เครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์

นายประภาส ทองรัก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A TOOL FOR GENERATING TEST CASE FOR RELATIONAL DATABASE CONSTRAINTS TESTING

Mr. Prapas Tongrak

ศูนย์วิทยทรัพยากร

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2009

จุฬาลงกรณ์มหาวิทยาลัย

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	เครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์
โดย	นาย ประภาส ทองรัก
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ นครทิพย์ พร้อมพูล)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.ภัทรชัย ลลิตโรจน์วงศ์)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประภาส ทองรัก : เครื่องมือสร้างกรณีทดสอบสำหรับการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ (A TOOL FOR GENERATING TEST CASE FOR RELATIONAL DATABASE CONSTRAINTS TESTING) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: รองศาสตราจารย์ ดร. ธาราทิพย์ สุวรรณศาสตร์, 121 หน้า.

ปัจจุบันแอพพลิเคชันของฐานข้อมูลมีความซับซ้อนเพิ่มขึ้น ส่งผลให้แอพพลิเคชันของฐานข้อมูลปัจจุบันมีการเปลี่ยนแปลง เช่น การเปลี่ยนแปลงกระบวนการทางธุรกิจ การเปลี่ยนแปลงสิทธ์การเข้าใช้งาน และบทบาทต่างๆ เป็นต้น การทดสอบแอพพลิเคชันของฐานข้อมูลจึงมีความท้าทาย เนื่องจากวิธีการและเครื่องมือทดสอบในการทดสอบแอพพลิเคชันของฐานข้อมูลปัจจุบันไม่สนใจการทดสอบเงื่อนไขของฐานข้อมูล วิทยานิพนธ์นี้จึงนำเสนอเครื่องมือสร้างกรณีทดสอบซึ่งสนับสนุนการทดสอบฐานข้อมูลและสร้างเครื่องมืออัตโนมัติสำหรับสร้างกรณีทดสอบให้เป็นไปตามข้อกำหนดซึ่งกำหนดจากผู้ใช้และฐานข้อมูลที่ทำการทดสอบ ข้อกำหนดเหล่านั้น ประกอบด้วยสคีมาของฐานข้อมูล เงื่อนไขบังคับบูรณภาพของเอนทิตี เงื่อนไขบังคับบูรณภาพของการอ้างอิงถึง และเงื่อนไขบังคับของโดเมน ทั้งนี้เป้าหมายของเครื่องมือนี้ เพื่อสร้างกรณีทดสอบซึ่งครอบคลุมทุกเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ภายใต้การทดสอบ และเป็นการช่วยให้นักพัฒนาและผู้ทดสอบสามารถสร้างกรณีทดสอบได้ง่ายขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....ประภาส ทองรัก.....
สาขาวิชา.....วิศวกรรมซอฟต์แวร์.....ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....
ปีการศึกษา....2552.....

4970427821 : MAJOR SOFTWARE ENGINEERING

KEYWORDS : SOFTWARE TESTING / TEST CASE / RELATIONAL DATABASE CONSTRAINTS

PRAPAS TONGRAK : A TOOL FOR GENERATING TEST CASE FOR RELATIONAL DATABASE CONSTRAINTS TESTING. THESIS ADVISOR: ASSOCIATE PROFESSOR TARATIP SUWANNASART, Ph.D., 121 pp.

Database applications become increasingly complex. Consequently, database application configuration must be changed, for business process change as well as, authorization and role change. Testing database applications is much more challenging because most methods and tools developed for application testing have not focused on database constraints. In this thesis, a tool for relational database constraints testing is proposed to support database testing and is constructed to automate test case generation according to criteria, supplied by tester and target database. The criteria consist of database schema, entity integrity constraints, referential integrity constraints, and domain constraints. The goal of the tool is to generate test cases that cover database constraints of the software under test and to assist and ease the database application developer or tester.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Department : Computer Engineering Student's Signature ปราศ ทองรัก

Field of Study : Software Engineering Advisor's Signature

Academic Year : 2009

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงด้วยความเมตตาและความช่วยเหลืออย่างยิ่งจากรองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์ อาจารย์ที่ปรึกษา ที่เสียสละเวลาช่วยให้คำปรึกษา ข้อคิดและคำแนะนำที่มีประโยชน์ต่องานวิจัย ตลอดจนความเอาใจใส่และความเชื่อมั่นที่อาจารย์มีให้ผู้วิจัย ซึ่งเป็นกำลังใจและเป็นแรงส่งเสริมให้ผู้วิจัยสามารถพัฒนางานวิจัยที่มีคุณภาพและมีคุณค่า

ขอขอบพระคุณ รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ ประธานกรรมการสอบวิทยานิพนธ์ ดร.อาทิตย์ ทองทักษ์ ผู้ช่วยศาสตราจารย์ นครทิพย์ พร้อมพูล และผู้ช่วยศาสตราจารย์ ดร. ภัทรชัย ลลิตโรจน์วงศ์ กรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาสละเวลาให้คำแนะนำสำหรับโครงร่างวิทยานิพนธ์และวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่าน ที่ประสิทธิ์ประสาทความรู้อันมีค่ายิ่งแก่ผู้วิจัย

ขอขอบคุณบุคลากรในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่าน ที่ให้ข้อมูล คำแนะนำและความช่วยเหลือในการดำเนินการทั้งในเรื่องการศึกษาและการสอบวิทยานิพนธ์ได้สำเร็จลุล่วง

ขอขอบคุณ เพื่อนๆ พี่ๆ และน้องๆ ทุกคนที่ผ่านเข้ามาในชีวิตของผู้วิจัย ที่ห่วงใยและให้ความช่วยเหลือในทุกๆ ด้านจนผู้วิจัยสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

ขอบคุณสมาชิกในห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ สำหรับน้ำใจ ความห่วงใย ความช่วยเหลือและคำแนะนำที่มีประโยชน์แต่ผู้วิจัย

ท้ายที่สุด ขอกราบขอบพระคุณบิดา มารดาและสมาชิกในครอบครัวทุกท่าน ที่ให้การสนับสนุน และให้กำลังใจแก่ผู้วิจัย เสมอมา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ญ

สารบัญภาพ..... ฑ

บทที่

1. บทนำ..... 1
   - 1.1 ความเป็นมาและความสำคัญของปัญหา..... 1
   - 1.2 วัตถุประสงค์ของการวิจัย..... 3
   - 1.3 ขอบเขตของการวิจัย..... 3
   - 1.4 ประโยชน์ที่คาดว่าจะได้รับ..... 4
   - 1.5 วิธีดำเนินการวิจัย..... 4
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 6
   - 2.1 ทฤษฎีที่เกี่ยวข้อง..... 6
     - 2.1.1 การทดสอบซอฟต์แวร์..... 6
     - 2.1.2 เอ็กซ์เอ็มแอล..... 7
     - 2.1.3 เอ็กซ์เอ็มแอลสคีมา..... 9
     - 2.1.4 ภาษาเอสคิวแอล..... 11
     - 2.1.5 เงื่อนไขของฐานข้อมูลเชิงสัมพันธ์..... 13
   - 2.2 งานวิจัยและเครื่องมือที่เกี่ยวข้อง..... 17
     - 2.2.1 วิทยานิพนธ์เรื่อง “การพัฒนาเครื่องมือซอฟต์แวร์สำหรับสร้างข้อมูลทดสอบ (Development of a Software Tool for Generating Test Data)”..... 17

