Evaluating the Use of an Interactive Software Tool for Learning BCNF Normalization

Full Paper

Georgia Koloniari
Department of Applied Informatics
University of Macedonia
Thessaloniki, Greece
gkoloniari@uom.gr

ABSTRACT

Normalization is a required task to ensure a relational database schema with no data repetition and anomalies. However, most students of database courses find it too theoretical and difficult to understand. In this paper, we present a study of an educational tool for BCNF normalization we have developed. The goal of the tool is twofold: to be used as a teaching aid by the instructor to make normalization more understandable and especially to be used as a self-training tool by students for practice. The results of our study show a positive feedback by the participating students on both design goals of our tool. The students also positively rate the tool with respect to usability and user-friendliness. Finally, our preliminary results show an improved performance of the students using the tool in their final exams.

CCS CONCEPTS

• **Applied computing → Computer-assisted instruction**; *Education*; • **Information systems** → *Relational database model*;

KEYWORDS

BCNF, educational tool, assessment, evaluation study

ACM Reference format:
Georgia Koloniari. 2017. Evaluating the Use of an Interactive Software Tool for Learning BCNF Normalization. In *Proceedings of BCI '17, Skopje, Macedonia, September 20–23, 2017,* 8 pages.
https://doi.org/10.1145/3136273.3136284

1 INTRODUCTION

In relational database design, normalization is required to eliminate data repetition and anomalies. Thus, it is included in any database systems course curricula. However, normalization is perceived by the majority of students as one of the most difficult topics in database courses. The topic is considered too theoretical by most students that have difficulties both understanding the methodology and the motivation behind normalization. Also, while there is a plethora of well-known CASE tools that automatically derive from the logical model the SQL code that implements the physical schema, the tools providing automated normalization are limited.

In [4], we first presented a software tool for automating the normalization process. In particular, given a relational table and a set of functional dependencies, the proposed tool outputs a set of normalized tables following the Boyce-Codd Normal Form (BCNF) [6]. While the tool can be used by application developers to automatically derive a normalized database schema, its main design goal is educational. The tool besides BCNF decomposition, supports additional separate functionalities such as candidate key evaluation, and most importantly an interactive step-by-step decomposition functionality in which a user is able to try alternative decompositions by selecting at each step a functional dependency and the corresponding table to be decomposed.

In this paper, we focus on the evaluation of the proposed software tool for educational purposes. We applied a pilot study of the use of the tool in the context of a database course on a group of students of the Applied Informatics Department of the University of Macedonia. We wanted to assess the value of our tool both when used as a teaching aid by educators to demonstrate the normalization process, but mostly as a self-training tool used by students to enable them to practice and better understand normalization on their own. Besides evaluating its educational value, our study also includes the evaluation of the tool as a software tool with respect to ease-of-use and user-friendliness, since we consider this an important factor in motivating the students to use the tool for self-training. Finally, our study also includes results on how the tool improved the performance of the group of students that participated in the pilot study as compared to their performance on the rest of the database course topics and also against other students that did not have access to the software tool. The rest of the paper is structured as follows. Section 2 discusses related work and Section 3 outlines the basic concepts and algorithms for normalization, and presents an overview of the normalization tool and its basic functionalities. Section 4 describes our study and Section 5 includes the derived results. Finally, Section 6 concludes the paper.

2 RELATED WORK

There are a number of tools for database normalization designed for educational purposes [1], [5], [9], [10], [12], [15], [16]. Two simple web-based tools are presented in [12] and [15]. Both support

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
BCI '17, September 20–23, 2017, Skopje, Macedonia
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5285-7/17/09…$15.00
https://doi.org/10.1145/3136273.3136284

only 3NF normalization, while the second additionally supports the evaluation of the closure and minimal cover of the functional dependencies. Both tools do not provide extensive explanations of the decomposition process. The tool, in [12], presents the decomposition results gradually, but without allowing user interaction. Another web-based tool, presented in [5], supports BCNF normalization but not directly, normalizing first to 3NF. This tool offers detailed explanations. All three web-based tools besides not allowing any user intervention, do not have particularly user-friendly GUIs, and no loading nor saving functionality. NormalDB [1] uses Prolog in its core and through a web-interface provides a user-friendly GUI that enables normalization to 3NF, BCNF and also denormalization. JMathNorm [16] supporting 2NF, 3NF and BCNF is written in Mathematica with a GUI in Java. While its main purpose is educational it can also be used for other purposes as it is simple and efficient. EDNA [9] is another educational tool that fully automates the normalization process and additionally derives SQL statements for implementing the corresponding physical schema. Finally, LBDN [10] is a web-based environment that supports decomposition to 2NF, 3NF and BCNF. It allows users to provide their own solutions that are evaluated by the system for validity, and it also provides a sample solution when required. However compared to all educational tools, ours is the only one that allows the user to direct the decomposition process and provides alternative solutions when users select the functional dependencies in different order.

