Intelligent Tutoring System for Database Normalization

Neha Mendjoge
D. J. Sanghvi
Mumbai, India
neha.mendjoge@djsce.ac.in

Abhijit R. Joshi
D. J. Sanghvi
Mumbai, India
abhijit.joshi@djsce.ac.in

Meera Narvekar
D. J. Sanghvi
Mumbai, India
meera.narvekar@djsce.ac.in

Abstract – **A database is said to be efficient, if it performs operations like delete, update and insert in an unambiguous way. These operations can be performed efficiently by normalizing the database. By normalization, the tables in a database are organized in a well-structured manner that reduces the generation of redundant data as well as ensures that the data dependencies make sense, i.e., all the functional dependencies are satisfied. It is observed that students find it difficult to understand the concepts of normal forms while learning a Database Management Course and also they find it difficult to obtain smaller well-structured relations by applying normalization concept. Very few commercial design tools are available to assist students in learning Normalization. Consequence is a big challenge for educators, in both, teaching as well as motivating students in learning Data Base Design. Keeping in mind all these aspects, we decided to develop an intelligent tutor, which will teach users the process of normalization and help them in normalizing the databases. While solving the problems, the tutor will also provide hints and feedback, if the user demands. The tutor also keeps track of the performance of the user to plan its teaching strategies.**

Keywords – ITS, normalization, database

I. INTRODUCTION

Most computer programming application requires a database to store data. Studies have shown that the vast majority of the content in the World Wide Web resides in the deep web sources, which store their content in backend databases. This has been growing by leaps and bounds. Database Normalization is used to organize the database in a well-structured manner so that the generation of redundant data is reduced and the data dependencies make sense, i.e., all the related data is stored in one table. It is important to store all the related data in one table to avoid data anomalies such as insert, update and delete. By applying the process of database normalization, we ensure that the data is stored in a way in which it will be efficient to insert, update and delete the data. Relational database is an important subject in the field of Computer Science and Information Technology curriculum. Database normalization is one of the topics, which is invariably a part of database courses. Keeping in mind all these aspects, it is important for students to apply this process of database normalization in a correct way. Very few commercial design tools are available to assist students in learning Normalization. These tools are covered in the next Section.

II. LITERATURE SURVEY

The tools or tutors currently available to learn database normalization are:

1. Automating the Normalization Process for Relational Database Model: The user is required to specify basic information associated with the database, which includes Table Name, Number of attributes, Attribute name along with its data type and constraints, which have been imposed on it and the functional dependencies associated with the table. Once all the input is accepted from the user, the relations are normalized up to 3NF according to a predefined algorithm. The user is provided with the normalized tables in different formats in a ready to use form [1].
2. A Web Based Relational Database Design Tool to Perform Normalization: It is a web based relational database design tool and shows each and every step in the normalization process. The user is required to input the name of the relation, number of attributes and the set of functional dependencies. Once this is done, the tool computes the set of all possible candidate keys, primary key, prime and non-prime attributes. The functional dependencies are also classified into full and partial FDs and finally the relation is decomposed into 3NF [2]. The tool also has a provision to show the decomposition of the table into 2NF and 3NF separately. The tool generates all the possible solutions as there can be multiple solutions to a given problem. This tool can normalize any relation whose set of FDs consist of 30 redundant attributes and can handle as many as 50 complex functional dependencies and computes all the possible solutions that exist.
3. A Web-based tool to enhance teaching/learning database normalization: It is a web based tool for database normalization. The tool requires the user to enter all the functional dependencies one by one. The tool decomposes the functional dependencies automatically, to satisfy the third normal form [3]. This tool can work for maximum 10 functional dependencies.
4. A Web-Based Environment for Learning Normalization of Relational Database Schemata (LDBN): The users are required to choose an assignment from a list of assignments submitted by teachers. 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 [4]. The user now had to determine the canonical cover of the functional dependencies, decompose the relation to 2NF, 3NF and BCNF and determine a primary key for each new relation/table.

