

# A Multi-tenant Database Framework for Software and Cloud Computing Applications

A Thesis Submitted for the Degree of  
Doctor of Philosophy in Computing Sciences

By

*Haitham Yaish*

Faculty of Engineering and Information Technology  
UNIVERSITY OF TECHNOLOGY, SYDNEY  
Australia  
July 2014

© Copyright by Haitham Yaish, 2014

# CERTIFICATE OF AUTHORSHIP/ ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Haitham Yaish

Date: 14/07/2014

# ACKNOWLEDGMENT

I wish to express my great thanks to all who gave me tremendous support and help during my PhD study primarily to the following.

First and foremost, I would like to express my ultimate acknowledgment to God, who has given me the inspiration and strength to accomplish this thesis beside the time that I spent working in the IT industry during the years of this study. Being a full-time PhD student and at the same time, full-time employee would be impossible without the assistance and the help of God.

I would like to express the deepest gratitude to my principal supervisor Dr. Madhu Goyal, and co-supervisor Dr. George Feuerlicht for their academic guidance, insight, and encouragement throughout every milestone in this study.

I would like to acknowledge my highest appreciation to my parents, Hashim and Sanaa, my brother Abdulatif, and my sisters Mona, Maha and May for their constant support, prayers, inspiration, and best wishes during this study and through all my life.

I would like to express my great appreciations to my wife, Heba. I could never have accomplished this thesis without her love, encouragement, patience, understanding, and prayers. She shared me all the good and difficult times during my study. I also would like to express my thanks and love to my sons, Yazan and Mohammad for doing their best to understand the situation of their father who had to be busy between study and work for such a long time.

I would like to express my appreciations of the support and help that I got from the Centre for Quantum Computation and Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology (FEIT) at the University of Technology, Sydney.

Finally, I should not forget to thank all the teachers and lecturers who taught me during the school years, Bachelor degree, Master degree, and PhD degree. The knowledge that I gained from all of them was essential to complete this study.

# TABLE OF CONTENTS

CERTIFICATE OF AUTHORSHIP/ ORIGINALITY .....	i
ACKNOWLEDGMENT .....	ii
Table of Contents .....	iii
LIST OF FIGURES .....	viii
LIST OF TABLES .....	xi
ABSTRACT .....	xiii
CHAPTER 1 Introduction.....	1
1.1 Background .....	1
1.2 Research Objectives.....	6
1.3 Research Contributions.....	8
1.4 Thesis Organisation .....	10
1.5 Publications Related to this Thesis .....	14
CHAPTER 2 Literature Review .....	16
2.1 Cloud Computing and Software as a Service .....	16
2.1.1 SaaS History.....	18
2.1.2 SaaS Model .....	19
2.1.3 SaaS Characteristics.....	19
2.1.4 SaaS Maturity Model .....	20
2.2 Multi-tenancy.....	21
2.2.1 Multi-tenant Architecture.....	22
2.2.2 Multi-tenant Configuration .....	22
2.3 Multi-tenant Database Management.....	23
2.4 Multi-tenant Database Schema Designs .....	26
2.4.1 Private Tables.....	26
2.4.2 Extension Tables.....	26
2.4.3 Universal Table .....	27

2.4.4 Pivot Tables .....	27
2.4.5 Chunk Table.....	28
2.4.6 Chunk Folding .....	28
2.4.7 XML Table.....	29
2.5 Multi-tenant Database Query Optimizer.....	30
2.5.1 Oracle Query Optimizer.....	31
2.5.2 SQL Server Query Optimizer .....	32
2.5.3 PostgreSQL Query Optimizer.....	33
2.5.4 Salesforce Query Optimizer.....	34
2.6 Multi-tenant Database Access Control .....	35
2.6.1 Siebel Systems Access Control.....	36
2.6.2 IBM DB2, Access Control.....	36
2.6.3 Salesforce Access Control .....	36
2.7 Big Data .....	37
2.7.1 RDBMS and SQL .....	38
2.7.2 NoSQL.....	40
2.7.3 Issues in RDBMS and NoSQL .....	41
2.8 Summary .....	43
<b>CHAPTER 3 Multi-tenant Database Framework Architecture .....</b>	<b>45</b>
3.1 EET Framework Overview Architecture .....	46
3.2 EET Framework Conceptual Architecture Design .....	48
3.2.1 Elastic Extension Tables.....	49
3.2.2 EET Schema Handler Service.....	50
3.2.3 EET Proxy Service.....	50
3.2.4 EET Query Optimizer Service.....	51
3.2.5 EET Access Control Service.....	52
3.2.6 Data Access Object .....	52
3.2.7 Object Relational Mapping.....	53
3.2.8 EET APIs .....	53
3.3 Summary.....	54
<b>CHAPTER 4 Multi-tenant Database Schema Design.....</b>	<b>56</b>
4.1 Elastic Extension Tables.....	57

