# Designing and Development of Database Testing Tool

Taranpreet Kaur
Department of Information Technology,
Guru Nanak Dev Engineering college.
Ludhiana, Punjab.

Sumeet Kaur Sehra
Department of Information Technology,
Guru Nanak Dev Engineering college.
Ludhiana, Punjab.

## ABSTRACT

Tool enable user to perform all the operations from the beginning of data definition which includes Creating, Altering, Dropping and Truncate of table to the data manipulation that includes Selecting, Inserting, Updating and deleting of data, which is followed by controlling them that is commit and reset. In Data Definition we have commands of Creating and dropping of databases followed by creation of tables within a database. And it also allows altering the contents of table either by adding column(s) or removing them. Along with this, user can truncate the table that is removing the whole content of a table. In data manipulation, user has the provision of selecting a specific amount of data from a table by using "where" clause in the command of view data. User can also view data from different table(s) of databases by using sub-queries and join commands according to the requirement.
User can insert, update and delete a number of data items whenever and whichever required. At any instance user has a provision of commit and reset of transaction.

## Keywords

MySQL, LAMP, PHP, JSP, GUI, XML, IDM, TPC-C .

## 1. INTRODUCTION

Database testing performs the tests to verify the exact data values that have been retrieved/ fetched from the database either by the desktop-based application or web-based application. Data must match exactly as per the records which are stored in the database. Testing of database is one of the most critical testing that requires tester to be proficient in writing queries and procedures as well as checking table(s). The main aim of testing is to find software failures so that the faults may be discovered and therefore corrected. Though testing cannot guarantee the proper functioning of a product under all conditions but it can only establish the specific conditions under which product does not function properly.

The scope of software testing includes not only examination of source code but also execution of that code under different conditions and in various environments as well as examining the various aspects of code: whether it does what is meant to be done or what it is supposed to do and do what it needs to be done. In the current scenario of software development, a testing organization may have separate department from the development team as there are various roles for the members of a testing department. The derived information from software testing may be used in order to correct the process(es) through which software is being developed.

**Table 1: Procedures vs. Functions**

| **Procedures Vs Functions** | |
|---|---|
| **Procedure** | **Function** |
| • Execute as a PL/SQL statement<br>• Do not contain RETURN clause in the header<br>• Can return none, one, or many values<br>• Can contain a RETURN statement | • Invoke as part of an expression<br>• Must contain a RETURN clause in the header<br>• Must return a single value<br>• Must contain at least one RETURN statement |

## 1.1 Test Cases for Database Testing

**Consider the key Points for writing the test cases for Database Testing:**

- The number of arguments being passed
- The data type of arguments being passed
- The order of the arguments passed
- The return value
- The data type of the return value
- Testing each and every Stored Procedure, Cursor, Trigger which are used in the application
- Testing the application with respect to functionality of the application
- Used joins between the back end tables.
- Testing of database should be done both from the front end and back end.

## 1.2 Types of Testing

### *1.2.1Black Box Testing in Database Testing*

This type of testing does not require any knowledge of the internal working of the application for the tester as there is no need to access the source code of application and the tester is oblivious to the software architecture. Basically, while performing black-box testing, a tester will only interact with the user interface of the system by giving inputs and examining the corresponding outputs without any kind of knowledge about how and where these inputs are being worked on. Thus, Black-box testing includes testing of interfaces and integration with database, which involves:

- Verification of incoming data items
- Data mapping including meta data
- Verification of outgoing data items from query functions
- Techniques used in case of Black- Box testing are boundary value analysis and equivalence class partitioning.

### *1.2.2White Box Testing in Database Testing*

This type of testing requires thorough know how of the internal structure as well logic of the source code. That's why it is also referred as a open-box or glass testing. While performing White-box test, tester must possess deep knowledge of the internal working of the source code as he has to have a very clear look inside the code of the application and therefore must figure out which module or function or even smaller unit is not behaving properly.