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ หน้า

2.2.2 วิทยานิพนธ์เรื่อง “การพัฒนาเครื่องมือสร้างข้อมูลทดสอบเพื่อสนับสนุนการทดสอบซอฟต์แวร์จากสคีมาของฐานข้อมูล (DEVELOPMENT OF A TEST DATA GENERATING TOOL TO SUPPORT SOFTWARE TESTING FROMDATABASE SCHEMA)”..... 18
2.2.3 DeZign for databases to create ER Diagrams..... 18
2.2.4 The Altova DatabaseSpy 2008……... 18
3. การวิเคราะห์และออกแบบเครื่องมือ..... 20
3.1 โครงสร้างการทำงานของเครื่องมือ…….. 20
3.1.1 บันทึกสคีมาของฐานข้อมูลและเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์..... 21
3.1.2 สร้างกรณีทดสอบตามเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์..... 23
3.1.3 ทดสอบฐานข้อมูลและแสดงผลจากการทดสอบฐานข้อมูล..... 24
3.1.4 การวิเคราะห์ผลการทดสอบและปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์……... 24
3.2 การวิเคราะห์และออกแบบเครื่องมือ……... 24
3.2.1 แผนภาพยูสเคส..... 25
3.2.2 แผนภาพคลาส..... 46
3.2.3 แผนภาพซีเควนซ์..... 50
4. การพัฒนาเครื่องมือ..... 61
4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ..... 61
4.2 โครงสร้างฐานข้อมูลของเครื่องมือ..... 61
4.3 โครงสร้างของเครื่องมือ..... 63
5. การทดสอบเครื่องมือ..... 71
5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ..... 71
5.2 การทดสอบเครื่องมือ..... 71

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ หน้า

5.3 ผลการทดสอบเครื่องมือกับระบบงานจริง..... 75

6. สรุปผลการวิจัย..... 77

6.1 สรุปผลการวิจัย..... 77

6.2 ข้อจำกัดของเครื่องมือสร้างกรณีทดสอบ..... 78

6.3 แนวทางในการพัฒนาต่อ..... 78

6.4 ผลงานที่เกี่ยวข้องกับการวิจัย..... 79

รายการอ้างอิง..... 80

ภาคผนวก..... 82

ภาคผนวก ก พจนานุกรมข้อมูลของฐานข้อมูลของเครื่องมือ..... 84

ภาคผนวก ข ตัวอย่างผลการทดสอบเครื่องมือสร้างกรณีทดสอบกับระบบงานจริง 89

ภาคผนวก ค คู่มือการติดตั้งและการใช้งานเครื่องมือ..... 103

ภาคผนวก ง ผลงานที่ตีพิมพ์..... 114

ประวัติผู้เขียนวิทยานิพนธ์..... 121

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

		หน้า
ตารางที่ 3.1	รายละเอียดของยูสเคสกำหนดการเชื่อมต่อกับฐานข้อมูลทดสอบ.....	26
ตารางที่ 3.2	รายละเอียดของยูสเคสกำหนดชื่อสคีมาของฐานข้อมูล.....	28
ตารางที่ 3.3	รายละเอียดของยูสเคสสคีมาของฐานข้อมูลซ้ำซ้อน.....	29
ตารางที่ 3.4	รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูล.....	29
ตารางที่ 3.5	รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูลโดยผู้ใช้.....	30
ตารางที่ 3.6	รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่ต้องการทดสอบ.....	31
ตารางที่ 3.7	รายละเอียดของยูสเคสเปิดสคีมาของฐานข้อมูล.....	32
ตารางที่ 3.8	รายละเอียดของยูสเคสลบสคีมาของฐานข้อมูล.....	33
ตารางที่ 3.9	รายละเอียดของยูสเคสเลือกสคีมาของฐานข้อมูล.....	33
ตารางที่ 3.10	รายละเอียดของยูสเคสแก้ไขสคีมาของฐานข้อมูล.....	34
ตารางที่ 3.11	รายละเอียดของยูสเคสปิดสคีมาของฐานข้อมูล.....	35
ตารางที่ 3.12	รายละเอียดของยูสเคสสร้างกรณีทดสอบ.....	36
ตารางที่ 3.13	รายละเอียดของยูสเคสกำหนดรายละเอียดสำหรับการสร้างกรณีทดสอบ.....	37
ตารางที่ 3.14	รายละเอียดของยูสเคสผลลัพธ์.....	38
ตารางที่ 3.15	รายละเอียดของยูสเคสสร้างแฟ้มเอกสารเอ็กซ์เอ็มแอล.....	38
ตารางที่ 3.16	รายละเอียดของยูสเคสการเพิ่มกลุ่มค่าทั่วไปใหม่.....	39
ตารางที่ 3.17	รายละเอียดของยูสเคสแก้ไขกลุ่มค่าทั่วไป.....	40
ตารางที่ 3.18	รายละเอียดของยูสเคสลบกลุ่มค่าทั่วไป.....	41
ตารางที่ 3.19	รายละเอียดของยูสเคสเลือกกลุ่มค่าทั่วไป.....	42
ตารางที่ 3.20	รายละเอียดของยูสเคสเลือกกรณีทดสอบ.....	43
ตารางที่ 3.21	รายละเอียดของยูสเคสเลือกฐานข้อมูลเชื่อมต่อ.....	44
ตารางที่ 3.22	รายละเอียดของยูสเคสเรียกดูผลลัพธ์การทดสอบ.....	45
ตารางที่ 3.23	รายละเอียดของยูสเคสจัดการผลลัพธ์การทดสอบ.....	45

หน้า

ตารางที่ 5.1 รายละเอียดของสคีมาของฐานข้อมูลระบบการจัดการการศึกษาคณะวิทยาศาสตร์..... 72

ตารางที่ 5.2 สรุปผลการทดสอบเครื่องมือทดสอบโดยการเพิ่มข้อผิดพลาดในสคีมาของระบบที่นำมาทดสอบ..... 76

ตารางที่ ก-1 พจนานุกรมของตาราง DatabaseSchema..... 84

ตารางที่ ก-2 พจนานุกรมของตารางTable..... 84

ตารางที่ ก-3 พจนานุกรมของตาราง DatabaseConstraints..... 85

ตารางที่ ก-4 พจนานุกรมของตาราง FieldType..... 86

ตารางที่ ก-5 พจนานุกรมของตาราง NormalDependency..... 86

ตารางที่ ก-6 พจนานุกรมของตาราง Category..... 87

ตารางที่ ก-7 พจนานุกรมของตาราง CategoryValue..... 87

ตารางที่ ก-8 พจนานุกรมของตาราง TestCase..... 87

ตารางที่ ก-9 พจนานุกรมของตาราง TestResult..... 88

ตารางที่ ข-1 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางคณาจารย์..... 90