4.1.1 Common Tenant Tables.....	57
4.1.2 Extension Tables.....	58
4.1.3 Virtual Extension Tables.....	63
4.2 Elastic Extension Tables Database Models .....	63
4.3 An Example to Compare Multi-tenant Database Schema Designs with Elastic Extension Tables.....	67
4.4 Performance Evaluations .....	76
4.4.1 Experimental Data Set and Setup .....	77
4.4.2 Experimental Result.....	81
4.5 Summary .....	89
CHAPTER 5 Multi-tenant Schema Handler Method .....	91
5.1 Elastic Extension Tables Schema Handler Service .....	92
5.1.1 Table Management.....	93
5.1.2 Column Management.....	94
5.1.3 Row Management .....	95
5.1.4 Relationship Management .....	95
5.1.5 Primary Key Management .....	96
5.1.6 Index Management.....	97
5.2 Sample Algorithms of Elastic Extension Tables Schema Handler Service.....	99
5.2.1 Creating Physical and Virtual Rows Algorithm .....	99
5.2.2 Updating Physical and Virtual Rows Algorithm .....	101
5.2.3 Deleting Physical and Virtual Rows Algorithm .....	102
5.3 Performance Evaluations .....	104
5.3.1 Experimental Data Set and Setup .....	105
5.3.2 Experimental Result.....	108
5.4 Summary .....	113
CHAPTER 6 Multi-tenant Database Proxy Method.....	115
6.1 Elastic Extension Tables Proxy Service .....	116
6.2 Elastic Extension Tables Proxy Service Algorithms .....	118
6.2.1 Single Table Query Algorithm.....	119
6.2.2 One-to-Many Query Algorithm .....	123
6.2.3 Union Query Algorithm.....	125
6.2.4 Join Query Algorithm .....	126

6.2.5 Targeted Tables Query Algorithm .....	128
6.3 Performance Evaluation.....	131
6.3.1 Experimental Setup.....	131
6.3.2 Experimental Data Set and Results.....	131
6.4 Summary.....	154
CHAPTER 7 Multi-tenant Query Optimizer Method.....	156
7.1 Elastic Extension Tables Query Optimizer Service.....	158
7.1.1 Query Access Control.....	159
7.1.2 Index Selection.....	161
7.1.3 Table Row Selection.....	162
7.1.4 Statistics.....	163
7.1.5 Multi-tenant Database.....	164
7.1.6 Generate Query.....	164
7.1.7 Execute Query.....	164
7.2 Performance Evaluation.....	165
7.2.1 Experimental Data Set and Setup.....	165
7.2.2 Experimental Results.....	167
7.3 Summary.....	170
CHAPTER 8 Multi-tenant Access Control Method.....	171
8.1 Elastic Extension Tables Access Control.....	172
8.1.1 Access Control Tables.....	172
8.1.2 Elastic Extension Tables Access Grants.....	175
8.2 Columns and Rows Access Grant Algorithms.....	177
8.2.1 Get User Roles Algorithm.....	177
8.2.2 Get User Columns Algorithm.....	179
8.2.3 Get User Insert Access Algorithm.....	180
8.2.4 Get User Update Access Algorithm.....	181
8.2.5 Get User Delete Access Algorithm.....	182
8.2.6 Get User Query Access Algorithm.....	183
8.3 Performance Evaluation.....	185
8.3.1 Experimental Data Set and Setup.....	185
8.3.2 Experimental Results.....	186

8.4 Summary .....	189
CHAPTER 9 Conclusions and Future Research.....	190
9.1 Conclusions.....	190
9.2 Future Research .....	196
ABBREVIATIONS .....	199
BIBLIOGRAPHY .....	201