White box testing mainly deals with the internal structure of the database application. But the specification details are kept hidden from the user.

- Performs testing of logical views and database triggers which supports database refactoring.
- Checks Referential Integrity rules.
- Validates database schema, data models, tables used in database etc.
- Execute module testing of database functions, views, SQL queries, triggers etc.
- Selects default values of table(s) in order to check consistency of database.
- Techniques such as condition coverage, cyclomatic complexity, statement coverage, decision coverage etc are used.

### *1.2.3 Grey Box Testing in Database Testing*

This type of testing involves limited know how of the internal working of the software application.In case of software testing, the term used: the more you know the better it is, holds a great weight-age while testing any application.

Mastering in system's domain always provides a tester an advantage over something with little or less domain knowledge. Unlike black-box test where tester has only access to user interface of the application, here in case of Grey-box testing tester is having access to database as well as to the design documents which allows him to prepare better data test and test scenarios while preparing the test plan.

## 1.3 Literature Review

Testing can be performed in application which is web-based or desktop-based and database used in the application can be SQL or Oracle.

Following are some of the research papers and the references which helped us to derive our conclusions and develop a technology to some extent.

Bogdan Walek and Cyril Klimeš illustrates that there are various tools for testing stored functions and procedures in the database, unit testing of databases, performance and load testing of Relational database but no tool for testing the logical model. Therefore, they proposed a tool for testing logical model of a database against its requirements for creation. Tool uses XML document for representation of requirement and for loading logical model of the implemented database JDBC methods are used. The proposed tool provides results in the form of tables with comparison of attributes and relationships, entities and tables.

Klaus Haller describes that White-box testing is the most crucial part of any software testing and quality assurance strategy. Testing of database applications demands the adoption of white-box testing, but it is not clear what adoption is required. So, rather than focusing on one problem and its solution, this paper highlights all of the main challenges being faced from a practitioner’s point of view. Beginning with a process of generic testing, it analyzes for each process step whether adoption is required or not and, if so, which adoptions are required, and redefining the concepts of test cases and its coverage. It discussed the state generation methods of database test and the problems associated with scheduling these test cases efficiently. Thereby, provides a road map for the emerging domain of testing database applications and enabling such testing useful for commercial development of software industry.

Harpreet Kaur and Dr. Gagan Gupta suggest that Testing is the most important parameter in Software Development Life Cycle. It main aims is to minimize the number of errors and thereby reduces the maintenance and overall cost of the software and producing a high quality software product. The aim of this paper is to conduct a comparative study of three automated tools namely Selenium free source, HP Quick Test Professional (QTP) and the Test Complete (TC) which are available in the market to determine their usability and effectiveness so that one can select a testing tool depending upon the type of application , budget and effectiveness required.

Raquel Blanco, Javier Tuya and Rubén V. Seco discuss that Testing any database application is a challenging task when it comes to designing of test cases where both user interaction and database have to be considered. It describes an approach which is specification-based in order to guide the designing of test inputs for a database application and to evaluate its test adequacy automatically. First of all, the system specifications are modeled: the structure of the database and the user interface are represented as Integrated Data Model (IDM) and the functional requirements are expressed as a set of business rules, written in IDM. Then, Criteria of MCDC is applied over the set of business rules that drives automatically test requirements, which helps in guiding the design of the test items. Finally, the adequacy of test inputs is evaluated automatically in order to determine whether the test requirements are covered or not. The results suggest that the test cases designed are able to detect faults that were located in the implementation of procedural code.

## 2. PROBLEM FORMULATION

**Data integrity**- The complete data belonging to each entity should be stored in the database. There should not be any missing data. Data integrity corresponds to the quality of data in the databases and to the level by which users examine the quality of data, reliability and integrity. Data integrity testing verifies that the data in the database is accurate and functions as expected within a given application.