5. Web Based E-Learning System for Data Normalization: This tool consists of several case studies, i.e., schemas for normalization. The learner can attempt to normalize the schema, getting feedback after each step. The information is retrieved dynamically from the database and the user's answers are checked against the solution stored in the database and accordingly the feedback is given [5]. New problems cannot be added to the database. Also, the solutions to exercises are statically stored in the database. So, the system checks the correctness of user solution by simply matching with the stored solution.

6. NORMIT: It is a web enabled tutor that provides users with a step by step guide or tutorial to perform normalization. It also provides users with certain practice problems that users can solve and submit to the system. The system then verifies the user answer with the stored teacher answer and accordingly provides feedback to the user [6].

   NORMIT is a problem solving environment that focuses on problem solving only. It does not provide in-depth information of the concepts. The system describes certain basic concepts, if it finds that the user is not able to apply the concept correctly to the given problem. The problems for normalization are predefined.

Having looked at the various tools available for learning normalization, we now analyze these tools. NORMIT is the only ITS among these tutors whereas others are web-based which lack intelligent component, viz., appropriate intervention and interaction. These web-based tutors accept the tables and functional dependencies from the user and on button-click generate the normalized tables. These tutors do not help the users in learning the process of database normalization to obtain smaller well-structured tables. In NORMIT, for most of the problems presented to the user, the answer is expected in the form of yes or no. Due to this, the user is not getting a feel of solving the normalization problem. The major drawback of NORMIT is that the tutor is available only to the students of University of Canterbury. Students find it difficult to understand the processes of normalization in one go. To understand the normalization process, one needs practice and right time feedback when stuck in between while solving a problem. The available tutors are not addressing these needs of the users. Considering the importance of the topic and needs of the users, it is decided to develop an Intelligent Tutor for learning Database Normalization Process. Our system provides a rich and effective learning environment that helps the user in learning database normalization process.

The rest of the paper is organized as follows. Section III discusses our approach for developing the tutor. Section IV throws light on the design and implementation details. Evaluation of the system is shown in section V and results are presented in section VI. The paper ends with conclusion and direction for future work.

III. OUR APPROACH

Students are always comfortable to learn on a 1:1 basis from a teacher where the students are taught a topic by the teacher and then students solve a few problems or questions to practice the topic. If the teacher feels that the student is deviating from the correct solution, the teacher immediately informs this to the student by providing a few hints. To incorporate these qualities of one to one teaching in our tutor, we have followed the teaching approach where the focus is on continuous practice, right time intervention and appropriate feedback [7].

Figure 1 shows a table named 'STUFF' having attributes H,I,J,K,L,M,N,O and functional dependencies FD1, FD2, FD3 and FD4. Using this data, the system generates queries for the user for obtaining the candidate key, decomposing to 2NF and decomposing to 3NF. This information is also used by the system to generate its own response. Let us consider the table shown in Figure 1, assuming that the system has asked the user to give the third normal form of it. Assume the response given by the user is H,I,J,K,L as table 1; J,M as table 2; K,N as table 3 and L,O as table 4.

STUFF(H,I,J,K,L,M,N,O)
FD1:H,I$\rightarrow$J,K,L
FD2:J$\rightarrow$M
FD3:K$\rightarrow$N
FD4:L$\rightarrow$O

Figure 1: Sample Question

To check the correctness of user's answer, system generates its own solution. For that the system has to first generate the candidate key using the functional dependencies. To do so, the system finds the closure of each of the LHS of the functional dependency. Closure of an attribute includes the attribute itself and the attributes on the RHS of the functional dependency. Now, from these set of attributes, the system checks if there is any other functional dependency with LHS as one or more of the attributes of the set. If yes, then the closure is updated to include the RHS of the newly considered functional dependency. If this closure gives all the attributes of the table then, the LHS becomes the candidate key. Now, to decide the candidate key, the system checks whether all the attributes of the table have been obtained in the closure. If so, the candidate key is the LHS of the first FD that was considered.

System uses this candidate key to decompose the table to second normal form. To do so, system checks for full functional dependency, partial functional dependency and no functional dependency. If there exists any functional dependency having attributes in the LHS, which are the candidate key, then such a FD are called as fully functional dependent on the candidate key. In Partial functional dependency, few attributes of the LHS match with the candidate key whereas in no functional dependency none of the attributes of the candidate key matches with any of the LHS of the functional dependencies. This information is used by the system to decompose the tables. Functional dependencies categorized as partially dependent are stored in a separate table, while functional dependencies categorized as full and no dependency are kept in one table.

