

The Development and Usage of a Relational Database Design Tool for Educational Purposes

*Miloš Bogdanović, Aleksandar Stanimirović, Nikola Davidović,
and Leonid Stoimenov*

Faculty of Electronic Engineering, University of Niš, Serbia

bogda@elfak.ni.ac.yu; alex@elfak.ni.ac.yu
davidovicnikola@yahoo.com; leni@elfak.ni.ac.yu

Abstract

Most universities where students study informational technologies and computer science have an introductory course dealing with the development and design of databases. These courses often include usage of database design tools. In this paper, the #EER tool is presented, the task of which is to make the process of relational databases design easier for the students and partially automatize it. The tool evolved due to the experience in using similar tools for educational purposes. It enables fast and efficient development of the relational database conceptual model and its automatized compilation into a relational model and further to data definition language (DDL) commands. #EER tool is based on the extended entity-relationship (EER) model for conceptual modeling of relational databases. Modular architecture of the tool, the development of which is based on the usage of the design patterns, along with the benefits that its usage brings, is also presented.

Keywords: Database design, educational tool, EER, conceptual data modeling

Introduction

Databases very often have an important role in the development of modern software systems. Nowadays, there is hardly any software system that doesn't store particular information in some kind of database. The situations in which different applications share same database often occurs. Because of that the process of database designing is extremely important as it needs to provide the solutions that could respond to different and often opposite requirements.

Due to the high importance which databases have in modern software systems, introductory database courses also gain importance. Therefore, it is necessary to devote a lot of attention to theoretical concepts of database development process. In order to focus attention of students to the phase of conceptual database design, it is of great importance to use an automated database design tool. Concerning its functionalities and capabilities, this tool would facilitate the overcoming

of the theoretical concepts of database design to the students.

The main aim of introductory database courses is comprehension of the database design process. The process of database design itself is divided into four phases (Connolly & Begg, 2002; Elmasri & Navathe, 2002):

- Collect and analyze demands

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

- Conceptual design
- Logical design
- Physical design

For this purpose of the greatest importance is the phase of conceptual design. In this phase the conceptual database model is designed to be independent from any kind of implemental details (hardware and software platforms, chosen DBMS, programming languages, client applications). The conceptual database model consists of the detailed description of entities, relationships, and constraints.

In the phase of conceptual design it is possible to use different conceptual database models. The basic function of the conceptual model is to provide means for the description and formulation of the demands for the databases which have been collected in the previous phase. The conceptual database model is considerably important because it prepares the foundation for the following phases in the process of database designing.

For the introductory database course, there appears to be more support for ER modeling than for UML modeling, and most of the Computer Science (CS) and Information Systems (IS) university curricula's use traditional ER modeling techniques as their primarily modeling technique in their undergraduate introductory database courses (Suleiman & Garfield, 2006). The primary focus on relational databases in introductory database courses and the complexity of UML class diagrams are compelling reasons to consider using (E)ER notation in the class instead of UML (Suleiman & Garfield, 2006).

Due to the great importance of the conceptual model in the database design process, this model has a significant part in teaching courses dealing with this aspect of informational technologies. Software tool which would lead to automatization of the process of conceptual database model development might have a considerable part in the education of students, especially if they are developing databases for the first time. This kind of tool would focus the attention of students to conceptual database design, speed up the process of database development, and cause students to spend more time on understanding the concepts of database design. Also, universities could customize the tool according to their needs and accepted (E)ER graphical notation.

The paper is organized according to the following method: in the section *Database Design Tools – State of the Art*, the arguments for the construction and usage of this kind of tool, the characteristics of the similar tools and the target groups of the users are listed. In the section *#EER – Database Design Tool*, the characteristics of the tool are presented. The section *The Architecture of #EER Tool* describes the general architecture of the tool. The section *Using the #EER in Educational Purposes* describes the way the tool has been used in Faculty of Electronic Engineering Niš.

Database Design Tools – State of the Art

In order to emphasize the advantages of the tools specially developed for educational purposes and therefore adapted for certain courses, we will present a short list of some of the most frequently used tools and their characteristics. At the moment, there are as many commercial as free-ware tools whose function is to support the process of database designing. Some of these tools offer support for whole process of database designing, while the others offer support just for particular phases in the process.

Probably the best example of a popular commercial database design tool would be Microsoft Visio (<http://office.microsoft.com/visio>). This tool is designed for the professionals that are involved in the process of developing and using the different types of systems. Visio is one of the

best and most frequently used tools. But if we consider its complexity, it probably isn't the right choice for students with no experience in database designing.

