

# Semistructured Database Design

---

---

# Web Information Systems Engineering and Internet Technologies

## *Book Series*

**Series Editor:** Yanchun Zhang, Victoria University, Australia

**Editorial Board:**

Robin Chen, AT&T

Umeshwar Dayal, HP

Arun Iyengar, IBM

Keith Jeffery, Rutherford Appleton Lab

Xiaohua Jia, City University of Hong Kong

Yahiko Kambayashi† Kyoto University

Masaru Kitsuregawa, Tokyo University

Qing Li, City University of Hong Kong

Philip Yu, IBM

Hongjun Lu, HKUST

John Mylopoulos, University of Toronto

Erich Neuhold, IPSI

Tamer Ozsu, Waterloo University

Maria Orłowska, DSTC

Gultekin Ozsoyoglu, Case Western Reserve University

Michael Papazoglou, Tilburg University

Marek Rusinkiewicz, Telcordia Technology

Stefano Spaccapietra, EPFL

Vijay Varadharajan, Macquarie University

Marianne Winslett, University of Illinois at Urbana-Champaign

Xiaofang Zhou, University of Queensland

# Semistructured Database Design

Tok Wang Ling  
Mong Li Lee

*National University of Singapore*

Gillian Dobbie

*The University of Auckland*

**Springer**

eBook ISBN: 0-387-23568-X  
Print ISBN: 0-387-23567-1

©2005 Springer Science + Business Media, Inc.

Print ©2005 Springer Science + Business Media, Inc.  
Boston

All rights reserved

No part of this eBook may be reproduced or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise, without written consent from the Publisher

Created in the United States of America

Visit Springer's eBookstore at:  
and the Springer Global Website Online at:

<http://ebooks.kluweronline.com>  
<http://www.springeronline.com>

# Contents

List of Figures	ix
List of Tables	xiii
Preface	xv
1. INTRODUCTION	1
1.1 Chapter Overview	3
2. DATA MODELS FOR SEMISTRUCTURED DATA	7
2.1 Document Type Definition	8
2.2 DOM, OEM and DataGuide	12
2.3 S3-graph	16
2.4 CM Hypergraph and Scheme Tree	18
2.5 EER and XGrammar	21
2.6 AL-DTD and XML Tree	24
2.7 ORA-SS	28
2.8 Discussion	32
3. ORA-SS	37
3.1 ORA-SS Schema Diagram	37
3.2 ORA-SS Data Instance Diagram	49
3.3 ORA-SS Functional Dependency Diagram	52
3.4 ORA-SS Inheritance Hierarchy Diagram	55
3.5 Discussion	57
4. SCHEMA EXTRACTION	59
4.1 Basic Extraction Rules	60
4.2 Schema Extraction Algorithm	62

4.3	Example	66
4.4	Discussion	74
4.5	Summary	75
5.	NORMALIZATION	77
5.1	Motivating Example	78
5.2	Background	82
5.3	A Normal Form For Semistructured Schemas	85
5.4	Converting Schemas into the Normal Form	89
5.5	Discussion	107
6.	VIEWS	111
6.1	Motivating Example	112
6.2	The Select Operator	116
6.3	The Drop Operator	117
6.4	The Join Operator	121
6.5	The Swap Operator	125
6.6	Design Rules for Identifier Dependency Relationship	132
6.7	Example of Designing View	134
6.8	Related Work	136
6.9	Summary	138
7.	PHYSICAL DATABASE DESIGN	139
7.1	Relational Database Physical Design	139
7.2	IMS Database Physical Design	141
7.3	Redundancy in ORA-SS Schema Diagram	143
7.4	Replicated NF in ORA-SS	146
7.5	Controlled Pairing in ORA-SS Schema Diagrams	150
7.6	Measure of Data Replication	153
7.7	Guidelines for Physical Semistructured Database Design	154
7.8	Storage of Documents in an Object Relational Database	158
7.9	Summary	160
8.	CONCLUSION	161
	Appendices	165

<i>Contents</i>	vii
References	169
Index	173
About the Authors	175

# List of Figures

2.1	Example XML document	9
2.2	A DTD for the document in Figure 2.1	10
2.3	A DTD for the document in Figure 2.1 without replication	11
2.4	A DOM tree for the document in Figure 2.1	14
2.5	An (a) OEM diagram and its (b) DataGuide for the document in Figure 2.1	15
2.6	An S3-Graph for the document in Figure 2.1	18
2.7	A CM Hypergraph and Scheme Tree for the schema in Figure 2.3	20
2.8	An EER diagram and XGrammar definition for Examples 2.7 and 2.8	22
2.9	An EER diagram and XGrammar definition representing ordering on student within course	23
2.10	A textual representation of the XML Tree in Figure 2.11	25
2.11	A diagram of the XML Tree in Figure 2.10	26
2.12	An AL-DTD schema for the XML Tree in Figures 2.10 and 2.11	28
2.13	An ORA-SS Instance Diagram for the document in Figure 2.1	30
2.14	An ORA-SS schema diagram for the document in Figure 2.1	31
2.15	An ORA-SS schema diagram showing binary and ternary relationships	33
2.16	An ORA-SS schema diagram showing ordering of students and hobbies	33
3.1	Object class <i>student</i> with attributes in an ORA-SS Schema Diagram	38