The system now proceeds to normalize the table to the third normal form. A table is in third normal form if it satisfies second normal form and there are no transitive dependencies in the table. To check for transitive dependency, the system uses the LHS and RHS of the functional dependencies. A transitive dependency means that one or more attributes in the RHS of one functional dependency is also present in the LHS of some other functional dependency. If a transitive dependency exists, the system decomposes the table further by storing attributes of the functional dependency having the transitive attribute in the LHS; in a separate table. Each of these tables are stored in a separate array by the system. Now, system compares the user response with its own. After comparing the arrays of the user solution and system solution, the system informs the user if the answer given by him/her is correct or incorrect. In this case, the feedback will be given to the user by the tutor model.

While solving the problem, the user may require some help from the tutor. This help is provided in the form of hints to the user. The tutor model analyzes where and why the user answer is deviating from the correct answer and accordingly provides the appropriate hint. There may be some questions that the user may not be able to solve in spite of obtaining the hints. In such scenarios, the tutor model will help the student in solving the question. A scaffold is designed having series of questions on a particular topic. The tutor helps the student in solving the question in a step-by-step manner by asking questions for each step. If the answer obtained from the user is correct, then the tutor moves to the next step, else it informs the user about the incorrect answer and ask the user to rethink and submit the answer again. Once the complete solution is obtained, then the next question is presented to the user.

Our tutor allows the user to practice the topics such as obtaining the candidate key or decomposition of given table to second normal form or third normal form as individual exercises. This kind of pedagogy is provided because it may happen that the user is well versed with the method for obtaining the candidate key, but lacks confidence in normalizing the table to second and third normal form. In such cases, the user can directly solve questions for second and third normal form. She/he does not have to go through the candidate key module.

In this way, adequate scaffolding, right time intervention and appropriate feedback help the user in learning the database normalization process. Now, let us see the design and implementation of our system.

IV. DESIGN AND IMPLEMENTATION

We can divide the system architecture into 3 parts as shown in Figure 2. These are the presentation of content to the user, the knowledge base and the processing logic.

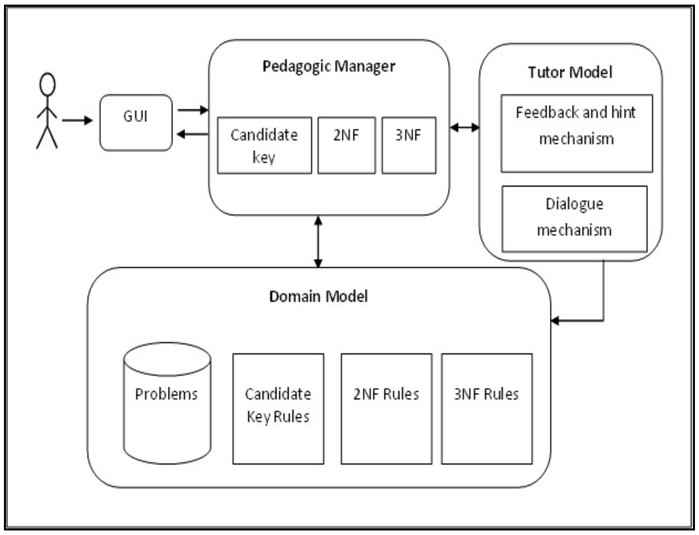


Figure 2: System Architecture

The user can access the system through the GUI. The domain model consists of a database consisting of the problems to be asked to the user and a set of rules pertaining to the candidate key, second normal form and third normal form. The pedagogic manager consists of modules for candidate key, second normal form and third normal form. The user can choose to solve problems from any of the modules present in the pedagogic manager. Since the user can make a mistake while solving a problem, the tutor model must inform the user of the committed mistake. To inform the user about his mistake, the tutor model makes use of the feedback and hint mechanism. This mechanism provides appropriate hints to indirectly tell the user of his/her mistake. The tutor model also makes use of the dialogue mechanism to help the user in correcting the mistake.