The following less complex and easier to use shareware tools are probably better choice:

- **DeZign for Databases**- the tool aimed to enable intuitive database design (<http://www.datanamic.com/dezign/index.html>). It is narrowly specialized for the problems of designing databases and is designed for professional database designers and administrators. For the purpose of conceptual database design, it supports the entity-relationship model (ER model) and its graphical notation. It also provides the integration with the most popular SQL and desktop databases: Oracle, ANSI Level2, dBase, IBM DB2, Informix, Ingres, etc.
- **Toad Data Modeler**- a freeware tool which is based on the ER model and its graphical notation (<http://www.casestudio.com/enu/default.aspx>). It enables the creation of complex ER diagrams for over 20 different database management systems. It offers possibilities to define the rules of the referential integrity and generalization of the detailed HTML and RTF reports.

The next generation of database design tools is based on the development of the intelligent teaching systems (ITS). The goal of these systems is to provide individualized pedagogical sessions to all students (Suraweera, 2002). A well known solution from this category is KERMIT - the Knowledge-based Entity Relationship Modeling Intelligent Tutor. This system is developed as a problem-solving environment in which students can practice their ER modeling skills with the individualized assistance of the system (Suraweera, 2002). The presented tool can also be compared to this class of educational tools.

The main characteristic of the described tools is that they are designed to cover the great number of different database design approaches. Basically, these tools are designed for database experts and professionals and can be very complex and immense. As a result, applying these tools in educational process is very difficult and requires additional activities which are related to the special training for using these tools.

Another common characteristic of these tools is that they support (E)ER model and its graphical notation (Chen, 1976) or UML modeling techniques (<http://www.uml.org/>) (Unified Modeling Language, 2007). If users want to use alternative notations, they have to use or develop plug-ins for the selected tool which will allow the usage of the desired notation. Also, these tools provide means for integration with the existing database management systems.

#EER – Relational Database Design Tool

The basic goal for realization of the #EER tool, whose architecture is shown in this paper, is to overcome the mentioned problems. Another aim was development of a tool in which the user can create a conceptual model for the relational database using the EER modeling technique. The basic idea behind this tool is to simplify and speed up development of the conceptual database model. This tool targets students as potential users. Most of the students meet with the problem of developing the relational database model for the first time. Students are the kind of users that have advanced knowledge about usage of computers and have experience in drawing various diagrams. The purpose of this tool is to provide students with means to develop relational database model without unnecessary and complex functionalities.

#EER is the tool which tends to automatize the process of database designing and to make it easier for the students. Because of the simplicity of the relational databases and their dominant role on the market, the tool is limited to designing only the adequate relational database models. ER model has a great modeling support in introductory database courses. For the need of the concep-

tual design EER (Extended Entity Relationship) model is chosen and its graphical notation in the form of EER diagram. The adequate graphical notation is used and was suggested by Elmasri and Navathe (2002) (See Figure 1.)

The tool itself is outlined in the form of mini developing environment for the relational database design and according to that gives the support to the user from the creation phase of conceptual model until the phase of automatic generation commands for creation of data (DDL commands) at the end of the designing phase.