# LIST OF FIGURES

Figure 1-1: The overall structure of the thesis .....	13
Figure 2-1: SaaS Maturity Levels (Shao 2011) .....	21
Figure 2-2: Separate Database Approach (Chong, Carraro & Wolter 2006).....	24
Figure 2-3: Shared Database - Separate Schema Approach (Chong, Carraro & Wolter 2006) .....	24
Figure 2-4: Shared Database - Shared Schema Approach (Chong, Carraro & Wolter 2006) .....	25
Figure 2-5: The architecture of Oracle query optimizer (Raza et al. 2010).....	31
Figure 2-6: The architecture of SQL Server query optimizer (Raza et al. 2010) .....	32
Figure 2-7: The architecture of PostgreSQL Query Optimizer (Dash et al. 2010).....	33
Figure 3-1: EET overview architecture.....	48
Figure 3-2: EET conceptual architecture design.....	49
Figure 4-1: Elastic Extension Tables .....	62
Figure 4-2: The Three EET Database Models .....	64
Figure 4-3: The EET Three Database Models Example.....	66
Figure 4-4: Private Tables.....	68
Figure 4-5: Extension Tables.....	69
Figure 4-6: Universal Table .....	69
Figure 4-7: Pivot Tables.....	70
Figure 4-8: Chunk Table.....	71
Figure 4-9: Chunk Folding.....	72
Figure 4-10: XML Table.....	72
Figure 4-11: Virtual Extension Tables (VET) .....	74
Figure 4-12: The data stored in the 'sales_person' CTT .....	74
Figure 4-13: The data stored in the 'db_table' ET .....	74
Figure 4-14: The data stored in the 'table_column' ET.....	75
Figure 4-15: The data stored in the 'table_row' ET .....	75
Figure 4-16: The data stored in the 'table_relationship' ET .....	75
Figure 4-17: The data stored in the 'table_index' ET .....	75
Figure 4-18: The data stored in the 'table_primary_key_column' ET .....	76
Figure 4-19: Universal Table Schema Mapping (Liao et al. 2012) .....	77
Figure 4-20: The virtual 'product' table structure. ....	78
Figure 4-21: Retrieving small numbers of rows (Exp. 4-1.1).....	82
Figure 4-22: Retrieving large numbers of rows (Exp. 4-1.1) .....	82
Figure 4-23: Retrieving rows using columns query filters (Exp.4-1.2).....	83
Figure 4-24: Retrieving rows using PK indexes (Exp. 4-1.3) .....	84
Figure 4-25: Retrieving rows using a custom index (Exp. 4-1.4).....	84

Figure 4-26: Inserting rows (Exp.4-2) .....	85
Figure 4-27: Updating rows (Exp. 4-3).....	86
Figure 4-28: Deleting rows (Exp.4-4).....	87
Figure 5-1: EET Schema Handler Service overview architecture .....	93
Figure 5-2: The product and the sales_fact tables' structures.....	106
Figure 5-3: Inserting rows experiment.....	109
Figure 5-4: Updating rows experiment.....	110
Figure 5-5: Deleting rows experiment.....	111
Figure 6-1: EETPS overview architecture .....	117
Figure 6-2: Targeted Tables example .....	129
Figure 6-3: Current Root Table and Current Targeted Table .....	130
Figure 6-4: The tables structures used in the experiments.....	133
Figure 6-5: The outputs of the Simple Query Experiment (Single Table) .....	133
Figure 6-6: The experimental results of retrieving 1 row from the Single Table function .....	134
Figure 6-7: The experimental results of retrieving 100 rows from the Single Table function .....	134
Figure 6-8: The outputs of the Simple-to-Medium Query Experiment (One-to-Many) .....	136
Figure 6-9: The experimental results of retrieving 1 row from the One-to-Many function .....	136
Figure 6-10: The experimental results of retrieving 100 rows from the One-to-Many function .....	137
Figure 6-11: The outputs of the Medium Query Experiment (Union).....	139
Figure 6-12: The experimental results of retrieving 1 row from the Union function .....	139
Figure 6-13: The experimental results of retrieving 100 rows from the Union function .....	140
Figure 6-14: The three left joins of The Left Join experiment .....	141
Figure 6-15: The output of the Medium-to-Complex Query Experiment (Left Join) .....	142
Figure 6-16: The experimental results of retrieving 1 row from the Left Join function .....	142
Figure 6-17: The experimental results of retrieving 100 rows from the Left Join 100 rows experimental results .....	143
Figure 6-18: The query filters of the Targeted Tables experiment.....	145
Figure 6-19: The outputs of the Complex Query Experiment (Targeted Tables)....	146
Figure 6-20: The experimental results of retrieving 1 row from the Targeted Tables function .....	146
Figure 6-21: The experimental results of retrieving 100 rows from the Targeted Tables function.....	147
Figure 6-22: The structures of the queries used in the experiments .....	148
Figure 6-23: The average experimental results of retrieving 1 row.....	149
Figure 6-24: The average experimental results of retrieving 100 rows.....	150

Figure 7-1: The EETQOS architecture and how it is orchestrated with EETPS and EET .....	159
Figure 7-2: The table structure of the 'product' table.....	167
Figure 7-3: The experimental results of retrieving data using filters and indexes...	168
Figure 8-1: EET Access Control Data Architecture .....	172
Figure 8-2: EET access control grants .....	175
Figure 8-3: Table columns access grant.....	176
Figure 8-4: Table rows access grant .....	177
Figure 8-5: The table structure of the 'product' table.....	186
Figure 8-6: Accessing data from the table columns experiment (Exp.8-1) .....	187
Figure 8-7: Accessing data from the table rows experiment (Exp.8-2).....	188