3.2	Representing binary relationship types in an ORA-SS Schema Diagram	40
3.3	Representing ternary relationship types in an ORA-SS Schema Diagram	40
3.4	Representing a binary and ternary relationship type in an ORA-SS Schema Diagram	42
3.5	Object classes with no identifier or a weak identifier in an ORA-SS Schema Diagram	44
3.6	Object classes with relationship types and attributes in an ORA-SS Schema Diagram	45
3.7	Referencing an object class in an ORA-SS Schema Diagram	46
3.8	Example of a recursive relationship in ORA-SS Schema Diagrams	47
3.9	Symmetric relationship in an ORA-SS Schema Diagram	47
3.10	Ordered object classes, attributes, and attribute values in an ORA-SS Schema Diagram	49
3.11	Disjunctive attribute and object classes in an ORA-SS Schema Diagram	50
3.12	ORA-SS Instance Diagram for document in Figure 2.1	51
3.13	An XML Document for the ORA-SS Instance Diagram in Figure 3.12	52
3.14	ORA-SS Schema Diagram for document in Figure 3.12	53
3.15	An DTD for the ORA-SS Schema Diagram in Figure 3.14	53
3.16	Functional dependency diagram enhancing the information in Figure 3.7	55
3.17	ORA-SS Schema Diagram and Inheritance Diagram	56
4.1	Example ORA-SS schema	60
4.2	Initial ORA-SS schema structure after Step 1	69
4.3	Final ORA-SS schema obtained after Step 2	74
4.4	DataGuide extracted from sample XML document	74
5.1	Example XML document with redundant information	78
5.2	An ORA-SS schema diagram for document in Figure 5.1	79
5.3	An ORA-SS schema diagram, where valid documents do not contain redundant information	80
5.4	A DTD for the schema diagram in Figure 5.3	80
5.5	Example XML document without redundant information	81
5.6	ORA-SS schema diagrams for example 5.4	87
5.7	ORA-SS schema diagrams for example 5.5	87

5.8	ORA-SS schema diagram that is not in NF	90
5.9	An NF ORA-SS schema diagram for Figure 5.8	91
5.10	Figures for Example 5.7 illustrating Algorithm ConvertNF	95
5.11	Figures for Example 5.7 illustrating Algorithm ConvertNF	96
5.12	Figures for Example 5.8 illustrating Algorithm ConvertNF	97
5.13	Figures for Example 5.8 illustrating Algorithm ConvertNF	99
5.14	Figures for Example 5.9 illustrating Algorithm ConvertNF	100
5.15	Figures for Example 5.10 illustrating Algorithm ConvertNF	102
5.16	Figures for Example 5.11 illustrating Algorithm ConvertNF	103
5.17	Figures for Example 5.11 illustrating Algorithm ConvertNF	104
5.18	Figures for Example 5.11 illustrating Algorithm ConvertNF	105
5.19	Figures for Example 5.11 illustrating Algorithm ConvertNF	106
6.1	A Supplier-Part-Project ORA-SS Schema Diagram	113
6.2	An Invalid XML View of the Supplier-Part-Project Schema in Figure 6.1	115
6.3	A Valid XML View of the Supplier-Part-Project Schema in Figure 6.1	115
6.4	An XML View of the Supplier-Part-Project Schema in Figure 6.1 obtained by the Selection Operator	117
6.5	An XML View of the Supplier-Part-Project Schema in Figure 6.1 obtained by the Drop Operator	118
6.6	ORA-SS source schema involving Project, Staff and Publication.	121
6.7	An ambiguous view of Figure 6.6.	122
6.8	A valid view of Figure 6.6. The new relationship type <i>jp</i> is derived by joining <i>js</i> and <i>sp</i> .	122
6.9	ORA-SS schema diagram on Project, Supplier, Part and Retailer	123
6.10	View of Figure 6.9 obtained by a join operation	124
6.11	ORA-SS schema of Supplier-Part-Project	126
6.12	View of Figure 6.11 obtained by a swap operation	126
6.13	Handling relationship types that are affected by a swap operation.	129
6.14	Handling relationship types that involve the descendants of $O_j$ .	129
6.15	ORA-SS schema of course-student-lecturer	131
6.16	An invalid view of Figure 6.15 after swapping <i>student</i> and <i>course</i>	131

