Visualization of Functional Dependences in a Web Environment

Nikolay Georgiev

September 2009
Bachelor's Thesis in Computing Science, 15 ECTS credits
Supervisor at CS-UmU: Stephen J. Hegner
Examiner: Per Lindström

Umeå University
Department of Computing Science
SE-901 87 UMEÅ
SWEDEN

Abstract

As the core of a previous thesis [1], a web-based tool, called LDBN(Learn DataBase Normalization), was developed. The purpose of this tool is to provide an interactive learning environment for the normalization of relational database schemata whose constraints are defined by functional dependencies (FDs). During the Fall term of 2008, LDBN was used in conjunction with the course Principles of Database Systems at the Department of Computing Science at Umeå University, and some important observations were made.

As a part of this thesis, some crucial extensions to LDBN, based upon these observations, are developed. The most significant extension is a tool for the visualization of FDs, based upon templates found in popular textbooks such as [5] and [8]. Often, such visual representations are much easier for humans to grasp than purely text-based representations. However, this extension does not compromise the existing capabilities of LDBN. In addition, we present some other shortcoming of the previous version of LDBN and our approach to deal with those. We also discuss problems that lie within educational and web-based software development.

Contents

Chapter 1

Introduction

One of the fundamental concepts to teach in a database design course is the concept of relation decomposition, which consists of dividing relations (or data tables) into smaller tables in order to reduce redundancy, eliminate wasted storage and more importantly reduce anomalies or inconsistencies due to data updates. The central tool in producing these decompositions and refinement of the database in what is called normal forms, is the theory of functional dependencies, often called normalization theory. Unfortunately, theory is not yet understood well by practitioners [3]. One of the reasons for this is the lack of good tools which could aid the students during the learning process of relational-database normalization [7]. Thus our learning environment was developed in order to give students the ability to easily and efficiently test their knowledge of the different normal forms in practice. The environment assists the students by providing them the following functionalities:

1. Allow the student to specify a candidate decomposition of a given relation.
2. Assess the correctness of the student's proposed decomposition relative to many factors; including:
   - Lossless-join property.
   - Dependency preservation.
   - Specification of keys.
   - Correctness of the Second Normal Form (2NF), Third Normal Form (3NF) and Boyce-Codd Normal Form (BCNF) decompositions.
3. Provide students with sample decompositions when needed.
4. Allow users to communicate with each other via comments/posts.

1.1 Organization of This Report

In the remaining sections of this chapter we introduce informally the key features and concepts of our web-based learning environment, called LDBN (Learn DataBase Normalization) [1]; compare it to a couple of other available web-based database normalization tools, and provide the reader with small glossary. In Chapter **??** we give definitions to relational-database normalization and to the different normal forms. In Chapter **??** we discuss some design issues regarding LDBN such as platform choice and others. Chapter **??** provides a formal description of our reference implementation of the learning environment, and Chapter 4 shows our conclusions.

1.2 Learning Database Normalization with LDBN

In this section we briefly introduce our reference implementation of the web-based learning environment, called LDBN. Figure **??** shows the overview of the most important part of the user interface (UI) - the *Solve Assignment* view/tab. Here students can test their knowledge on the subject of relational-database normalization. The first thing the reader may notice is the fact that LDBN runs within a browser. The client side of LDBN is written in JavaScript following the AJAX techniques (more about this in Chapter **??**). Furthermore, LDBN is assignment driven. This means students have to first choose an assignment from a list with assignments, submitted by other users (lecturers). Such a list is shown in Figure **??**. An assignment consists of a relational-database schema in universal-relation form (URF), i.e., all the attributes in a single relation and a set of FDs on the attributes. After an assignment has been loaded, we require the students to go through the following steps in LDBN:

1. Determine a minimal cover of the given FDs, also known as a canonical cover.
2. Decompose the relational schema which is in URF into 2NF, 3NF and BCNF.
3. Determine a primary key for each new relation/table.