Regarding evaluation, the tools in [1], [5], [10], [15] report no evaluation results of their application. The web-based tool in [12] was applied in a single group of 45 students of two sections of a junior level systems analysis and design course using a pretest-posttest design for the evaluation. The students were called to compare the textbook method in BCNF normalization and the use of the web-based tool and results reported that students preferred the use of the tool. JMathNorm [16] was applied during a Database Systems course offered to about 25 third year computer engineering majors. No detailed evaluation results are presented, but the authors report that during the course evaluation the students characterized the tool as useful. Finally, EDNA [9] was applied in a sample of 18 students and a limited qualitative evaluation reported positive feedback. Our evaluation focuses besides the educational value, to the ease of use and user-friendliness of the software tool as well. In addition, the performance of the students in the final exams on the normalization topic is evaluated and compared against other students that followed the traditional classroom teaching approach.

There are also a number of database normalization tools that are designed not as educational tools but to provide aid to database designers. Micro [8] supports 2NF, 3NF and BCNF, while Normalizer [2] and RDBNorma [7] only support 2NF and 3NF. All three automate the normalization process and focus mostly on time and space efficiency. In recent years, the focus has shifted to applying normalization in new domains such as XML databases [14] and document-based databases [11].

3 NORMALIZATION EDUCATIONAL TOOL

In this section, we briefly describe the basic concepts used in normalization and the corresponding algorithms involved. Then, we proceed with an overview of the normalization tool, which was first presented in [4].

3.1 Relational Database Design

A fundamental concept in normalization is that of *functional dependency* (FD) which defines a constraint between subsets of attributes in a relation. For two set of attributes X and Y belonging to relation R, we say that X functionally determines Y, $X \rightarrow Y$ if and only if, each value of X is associated strictly with a single value of Y.

A FD $X \rightarrow Y$ is *trivial* if and only if $Y \subseteq X$. The non-trivial functional dependencies are divided into *completely non-trivial* and *partially non-trivial*. A FD $X \rightarrow Y$ is completely non-trivial when $X \cap Y = \emptyset$, otherwise, it is considered as partially non-trivial.

Given a relation R and a set F of functional dependencies, the closure of a set of attributes X of R, denoted as X^+, is the set of all attributes that are functionally dependent from X as they can be derived by using Armstrong's axioms [3] for dependency deduction.

The attributes closure is required to determine the candidate keys of a relation R. A set of attributes X is a candidate key for R if $X^+ = R$ and $\nexists Y : Y \subset X$ such that $Y^+ = R$. The trivial algorithm in the literature, initially examines the closure of each attribute one by one and then in combinations of two, three and so on, until all candidate keys are found. If a candidate key X is found, then all its supersets are omitted for further examination as they are determined as superkeys.

In our tool, simple optimizations are applied to enhance the trivial algorithm, by considering only candidate keys that contain all attributes that do not appear in the right side of any of the FDs in F, if such attributes exist. We also used an efficient binary scheme to enumerate all possible attribute combinations for consideration.

Data: R, F
Result: Set of relations derived from R in BCNF
for *each* $X \rightarrow Y \in F$ **do**
  if $X^+ \neq R$ **then**
    $R1 \leftarrow X \cup Y$;
    $R2 \leftarrow R - Y$;
    return R1,R2 ;
  end
end
return R ;

Algorithm 1: BCNF Decomposition

Edgar Codd, along with Raymond Boyce, in 1974, redefined the 3rd Normal Form, upgrading it into the "3.5" normal form [6]. This normal form is well known as *Boyce-Codd Normal Form* or simply BCNF. While 3NF is also very popular, we selected to focus our tool on BCNF as it ensures that all redundancies based on functional dependencies have been removed from the relational schema.

Given a relation R and a set of FDs F, for R to conform to BCNF, every functional dependency in F must contain a superkey for R in its left side. If there is a dependency that violates this rule, then R is not in BCNF and must be decomposed. Algorithm 1 selects one of the FDs that violates BCNF and R is split into two new relations satisfying the lossless join property. The resulting relations need to be examined in turn using Alg. 1 recursively, and more decompositions may arise until all remaining relations are in BCNF.

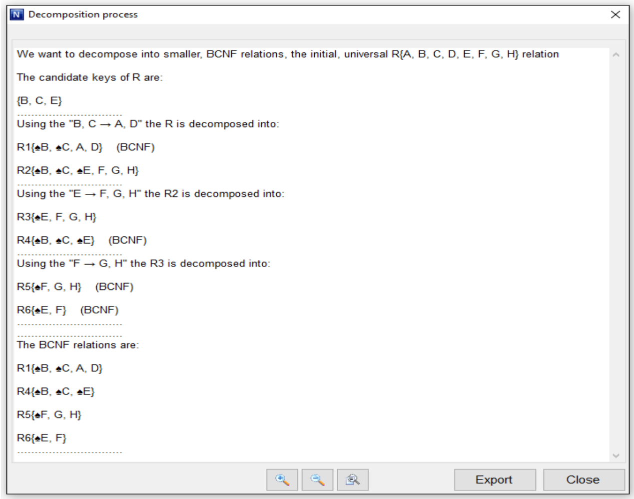

Figure 1: Automatic decomposition.

