

LB/DON/106/2016

IT 01/139

LIBRARY
UNIVERSITY OF MORATUWA, SRI LANKA
MORATUWA

Migrate and Transfer Schema and Data across Multiple Databases

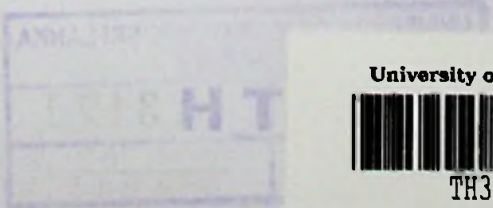
I. Milinda Wijewardana

139179 E


Dissertation submitted to the Faculty of Information Technology,
University of Moratuwa, Sri Lanka for the partial fulfillment of the
requirements of the Degree of Master of Science in

Information Technology.

April 2016



University of Moratuwa



TH3174

004¹⁶
004 (043)

TH 3174

+
1 DVD - ROM

(TH 3160 - TH 3180)

TH 3174

Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

I. M. Wijewardana

Name of Student

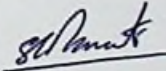


Signature of Student

Date: 26-04-2016

Supervised by

Name of Supervisor



Signature of Supervisor

Date: 26/04/2016

Acknowledgements

I would like to thank my thesis supervisor Mr. S. C. Premaratne of the Faculty of Information Technology at University of Moratuwa.

Abstract

In instances when software is made to interact with multiple databases the upgrade of the software requires data migration. In addition, there are also software's that move schema and data between databases through migration. Database migrations normally take very long time to execute and it is an error prone process. Many commercial and open sourced database migration products are available, but the effectiveness of a tool always depends on features that it supports, however these features do not fully support for database migration projects' requirements. Based on our studies, we design and implement the solution which automates database migration for migrating and transferring schema and data through different types of vendor databases such as MySQL, Oracle, PostgreSQL and MS SQL Server. The solution is developed on top of Java technologies such as NetBeans Platform APIs and Java Database Connectivity API. Creating a Java Swing application on top of NetBeans Platform can largely reduce development time. The NetBeans Platform is an open source framework for Java Swing application. The Java Database Connectivity API is the main standard for database independent connectivity among a Java application and different types of vendor databases. The proposed solution supports project based mechanism to manage the migrations and it caters simple user-friendly wizard based GUI for migration. The solution also provides a set of automated database migration features, which allows migration of complicated database schema and data from one vendor database to another vendor database. The solution automates up to ninety percent of the manual work of database migration. It gives full migration solution for different types of vendor databases such as MySQL, Oracle, PostgreSQL and MS SQL Server. This automated database migration solution helps to save migration effort and costs involved in database migration projects.

Contents

Chapter 1	1
Introduction.....	1
1.1 Prolegomena	1
1.2 Background and Motivation	2
1.3 Problem Definition.....	2
1.4 Aim and Objective	3
1.5 Hypothesis.....	3
1.6 Structure of the Thesis	4
1.7 Summary	4
Chapter 2	5
Current Development in Database Migration.....	5
2.1 Introduction.....	5
2.2.1 FlySpeed Database Migration Tools.....	5
2.2.2 ESF Database Migration Toolkit	6
2.2.3 SwisSQL - Data Migration Tool.....	6
2.2.4 MySQL Workbench - Database Migration.....	7
2.2.5 Microsoft SQL Server Migration Assistant (SSMA)	7
2.2.6 Oracle SQL Developer.....	8
2.3 Problem definition – Research Question	8
2.4 Summary	11
Chapter 3	12
Technology Adopted for Database Migration	12
3.1 Introduction.....	12
3.2.1 NetBeans Platform Features	12
3.2.2 Module System	13
3.2.3 Lifecycle Management.....	13
3.2.4 Pluggability, Service Infrastructure, and File System	13
3.2.5 Miscellaneous Features, Documentation, and Tooling Support	14
3.3 MySQL Workbench.....	14
3.4 pgAdmin	15
3.5 SQL Server Management Studio	16
3.6 Oracle SQL Developer.....	17
3.7 The Java Database Connectivity (JDBC)	18
3.8 Summary	19

Chapter 4	20
A Novel Approach to Database Migration	20
4.1 Introduction.....	20
4.2 Hypothesis.....	20
4.3 Inputs.....	20
4.4 Outputs.....	21
4.5.1 Process for Application Frontend	21
4.5.2 Process for Application Backend.....	22
4.6 Users	23
4.7 Features	23
4.8 Summary.....	24
Chapter 5	25
Solution Design.....	25
5.1 Introduction.....	25
5.2 Wizard Architecture.....	25
5.3 Database Connectivity Architecture	29
5.4 Summary.....	30
Chapter 6	31
Implementation of the solution	31
6.1 Introduction.....	31
6.2 Implementation	31
6.3 Summary	42
Chapter 7	43
Product Evaluation.....	43
7.1 Introduction.....	43
7.2 Squish Test Automation Suite	43
7.3 Summary	44
Chapter 8	45
Conclusion	45
8.1 Introduction.....	45
8.2 Results.....	45
8.3 Further Work.....	50
8.4 Summary	50
Chapter 9	51
Reference	51

List of Tables

	Page
Table 2.1: Summary of Limitation	08
Table 2.2: Summary of Database Migration Tools	11
Table 8.1: Row count of each table	47
Table 8.2: Time taken to migrate data for each table with respect to different databases	48

List of Figures

Figure 3.1: MySQL Workbench main screen window	15
Figure 3.2: PostgreSQL server with pgAdmin III	16
Figure 3.3: Microsoft SQL Server Management Studio	17
Figure 3.4: SQL Developer main window	18
Figure 4.1: First step of a sample wizard	22
Figure 4.2: The Application, JDBC API, JDBC Driver and Database can be represented	23
Figure 5.1: Architecture of a wizard	26
Figure 5.2: Sequence diagram for the wizard objects	28
Figure 5.3: JDBC Architectural Diagram	29
Figure 6.1: Splash screen	31
Figure 6.2: The application on top of the NetBeans Platform	32
Figure 6.3: Clicking the menu item, it shows the wizard	32
Figure 6.4: Welcome screen	33
Figure 6.5: Create a Project	33
Figure 6.6: Select a project path	34
Figure 6.7: Configuring a new project	34
Figure 6.8: Select a Source Database	35
Figure 6.9: Select a specific vendor source database	35
Figure 6.10: Clicking on "Test Connection" button, the dropdown will be populated with all database names	36
Figure 6.11: Source Database with all inputs	36
Figure 6.12: Select a Target Database	37
Figure 6.13: The user selects the tables and/or data from the source database	38

Figure 6.14: The user can select the copy criteria	38
Figure 6.15: The step 6 shows the default data type mappings	39
Figure 6.16: The table mapping	39
Figure 6.17: Database migration summary	40
Figure 6.18: The wizard summarizes the selection	40
Figure 6.19: The last screen of the wizard is showed	41
Figure 6.20: When the user clicks the migrate button, the actual migration takes place	41
Figure 8.1: MySQL Database	45
Figure 8.2: PostgreSQL Database	46
Figure 8.3: MS SQL Server Database	46
Figure 8.4: Oracle Database	47
Figure 8.5: Total time (Seconds) taken to migrate data	48
Figure 8.6: Time taken to migrate data for each table with respect to different databases	49