The pedagogic manager as shown in

Figure 2 allows the user to select either candidate key or 2NF or 3NF problems in any order. The tutor presents a problem to the user. The user is now expected to enter the answer. At the same time, the system also generates the solution to the same question dynamically. To generate these solutions dynamically, it uses the rules to generate solutions for candidate key, second normal form and third normal form. The solution submitted by the user is verified with the one, which is generated by the tutor. The tutor model consists of the feedback and hint mechanism and dialogue mechanism. Interaction in e-learning is in the form of intervention by the tutor in the form of feedback and hint provided to students when they make mistake while solving a problem [5]. The feedback mechanism is used to inform the user whether the answer submitted by them is correct or not. The hint mechanism, on the other hand, can be used by the user to obtain clues regarding the correct answer. The user uses the given clue and his knowledge to reach to the correct answer.

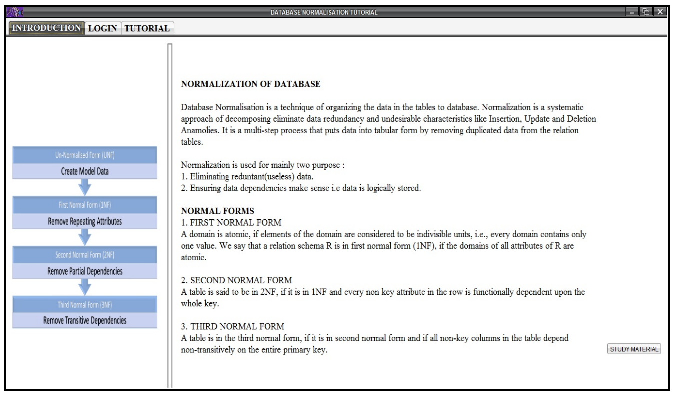
But there can be users, who want the tutor to teach them how to obtain the correct answer. In such scenarios, the user can use the dialogue mechanism. In this mechanism, the tutor and user hold a dialogue with each other, which will teach the user the steps to obtain the correct answer. This dialogue mechanism enacts like the human tutor who teaches the students the solution in a step-by-step manner.

Let's consider an example. Let's say, the user selects to solve a problem on third normal form. Assume the question is as given in Figure 1. The user must now enter his/her answer, i.e., tables that satisfy the third normal form. At the same time, the system uses the 3NF rules stored in the domain model to dynamically generate the solution. The tutor model now uses the Feedback mechanism to compare the solution given by the user and the one generated by the system. If the user's answer matches with the system's answer, the user is notified that his/her answer is correct, else he/she is notified that the answer is incorrect. To correct his answer, the user can take help of hint mechanism. Through this mechanism the user interacts with the tutor. During interaction, the tutor provides some clues so that the user is able to give the right answer. For generating these hints, the system makes use of the 3NF rules. Some users may require additional help from the tutor to obtain the correct answer. For this, the tutor provides a Dialogue mechanism. In the dialogue mechanism, the tutor holds a conversation with the user by asking questions. If the answer given by the user is correct, then the tutor asks the next question. This takes the user one step closer to obtain the correct answer. For example, if the user needs help in obtaining the correct answer for third normal form, then the tutor first asks the user to submit the tables that satisfy the second normal form. This is to check whether the user understood the concept of second normal form. If not, then the tutor asks the user to practice the second normal form. However, if the answer to second normal form is correct, then the tutor asks the user to check for transitive dependency in the first table given by the user. If the user says that a transitive dependency exists in the first table, then the tutor informs the user that in third normal form a table cannot have any transitive dependency and hence, he/she must split the table. The tutor asks the user to enter the split table again to verify whether the user has understood where and why the table must split. If the answer entered is correct, then the process is repeated till all the tables are exhausted. Through the dialogue mechanism, the system teaches the concepts which were not clear to the user there by overcoming the doubt in user's mind about a particular topic resulting in the correct answer from user.

Having looked at the working of the tutor, now let us take a walk through our tutor as a user.