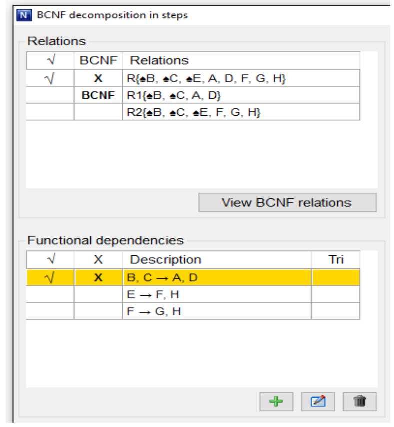

Figure 2: Selection of second step for decomposition.

3.2 Tool Overview

The normalization tool (first presented in [4]) is designed for educational purposes intended to be used both as a teaching aid by instructors and students themselves for self-training. For the latter purpose, we focused particularly in ease of use and user-friendliness, so as to motivate the students in using it.

Implemented in C#, our software tool provides a self-explanatory GUI with a windowing environment equipped with typical menus and command buttons to ensure that users will familiarize themselves quickly with its main operations. Mechanisms enforce checks to user input so as to reduce the possibility of human error, and provide error messages that guide the user towards the right direction. Finally, the results of all operations are explained in detail.

The basic use of our tool is: Given a relation R and a set of functional dependencies F to derive a set of appropriate normalized relations in BCNF. We demonstrate its functionalities through a running example: consider a relation $R(A, B, C, D, E, F, G, H)$ and a set of FDs $F = \{BC \longrightarrow AD, E \longrightarrow FH, F \longrightarrow GH\}$.

The tool creates database schemas consisting of a single relation R and a set of functional dependencies. The user first creates a new relation by enumerating its attributes and optionally specifying a type and a description for each. After defining at least two attributes, the user can define one by one the FDs. The specification of the dependencies is assisted by the GUI enabling the user to select the appropriate attributes for the left and right part of each FD. User-defined schemas are saved and can later be loaded, either to continue their editing or to perform normalization. All results can be exported in text or html format so that students and instructors can easily edit them, add notes or any other comment they want.

The basic supported operations are: (a) closure evaluation, (b) candidate key evaluation, (c) decomposition and (d) step-by-step decomposition.

Closure Evaluation. The user selects a set of attributes and their closure is evaluated. The results demonstrate the process in detail to assist the students in understanding how they are produced. The purpose of this separate operation is to enable students to find the candidate keys for the relation without using the automatic option or to understand why a combination they consider a key is not. For this reason, messages that explain superkeys are also included as this is one of the subtleties many students have difficulty understanding.

Candidate Key Evaluation. This functionality produces a result screen that enumerates all candidate keys. While performing normalization, our tool selects a candidate key as a primary key for each relation and denotes all primary attributes in the result screens with a spade symbol. If there is more than one candidate key, the shortest one is chosen and between keys with the same length the choice is random.

Decomposition. The "automatic" decomposition presents the normalization process in detail but without giving the user any chance to intervene (Fig. 1). In this case, the FDs are examined in the order they were defined by the user.

Step-by-Step Decomposition. The most important functionality of our tool is the interactive step-by-step decomposition in which the user guides the normalization process. When applying step-by-step decomposition, a new window pops up containing a table with the initial, universal relation and a list of the FDs in which trivial dependencies are marked. The user needs to select one relation and one functional dependency to initiate decomposition. By default at the first step, the initial, universal relation is selected, so the user needs to choose a functional dependency. The preview command enables us to check the effect of the selected FD before applying the decomposition. We can also edit our dependencies during this process so as to derive alternative decompositions on the fly. This enables instructors to timely address different students' questions.

Whenever a relation is decomposed, two new relations are created. Figure 2 shows the result of decomposing R using $B, C \longrightarrow A, D$ at the first step. In our example, the $R1$ relation is in BCNF, and it can no longer be decomposed, though we can still preview a new decomposition that creates a corresponding error message. The $R2$ relation is not in BCNF and since there are two more FDs available, we can decompose following the same process. In case a relation cannot be decomposed based on a chosen FD, a message explains why. For the given example, we require three decompositions in total resulting in the six tables in Fig. 3.

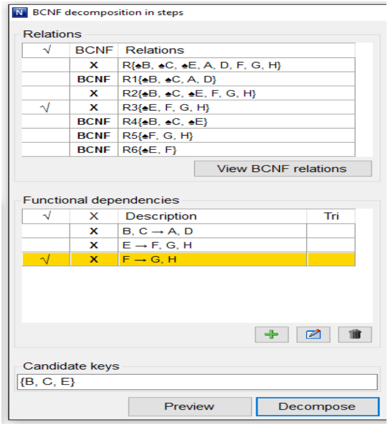

Figure 3: Final step for the decomposition.

Note that if one wants to normalize an entire database schema and not just a single table she can do it by normalizing each table. Therefore, the tool can also be used to check whether a table or an entire database schema is already in BCNF.