Data Integrity testing involves:

- Checking whether or NOT any blank value and default value can be retrieved from database.
- Validating each and every value if it is successfully saved to database.
- Ensuring the data compatibility against old hardware and old versions of operating system.
- Verifying data in data tables can be modified or deleted.
- Running data tests for data files, including clip art, tutorials etc.

## 2.1 Correctness of the data stored in the database

### Data type Testing

**Data size testing**- tests the size of data size only at the front end, but it is essential to test it at back end separately.

### Database Performance

**Data Security**- In case of data migration checks Correctness and completeness of data.

If you are good in database also test for **Stored Procedure**, here one need to test Every Procedure separately, which covers:
The number of arguments being passed.

The order and data types of agruements passed. The data type of the return value.

**Input Item verification**- In this, verifies the input items such as text box, combination box, active X controls. Generally, developer is asked to test this during unit testing of the module.

## 3. OBJECTIVES AND METHODOLOGY

- To study different techniques for Database Test
- To develop a tool for Database Test
- To compare the proposed tool with the existing Database testing.

## 3.1 Methodology

### Tool for Data Base Testing Using:

**Swing**, is the primary Java GUI widget kit. It is a part of Oracle's Java Foundation Classes (JFC)—an API(Application Programming Interface) which is used for providing a graphical user interface (GUI) to the Java programs. Swings were developed in order to provide a more sophisticated set of GUI components as compared to the earlier Abstract Window Toolkit (AWT). Swings provides a native look and feel that emulates the feel and look of several different platforms, and also supports a plug-in look which allows applications to have a look and feel unrelated to the underlying platform. It comprises of more powerful and flexible components as compared to AWT. In addition to the known components such as buttons, check box and labels, Swing also allows the use of several advanced components like trees, tabbed panel, tables, scroll panes, and lists. Unlike AWT, Swing components are not implemented by platform-specific code. Rather they are written entirely in Java and therefore are platform-independent. The term lightweight is used for describing such elements.

## 4. RESULTS

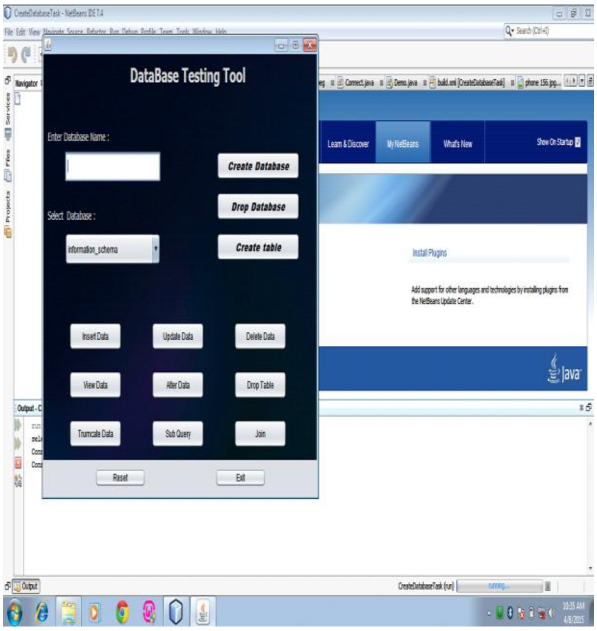
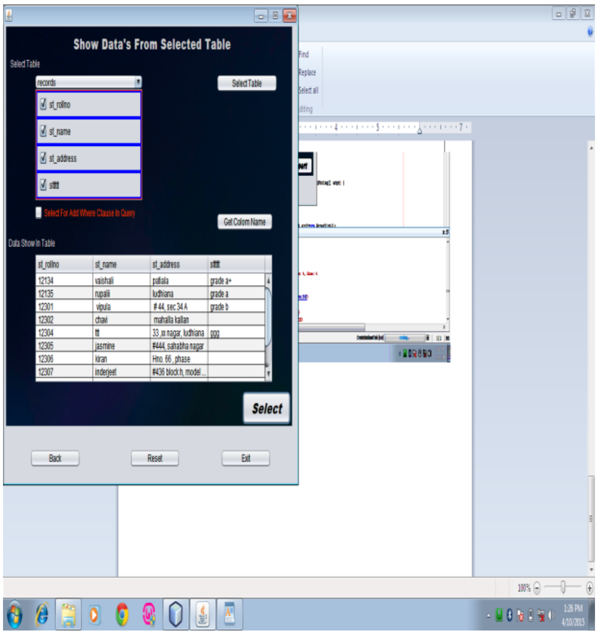
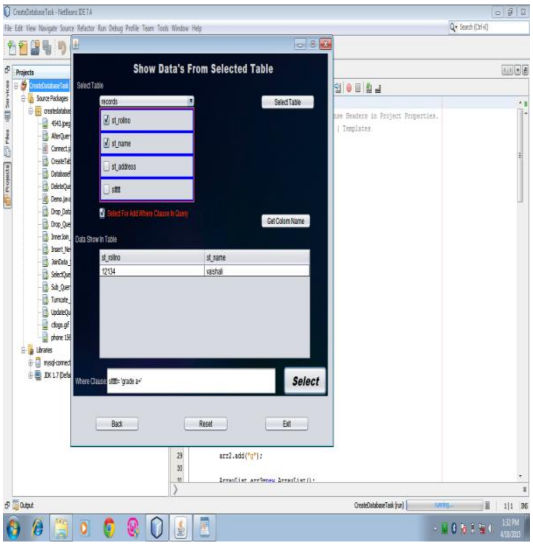