ตารางที่ ข-2 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางตำแหน่ง..... 91

ตารางที่ ข-3 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางคณะ.. 91

ตารางที่ ข-4 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางสาขาวิชา..... 91

ตารางที่ ข-5 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางระดับ.. 91

ตารางที่ ข-6 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางวิชาเรียน..... 92

ตารางที่ ข-7 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางกลุ่มเรียน..... 92

ตารางที่ ข-8 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางหลักสูตร..... 92

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

ตารางที่ ข-9 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางวันหยุด..... 92
ตารางที่ ข-10 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางกำหนดการ..... 93
ตารางที่ ข-11 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางคณาจารย์..... 95
ตารางที่ ข-12 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางคำนำหน้าชื่อ..... 95
ตารางที่ ข-13 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางวิชาเรียน..... 96
ตารางที่ ข-14 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางคณะ..... 96
ตารางที่ ข-15 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางห้องเรียน..... 96
ตารางที่ ข-16 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางสาขาวิชา..... 96
ตารางที่ ข-17 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางอุปกรณ์อำนวยความสะดวก..... 97
ตารางที่ ข-18 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางการใช้ห้องเรียน..... 97
ตารางที่ ข-19 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางข่าว..... 99
ตารางที่ ข-20 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางคณะ..... 99
ตารางที่ ข-21 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางคำนำหน้าชื่อ..... 100
ตารางที่ ข-22 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางตำแหน่ง..... 100

หน้า

ตารางที่ ข-23 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางตำแหน่ง..... 101
ตารางที่ ข-24 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางสาขาวิชา..... 101
ตารางที่ ข-25 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางประเภทของข่าว..... 101

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

		หน้า
รูปที่ 2.1	ตัวอย่างเอกสารเอ็กซ์เอ็มแอล	8
รูปที่ 2.2	ตัวอย่างเอ็กซ์เอ็มแอลสคีมา	10
รูปที่ 2.3	ความสัมพันธ์ระหว่างตาราง Customersตาราง Cus_Orders และตาราง Order_Details	15
รูปที่ 2.4	ผลการลบระเบียนตารางหลักที่ใช้คีย์เวิร์ด ON DELETE CASCADE	15
รูปที่ 2.5	ความสัมพันธ์ระหว่างตาราง Salesrep และตาราง Cus_Orders	16
รูปที่ 2.6	ผลการลบระเบียนตารางหลักที่ใช้คีย์เวิร์ด ON DELETE SET NULL	16
รูปที่ 3.1	โครงสร้างการทำงานของเครื่องมือ	20
รูปที่ 3.2	แผนภาพยูสเคสโดยรวม	25
รูปที่ 3.3	ยูสเคสภายในแพ็คเกจการจัดการสคีมาของฐานข้อมูล	27
รูปที่ 3.4	ยูสเคสภายในแพ็คเกจการสร้างกรณีทดสอบ	35
รูปที่ 3.5	ยูสเคสภายในแพ็คเกจการจัดการกลุ่มค่าทั่วไป	39
รูปที่ 3.6	ยูสเคสภายในแพ็คเกจการประมวลผลกรณีทดสอบ	42
รูปที่ 3.7	ยูสเคสในแพ็คเกจการจัดการผลลัพธ์การทดสอบ	44
รูปที่ 3.8	แผนภาพคลาสของเครื่องมือสร้างกรณีทดสอบ	46
รูปที่ 3.9	คลาส TestCaseTool	47
รูปที่ 3.10	คลาส DatabaseSchema	47
รูปที่ 3.11	คลาส Table	48
รูปที่ 3.12	คลาส DBConstriants	48
รูปที่ 3.13	คลาส Logicaldependency	49
รูปที่ 3.14	คลาส DBConnection	49
รูปที่ 3.15	คลาสTestCaseGen	49
รูปที่ 3.16	คลาส TestResult	50
รูปที่ 3.17	คลาสCategory	50
รูปที่ 3.18	แผนภาพซีเควนซ์แสดงการกำหนดชื่อสคีมาของฐานข้อมูล	51
รูปที่ 3.19	แผนภาพซีเควนซ์แสดงการสร้างสคีมาของฐานข้อมูลโดยผู้ใช้ระบุ	52

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

รูปที่ 3.20 แผนภาพซีเควนซ์แสดงการกำหนดการขึ้นต่อกันเชิงตรรกะ..... 53
รูปที่ 3.21 แผนภาพซีเควนซ์แสดงการกำหนดความคงสภาพในการอ้างอิง..... 53
รูปที่ 3.22 แผนภาพซีเควนซ์แสดงการเปิดสคีมาของฐานข้อมูล..... 54
รูปที่ 3.22 แผนภาพซีเควนซ์แสดงการปิดสคีมาของฐานข้อมูล..... 55
รูปที่ 3.23 แผนภาพซีเควนซ์แสดงการลบสคีมาของฐานข้อมูล..... 55
รูปที่ 3.24 แผนภาพซีเควนซ์แสดงการสร้างกรณีทดสอบ..... 56
รูปที่ 3.25 แผนภาพซีเควนซ์แสดงการประมวลผลกรณีทดสอบ..... 57
รูปที่ 3.26 แผนภาพซีเควนซ์แสดงการกำหนดการเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ..... 58
รูปที่ 3.27 แผนภาพซีเควนซ์แสดงการเพิ่มกลุ่มค่าทั่วไป..... 59
รูปที่ 3.28 แผนภาพซีเควนซ์แสดงการลบกลุ่มค่าทั่วไป..... 59
รูปที่ 3.29 แผนภาพซีเควนซ์แสดงการจัดการผลลัพธ์การทดสอบ..... 60
รูปที่ 4.1 โครงสร้างฐานข้อมูลของเครื่องมือ..... 62
รูปที่ 4.2 แผนภาพส่วนประกอบของเครื่องมือ..... 63
รูปที่ 4.3 หน้าจอหลัก..... 65
รูปที่ 4.4 หน้าจอสร้างสคีมาของฐานข้อมูลใหม่..... 65
รูปที่ 4.5 หน้าจอสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่ต้องการทดสอบ..... 66
รูปที่ 4.6 หน้าจอเปิดสคีมาของฐานข้อมูล..... 66
รูปที่ 4.7 หน้าจอลบสคีมาของฐานข้อมูล..... 67
รูปที่ 4.8 หน้าจอการรับข้อมูลเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์..... 67
รูปที่ 4.9 หน้าจอการสร้างกรณีทดสอบ..... 68
รูปที่ 4.10 หน้าจอการประมวลผลกรณีทดสอบ..... 68
รูปที่ 4.11 หน้าจอการกำหนดหรือระบุกลุ่มค่าทั่วไป..... 69
รูปที่ 4.12 หน้าจอการจัดการผลลัพธ์การทดสอบ..... 69
รูปที่ 4.13 หน้าจอเกี่ยวกับเครื่องมือ..... 70
รูปที่ 5.1 ตัวอย่างของสคีมาของฐานข้อมูล..... 73
รูปที่ 5.2 ตัวอย่างของกรณีทดสอบที่สร้างขึ้น..... 74
รูปที่ 5.3 ตัวอย่างของสคีมาของฐานข้อมูลที่มีการแก้ไขแล้ว..... 74