4 THE STUDY

The study was conducted at the Applied Informatics Department of the University of Macedonia. Studies at the department last four academic years and at the 3rd semester students select between two orientations, Applied Informatics and Technology Management. We applied our study in the context of an introductory course to database managements systems, "Databases I", which is offered at the 3rd semester and is compulsory for students in both orientations. The teaching goals of the course consist of students learning to design an appropriate database schema based on application requirements and also learning to implement and manage a database in any commercial or open-source relational database management system.

The contents of the course include conceptual database design using the Entity-Relationship (ER) model, relational database design, relational algebra, SQL and normalization. Appropriate tools are demonstrated and used for both the ER and relational model, and large emphasis is given on the implementation and practice on open-source database management systems. With respect to normalization, a *bottom-up* approach to database design is presented where normalization is the main mechanism that automates the design of a relational schema by gradually decomposing a universal relation into a set of smaller tables according to a specific normal form. The students are introduced to functional dependencies and multi-valued dependencies and all normal forms from First Normal Form (1NF) to Fourth Normal Form (4NF).

The main goal of the study we performed is to evaluate the usefulness of the proposed normalization tool as an educational tool. Thus, the conducted study aimed at answering the following questions.

Question 1: How difficult/useful/interesting do the students find the topic of normalization? We are especially interested in where the students place the normalization process with respect to each of the above factors compared to the other topics included in the database course. Our experience indicates that the students consider the topic too theoretical and therefore usually focus on topics they consider more practical such as implementation issues and database management using SQL. As tools are used to demonstrate and practice on all other topics besides relational algebra and normalization, this could indicate that using software tools in the educational process, both to demonstrate the processes and also to enable student practice would better motivate the students.

Question 2: Is students' interest in normalization increased by using the normalization tool?

Question 3: To what extent do the students consider the tool useful (i) as a teaching aid used by the instructor and (ii) as a practice tool used for self-learning?

Question 4: Does the use of our tool improve the performance of the students in their exams?

To answer the above questions we conducted our pilot study based on an extensive questionnaire. In particular, our study was conducted during the Fall Semester of 2016 when the Database I course was taught. After the topic of normalization was presented in three lectures using the traditional teaching methods in class, a three hour laboratory study was conducted using the normalization tool. 40 students that volunteered participated in the study, 10 (25%) of who were women and 30 (75%) were men. Before the start of the evaluation, the participants were asked to assess their own knowledge on the various tasks involved in the BCNF normalization process, denoted as T1 to T8 in Table 1, using a typical 5-point Likert-scale from *None* to *Excellent*. As shown in Fig. 4, the majority of the students consider that they have an overall fair knowledge of normalization, 35% of the total answers were fair, while 37% consider they have very good or excellent knowledge.

Table 1: Preliminary Questions

T1	Computing the closure of a single attribute.
T2	Computing the closure of a combination of attributes.
T3	Computing a candidate key.
T4	Computing all candidate keys.
T5	Checking if a relation satisfies BCNF.
T6	Choosing a functional dependency for decomposition.
T7	Decomposing a relation.
T8	Overall BCNF normalization.

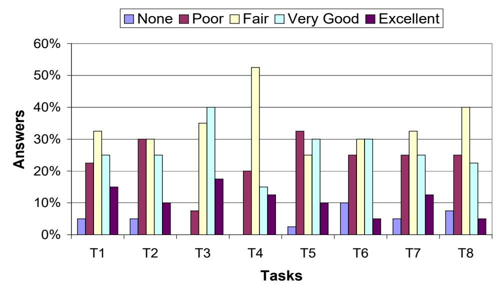

Figure 4: Understanding of various tasks of normalization.

Candidate key evaluation and choosing a functional dependency based on which to perform decomposition seem to be more difficult for students compared to other tasks. Also, most students find even more difficult repetitively combing the tasks in an overall normalization process as only 23% consider they have very good or excellent knowledge of the task, while 25% believe to have only poor knowledge.

The most important finding from this preliminary test was the discovered inconsistencies. For instance, while for closure evaluation of a combination of attributes 30% of the students answered with Poor, only 7% answered the same for the evaluation of a candidate key, and just 20% answered the same for all candidate keys evaluation. As candidate key evaluation requires computing the closure of combinations of attributes this reveals that the students have problems understanding the entire process and how the components are combined and a large number of them believe they have a better grasp of the topic than they actually have.

Next, the students were given a sheet of practice exercises that they were asked to complete using the normalization tool. The exercise sheet included a number of simple tasks the students were asked to perform using particular functionalities of the normalization tool. For instance, they were asked to create a new database schema, use and alter an existing schema, use the tool to evaluate candidate keys, BCNF decomposition, export their results and so on. Furthermore, the exercises included ones where the students were asked first to evaluate their own solution to a normalization problem and then use the tool to verify their solution. Afterwards, the instructor also showcased a brief case-study using the normalization tool. Finally, at the end of the pilot practice the students completed an extensive questionnaire for the tool's evaluation.

5 RESULTS

Our questionnaire consisted of three parts. The first evaluates how the students view normalization, the second measures the usefulness and user-friendliness of the normalization tool as a software tool, and the third focuses on its educational value.