**Figure 1: Overlook of Database Testing Tool**



| st_rollno | st_name | st_address | stttt |
|---|---|---|---|
| 12134 | vaishali | patiala | grade a+ |
| 12135 | rupali | ludhiana | grade a |
| 12301 | vipula | # 44, sec 34 A | grade b |
| 12302 | chavi | mahalla kallan | |
| 12304 | ttt | 33 ,xx nagar, ludhiana | ggg |
| 12305 | jasmine | #444, sahabha nagar | |
| 12306 | kiran | Hno. 66 , phase | |
| 12307 | inderjeet | #436 block h, model ... | |

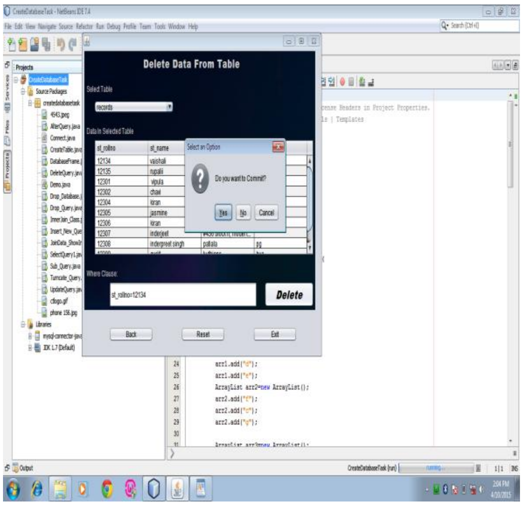
**Figure 2: Viewing data from a table without 'Where' clause**



| st_rollno | st_name |
|---|---|
| 12134 | vaishali |

Where Clause: stttt= 'grade a+'