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

รูปที่ ข-1 แสดงตัวอย่างของกรณีทดสอบของระบบการจัดการศึกษาคณะวิทยาศาสตร์... 93

รูปที่ ข-2 แสดงตัวอย่างของกรณีทดสอบของระบบการจัดการศึกษาคณะวิทยาศาสตร์ (ต่อ)…….. 94

รูปที่ ข-3 แสดงตัวอย่างของกรณีทดสอบของระบบการจองการใช้ห้องปฏิบัติการ..... 97

รูปที่ ข-4 แสดงตัวอย่างของกรณีทดสอบของระบบการจองการใช้ห้องปฏิบัติการ(ต่อ).... 98

รูปที่ ข-5 แสดงตัวอย่างของกรณีทดสอบของระบบระบบการจัดฐานข้อมูลและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี..... 102

รูปที่ ค-1 หน้าจอเมนูการสร้างสคีมาของฐานข้อมูล..... 104

รูปที่ ค-2 หน้าจอเมนูการกำหนดชื่อสคีมาของฐานข้อมูล..... 104

รูปที่ ค-3 การอ่านสคีมาของฐานข้อมูล..... 105

รูปที่ ค-4 การกำหนดการเชื่อมต่อกับฐานข้อมูลเพื่อสร้างสคีมาของฐานข้อมูล..... 105

รูปที่ ค-5 เมนูการเปิดสคีมาของฐานข้อมูล..... 106

รูปที่ ค-6 หน้าจอการกำหนดรายละเอียดเงื่อนไขบังคับของเขตข้อมูล..... 107

รูปที่ ค-7 หน้าจอเมนูการสร้างกรณีทดสอบ..... 107

รูปที่ ค-8 หน้าจอการกำหนดรายละเอียดการสร้างกรณีทดสอบ..... 108

รูปที่ ค-9 หน้าจอเมนูการประมวลกรณีทดสอบ..... 108

รูปที่ ค-10 หน้าจอการกำหนดรายละเอียดการประมวลผลกรณีทดสอบ..... 109

รูปที่ ค-11 หน้าจอการเมนูการเพิ่มกลุ่มค่าทั่วไป..... 109

รูปที่ ค-12 หน้าจอการกำหนดรายละเอียดของการเพิ่มกลุ่มค่าทั่วไป..... 110

รูปที่ ค-13 หน้าจอการเมนูการแก้ไขกลุ่มค่าทั่วไป..... 110

รูปที่ ค-14 การเลือกกลุ่มค่าทั่วไปที่ต้องการแก้ไข..... 111

รูปที่ ค-15 หน้าจอการแก้ไขกลุ่มค่าทั่วไป..... 111

รูปที่ ค-16 หน้าจอการเมนูการลบกลุ่มค่าทั่วไป..... 112

รูปที่ ค-17 การเลือกกลุ่มค่าทั่วไปที่ต้องการลบ..... 112

รูปที่ ค-18 หน้าจอเมนูการจัดการผลลัพธ์ของการทดสอบ..... 113

รูปที่ ค-19 หน้าจอเกี่ยวกับข้อมูลของเครื่องมือ..... 113

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทดสอบซอฟต์แวร์ (Software Testing) เป็นขั้นตอนหนึ่งที่มีความสำคัญในกระบวนการพัฒนาซอฟต์แวร์ เพื่อให้ซอฟต์แวร์ที่พัฒนามีประสิทธิภาพน่าเชื่อถือ และถูกต้องตรงตามความต้องการของผู้ใช้งาน การทดสอบซอฟต์แวร์เป็นขั้นตอนที่ใช้แรงงาน (Effort) ระยะเวลา (Time) และค่าใช้จ่าย (Cost) ที่สูงพอกับการพัฒนาซอฟต์แวร์ ทั้งนี้เนื่องจากการทดสอบซอฟต์แวร์ครอบคลุมตั้งแต่กระบวนการวางแผนการทดสอบ และการประมวลผล (Execute) เพื่อค้นหาข้อผิดพลาด (Error) ที่อาจจะเกิดขึ้น จนได้ผลลัพธ์เพื่อนำมาวิเคราะห์หาสาเหตุและแก้ไขซอฟต์แวร์ให้มีความถูกต้องตรงความต้องการของซอฟต์แวร์นั้น ในปัจจุบันซอฟต์แวร์ถูกนำไปประยุกต์ใช้กับงานหลายด้าน อาทิ ซอฟต์แวร์ช่วยงานทางด้านการแพทย์ หรือซอฟต์แวร์ที่ควบคุมการทำงานของระบบที่ต้องการความถูกต้องสูง หากซอฟต์แวร์เหล่านี้มีการพัฒนาแต่ขาดการการทดสอบอย่างละเอียดไม่ครอบคลุมทุกกรณีที่อาจจะเกิดขึ้นแล้ว ย่อมส่งผลให้เกิดความเสียหายอย่างรุนแรงตามมา ดังนั้นการทดสอบซอฟต์แวร์จึงต้องมีประสิทธิภาพเพียงพอที่จะช่วยลดความเสี่ยงที่อาจจะเกิดขึ้นในระหว่างการใช้งานซอฟต์แวร์

ในปัจจุบันแอพพลิเคชันของฐานข้อมูล (Database Application) มีการพัฒนาอย่างต่อเนื่อง ทั้งนี้อาจเกิดจากการที่แอปพลิเคชันของฐานข้อมูลมีการเปลี่ยนแปลงไม่ว่าในกรณีต่างๆ อาทิ กระบวนการทางธุรกิจมีการเปลี่ยนแปลง เช่น ภาษีมูลค่าเพิ่มมีการปรับเปลี่ยน หรือการกำหนดสิทธิ์ในการเข้าฐานข้อมูลมีการเปลี่ยนแปลง เป็นต้น การเปลี่ยนแปลงดังกล่าวย่อมส่งให้แอปพลิเคชันของฐานข้อมูลนั้นมีความซับซ้อนมากยิ่งขึ้น