5.1 Importance of Normalization

We questioned how the students view the database course and recorded the answers on a 5-point Likert scale. In particular, the two first questions were on how interesting and useful and on how difficult, starting from *Not at All* to *Very Much*, the students find the database course compared to their other courses. As Fig. 5(left) shows, 94% of the students rate databases with a 4 to 5 of the Likert-scale with respect to usefulness, but 41% also give the same rate to its difficulty Fig. 5(center).

The second part of questions concerned the topics covered in the database course again with respect to their usefulness and difficulty. As we expected SQL is considered the most useful subject in the database course, with *mean* (μ) equal to 4.53 and *standard deviation* (σ) equal to 0.93 ((Fig. 6(left)), as it is the most practical. However, 62.5% of the students also rated normalization with a 4 to 5 likert-scale ($\mu = 3.68$, $\sigma = 0.94$). While encouraging these results are considered slightly biased as the students who participated in the study volunteered showing their interest in the topic. However, confirming our intuition normalization is perceived as the most difficult topic (Fig. 6(center)), 56% of the students rate it as *much* and *very much* difficult ($\mu = 3.64$, $\sigma = 0.78$) indicating that the traditional teaching methods are not sufficient to help students understand the subject. Finally, Fig. 6(right) shows how difficult the students find the different normal forms from 1NF to 4NF as well as the concepts of functional dependencies (FDs) and multi-valued dependencies (MVDs). 4NF is considered the most difficult of all ($\mu = 3.53$, $\sigma = 0.25$), with BCNF a close second ($\mu = 3.42$, $\sigma = 0.8$). In general, the more strict the normal form the harder the students find it. In particular, based on our experience, in BCNF, the use of the multiple candidate keys is what usually confuses students.

5.2 Usability and Ease of Use

Table 2 presents the largest part of the questionnaire, questions $Q1$ to $Q28$ and their corresponding results. The responses follow a typical 5-point Likert-scale from *Strongly Disagree* to *Strongly Agree*. We report the percentage for each response, along with its mean, median, standard deviation ($StdDev$) and confidence ($Conf$).

We consider user-friendliness and usability very important as we believe it plays an important role in convincing students to use the tool for practice. Thus, following the USE questionnaire [13], queries $Q1$ to $Q8$ (Table 2) aim at evaluating the ease of use and user-friendliness of the normalization tool, while queries $Q9$ to $Q15$ measure usability and user-satisfaction.

Overall the response was positive with respect to both user-friendliness and ease of use and usability. Only $Q7$ has a median below 4 (i.e. 3), while just 4 out of the first 15 queries have also a mean below 4. The students seem to expect more continuous support based on the responses of $Q7$, so we plan to integrate short instruction texts in the tool to aid students in selecting the next operation, especially on the interactive step-by-step process.

To better assess how easily the students familiarize themselves with the tool, they were provided with no initial instructions but had access to its manual and could ask assistance from the instructor. As the results to $Q4$ and $Q5$ show, most students learned to use the tool quickly without problems, and 67.5% did not require any assistance based on the responses of 4 and 5 on the Likert-scale to query $Q8$.

Particularly important is also that the students consider that the tool saves them time and is useful, according to queries $Q10$ and $Q11$. More than 50% responded with 5 out of the Likert-scale, which indicates that they are likely to use it.

The students were also asked to answer how much they considered that the use of the tool would help them in their performance in the final exams in the database course. As shown in Fig. 5(right), 70% of the students think that the tool will help them in their exams showcasing the usefulness of the tool and their satisfaction with it.

5.3 Educational Value

Questions $Q16$ to $Q24$ aim at assessing how the students view the tool with respect to the learning process, while $Q25$ to $Q28$ aim at evaluating the students' view with respect to the use of educational tools in the teaching process in general.

As shown by the results to these last 4 questions, the students view educational tools positive in general, and most encouraging is that they found them especially useful for their practice, as $Q26$ has