V. WALKTHROUGH

This research work focuses on building a framework for teaching database normalization. In this research, we have emphasized on developing algorithms for obtaining the candidate key, second normal form, third normal form and generating appropriate feedback and hint. Now, let us take a walkthrough our tutor as a user. The snapshot given in Figure 3 briefly introduces the normalization process with the help of flow chart. As seen in the menu bar, there are three tabs viz., Introduction, Login and Tutorial. Initially the tutorial tab will be inactive. After logging in to the system, the tutorial tab will be activated. After clicking on the tutorial tab, the three sub-tabs namely, candidate key, second normal form and third normal form are displayed (see Figure 4). These tabs are used for practicing and in turn learning the normalization concepts.



NORMALIZATION OF DATABASE

Database Normalisation is a technique of organizing the data in the tables to database. Normalization is a systematic approach of decomposing eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anamolies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purpose :
1. Eliminating reduntant(useless) data.
2. Ensuring data dependencies make sense i.e data is logically stored.

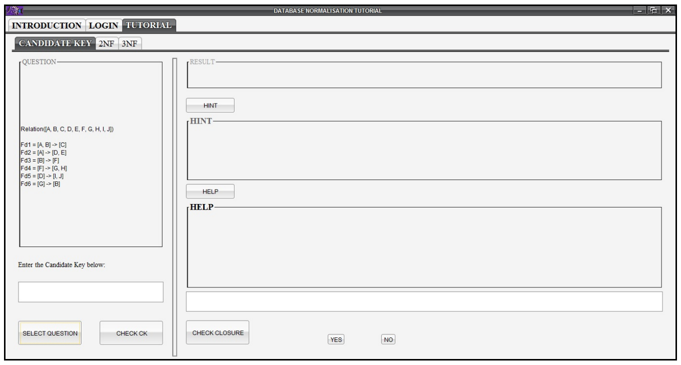
NORMAL FORMS
1. FIRST NORMAL FORM
A domain is atomic, if elements of the domain are considered to be indivisible units, i.e., every domain contains only one value. We say that a relation schema R is in first normal form (1NF), if the domains of all attributes of R are atomic.

2. SECOND NORMAL FORM
A table is said to be in 2NF, if it is in 1NF and every non key attribute in the row is functionally dependent upon the whole key.

3. THIRD NORMAL FORM
A table is in the third normal form, if it is in second normal form and if all non-key columns in the table depend non-transitively on the entire primary key.

Figure 3: Introduction Tab

The user is allowed to choose any tab for mastering the concepts of normalization, i.e., if a user wants to practice only second normal form, then it is not mandatory for him/her to take the lessons on candidate key. On clicking sub-tab viz., candidate key, 2NF and 3NF the user can see a screen which is divided into 4 areas, namely, question selection frame, result frame, hint frame and help frame. On clicking on the 'Select Question' button, the system selects a question randomly from the database and displays it in the question frame. Now, the user is required to enter his/her answer. On clicking the 'Check' button, the system compares the user's answer with its own and accordingly informs the user by displaying the message in the 'Result' frame.



Relation([A, B, C, D, E, F, G, H, I, J])

Fd1 = [A, B] -> [C]
Fd2 = [A] -> [D, E]
Fd3 = [B] -> [F]
Fd4 = [F] -> [G, H]
Fd5 = [D] -> [I, J]
Fd6 = [G] -> [B]

Enter the Candidate Key below:

Figure 4: Tutorial Tab

If the user answer is not correct, he/she may take the help of the tutor by pressing the 'HINT' button. The system provides maximum 4 hints but these hints are displayed one at a time to the user. Based on the given hint, the user tries to solve the question again and gives his/her answer. The system verifies the user answer and displays the message accordingly in the result frame. Still the answer is incorrect, then on clicking the hint button, the system displays the next hint. On the basis of this hint, user again gives his/her response. If the answer is not correct, the process continues. If the user exhausts all 3 hints, then the 4th hint which is the correct answer of the question is displayed in the hint frame.