# LIST OF TABLES

Table 4-1: The query execution times of retrieving rows without using query columns filters experiment (Exp. 4-1.1).....	83
Table 4-2: The query execution times of retrieving rows using columns query filters experiment (Exp. 4-1.2).....	83
Table 4-3: The query execution times of retrieving rows using primary key indexes experiment (Exp. 4-1.3).....	84
Table 4-4: The query execution times of retrieving rows using custom index experiment (Exp. 4-1.4).....	85
Table 4-5: The query execution times of inserting rows experiment (Exp. 4-2).....	85
Table 4-6: The query execution times of updating rows experiment (Exp. 4-3).....	86
Table 4-7: The query execution times of deleting rows experiment (Exp. 4-4).....	87
Table 4-8: The experiments queries.....	87
Table 5-1: The query execution times of inserting rows experiment (Exp. 5-1).....	109
Table 5-2: The query execution times of updating rows experiment (Exp. 5-2).....	110
Table 5-3: The query execution times of deleting rows experiment (Exp. 5-3).....	111
Table 5-4: The experiments queries.....	111
Table 6-1: The query execution times of retrieving 1 row from the Single Table experiment (Exp. 6-1).....	134
Table 6-2: The query execution times of retrieving 100 rows from the Single Table experiment.....	135
Table 6-3: The query execution times of retrieving 1 row from the One-to-Many experiment.....	137
Table 6-4: The query execution times of retrieving 100 rows from the One-to-Many experiment (Exp. 6-2).....	137
Table 6-5: The query execution times of retrieving 1 row from the Union.....	140
Table 6-6: The query execution times of retrieving 100 rows from the Union experiment (Exp. 6-3).....	140
Table 6-7: The query execution times of retrieving 1 row from the Left Join experiment (Exp. 6-4).....	143
Table 6-8: The query execution times of retrieving 100 rows from the Left Join experiment (Exp. 6-4).....	143
Table 6-9: The query execution times of retrieving 1 row from the Targeted Tables experiment (Exp. 6-5).....	147
Table 6-10: The query execution times of retrieving 100 rows from the Targeted Tables experiment (Exp. 6-5).....	147
Table 6-11: The average experimental results of retrieving 1 row in milliseconds..	149

Table 6-12: The average experimental results of retrieving 100 rows in milliseconds	150
Table 6-13: The experiments queries	151
Table 6-14: The experiments queries details	151
Table 7-1: The query execution times of retrieving data using filters and indexes	169
Table 7-2: The experiments queries	169
Table 8-1: The query execution times of Exp.8-1 and Exp.8-2	188
Table 8-2: The experiments queries	188

# ABSTRACT

Cloud Computing is a new computing paradigm that transforms accessing computing resources from internal data centres to external service providers. This approach is rapidly becoming a standard for offering cost effective and elastic computing services that are used over the internet. Software as a service (SaaS) is one of the Cloud Computing service models that exploits economies of scale for SaaS service providers by offering the same software and computing environment for multiple tenants. This contemporary multi-tenant service requires a multi-tenant database design that can accommodate data for multiple tenants in one single database schema. Due to multi-tenant database resource sharing in this service, the multi-tenant schema should be highly secured, optimized, configurable, and extendable during runtime execution to fulfil the applications' requirements of different tenants. However, traditional Relational Database Management Systems (RDBMS) do not support such multi-tenant database schema capabilities, and it is a significant challenge to enable RDBMS to support these capabilities. Therefore, one solution is using an intermediate software layer that mediates multi-tenant applications and RDBMS, to convert multi-tenant queries into regular database queries, and to execute them in a RDBMS. Developing such a multi-tenant software layer to manage and access tenants' data is a hard and complex problem to solve and has significant complexities that involve longer development lifecycle.

There are two main contributions of this thesis. Firstly, a proposal for a novel multi-tenant schema technique called Elastic Extension Tables (EET). Secondly, a proposal for a multi-tenant database framework prototype to implement EET schema

in a RDBMS. This approach can be used to develop a software layer that mediates software applications and a RDBMS. This software layer aims to facilitate the development of software applications, and multi-tenant SaaS and Big Data applications for both cloud service providers and their tenants.

Extensive experiments were conducted to evaluate the feasibility and effectiveness of EET multi-tenant database schema by comparing it with Universal Table Schema Mapping (UTSM), which is commercially used. Significant performance improvements obtained using EET when compared to UTSM, makes the EET schema a good candidate for implementing multi-tenant databases and multi-tenant applications. Furthermore, the prototype of the EET framework was developed, and several experiments were performed to verify the practicability and the effectiveness of using this framework that based on EET multi-tenant database schema. The results of the experiments indicate that the EET framework is suitable for the development of software applications in general, and multi-tenant SaaS and Big Data applications in particular.