Table 2: Questionnaire Results

		Mean (μ)	StdDev (σ)	Median	Conf	Strongly Disagree (%)	Disagree (%)	Neither Agree Nor Disagree (%)	Agree (%)	Strongly Agree (%)
Q1	It is easy to use.	4.2	0.65	4	0.2	0	2.5	5	62.5	30
Q2	It is user-friendly.	4.1	0.63	4	0.2	0	0	15	60	25
Q3	The messages, dialogues and menus are written simply and comprehensibly.	4.18	0.71	4	0.22	0	0	17.5	47.5	35
Q4	I learned to use it easily and quickly.	4.48	0.6	5	0.19	0	0	5	42.5	52.5
Q5	I can use it without written instructions.	4.15	0.95	4	0.29	0	7.5	15	32.5	45
Q6	The volume of the text and provided information does not create confusion.	3.45	1.13	4	0.35	10	7.5	22.5	47.5	12.5
Q7	It provides continuous support for the various operations.	3.4	0.84	3	0.26	0	12.5	45	32.5	10
Q8	I did not need help/guidance to use it.	3.58	1.08	4	0.34	5	15	12.5	52.5	15
Q9	I am satisfied with it.	4.15	0.7	4	0.22	0	0	17.5	50	32.5
Q10	It is useful.	4.53	0.64	5	0.2	0	2.5	0	40	57.5
Q11	It saves me time when I use it.	4.5	0.64	5	0.2	0	2.5	0	42.5	55
Q12	It does what I expected it to do.	4.08	0.66	4	0.2	0	0	17.5	57.5	25
Q13	It requires the minimum number of steps to accomplish what I want.	4.08	0.66	4	0.2	0	0	17.5	57.5	25
Q14	All its operations are fully integrated.	3.75	0.81	4	0.25	0	5	32.5	45	17.5
Q15	I would recommend it to a fellow student.	4.63	0.63	5	0.19	0	0	7.5	22.5	70
Q16	It helps solving normalization problems.	4.35	0.66	4	0.20	0	2.5	2.5	52.5	42.5
Q17	It makes me more confident about my solutions to normalization problems.	4.43	0.71	5	0.22	0	0	12.5	32.5	55
Q18	It helps me find my mistakes easily.	4.2	0.69	4	0.21	0	2.5	7.5	57.5	32.5
Q19	It provides sufficient explanations for my mistakes.	3.7	0.91	4	0.28	0	10	30	40	20
Q20	The solutions it provides are clear and easily understood.	4.01	0.66	4	0.2	0	2.5	10	65	22.5
Q21	It helps the instructor in teaching normalization.	4.15	0.76	4	0.24	0	0	22.5	40	37.5
Q22	It helps me practice on the normalization process on my own.	4.6	0.59	5	0.18	0	0	5	30	65
Q23	It makes the subject of normalization more interesting.	3.63	1.03	4	0.32	2.5	10	32.5	32.5	22.5
Q24	It makes the subject of normalization more easily understood.	4.13	0.56	4	0.17	0	0	10	67.5	22.5
Q25	It helps the instructor in teaching.	4.25	0.71	4	0.22	0	0	15	45	40
Q26	It helps me practice on my own	4.45	0.68	5	0.21	0	0	10	35	55
Q27	It helps me in the learning process.	4.28	0.75	4	0.23	0	2.5	10	45	42.5
Q28	It makes learning more interesting.	4.05	0.93	4	0.29	0	10	10	45	35

Table 3: Basic Functionalities

	Mean (μ)	StdDev (σ)	Median	Conf	Not at All (%)	Slightly (%)	Averagely (%)	Much (%)	Very Much (%)
Closure Evaluation.	4.05	0.85	4	0.26	0	5	17.5	45	32.5
Candidate Key Evaluation.	4.55	0.6	5	0.19	0	0	5	35	60
Automatic Decomposition.	4.58	0.64	5	0.2	0	2.5	0	35	62.5
Step-by-Step Decomposition.	4.35	0.95	5	0.29	2.5	2.5	10	27.5	57.5
Closure Evaluation.	4.25	0.81	4	0.25	0	2.5	15	37.5	45
Candidate Key Evaluation.	4.25	1	5	0.31	2.5	5	10	30	52.5
Automatic Decomposition.	4.15	0.83	4	0.26	0	2.5	20	37.5	40
Step-by-Step Decomposition.	4	0.85	4	0.26	0	5	20	45	30

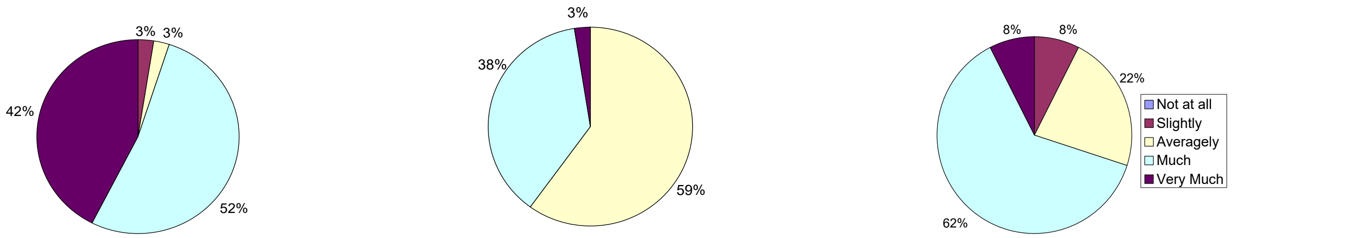

Figure 5: (left) Usefulness and (center) difficulty level of databases, and (right) level of improvement in students performance.