Through 'Hint' section, if the user is not able to give the correct answer to the generated question, he/she can learn the topic by carrying out the dialogue with the system. To carry out the dialogue with the system, he/she needs to press the 'HELP' button. Various scaffolds are designed to teach the normalization topics to the user wherein the user is supposed to give the answers to the question generated by the system.

Let us say that the user wants to practice problems on finding the candidate key. On clicking the Select Question button, as shown in figure 4, a question is presented to the user. The user enters the answer for candidate key in the given text box. Let's say the answer entered by the user is 'A'. By clicking the 'CHECK CK' button, the system displays the message in the RESULT frame. Since the answer is incorrect, in this case, it displays "Wrong answer!!! Try again". Now, the user takes hints from the tutor. Since the user has clicked the HINT button for the first time, the level-1 hint is displayed. The hint is "Please check all the functional dependencies". Using this hint, the user tries to solve the question again. If the answer is incorrect, then the user can again take help from the system. The system provides in all four hints to the user. These are basically four levels as follows:

Level 1: It is a generalized hint, which asks the user to check if he or she has considered all the functional dependencies.

Level 2: This hint is with respect to the answer entered by the user. This hint tells the user how close his/her answer is to the actual answer.

Level 3: This is an indirect way of telling the user, which functional dependency can give the correct answer.

Level 4: This level discloses the answer to the user.

If the user finds that hint section is not appropriate to solve the question, then he/she can use the 'HELP' section. On clicking on the 'Help' button, user can carry out a dialogue with the system. Consider again the question generated in Figure 4 for which the user wants the help from the system to solve it. After clicking the help button by the user, the system asks a few questions, i.e., "input the fd no. you want to consider". Let us recall that to generate the candidate key we have to begin by choosing a function dependency from the ones given in the question. Hence, the first question generated by the tutor is "input the fd no. you want to consider". Assuming that the user enters fd no. 1. Now, the second question by the system is "find the closure of FD 1". The user must now enter the closure of FD 1, i.e., A,B $\rightarrow$ C. The answer entered by the user is a,b,c. If the answer is correct, the system informs the user by displaying "CORRECT ANSWER". Now, the system asks the user to check the transitive dependency between FD 1 and FD 2.

If the transitive dependency exists, then the user must again find the closure. Now, for FD 1 and FD 2, there is a transitive dependency due to attribute A. Hence, the closure will now be a,b,c,d,e. The response given by the user is also a,b,c,d,e. Since the answer is correct, then the system again asks the user to check the transitive dependency with the next functional dependency, i.e., FD 3. Again, there is a transitive dependency due to attribute B. The user responds with a,b,c,d,e,f as his answer. Since this answer is correct, the system again asks the user to check the transitive dependency with the next functional dependency, i.e., FD 4.

The user now responds with the a,b,c,d,e,f,g,h. Since the answer is correct, system asks the user to check for transitive dependency with FD 5. This transitive dependency is due to attribute D. To this question, the user responds with his answer as a,b,c,d,e,f,g,h,i,j. This time also the user answer is correct. Since all the functional dependencies are exhausted, the system now checks whether all the attributes of the table given in the question are obtained. This check is necessary because an attribute can become a candidate key if and only if it can produce all the other attributes of the table. The system now informs the user that all the attributes of the table are obtained by starting with the first functional dependency hence, the candidate key is the LHS of FD 1, i.e., A, B.

Thus, the scaffolding model helps the user in understanding the steps to be followed while obtaining the candidate key. Similar scaffolding process is also designed for 2NF and 3NF tutoring. Let us now take a look at the results obtained during the research and based on these results the analysis is carried out which is presented in the next section.

VI. RESULTS & ANALYSIS

We devised a strategy of conducting a pre-test and post-test on normalization and then compared the scores obtained by the learner in both tests. To carry out the test, 10 students were identified. These students were first asked to give the pre-test consisting of 3 questions. For each question, the students were required to find the candidate key, decompose the table to second normal form and third normal form. For each student, we recorded whether for each question he could find the correct answer for candidate key, second normal form and third normal form. The data in Table 1 shows how many students were able to get the correct answers.