ในการทดสอบซอฟต์แวร์ สิ่งที่จำเป็นสำหรับผู้ทดสอบเพื่อใช้ในการทดสอบคือกรณีทดสอบ (Test Case) ซึ่งสามารถออกแบบกรณีทดสอบโดยแบ่งออกเป็น 2 ประเภทคือ แบบไวท์บอกซ์ (White-box testing) และแบบแบล็กบอกซ์ (Black-box testing) สำหรับการออกแบบกรณีทดสอบแบบแบล็กบอกซ์ เป็นการออกแบบโดยเน้นเรื่องของฟังก์ชันการทำงานของซอฟต์แวร์ตลอดจนกระบวนการต่างๆ ทำงานของซอฟต์แวร์ ทั้งนี้การออกแบบกรณีทดสอบค่อนข้างใช้ระยะเวลานานและมีค่าใช้จ่ายที่สูง เพื่อให้กรณีทดสอบที่สร้างขึ้นครอบคลุมทุกฟังก์ชันของการทำงานของซอฟต์แวร์ การสร้างกรณีทดสอบอัตโนมัติโดยใช้เครื่องมือซอฟต์แวร์นั้น จะช่วยลดค่าใช้จ่ายระยะเวลา ตลอดจนความถูกต้องในการพัฒนาและทดสอบซอฟต์แวร์

จากการศึกษาเครื่องมือที่ช่วยในการสร้างกรณีทดสอบฐานข้อมูลในปัจจุบัน พบว่าโดยเครื่องมือช่วย สร้างกรณีทดสอบเหล่านั้นมุ่งเน้นการสร้างกรณีทดสอบขึ้นเพื่อทำการทดสอบตามฟังก์ชันการทำงานของฐานข้อมูลเป็นส่วนใหญ่ เช่น กรณีของการสั่งซื้อสินค้าผ่านเว็บไซต์ กรณีทดสอบที่ถูกสร้างขึ้นนั้น จะเป็นกรณีทดสอบที่ผู้ซื้อสินค้าสามารถทำการสั่งซื้อสินค้าผ่านเว็บไซต์ได้หรือไม่เท่านั้น เป็นต้น ทั้งนี้การสร้างกรณีทดสอบสำหรับทำการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ยังเป็นอีกหนึ่งความสำคัญที่ผู้ทดสอบควรให้ความสนใจเช่นเดียวกัน ทั้งนี้เนื่องจากในขั้นตอนของการสร้างฐานข้อมูลขึ้น ฐานข้อมูลที่ถูกสร้างขึ้นอาจไม่เป็นไปตามเงื่อนไขที่ได้ทำการกำหนดไว้ในขั้นตอนการออกแบบ ทำให้ฐานข้อมูลที่ทำการออกแบบไม่ถูกต้องตามจุดประสงค์ที่แท้จริงที่ผู้ออกแบบได้กำหนดไว้

โดยปกติการแก้ไขและการจัดการข้อมูลของฐานข้อมูลจะอาศัยคำสั่งการเพิ่มข้อมูล (Insert) การลบข้อมูล (Delete) และการปรับปรุงข้อมูล (Update) โดยระบบบริหารจัดการฐานข้อมูล (Database Management Systems) จะทำหน้าที่ตรวจสอบความสมบูรณ์และความถูกต้องของข้อมูลที่กระทำกับฐานข้อมูลว่าสามารถดำเนินการกับข้อมูลได้หรือไม่ ซึ่งระบบบริหารจัดการฐานข้อมูลจะอาศัยเงื่อนไขข้อบังคับของฐานข้อมูลมาช่วยในการป้องกันข้อมูลในฐานข้อมูลไม่ให้สูญเสียความสมบูรณ์และความถูกต้องไป

นอกจากนี้ผู้วิจัยได้ทำการศึกษาเครื่องมือช่วยในการออกแบบฐานข้อมูลเชิงสัมพันธ์ พบว่าเครื่องมือที่ช่วยในการออกแบบฐานข้อมูลสามารถที่จะทำการกำหนดเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ได้บางในบางกรณี แต่ยังไม่ครอบคลุมเงื่อนไขบังคับบูรณภาพของการอ้างอิงถึง (Referential Integrity Constraints) ซึ่งอาจทำให้ผู้ใช้หรือผู้พัฒนามองข้ามการระบุหรือให้ความสำคัญของเงื่อนไขดังกล่าว ส่งผลให้ฐานข้อมูลที่สร้างขึ้นไม่ตรงตามความต้องการ

วิทยานิพนธ์นี้จึงออกแบบและพัฒนาเครื่องมือช่วยสร้างกรณีทดสอบสำหรับทำการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ โดยกรณีทดสอบที่สร้างขึ้นจะอยู่ในรูปแบบของคำสั่งการเพิ่มข้อมูล การลบข้อมูล และการปรับปรุงข้อมูล จากนั้นกรณีทดสอบจะถูกบันทึกลงไปในฐานข้อมูลของเครื่องมือทดสอบ เพื่อใช้ในการทดสอบฐานข้อมูล เมื่อทำการทดสอบแล้วเครื่องมือทดสอบจะทำการบันทึกผลลัพธ์การทดสอบและแสดงผลลัพธ์การทดสอบให้ผู้ใช้ทราบ ในกรณีที่พบข้อผิดพลาดขึ้นเครื่องมือทดสอบสามารถทำการแก้ไขปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ให้เป็นไปตามที่มีการออกแบบไว้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบและพัฒนาเครื่องมือช่วยสร้างกรณีทดสอบสำหรับทำการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์

1.3 ขอบเขตของการวิจัย

1. เครื่องมือสามารถรับข้อมูลเกี่ยวกับทดสอบฐานข้อมูลได้ 2 ลักษณะดังนี้

1.1 การรับสคีมาของฐานข้อมูลจากฐานข้อมูลที่จะทำการทดสอบโดยตรง โดยข้อมูลนำเข้าของเครื่องมือทดสอบ ได้แก่ ตารางข้อมูล และเขตของข้อมูล

1.2 การรับข้อมูลเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ (Relational Database Constraints) ที่ทำการทดสอบ ซึ่งประกอบด้วย 3 เงื่อนไข คือ เงื่อนไขบังคับบูรณภาพของเอนทิตี (Entity Integrity Constraints) เงื่อนไขบังคับบูรณภาพของการอ้างอิงถึงและ เงื่อนไขบังคับของโดเมน (Domain Constraints) โดยผู้ใช้เป็นผู้กำหนดขึ้นตามการออกแบบฐานข้อมูลผ่านเครื่องมือทดสอบและเครื่องมือทดสอบจะทำจัดเก็บไว้ในรูปแบบของแฟ้มเอกสารเอ็กซ์เอ็มแอล

2. สคีมาของฐานข้อมูลและเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ที่จะมาใช้ในการสร้างกรณีทดสอบอยู่ในรูปแบบของแฟ้มเอกสารเอ็กซ์เอ็มแอล

3. ฐานข้อมูลที่ทำการทดสอบจะต้องเป็นฐานข้อมูลเชิงสัมพันธ์

4. เครื่องมือทดสอบสามารถอ่านสคีมาของฐานข้อมูลได้ 3 ชนิด คือ ฐานข้อมูลมายเอสคิวแอล (MySQL database) ฐานข้อมูลออราเคิล (Oracle Database) และฐานข้อมูลไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ 2000 (Microsoft SQL Server 2000)