Figure 3: Viewing table with 'Where' clause

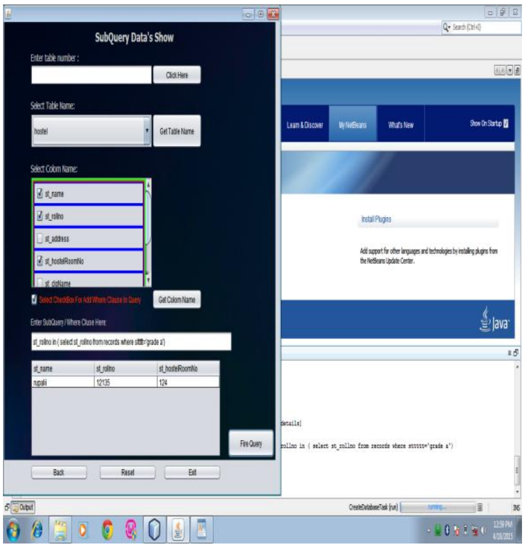


| st_rollno | st_name |
|---|---|
| 12134 | vaishali |
| 12135 | rupalii |
| 12301 | vipula |
| 12302 | chawl |
| 12304 | kiran |
| 12305 | jasmine |
| 12306 | kiran |
| 12307 | inderjeet |
| 12308 | inderpreet singh |

Do you want to Commit?

Where Clause: st_rollno=12134

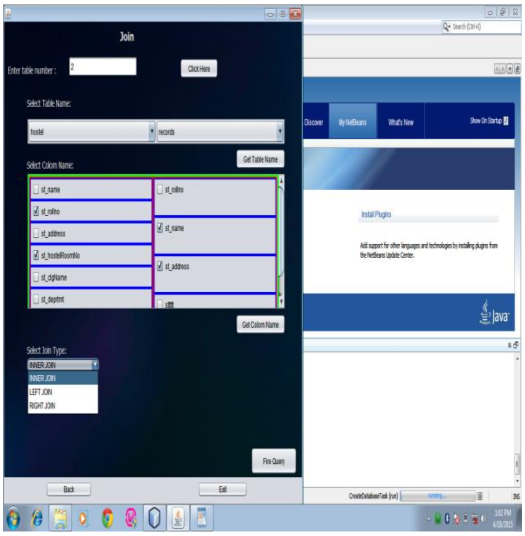
Figure 4: Deleting data from the table



st_rollno in ( select st_rollno from records where stttt='grade a')

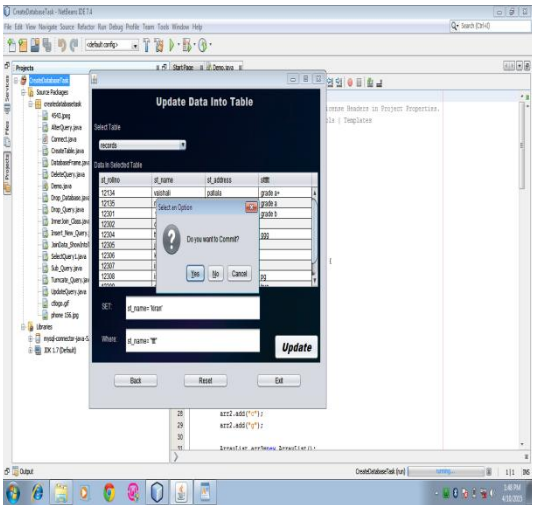
| st_name | st_rollno | st_hostelRoomNo |
|---|---|---|
| rupalii | 12135 | 124 |

Figure 5: After firing SubQuery



Enter table number : 2

Figure 6: Joins



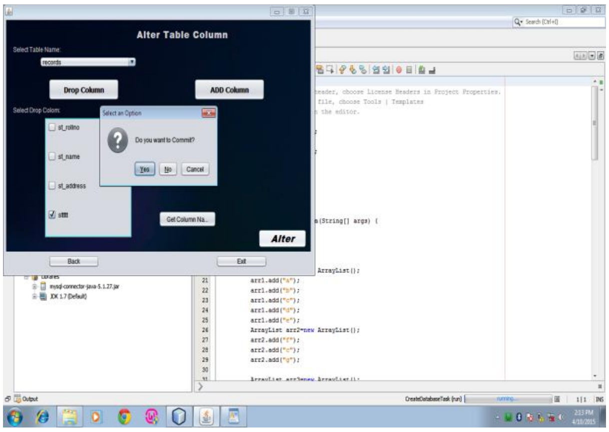
| st_rollno | st_name | st_address | stttt |
|---|---|---|---|
| 12134 | vaishali | patiala | grade a+ |
| 12135 | | | grade a |
| 12301 | | | grade b |
| 12302 | | | |
| 12304 | | | ggg |
| 12305 | | | |
| 12306 | | | |
| 12307 | | | |
| 12308 | | | pg |

Do you want to Commit?