Table 1: Analysis of Pre-Test

Q1			Q2			Q3		
CK	2NF	3NF	CK	2NF	3NF	CK	2NF	3NF
6	6	3	5	5	2	4	2	2

Later, the same sets of students were asked to use the tutor for a day after, which they were asked to take a test. The scores of post-test are also recorded and shown in Table 2.

Table 2: Analysis of Post-Test

Q1			Q2			Q3		
CK	2NF	3NF	CK	2NF	3NF	CK	2NF	3NF
8	7	6	7	7	5	6	5	3

The analysis of pre-test and post-test is presented in the form of graphs as shown in the Figure 5.

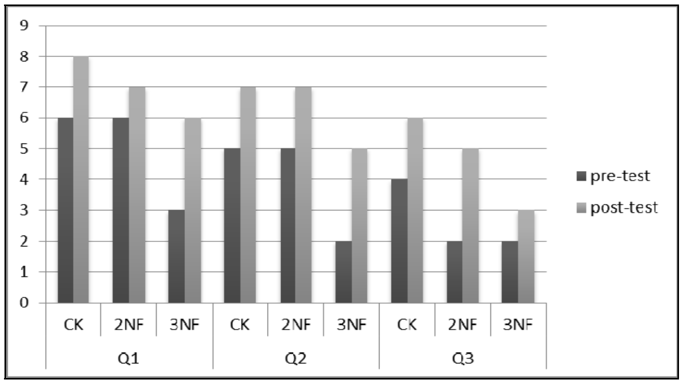


Figure 5: Analysis of Pre-Test and Post-Test

The results obtained are indeed very encouraging. From the graph shown in Figure 5, we can say that the students are able to solve the questions in a correct manner after studying and practicing the concepts using the tutor. As seen in Figure 5, the performance of students is improved after using the tutor, i.e., the post-test result is better than the pre-test result.

VII. CONCLUSION AND FUTURE SCOPE

In this research, we have developed an ITS to teach normalization process. The user can use our system to learn the database normalization concepts, namely, candidate key, second normal form and third normal form. Currently, the problems on the above mentioned topics are stored in the database. The system selects any problem randomly from the database and presents it to the user. At the same time, it generates its own solution to verify the correctness of the user's answer. The system also provides appropriate hints to the user and a dialogue mechanism to teach the user to find closure and normalize the schema to second and third normal form.

In future, the proposed system can be expanded to consider dynamic problem generation and normalize to the higher normal forms like BCNF, fourth normal form and fifth normal form.

The pedagogical model and student model also need improvement in terms of decision regarding the next question to be presented to the user. This decision would depend on the concepts that the user has already understood.

ACKNOWLEDGMENT

Dr. Sasikumar, Associate Director, CDAC, Mumbai, has supported and helped throughout the project with his patience and knowledge. Without him this project would not have been completed.

REFERENCES

[1] P. B. Alappanavar, R. Grover, S. Hunjan, D. Patil, D., Y. Girnar, "Automating The Normalization Process For Relational Database", International Journal of Engineering Research and Applications, Vol. 3, Issue 1, January -February 2013, pp.1826-1831.

[2] R. Vangipuram, R. Velpula, V. Sravya, "A Web Based Relational Database Design Tool to Perform Normalization", International Journal of Wisdom Based Computing, Vol. 1(3), December 2011.

[3] Kung, Hsiang-Jui and Tung, Hui-Lien, "A Web-Based Tool to Enhance Teaching/Learning Database Normalization" (2006). SAIS 2006 Proceedings. Paper 43.

[4] N. Georgiev, "A Web-Based Environment for Learning Normalization of Relational Database Schemata" , Master's Thesis in Computing Science, Department of Computing Science, Umea University, September 2008.

[5] Sara Javanbakht Yousefi, "Web Based E-Learning System for Data Normalization", Dublin Institute of Technology, Poster 2010.

[6] A. Mitrovic, "NORMIT, A Web-enabled tutor for database normalization", Auckland, New Zealand. Proceedings of the International Conference on Computers in Education ICCE, 2002, 3-6 Dec 2002. 1276-1280.

[7] A. R. Joshi, "Building an Intelligent Environment for Learning Indian Language: Marathi", Ph.D. thesis. NMIMS University, 2012.