distributed and Dependable Systems

Supervisor of the bachelor thesis: RNDr. Pavel Parízek, Ph.D.

Study programme: Computer Science

Study branch: Programming and Software Systems

Prague 2019

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date

signature of the author

Dedication.

Title: Extracting Information from Database Modeling Tools

Author: Denis Drobny

Department: Department of Distributed and Dependable Systems

Supervisor: RNDr. Pavel Parízek, Ph.D., Department of Distributed and Dependable Systems

Abstract: Abstract.

Keywords: key words

Contents

1	Introduction	3
1.1	Goals	5
1.2	Overview of Chapters	5
2	Databases	6
3	Data Models and Abstraction Layers	7
3.1	Data Modeling	7
3.2	Abstraction Layer Types	7
3.2.1	Physical Data Model	7
3.2.2	Logical Data Model	7
3.2.3	Conceptual Data Model	7
3.3	Relations Between the Layers	7
3.3.1	Maps-to Relation	7
3.4	Possible Construction of a Model	7
3.4.1	Modeling	7
3.4.2	Reverse-Engineering	7
3.4.3	Generating	7
3.4.4	Importing	7
4	Modeling Tools	8
4.1	ER/Studio	8
4.2	PowerDesigner	8
5	Data Lineage	9
5.1	Theory	9
5.2	Manta Flow	9
5.2.1	Supported Database Technologies	9
6	Analysis & Design of the Solution	10
6.1	Requirements	10
6.2	Analysis of the Problem	10
6.3	Architecture of the System	10
7	Implementation	11
7.1	Extensibility	11
7.2	Technologies	11
7.3	Testing	11
	Conclusion	12
	Bibliography	13
	List of Figures	14
	List of Tables	15

List of Abbreviations	16
A Attachments	17
A.1 Building	17
A.2 User Documentation	17
A.2.1 Tutorials	17
A.3 Cooperation with Manta Flow	17

1. Introduction

A *Database* is a collection of related data. By data, we mean known facts that can be computerized and that have implicit meaning as stated in Fundamentals of Database Systems [1]. We will consider that a database stores data relevant to an enterprise at a host that can be accessed via network. Databases became deep-rooted in every business. Independently of the field that a company is focused on we can enumerate many reasons for considering a database storage deployment a good idea. Let us show some examples of how databases are used through various business domains.

- Social Media

Every piece of information that has ever been published on social media, from photo through like or comment to friendship establishment, was stored somewhere and that place is a database. Usually the database that a social platform uses does its job in a background. Nevertheless there may occur events when the data storage reminds of its presence as it did on the most recent outage of Facebook. [2].

- Healthcare

Easy accessibility of large amount of patient's data is a main reason to deploy a database at doctor's office or a healthcare organization [3]. High discretion is a requirement when managing data of such sensitiveness.

- Finances

complete

- E-commerce

Every company that sells products online should use a database. The bare minimum is to store offered products themselves and keeping track of purchases that were done by users.

And the list goes on.

A *Data Model* is a description of data, data relationships, data semantics, and consistency constraints.

A *Database Model* is a particular type of a data model that determines in what manner can be data in database organized, stored, represented and accessed in a database.

There are multiple kinds of specific database models known. According to Fundamentals of Database Systems [4] the main categorization of database models nowadays, according to what do they describe and on what type of user are aimed, is following:

- Conceptual Data Models (High-Level)

Reproduces real world objects along with their relationships and should be close to how business end-users perceive them.

- Representational Data Models (Implementation)

In the middle between the two other model types there are representational data models which on the one hand are comprehensible by end-users and on the other hand are not too abstract so that they can be used as documentation for an actual database implementation of the modeled data.

- Physical Level Data Models (Low-Level)

In contrast to conceptual models the physical ones are tied with how data are stored physically at storage media showing all specific internal details that may be overwhelming in the case that the reader is a computer specialist.

Once the decision is made and the usefulness of a database for our thing is proved there may be still a long way until everything runs as expected and we can use all the advantages that a data storage brings. The database design phase comes in place then. By the nature of the problem, a top-down approach is followed. A database designer comes in handy for the process.

The first thing that is needed to do is to discuss with an expert in the domain **what domain** in order to identify and collect requirements for the designed system.

The debate should be translated to a conceptual model by the designer. **that is an instance of what data model**

Once the abstract model is created a movement towards its implementation is desired. That is when the development of a logical model takes place and the high level view is turned into system-specific model that is the logical one.

Finally, the organization of the database itself is figured out and captured in a physical model of the analyzed system.

Given the physical model a database can be deployed straightforwardly.

A *Database Schema* defines how is the database described in a data model constructed, represents an instance of a data model.

A *Diagram* is a graphical visualization of a data model.

A *Data Modeling Tool* is a software that allows a database designer to create data models. End user may use the tools for interactive previewing of the models' diagrams.

Data Lineage provides a picture of how data moves in some system. It is a description of how data go from an origin through their transformations until they reach a destination. The ability of seeing graphically how data are used, what for, and what are the consequences of the usage in a system is a powerful tool for error tracing.

The process of development and deployment of a database consists of multiple

stages. At the beginning there is a high level view of why the database is needed and what purpose will it serve. Hopefully in some time the result is that the data described in the initial step are stored physically at some server. This way the data can be accessed and processed. What we want to achieve in this work is to make use of the individual steps taken during the design process, and make operations on data as transparent and traceable as possible even for business users that don't have a technical background.

1.1 Goals

- Develop a component that extracts metadata from database models that were created using SAP PowerDesigner
 - Develop a component that extracts metadata from database models that were created using ER/Studio
- complete the list

1.2 Overview of Chapters

Introduction to each of the following chapters

2. Databases

3. Data Models and Abstraction Layers

3.1 Data Modeling

3.2 Abstraction Layer Types

mention the types from introduction

move to the chapter about databases in comparison with the historical proposition In 1975 American National Standards Institute [5] introduced its database architecture consisting of three layers:

- External Level
- Conceptual Level
- Physical Level

3.2.1 Physical Data Model

3.2.2 Logical Data Model

3.2.3 Conceptual Data Model

3.3 Relations Between the Layers

3.3.1 Maps-to Relation

3.4 Possible Construction of a Model

3.4.1 Modeling

Each layer

3.4.2 Reverse-Engineering

Physical layer

3.4.3 Generating

Downwards

3.4.4 Importing

Each layer

4. Modeling Tools

4.1 ER/Studio

4.2 PowerDesigner

5. Data Lineage

5.1 Theory

5.2 Manta Flow

5.2.1 Supported Database Technologies

6. Analysis & Design of the Solution

6.1 Requirements

6.2 Analysis of the Problem

6.3 Architecture of the System

7. Implementation

7.1 Extensibility

Description of the common structure of the solution - what to do when a programmer wants to write a connector for another modeling tool.

7.2 Technologies

7.3 Testing

Conclusion

Bibliography

- [1] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems (7th Edition)*. Pearson, 2015.
- [2] Facebook blames 'database overload' for most severe outage in its history as users continue to report problems with Instagram and WhatsApp more than 14 hours after global meltdown.
- [3] HealthCare.gov's Heart Beats For NoSQL.
- [4] Abraham Silberschatz Professor, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. McGraw-Hill Education, 2010.
- [5] SPARC-DBMS Study Group. Interim Report: ANSI/x3/SPARC Study Group on Data Base Management Systems, 1975.

List of Figures

List of Tables

List of Abbreviations

A. Attachments

A.1 Building

A.2 User Documentation

A.2.1 Tutorials

A.3 Cooperation with Manta Flow