5. เครื่องมือทดสอบจะสร้างกรณีทดสอบสำหรับทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ ทั้งนี้กรณีทดสอบที่สร้างขึ้นจะอยู่ในรูปแบบของคำสั่งการเพิ่ม การลบ และการปรับปรุง โดยจัดเก็บกรณีทดสอบในลักษณะของแฟ้มเอกสารเอ็กซ์เอ็มแอล เพื่อใช้สำหรับทดสอบฐานข้อมูล

6. เครื่องมือทดสอบสามารถแสดงผลลัพธ์ของการทดสอบ โดยผลลัพธ์ของการทดสอบจะแสดงถึงรายละเอียดของการทดสอบ ได้แก่ กรณีทดสอบที่ทำการทดสอบ ผลของการ

ทดสอบ และครั้งที่ของการทดสอบ ทั้งนี้ผลลัพธ์ของการทดสอบจะถูกจัดเก็บลงฐานข้อมูลของเครื่องมือและแสดงผลลัพธ์ของการทดสอบในลักษณะของแฟ้มเอกสารเอ็กซ์เอ็มแอล

7. ในกรณีที่พบข้อผิดพลาดจากการทดสอบ เครื่องมือทดสอบสามารถปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ให้เป็นไปตามที่กำหนดไว้ในขั้นตอนการออกแบบ

8. เครื่องมือทดสอบรองรับคีย์หลักของตารางหลักที่ทำการอ้างอิงคีย์อ้างอิงในตารางลูกได้ 2 ระดับ

9. เครื่องมือทดสอบสามารถบันทึกข้อมูลที่สร้างขึ้นได้ 2 รูปแบบ คือ

9.1 บันทึกข้อมูลในลักษณะของแฟ้มเอกสารเอ็กซ์เอ็มแอล

9.2 บันทึกข้อมูลลงในฐานข้อมูล

10. เครื่องมือสามารถใช้งานบนระบบปฏิบัติการไมโครซอฟท์วินโดวน์ เอ็กซ์พี (Microsoft Window XP) ได้

11. เครื่องมือทดสอบที่พัฒนาเสร็จแล้วจะถูกนำไปทดสอบกับระบบงานอย่างน้อย 3 ระบบ โดยนำไปใช้สร้างกรณีทดสอบตามเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ของระบบงานนั้น และพิจารณากรณีทดสอบที่สร้างขึ้นว่าเป็นไปตามการเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์นั้นหรือไม่

1.4 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำเครื่องมือสร้างกรณีทดสอบที่นำเสนอไปใช้เป็นทางเลือกในการสร้างกรณีทดสอบสำหรับทดสอบความถูกต้องของการกำหนดเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์

1.5 วิธีดำเนินการวิจัย

1.สำรวจและศึกษาเครื่องมือสร้างกรณีทดสอบสำหรับทดสอบฐานข้อมูลที่มีอยู่ในปัจจุบัน

2. ศึกษารูปแบบของการกำหนดเงื่อนไขข้อบังคับบูรณภาพของฐานข้อมูล

3. ศึกษาวิธีการเขียนและการใช้งานภาษาเอ็กซ์เอ็มแอล

4. กำหนดขอบเขตความสามารถที่เครื่องมือทดสอบจะทำได้

5. ออกแบบและกำหนดวิธีการสร้างกรณีทดสอบของเครื่องมือที่จะพัฒนา

6. พัฒนาเครื่องมือสร้างกรณีทดสอบตามที่ได้ออกแบบไว้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

7. ทดสอบเครื่องมือสร้างกรณีทดสอบตามขอบเขตที่ได้กำหนดไว้
8. สรุปผลการวิจัย และข้อเสนอแนะ
9. จัดทำรายงานวิทยานิพนธ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์นี้ ได้แก่ การทดสอบซอฟต์แวร์ เอ็กซ์เอ็มแอล เอ็กซ์เอ็มแอลสคีมา ภาษาเอสคิวแอล และเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ โดยมีรายละเอียดดังนี้

2.1.1 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ [1] หมายถึง กระบวนการในการใช้งานหรือประเมินค่าซอฟต์แวร์ทั้งการทำด้วยมือ หรือทำอย่างอัตโนมัติ เพื่อที่จะตรวจสอบว่าซอฟต์แวร์เป็นไปตามความต้องการของซอฟต์แวร์ (Software requirements) หรือเพื่อที่จะระบุความแตกต่างระหว่างผลลัพธ์ที่คาดหวังกับผลลัพธ์ที่ได้จริงจากซอฟต์แวร์

เป้าหมายของการทดสอบซอฟต์แวร์ [2] มีดังนี้

1. สามารถป้องกันความผิดพลาด (Bug) ไม่ให้เกิดขึ้นกับซอฟต์แวร์ได้
2. ถ้าการทดสอบไม่สามารถป้องกันความผิดพลาดไม่ให้เกิดขึ้นได้ ก็ควรจะสามารถบอกได้ว่าความผิดพลาดนั้นจะส่งผลกระทบต่อซอฟต์แวร์อย่างไร
3. การทดสอบควรจะบอกแนวทางแก้ไขความผิดพลาดที่ชัดเจน เพื่อให้สามารถแก้ไขความผิดพลาดได้อย่างง่ายดาย

การทดสอบซอฟต์แวร์แบ่งเป็นระดับได้ 4 ระดับ [2] ดังนี้

1. การทดสอบหน่วยโปรแกรม (Unit Testing) คือการทดสอบหน่วยโปรแกรมหลังจากที่เขียนโปรแกรมในหน่วยนั้นเสร็จแล้ว
2. การทดสอบแบบบูรณาการ (Integration Testing) คือการทดสอบซอฟต์แวร์โดยนำแต่ละหน่วยโปรแกรมที่ผ่านการทดสอบหน่วยโปรแกรมมาแล้วมาทำงานร่วมกัน
3. การทดสอบระบบ (System Testing) คือการทดสอบว่าซอฟต์แวร์ที่พัฒนาได้นั้นทำงานได้ตรงกับข้อกำหนดความต้องการของลูกค้า (Customer requirements) หรือไม่
4. การทดสอบการตรวจรับ (Acceptance Testing) คือการนำเอาซอฟต์แวร์ไปทดสอบกับสิ่งแวดล้อมทางปฏิบัติที่ซอฟต์แวร์จะถูกนำไปใช้งานจริงเพื่อยืนยันว่าซอฟต์แวร์นั้นสามารถทำงานได้ตามความต้องการที่กำหนดไว้จริงๆ

2.1.2 **เอ็กซ์เอ็มแอล** (XML – eXtensible Markup Language) [3]