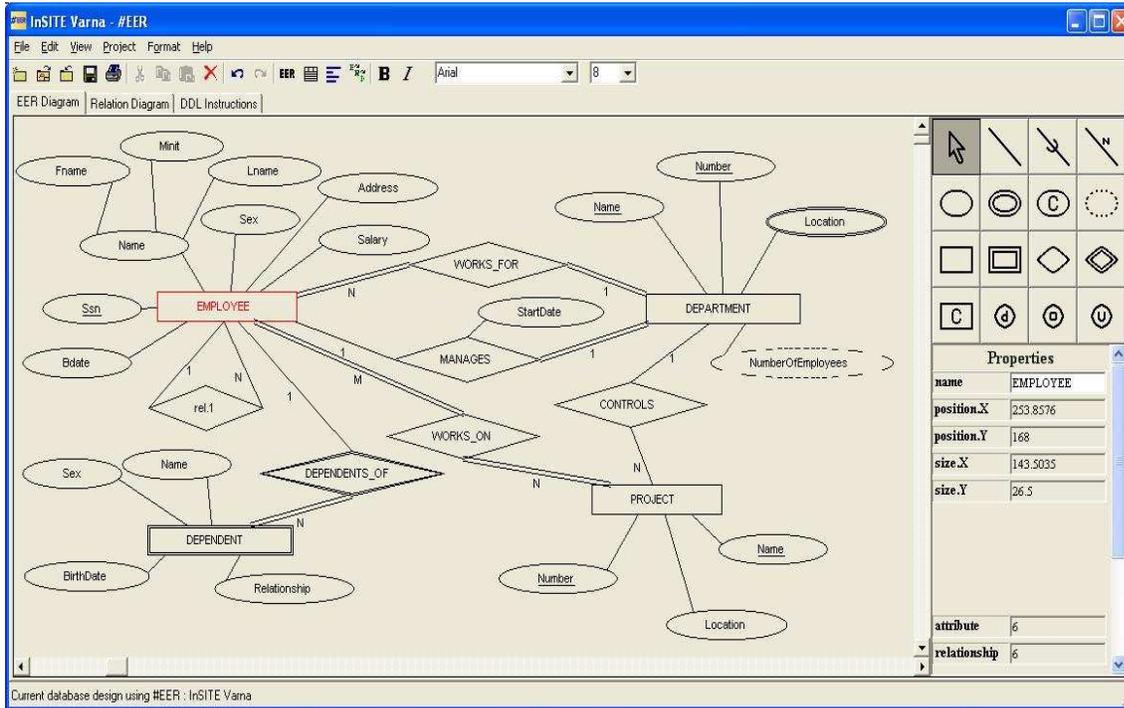


Figure 1. EER diagram of the COMPANY database (example taken from Elmasri & Navathe, 2002)

#EER is carried out as the standard Windows desktop application. Along with that it offers most of the standard Windows graphical elements (menu bars, dialogs, tool bars) as well as the standard functions of the Windows applications (cut, copy, paste, undo, redo...). Also, the tool provides a wide range of functions giving the support to the process of relational database designing.

- **Drawing EER diagram**- fast and easy outlining of the complete EER database diagram, the selection of the individual elements or group of elements in EER diagram, the simple manipulation and modification of the diagram elements and similar.
- **Automatic compilation of EER model into relational model** - automated compilation EER model in relational model and generation of relational diagram underlying derived relational model.
- **Creation of DDL commands** – automatic generation SQL commands based on derived relational model.
- **Printing of the designed project** - the possibility of optimized printing of the whole project or its particular parts (models, diagrams and etc.)

- **Export of the EER diagrams into standardized graphical formats** - the possibility to export EER diagrams in TIFF, JPG, BMP, PNG, EMF, WMF and the rest of the usual graphical formats.
- **Export of the Generated DDL commands into TXT file** - generated SQL commands can be saved in the textual file for the possible processing and usage in creating database

The Architecture of #EER Tool

The general architecture of the #EER tool is shown in Figure 2 in the form of general block diagram. That the application has a modular architecture can be seen in the figure.

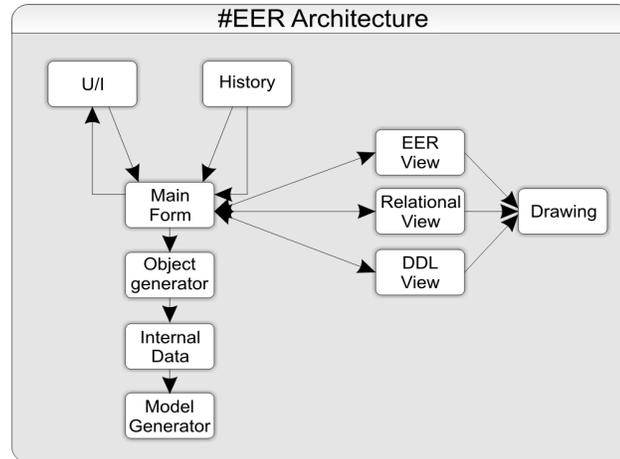


Figure 2. Architecture of the tool

All the actions of the users are registered at the main form of the tool – *MainForm* – which contains the area for outlining the EER diagram. Besides this main area, the main form of the tool contains the areas for displaying the generated relational database model and as well the area displaying DDL commands generated on the account of derived relational model. Every area is outlined separately based on the data that are linked to the observed area. Managing the drawings of the different parts of the project is centralized and implemented by the singleton class that changes the way of the outlining depending on the currently active working domain. The outlining class is implemented with the usage of observer and singleton design patterns (Gamma, Helm, Johnson, & Vlissides, 1995).

The application implements all the standardized Windows functionalities. *I/O module* is in charge of one part of these functionalities while the other part of the basic functionalities is realized through *History module* of the tool. These two modules enable the export of EER diagrams in standardized graphic formats, printing, and the possibility to use undo and redo operations, etc.

The drawing of the database diagram is reflected in the creation of objects that illustrate all previous user actions. *Object Generator module* is responsible for the creation of objects. This module is implemented by the usage of Builder design pattern (Gamma et al., 1995). The objects that have been built by the *Object generator module* show the internal (system) representation of the EER database diagram. Elements are graphically represented in the drawing area. Every type of element that can be a part of an EER module is illustrated by one type of the application object. These objects are connected and conditioned by each other and are saved in the operative memory for the purpose of generation of the relational model and DDL commands. Flyweight design pattern (Gamma et al., 1995) is used in order to generate the new objects. *Internal data module* of the tool is in charge of storing and organizing internal data structures.

