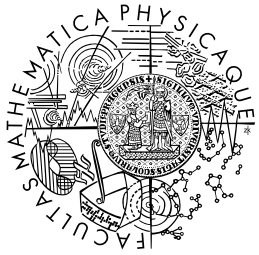

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

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 Drobný

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

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. Structure of database vs modeling tools

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 in use, 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 Flow is an internal logic of how data moves in some system across its components.

Data Lineage provides a picture of how data moves in some system across its components. 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 as we have seen. 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
- Provide a description by means of a programming language for a general scenario of metadata extraction from a data modeling tool output and passing the information to a data lineage tool
- Propagate data lineage acquired by analysis of how is database used and constructed to more abstract data models than is the physical one, to the logical and the conceptual models.

Introduction to each of the following chapters once the final organization is known

2. Databases

A standalone database is not very useful. To take the full advantage of it we need some means to define, create, maintain and control access to the database. That is purpose of a software called *Database Management System (DBMS)*.

2.1 Database Model Types

We already described why we want to use a database and roughly mentioned what are the pieces of data that we want to save there. Now let's take a look at what are differences between in database implementations and what to take in account when comparing database technologies. That may be helpful when choosing the best suitable option for some specific data set to store or to see how storing of great amount of structured information can be approached.

The basic division of database model types is simple and binary - they are either Relational or Non-Relational.

There are Database Management Systems build around both, Relational Database Management System (RDBMS).

BASE, ACID

2.1.1 Relational

A *Relational Database* is a set of tables. A table consists of rows (or records) and columns. We can see such table as an object whose attributes are represented by columns and instances by rows. The important thing is that relational tables tables carry both data that need to be stored by user and the relationships between the data as well. To store an atomic piece of data about instance a proper column is filled with a value. Whereas to capture a relationship between objects the concept of primary and foreign keys is used. They are known also as SQL databases by the query language used in RDBMS for managing data.

is it necessary to explain keys?

The Most Used Relational Database Management Systems [1]

- Oracle
- MySQL
- Microsoft SQL Server
- PostgreSQL
- IBM Db2

[1] The database technologies usage statistics are based on data from the most up to date version of website db-engines.com [5].

Advantages [6] [7]

- Designed for managing structured data
- ACID compliance - database transactions are Atomic, Consistent, Isolated, Durable
- The technology is mature, well-established with large eco-system and many developers have experiences with SQL and define RDBMS
- Data integrity is enforced

Disadvantages

- Problems managing data that are unstructured or semi-structured
- Data is normalized in order to achieve reduction of data redundancy, therefore stored objects may not have one-to-one mapping with the tables that represent them in memory. Also meaning lots of expensive (in terms of speed) joins when fetching such objects.

2.1.2 Non-Relational

A *Non-Relational Database*, is any database that does not follow the relational paradigm. They are younger and were invented to overcome limitations that relational engines have. The ultimate aim is to be more effective when coping with Big Data - data that is fast growing and their structure may not be defined strictly (unstructured, semi-structured information) [8]. There are multiple ways that these requirements can be met so we will introduce more precise division [9]. They are also commonly referred as NoSQL databases as the opposite of SQL databases.

Description of the types?

Non-Relational Database Types The most used DBMS is listed with each type.

- Key-value stores [10]
  Redis
- Wide column stores
  Cassandra
- Document stores
  MongoDB
- Graph databases
  Neo4j
- Search engines
  Elasticsearch

Advantages [6] [7]

- Elastic scaling, new cluster nodes can be added easily
- No strict database schema is required, bigger flexibility when changing inserted data format

Disadvantages

- Weaker data consistency mode - BASE (Basically Available, Soft state, Eventual consistency) in contrast to stronger ACID in RDBMS
- Lack of built-in data integrity
- Join operation is hard and may be even not supported

2.1.3 Conclusion

By the described properties of the respective systems, hopefully, a reader has some image of in what situation is reasonable to use the more traditional Relational design or look around for one of the Non-Relational databases. To sum it up, if the ACID principle is required by a user and business rules should be enforced the SQL databases are the ones to choose. Enterprises should be cautious and their first choice would be a Relational Database. In contrast when storing heterogeneous data or big volumes of it, consistency is not a priority and the system is extensively distributed some of the Non-Relational databases may be the right one.

Move to the modeling chapter?

However, in this work we will focus only on databases that are of the Relational kind.

The main reason behind this is that since NoSQL Databases have flexible schema or are schema-less (there is no point in determining a database schema when data types of attributes or keys) modeling of these databases quite a new discipline and is hard to find an intersection among different approaches to NoSQL modeling. Also concepts of higher abstraction models are omitted. [11]

The second thing to consider is that once a database is Relational we more or less know what to expect from it. The structure of these databases . So a tool that would extract metadata from relational data models is potentially more powerful as it can be applied to more database technologies than a similar tool aimed for some specific type of Non-Relational database.

Lastly, despite the Non-Relational may be growing in numbers and becoming a serious player the Relational still are, and in the near future will be, far more widely used.

2.2 Query Language

2.3 Means of Database Access

Drivers Connection strings(ODBC,JDBC)

3. Data Models

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 [12] 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] DB-Engines Ranking.

[6]

[7]

[8] What Is A Non Relational Database.

[9] The Types of Modern Databases.

[10] What Is a Key-Value Database?

[11]

[12] 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