เอ็กซ์เอ็มแอล เป็นภาษาที่ใช้เป็นมาตรฐานในการอธิบายข้อมูล เอ็กซ์เอ็มแอลกำเนิดขึ้นมาจากสาเหตุที่ว่า ในปัจจุบันการจัดเก็บข้อมูลที่มีโครงสร้างเป็นลำดับชั้น (Hierarchy) ถูกนำไปใช้ในการสื่อสารระหว่างโปรแกรมต่างๆ บนอินเตอร์เน็ตมากขึ้น และเนื่องจากข้อมูลเหล่านี้มักจะผูกติดกับซอฟต์แวร์ที่สร้างข้อมูลหรือระบบปฏิบัติการที่ใช้สร้างข้อมูลนั้น จึงทำให้การส่งผ่านข้อมูลระหว่างโปรแกรมที่อยู่ต่างระบบปฏิบัติการนั้นค่อนข้างทำได้ยาก ด้วยเหตุนี้จึงจำเป็นต้องมีการกำหนดภาษากลางขึ้นมาเพื่อเป็นมาตรฐานที่จะใช้ในการนิยามและอธิบายข้อมูล และภาษากลางนั้นก็คือเอ็กซ์เอ็มแอลนั่นเอง การนำเอ็กซ์เอ็มแอลไปใช้งานนั้น นอกจากจะใช้ตัวภาษาเอ็กซ์เอ็มแอลในการนิยามและอธิบายข้อมูลแล้ว ยังต้องมีการประกาศใช้ ดีทีดี (DTD – Document Type Declarations) เพื่อนิยามโครงสร้างของข้อมูลในเอกสารเอ็กซ์เอ็มแอลด้วย

ภาษาเอ็กซ์เอ็มแอลเป็นภาษามาร์คอัพ (Markup Language) ดังนั้นข้อมูลภายในเอกสารเอ็กซ์เอ็มแอลจึงประกอบไปด้วยอิลีเมนต์ (Elements) และแอททริบิวต์ (Attribute) ต่างๆ ของแต่ละอิลีเมนต์ที่ใช้ในการอธิบายข้อมูล โดยแต่ละอิลีเมนต์จะเขียนแทนด้วยแท็ก (Tag) ซึ่งเมื่อจบการใช้งานอิลีเมนต์ต้องมีแท็กปิดเสมอ เอกสารเอ็กซ์เอ็มแอลนั้นสามารถแบ่งได้เป็น 2 ชนิดคือ

1.) เอกสารเอ็กซ์เอ็มแอลที่มีรูปแบบถูกต้อง (Well-Formed XML Document) คือ เอกสารเอ็กซ์เอ็มแอลที่มีคุณสมบัติถูกต้องตามกฎข้อบังคับพื้นฐานของการเขียนเอกสารเอ็กซ์เอ็มแอล โดยทุกๆ เอกสารเอ็กซ์เอ็มแอลจะต้องมีรูปแบบถูกต้องเป็นอย่างน้อย สำหรับกฎข้อบังคับพื้นฐานของการเขียนเอกสารเอ็กซ์เอ็มแอลมีดังนี้

- เอกสารต้องมีอิลีเมนต์ระดับบนสุดเพียงอิลีเมนต์เดียวเท่านั้น (ซึ่งเรียกว่า อิลีเมนต์เอกสาร (Document Element) หรือ อิลีเมนต์ราก (Root Element)) โดยอิลีเมนต์อื่นๆ ทั้งหมดจะต้องซ้อนอยู่ภายใน เช่น จากรูปที่ 2.1 อิลีเมนต์รากคือ <BookStore>
- อิลีเมนต์ต้องซ้อนอยู่อย่างถูกต้องนั้น หมายความว่าถ้ามีอิลีเมนต์ซ้อนอยู่ภายในอิลีเมนต์อื่นๆ ต้องปิดอิลีเมนต์นั้นภายในอิลีเมนต์เดียวกับที่อิลีเมนต์นั้นซ้อนอยู่ เช่น จากรูปที่ 2.1 อิลีเมนต์ <Book> ซ้อนอยู่ในอิลีเมนต์ <BookStore> อย่างถูกต้อง
- แต่ละอิลีเมนต์ต้องมีแท็กเริ่มต้นและแท็กปิดท้าย ถ้าแท็กไม่ครบคู่จะทำให้บราวเซอร์ (Browser) ไม่เข้าใจคำสั่งของอิลีเมนต์นั้น
- ชื่อของอิลีเมนต์ในแท็กเริ่มต้นจะต้องมีชื่อตรงกับชื่อของแท็กปิดท้ายเท่านั้น

- ชื่อของอิลีเมนต์มีลักษณะการแยกระหว่างตัวอักษรใหญ่และตัวอักษรเล็ก (Case-sensitive) ซึ่งอันที่จริงข้อความทั้งหมดภายในเอกสารจะมีลักษณะเป็นการแยกระหว่างตัวอักษรใหญ่และตัวอักษรเล็ก ทั้งหมด

```
<?xml version="1.0"?>
<!DOCTYPE BookStore
        [
        <!ELEMENT BookStore (Book+)>
        <!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>
        <!ELEMENT Title (#PCDATA)>
        <!ELEMENT Author (#PCDATA)>
        <!ELEMENT Date (#PCDATA)>
        <!ELEMENT ISBN (#PCDATA)>
        <!ELEMENT Publisher (#PCDATA)>
        ]
        >
<BookStore>
    <Book>
        <Title>My Life and Times</Title>
        <Author>Paul McCartney</Author>
        <Date>1998</Date>
        <ISBN>1-56592-235-2</ISBN>
        <Publisher>McMillin Publishing</Publisher>
    </Book>
    <Book>
        <Title>Illusions The Adventures of a Reluctant Messiah</Title>
        <Author>Richard Bach</Author>
        <Date>1977</Date>
        <ISBN>0-440-34319-4</ISBN>
        <Publisher>Dell Publishing Co.</Publisher>
 </Book>
</BookStore>
```

DTD

XML Data

รูปที่ 2.1 ตัวอย่างเอกสารเอ็กซ์เอ็มแอล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.) เอกสารเอ็กซ์เอ็มแอลที่ถูกต้องสมบูรณ์ (Valid XML Document) คือเอกสารเอ็กซ์เอ็มแอลที่มีรูปแบบถูกต้องที่มีการเพิ่มความต้องการ (Requirement) ที่เป็นส่วนเสริม 2 ข้อเข้าไปในเอกสารคือ

- ต้องมีการประกาศการใช้ดีทีดีสำหรับเอกสาร ซึ่งข้อมูลภายในดีทีดีจะเป็นตัวกำหนดโครงสร้างของเอกสารนั้น เช่น จากรูปที่ 2.1 อิลีเมนต์ <!DOCTYPE> คืออิลีเมนต์ที่ประกาศการใช้ดีทีดี โดยภายในดีทีดีอธิบายว่าอิลีเมนต์ <BookStore> ต้องประกอบด้วยอิลีเมนต์ <Book> ตั้งแต่ 1 อิลีเมนต์ขึ้นไป และอิลีเมนต์ <Book> ต้องประกอบด้วยอิลีเมนต์ <Title> อิลีเมนต์ <Author> อิลีเมนต์ <Date> อิลีเมนต์ <ISBN> และอิลีเมนต์ <Publisher> โดยทั้ง 5 อิลีเมนต์จะต้องมีเนื้อหาภายในอิลีเมนต์เป็นข้อมูลแบบอักขระเท่านั้น เป็นต้น