SET: st_name= 'kiran'

Where: st_name= 'ff'

Figure 7: Update data into table



Do you want to Commit?

Figure 8: Altering Table (Dropping column)

## 5. CONCLUSION AND FUTURE SCOPE

It checks the values of database table(s) by firing different queries. Database testing tool enables user to create a database which is error-free as this tool is implemented on the basics of EF Codd Rules which are the grounds of any database. Secondly, user can check/ test various tables of any exiting database by importing it and firing different queries on its own. This tool is user friendly, thus does not require thorough technical knowledge of the various syntax used in any database system. Therefore, any individual who is having basic knowledge can use the database testing tool in order to test the relational database(s) and there after obtain a database which satisfies the ACID properties: Atomicity, Consistency, Isolation and Durability which are essential from the testing point of view for any relational database system. Thus the tables so formed are isolated and atomic in nature as well as maintains the consistency with each other. Therefore, ensures the correctness of data items, enabling the proper functioning of the database application.

## 5.1 Future Scope

NoSQL databases which in simple terms are known as Non-Relational and Distributed databases that can be used as a database for testing. Thus, suggests proposing a tool for testing that works with NoSQL database as a future work.

## 6. ACKNOWLEDGEMENT

I wish to express sincere gratitude and indebtedness to my respected supervisors, Er.Sumeet Kaur Sehra, Assistant Professor, Department of I.T for his valuable guidance, attention-grabbing views and obliging nature which led to the successful completion of this study. I am thankful to worthy Director Dr. Manohar Singh Saini for providing various facilities and opportunities, which lead to successful completion of thesis work. I wish to express my deepest gratitude to Er. Akshay Girdhar (H.O.D, Information Technology) for helping me with his experienced and professional ideas to reach where I am today and still showing the light to excel.

I lack words to express my cordial thanks to the members of Departmental Research Committee, Er. Amanpreet Singh Brar (HOD, CSE) and Dr. Parminder Singh for their useful comments and constructive suggestions during all the phases of the present study as well as critically going through the manuscript.

## 7. REFERENCES

[1] Chan, W. K. and Cheung, S. C.(2005), "Fault-Based Testing of Database Application Programs with Conceptual Data Model" Proceedings of the 5th International Conference on Quality Software (QSIC 2005), IEEE Computer Society Press, Los Alamitos, California

[2] Deng, Y. Frankl, P. G. and Chays, D. Testing database transactions with AGENDA. In *Proceedings of the 27th International Conference on Software Engineering* (*ICSE 2005*), pages 78–87. IEEE Computer Society Press, Los Alamitos, California, 2005.

[3] Deng, Y. Frankl, P. G. and Wang, J. "Testing web database applications" *ACM SIGSOFT Software Engineering Notes*, 29 (5): 1–10, 2004.

[4] Haller, K. (2009), "White-Box testing for Database-driven applications: A Requirement Analysis", Zurich, Switzerland.

[5] Howard, P.(July 2012), "Exploring Successful Approaches to Test Data Management - A White Paper by Bloor Research"

[6] Hall, G.C. Saucer, B. A. Brown, J. S. and Reynolds, M. W. and Casale, R. L. "Guidelines for Good Database Selection and use in Pharmacoepidemiology Research" Bourke Published online in Wiley Online Library DOI: 10.1002/pds.2229

[7] Iacono, J.C. Brown, A. and Holtham, C. "The use of the Case Study Method in Theory Testing: The Example of Steel eMarketplaces" Cass Business School, London, UK

[8] Just, R. Jalali, D. and Michael D. "Defects4J: A Database of Existing Faults to Enable Controlled Testing Studies for Java Programs" Computer Science and Engineering University of Washington.