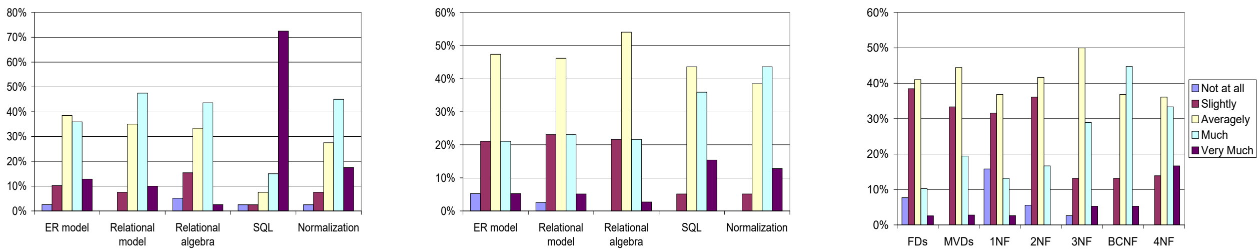

Figure 6: (left) Usefulness and (center) difficulty of topics in the database course, and (right) of the various normal forms.

a very high mean of 4.45 and a median of 5. Our tool seems to be especially well-perceived as question *Q22* has an even higher mean of 4.6 and 95% of the students responded with a score of >4 on the Likert-scale. In addition, it helps students in solving normalization problems (*Q*16) according to 94.5% who answered with 4 to 5 on the Likert-scale, and the solutions it provides are well understood (*Q*20) with 87.5% of the responses being ≥ 4. However, while the students consider that the tool helps them significantly in finding their mistakes (*Q*18) with 90% answering with 4 to 5, they would prefer more extended explanations on their mistakes as only 60% find them fully satisfactory (*Q*19).

As a teaching aid to the instructor, the tool is also considered useful, though not as useful as for self-training, as the responses to question *Q21* indicates with a mean of 4.15, and 77.5% of the responses from 4 to 5 on the Likert-scale.

Furthermore, the use of the tool made the subject of normalization more easily understood with respect to the traditional teaching methods of solving normalization problems on the board (*Q*24 with a mean of 4.13), while it also makes it more interesting for around half of the participating students (*Q*23).

We also wanted to evaluate the basic functionalities of our tool separately. Table 3 summarizes our results using another 5- point Likert-scale. The first four lines of Table 3 assess how useful the students found each of the separate functionalities, while the last four lines assess to what degree the students understand each functionality and the solutions it produces. Overall all functionalities were evaluated very positively, but automatic decomposition seems to be the one the students consider the most useful with a mean of 4.58. This indicates that the students seem to prefer to overview the entire solution of their normalization problems rather than driving the decomposition process themselves. However the interactive step-by-step solution also rates very positively with a mean of 4.35 and 85% of the responses to from 4 to 5 on the Likert-scale. Based on the responses of the students regarding its ease of use, our intuition is that if we integrate more short instructions in each step of the interactive process more students will be motivated to use it, as some found it slightly difficult to use. With respect of how understandable the functionalities and their solutions are, all score similarly with a mean of 4 to 4.25, with the step-by-step decomposition again scoring the lowest confirming our intuition for a need of more instructions.

Table 4: Performance in the Final Examination

	Control Group		Experimental Group		t-test	
	Mean	Std Dev	Mean	Std Dev	t	p
BCNF	3.94	3.13	6.5	3.4	5.26	0.0016
Other	3.55	1.79	4.71	1.78	2.43	0.02
Total	3.61	1.8	4.98	1.87	2.88	0.0064

5.4 Performance Evaluation

In the final part of our study, we wanted to evaluate whether the use of the tool improved the performance of the students in their final exams on the database course. To this end, we compared the performance of the group of 40 students that participated in the pilot study to that of a control group consisting of 227 students that take the same course. We measured their performance in the BCNF subject contributing 15% in the final examination as opposed to their performance in the other subjects which were an ER design, a transformation from an ER to a relational schema, and an SQL subject which contributed 20%, 15% and 50% respectively to the total grade. We normalized (i) their total performance, (ii) their performance in the BCNF subject and (iii) in the other topics except BCNF, to a 0 to 10 scale and report our results in Table 4.

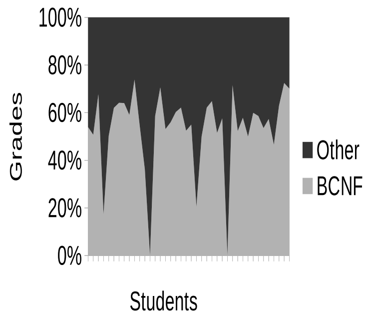

Figure 7: Performance of students in BCNF and other topics.

The mean of the experimental group in BCNF is 6.5, much greater than the corresponding 3.94 of the control group. Performing also a t-test we can see a t of 5.26 and $p = 0.0016$ showcasing that our tool did in fact have an impact on the performance of the experimental group. However, the results in the other subjects also show that the experimental group outperforms the control group with a mean of 4.71 compared to 3.55. The corresponding t-test also has $t = 2.43$ and $p = 0.02$ showing that the control group seems to overall have a better performance. This is explained as the students that volunteered for the pilot study are more diligent students in general and interested in the database course in particular. Therefore, further study is required to better assess the impact of the tool in the student's performance. These first results are however very encouraging. First of all t is much larger when evaluating the performance of the groups in the BCNF subject. Further, based on Fig. 7 that shows the percentage of the total grade of each student of the experimental group that is from the BCNF subject as opposed to other subjects, we can see that the BCNF subject in most cases accumulated the largest grade percentage with respect to the other topics of the exam.