The relational database model is created according to memory representation of EER diagram and presented using relational database diagram. After that the relational database model is translated into series of DDL commands (using SQL). Generalization of the relational model and DDL database commands is implemented through Model generator module of the tool. This module implements the adopted Factory Method design pattern (Gamma et al., 1995).

Using the #EER in Educational Purposes

The presented tool has been used in the past two years as an auxiliary educational resource in the Faculty of Electronic Engineering, University of Nis. The tool was used in two courses: Databases and Advanced Databases. The first one usually gathers 120 students. It consists of obligatory laboratory exercises, projects, and final exam.

This tool was used by instructors in order to present basic concepts of EER database model. Students also use this tool in order to:

1. learn basic concepts of EER database model and draw EER diagrams
2. translate EER database model into relational model
3. translate relational model into DDL commands

One of the tool's built-in mechanisms is the EER-to-relational model and a relational model-to-DDL commands translator. By using this mechanism, students with no previous knowledge concerning relational model and DDL commands are able to generate them from the EER model they have drawn. Once students generate relational model and DDL commands, they can observe generated output thus learning model translating methods.

The usage of the tool was obligatory in laboratory exercises. On the other hand, students had the opportunity to choose the tool they would use in order to develop their projects. Instructors suggested the usage of #EER tool because of the supported notation. Their task in the projects was to develop a database with given characteristics. Our statistics shows that over 50% of students used our tools in their projects.

Conclusion

Using modern commercial or freeware tools instead of specially designed tools in educational purposes, having in mind the adaptability of the commercial tools to the different project requirements, can bring as many advantages as perplexities. Having the tools specialized for the certain type of the problem can have a number of advantages: the tool is optimized, intuitive at the level of the problem it is dealing with, easy to use for the designers familiar with the notation the tool uses. #EER is an example of such a tool- a tool that is used by a certain group of people dealing with the development of specific type of databases – the relational databases – and it draws all the attention to itself. With that particular characteristic it directs all its users towards the narrow extent of the possible technologies and notations; therefore, it is made easy to overcome and use. Database design tool developed for educational purposes can be seen as the next logical step after the students have overcome the theoretical concepts. This kind of tool gives students the chance to test their knowledge of the database design in an easy and fast way.

Database design tools are a suitable way for demonstrating and explaining database design concepts to students. One can use any of the tools currently present on the market. However, commercial tools are sometimes complex and hard to use for beginners in database designing. Based on our two years experience, a better way would be usage of the tools that only have functionalities students need. By doing so, all of the student's attention is drawn to the design concepts rather than the ways the tool can be used.

This tool can be downloaded from <http://gislab.elfak.ni.ac.yu/SharpEER>

References

- Connolly, T., & Begg, C. (2002). *Database systems* (4th ed.). ISBN 0-321-21025-5, Addison-Wesley.
- Elmasri, E., & Navathe, S. (2002). *Fundamental of database systems* (4th ed.). ISBN 0-321-1222-6. Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns – Elements of reusable object-oriented software*. Addison-Wesley.
- Chen, P. P. (1976, March). The entity-relationship model--Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9-36. Retrieved from <http://bit.csc.lsu.edu/~chen/pdf/erd-5-pages.pdf>
- Suleiman, J., & Garfield, M. (2006). Conceptual data modeling in the introductory database course: Is it time for UML? *Journal of Information Systems Education*, 17(1), 93-100.
- Suraweera P. (2002). *An intelligent teaching system for database modeling*, University of Canterbury.
- Unified Modeling Language*. (2007). Object Management Group. Retrieved May 2007 from <http://www.uml.org/>

Biographies



Miloš Bogdanović received the BSc degree in computer science at the University of Niš, Serbia. His PhD research is concerned with spatial data modeling as well as development of WebGIS applications. He is working as a teaching assistant with the Department of Computer Science.



Aleksandar Stanimirović received the BSc, MSc degrees in computer science at the University of Niš, Serbia. His PhD research is concerned with spatial data modeling as well as design and development of methodologies and tools for managing spatial data. Since 2000 he is working as a teaching assistant with the Department of Computer Science. He is a student member of IEEE.



Nikola Davidović received the BSc degree in computer science at the University of Niš, Serbia. His PhD research is concerned mobile technologies and Web services. He is working as a teaching assistant with the Department of Computer Science.



Leonid Stoimenov received the BSc, MSc and PhD degrees in computer science at the University of Niš, Serbia. He is currently an associate professor in the Department of Artificial Intelligence at Faculty of Electronic Engineering at this University. His research interests in computer science include GIS, databases, mediators, ontologies and semantic interoperability. He is a member of IEEE and representative in AGILE association of GIS laboratories in Europe.