The task of checking a potential solution involves many subtasks, which may be performed in any order. In addition to this, a partial or complete solution can be submitted at any given time by pressing the *Check Solution* button. After that the system analyzes the solution by performing the following checks:

1. Correctness of the minimal cover of the given FDs.
2. Correctness of the FDs associated with each relation R; that is, if the FDs are actually in the embedded closure of F_R^+ for this relation. See Section **??** for more details on a closure of a set of FDs.
3. Losses-join properly for every schema in the decomposition.
4. Dependency preservation for every decomposition.
5. Correctness of the key of each relation.
6. Correctness of the decomposition, i.e., if the decomposition is really in 2NF, 3NF and BCNF.

A dialog with the result is shown to the user. In case of an error the system offers feedback in form of small textual hints, indicating where the error might be. Such a dialog is shown in Figure **??**. In this case we can see that the user has made a correct decomposition for 2NF and 3NF, but his/her decomposition for BCNF has some errors, namely the key of the relation R4 is incorrect. The dialog shows that the decomposition does not satisfy the dependency-preservation property, but in the case of BCNF this is not always possible, therefore it is only a warning.

Additional features of LDBN include creating an assignment, which can be done only by registered users. This restriction is necessary in order users to be able to distinct assignments provided by trusted users, e.g. the database course lecturers. Registered users have also the ability to leave textual comments for every assignment. On the one hand, such comments ensure that user can easily communicate and share ideas with each other, and one the other hand, comments could also decrease the amount of workload for the lecturers in terms of giving an explanation to difficult decomposition.

More detailed and formal description of the features of LDBN will be given in Chapter **??**.

1.3 Glossary

2NF, 3NF, BCNF *Second Normal Form, Third Normal Form, Boyce-Codd Normal Form.* See Section **??** for more details.

AJAX *Asynchronous JavaScript And XML* Asynchronous JavaScript And XML is a group of interrelated web development techniques used for creating interactive web applications, for more details see Section **??**.

API An *Application Programming Interface* is a set of functions, procedures or classes that an operating system, library or service provides to support requests made by computer programs [2].

CSS *Cascading Style Sheets* is a stylesheet language used to describe the presentation of a document written in HTML.

DBMS A *Database Management System* is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database.

GWT *Google Web Toolkit* is an open source Java software development framework that allows web developers to create AJAX applications in Java. More details in Section **??**.

LDBN *Learn Database Normalization* is our reference implementation of the web-based environment for learning normalization of relational database schemata. We often refer to LDBN as *our learning environment* or *our implementation.*

ODBC *Open Database Connectivity* provides a standard software API method for using database management systems.

RPC *Remote Procedure Call* is an inter-process communication technology that allows a computer program to cause a subroutine or procedure to execute in another address space.

SQL *Structured Query Language* is a computer language designed for the retrieval and management of data in relational database management systems, database schema creation and modification, and database object access control management.

XMLHttpRequest is an API that can be used by JavaScript and other web browser scripting languages to transfer asynchronously XML and other text data between a web server and a browser.

Chapter 2

Motivation

The motivation of the project goes here!

Chapter 3

Approach

Chapter 4

Conclusions

Chapter 5

Acknowledgements

I would like to thank my supervisor, Stephen J. Hegner, who has supported me throughout my thesis with his patience and knowledge while allowing me the room to work in my own way. Without him this report would not have been completed.

I would also like to thank Michael Höfling for all his help during my stay in Sweden. Last but not least, I thank my family for their love and support.

References

[1] LDBN - Learn DataBase Normalization. `http://ldbnonline.com/` (last visited August 2009).

[2] QuickStudy: Application Programming Interface (API). `http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=43487` (last visited August 2009).

[3] J. Biskup. Achievements of Relational Database Schema Design Theory Revisited. *Semantics in Databases*, (1358):29–54, 1998. `http://ls6-www.informatik.uni-dortmund.de/uploads/tx_ls6ext/Biskup_1998.ps.gz` (last visited August 2009).

[4] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems (5th Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[5] H. Kung and H. Tung. A web-based tool to enhance teaching/learning database normalization. In *Ninth Annual Conference of the Southern Association for Information Systems (SAIS)*, Jacksonville, FL, USA, March 2006.

[6] A. Kemper und A. Eickler. *Datenbanksysteme: Eine Einführung, 6. Auflage*. Oldenbourg Wissenschaftsverlag GmbH, München, Deutschland, 2006.