5.5 Discussion

We summarize our results with respect to the four research questions we posed for our study. Regarding the 1st question, how students perceive normalization, our findings are that though fairly interesting they find the subject of normalization rather difficult and thus any effort such as our normalization tool that enables them to practice more is a step towards the right direction. Regarding the 2nd question of whether the use of the tool makes the subject more interesting, according to the responses of question $Q23$ our tool succeeds to a degree as half of the participating students answered with 4 to 5 on the Likert-scale. Since the students find the subject interesting even without the use of the tool, we intend to focus on improving the explanations and guidance the tool provides so as to help the students better understand what they perceive as a difficult subject. With respect to the 3rd question, we received positive feedback on both accounts. The students considered the tool as extremely useful for self-training but they also found merit in its use as a teaching aid. Regarding the 4th question, the primary results we derived for the improvement in the students performance in BCNF normalization problems as demonstrated in their exams were very encouraging, though we must not ignore the limitation of the study as the participating students were diligent ones.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a study of an educational tool for relational database normalization to the Boyce-Codd Normal Form. The tool is designed to both assist educators in explaining database normalization, and also enable self-training by the students by providing a simple and easy to use interface, thorough explanations and instructions for each operation. The tool was applied to a group of students and was evaluated both with respect to its usability and user-friendliness, as well as to its educational value. The results attained through anonymous questionnaire show positive feedback by the students that consider that the tool helps them understand a topic they consider difficult and improves particularly the self-training process. We also studied the effect of the tool in the students performance and reported significant improvement. As the tool is currently limited to BCNF decompositions, we plan to extend its functionality to support other normal forms, i.e., 2NF, 3NF and also 4NF with the addition of multi-valued dependencies. Furthermore, from the suggestions we gathered from the students participating in the study, we plan to implement a functionality that enables students to submit their solutions and checks them for validity, as well as enhancing the instructions during the step-by-step decomposition process.

ACKNOWLEDGMENTS

Partially funded by the Graduate Program of the Department of Applied Informatics, "M. Sc. in Applied Informatics" of the University of Macedonia.

REFERENCES

[1] L. Ahmedi, N. Jakupi, and E. Jajaga. 2012. NORMALDB - A Logic-Based Interactive e-Learning Tool for Database Normalization and Denormalization. In *eL&mL, The Fourth Int. Conf. on Mobile, Hybrid, and On-line Learning*. 44–50.
[2] N. Arman. 2006. Normalizer: A case tool to normalize relational database schemas. *Information Technology Journal* 5, 2 (2006), 329–331.
[3] W. Armstrong. 1974. Dependency Structures of Data Base Relationships. In *IFIP Congress*.
[4] Stefanidis C. and Koloniari G. 2016. An interactive tool for teaching and learning database normalization. In *Proceedings of the 20th Pan-Hellenic Conference on Informatics, Patras*. Article 18.
[5] R. Cho. n.d.. Relational Database Tools. (n.d.). http://raymondcho.net/RelationalDatabaseTools/RelationalDatabaseTool
[6] E. F. Codd. 1974. Recent Investigations into Relational Data Base Systems. In *IFIP Congress*.
[7] Y. V. Dongare, P. S. Dhabe, and S. V. Deshmukh. 2011. RDBNorma: - A semi-automated tool for relational database schema normalization up to third normal form. *CoRR* abs/1103.0633 (2011).
[8] H. Du and L. Wery. 1999. Micro: A normalization tool for relational database designers. *Journal of Network and Computer Applications* 22, 4 (1999), 215 – 232.
[9] A. Floyd and H. Du. 2014. Edna: a Software Tool for Verifying Normalisation of Relations during Logical Database Design Process. In *12th HEA STEM Workshop on Teaching, Learning and Assessment of Databases*. 51–61.
[10] N. Georgiev. 2008. A Web-Based Environment for Learning Normalization of Relational Database Schemata. masters thesis, Umea university, Sweden. (2008).
[11] Y. Gu, S. Shen, J. Wang, and J. U. Kim. 2015. Application of NoSQL database MongoDB. In *2015 IEEE Int. Conf. on Consumer Electronics - Taiwan*. 158–159.
[12] H. J. Kung and H. L. Tung. 2006. A Web-based tool to enhance teaching/learning database normalization. In *SAIS 2006 Proceedings*. 251–258.
[13] A. M. Lund. 2001. Measuring Usability with the USE Questionnaire. *Usability interface* 8, 2 (2001), 3–6.
[14] W.Y. Mok, J. Fong, and K. Wong. 2015. Designing Redundancy-Free XML Schema: A Smallest Closure Approach. *International Journal of Cooperative Information Systems* 24, 04 (2015).
[15] Database Design Tool. n.d.. (n.d.). http://uisacad5.uis.edu/cgi-bin/mcrem2/database_design_tool.cgi
[16] A. Yazici and Z. Karakaya. 2007. JMathNorm: A Database Normalization Tool Using Mathematica. In *Computational Science – ICCS 2007, Part II*. 186–193.