[9] Jovanovich and Irena, "Software Testing Methods and Techniques," May 26, 2008.

[10] Kundu, S. (2012), "Web Testing: Tool, Challenges and Methods" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012 ISSN (Online): 1694-0814

[11] Kumar, P. S. and Siva, N.S.(2014), "Automation of Software Testing in Agile Development-An Approach and Challenges with Distributed Database Systems" International Global Journal for Research Analysis Vol.3 Issue 7 July 2014 ISSN NO-2277-8160

[12] Kaur, H.(2013), "Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete." International Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 3, Issue 5, Sep-Oct 2013, pp.1739-1743

[13] Manjunath, T.N. Hegadi, R.S. and Archana, R. A(2012) "A study on sampling techniques for data testing" International Journal of Computer Science And Communication Vol.3 No.1 January-June 2012 P.P 13-16

[14] Mu, K. Yuriy, S. and Meliou, B.A. "Data Debugging with Continuous Testing" Computer Science & Engineering School of Computer Science University of Washington University of Massachusetts Seattle, WA, USA

[15] Khan, M. E. (2010), "Different Forms of Software Testing Techniques for Finding Errors,"IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 1, May 2010.

[16] Orso, A. and Rothermel, G. "Software Testing: A Research Travelogue (2000–2014)" Georgia Institute of Technology Atlanta, GA, USA

[17] Paygude, P. and Devale, P. R. "Automated Data Validation Testing Tool for Data Migration Quality Assurance" International Journal of Modern Engineering Research (IJMER) Vol.3, Issue.1, Jan-Feb. 2013 pp-599-603 ISSN: 2249-6645

[18] Qu, X. Cohen, M. B. and Woolf, K. M. "Combinatorial Interaction Regression Testing: A Study of Test Case Generation and Prioritization" Department of Computer Science and Engineering University of Nebraska-Lincoln

[19] Raquel Blanco, R. Tuya, J. and Seco, R. V. "Test adequacy evaluation for the user-database interaction: a

specification-based Approach" University of Oviedo Gijón, Spain

[20] Sharma, V. and Agrawal, A. P.(2013), "Regression Testing for Data-Driven Applications" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-1, June 2013

[21] Shea, B. (2000, May/June). Avoiding Scalability Shock. Software Testing & Quality Engineering Magazine, 42-46.

[22] Tuteja, M. and Dubey, G.(2012) "A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012

[23] Trivedi, S. H. (2012), "Software Testing Techniques" International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 10, October 2012 ISSN: 2277 128X

[24] Willmor, D. and Embury, M. ,S. (2005), "Exploring test adequacy for database system", Informatics Process Group, School of Computer Science.

[25] Walek, B. and Klimes, C. (2012), "A tool for database testing and optimization", International Journal of Computer and Communication Engineering, Vol. 1, No. 3, September 2012.

[26] Weyuker, E & Vokolos, F. (2000). Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study. IEEE Transactions on Software Engineering, 25 (12), 1147-1156.

[27] Walsham G (1993) „Interpreting Information Systems in Organizations" Wiley Series on IS

[28] Xu, Z. Y. Kim, K. M, and G. Rothermel. A hybrid directed test suite augmentation technique. In Proceedings of the International Symposium on Software Reliability Engineering, 2011.

[29] Yin, R. (1994, 2003),"Case Study Research Design and Methods" second and third edition Applied Social Research Methods Series, Volume 5, Sage Publications.

[30] Software Testing Research Survey Bibliography. http://people.engr.ncsu.edu/txie/testingresearchsurvey.htm