6.17	A valid reversible view of Figure 6.15 after swapping <i>student</i> and <i>course</i>	132
6.18	ORA-SS schema containing an IDD relationship type	133
6.19	ORA-SS schema of a view that swaps <i>employee</i> and <i>child</i>	133
6.20	ORA-SS schema of a view that drops <i>employee</i>	133
6.21	Example ORA-SS schema	135
6.22	View of Figure 6.21 obtained by a join and a drop operator	135
6.23	View obtained by swapping <i>part</i> and <i>project'</i> in Figure 6.22	136
6.24	View obtained by swapping <i>employee</i> and <i>child</i> in Figure 6.23	137
7.1	Database design using IMS	142
7.2	Using logical parent pointers to remove redundancy	142
7.3	Physical pairing in IMS	143
7.4	Many to many relationship type	144
7.5	Symmetric relationship type	144
7.6	Relationship type nested under many to many relationship type	145
7.7	Precomputed derived and aggregate attributes	146
7.8	Replication of references for recursive query	146
7.9	Duplication of staff information in document	147
7.10	NF ORA-SS schema diagram	148
7.11	Replicated NF ORA-SS schema diagram with allowed replication of relatively stable attributes, <i>name</i> and <i>birthdate</i>	149
7.12	NF ORA-SS Schema Diagram	150
7.13	Symmetric relationship type	150
7.14	Repeating relationship type	152
7.15	Cost of physical storage design	155
7.16	Resulting ORA-SS Schema with controlled replication	157
7.17	Mapping ORA-SS Schema Diagram to object relational model	159
8.1	Steps in the Design of Repositories for Semistructured Data	162

# List of Tables

2.1	Essential concepts of a data model for semistructured data	8
2.2	Features supported in XML Data Models	34
4.1	Object class tables <i>course</i> , <i>lecturer</i> , <i>student</i> and <i>tutor</i>	70
4.2	Final object class tables <i>student</i> and <i>tutor</i>	71
4.3	Relationship type tables <i>cst</i> and <i>cl</i>	72
4.4	Final relationship type tables <i>cs</i> , <i>cst</i> and <i>cl</i>	73

# **Preface**

## **About This Book**

The work presented in this book came about after we recognized that ill-designed semistructured databases can lead to update anomalies, and there is a strong need for algorithms and tools to help users design storage structures for semistructured data. We have been publishing papers in the design of databases for semistructured data since 1999, and believe that after a number of attempts we have defined a data model that captures the necessary semantics for representing the semantics that are necessary in the design of good semistructured databases.

This book describes a process that initially takes a hardline approach against redundant data, and then relaxes the approach for gains in query performance. The book is suited to both researchers and practitioners in the field of semistructured database design.

Some of the material in this book has been published at international conferences. The material in Chapter 5 was originally based on work presented in [Wu et al., 2001a] and Chapter 6 was originally based on [Chen et al., 2002]. The material in Chapter 3 was published as a technical report at the National University of Singapore [Dobbie et al., 2000].

## **Use of the Book**

The target audience of this book is practitioners who design semistructured data file organizations or semistructured databases, researchers who work in the area of semistructured data organization, and students with an interest in the design of storage organizations for semistructured data. The material is as relevant for file organizations as it is for databases since inconsistencies can also exist in data files.

## Major Contribution

This major contributions of this book are:

- a comparison of data models for the purpose of designing storage organizations for semistructured data,
- the introduction of a data model, called Object Relationship Attribute Data Model for SemiStructured Data, or ORA-SS, which represents what we believe are the necessary semantics for the design of storage organizations for semistructured data,
- an algorithm for the extraction of a schema from a semistructured data instance, such as an XML document,
- a normalization algorithm for semistructured schemas,
- a set of rules for the validation of views created on an underlying semistructured instance,
- an algorithm for the denormalization of semistructured schemas.

## Acknowledgements

This work has been supported by the following grants:

National University Of Singapore Academic Research Fund  
R-252-000-093-112, Building a semi-structured data repository  
R-252-100-105-112, Integrating Data Warehouses on the Web

University of Auckland  
Research and Study Leave Grant  
Staff Research Fund, Semistructured database design

Our special thanks goes to the following students: Yabing Chen, Xiaoying Wu, Wei Ni, Yuanying Mo, Xia Yang, Wai Lup Low, Lars Neumann.

TOK WANG LING, MONG LI LEE AND GILLIAN DOBBIE