- ส่วนต่างๆ ภายในเอกสารต้องถูกต้องตามโครงสร้างที่กำหนดในดีทีดี จากรูปที่ 2.1 จะเห็นได้ว่าเนื้อหาภายในอิลีเมนต์ <BookStore> นั้นเป็นไปตามดีทีดี

2.1.3 **เอ็กซ์เอ็มแอลสคีมา** (XML Schemas) [4]

เอ็กซ์เอ็มแอลสคีมาเป็นรูปแบบคำสั่งที่ใช้นิยามและอธิบายโครงสร้างข้อมูลในเอกสารเอ็กซ์เอ็มแอลเหมือนกับดีทีดี เอ็กซ์เอ็มแอลสคีมาถูกสร้างขึ้นมาด้วยสาเหตุที่ว่าดีทีดีนั้นยังไม่สามารถใช้อธิบายโครงสร้างข้อมูลในเอกสารเอ็กซ์เอ็มแอลได้อย่างมีประสิทธิภาพ เนื่องจากไวยากรณ์ที่ใช้อธิบายข้อมูลของดีทีดีนั้นแตกต่างจากภาษาเอ็กซ์เอ็มแอล ทำให้ผู้ใช้ต้องจำไวยากรณ์ถึง 2 อย่างเพื่อใช้งานเอ็กซ์เอ็มแอล และดีทีดีนั้นกำหนดชนิดข้อมูลมาให้ใช้งานน้อยมากเพียง 10 ชนิดเท่านั้น ทำให้การใช้งานเอ็กซ์เอ็มแอลไม่มีความยืดหยุ่นเท่าที่ควร ดังนั้นเอ็กซ์เอ็มแอลสคีมาจึงถูกสร้างขึ้นมากำจัดข้อบกพร่องเหล่านี้ให้หมดไป

ความสามารถของเอ็กซ์เอ็มแอลสคีมา ได้แก่

1.) สนับสนุนชนิดข้อมูลทั้งหมด 44 ชนิด และผู้ใช้สามารถสร้างชนิดข้อมูลเพิ่มเติมเองได้

2.) ใช้ไวยากรณ์แบบเดียวกันกับภาษาเอ็กซ์เอ็มแอล

3.) มีความเป็นเชิงวัตถุ (Object-oriented) คือสามารถสร้างชนิดข้อมูลใหม่โดยสืบทอด (Extend) หรือจำกัด (Restrict) รูปแบบการใช้งานของชนิดข้อมูลเดิมที่มีอยู่ได้

4.) สามารถแสดงข้อมูลในแบบเซต (Set) ได้

5.) สามารถกำหนดให้อิลีเมนต์หลายๆ อันมีชื่อเดียวกัน แต่มีเนื้อหาที่ต่างกันได้

ตัวอย่างของเอ็กซ์เอ็มแอลสคีมาที่อธิบายโครงสร้างของอิลีเมนต์ <BookStore> แบบเดียวกับดีทีดีในรูปที่ 2.1 แสดงได้ดังรูปที่ 2.2

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xsd:element name="BookStore">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Book" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Book">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
       <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Title" type="xsd:string"/>
  <xsd:element name="Author" type="xsd:string"/>
  <xsd:element name="Date" type="xsd:string"/>
  <xsd:element name="ISBN" type="xsd:string"/>
  <xsd:element name="Publisher" type="xsd:string"/>
 </xsd:shema>
```

รูปที่ 2.2 ตัวอย่างเอ็กซ์เอ็มแอลสคีมา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.1.4 ภาษาเอสคิวแอล [5]

ภาษาเอสคิวแอล หรือภาษาสอบถามเชิงโครงสร้าง (Structured Query Language: SQL, อ่านอย่างย่อว่า เอสคิวแอล, ซีเควล, ซีควล) เป็นภาษาทางด้านฐานข้อมูล ที่สามารถสร้างและปฏิบัติการกับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) โดยเฉพาะ และเป็นภาษาที่มีลักษณะคล้ายกับภาษาอังกฤษ ภาษาเอสคิวแอลถูกพัฒนาขึ้นจากแนวคิดของความสัมพันธ์แคลคูลัส (Relational Calculus) และความสัมพันธ์พีชคณิต (Relational Algebra) เป็นหลัก ภาษาเอสคิวแอลเริ่มพัฒนาครั้งแรกโดยบริษัทไอบีเอ็ม ใช้ชื่อเริ่มแรกว่า “ซีเควล” (Sequel) ต่อมาได้เปลี่ยนชื่อเป็น“เอสคิวแอล” (SQL) หลังจากนั้นภาษาเอสคิวแอลได้ถูกนำมาพัฒนาโดยผู้ผลิตซอฟต์แวร์ด้านระบบจัดการฐานข้อมูลเชิงสัมพันธ์จนเป็นที่นิยมกันอย่างแพร่หลายในปัจจุบัน โดยผู้ผลิตแต่ละรายก็พยายามที่จะพัฒนาระบบจัดการฐานข้อมูลของตนให้มีลักษณะโดดเด่นเฉพาะขึ้นมา ทำให้รูปแบบการใช้คำสั่งเอสคิวแอลมีรูปแบบที่แตกต่างกันไปบ้าง เช่น ออราเคิล หรือเอสคิวแอลเซิร์ฟเวอร์ของไมโครซอฟท์เป็นต้น

ดังนั้นในปี ค.ศ. 1986 ทางด้านสถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา หรือแอนซี (American National Standards Institute: ANSI) จึงได้กำหนดมาตรฐานของภาษาเอสคิวแอลขึ้น

1.) ประเภทของคำสั่งของภาษาเอสคิวแอล

ภาษาเอสคิวแอลเป็นภาษาที่ใช้งานได้ตั้งแต่ระดับเครื่องคอมพิวเตอร์ส่วนบุคคลไปจนถึงระดับเมนเฟรม ประเภทของคำสั่งในภาษาเอสคิวแอลแบ่งออกเป็น 3 ประเภท คือ

1.1 ภาษาสำหรับการนิยามข้อมูล (Data Definition Language: DDL) ประกอบด้วยคำสั่งที่ใช้ในการกำหนดโครงสร้างข้อมูลว่ามีสดมภ์อะไร แต่ละสดมภ์เก็บข้อมูลประเภทใด รวมถึงการเพิ่มสดมภ์ การกำหนดดัชนี การกำหนดวิว (View) หรือตารางเสมือนของผู้ใช้ เป็นต้น

1.2 ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML) ประกอบด้วยคำสั่งที่ใช้ในการเรียกใช้ข้อมูล การเปลี่ยนแปลงข้อมูล การเพิ่มหรือลบข้อมูล เป็นต้น

1.3 ภาษาควบคุม (Data Control Language: DCL) ประกอบด้วยคำสั่งที่ใช้ในการควบคุม การเกิดภาวะพร้อมกัน หรือการป้องกันการเกิดเหตุการณ์ที่ผู้ใช้หลาย

คนเรียกใช้ข้อมูลพร้อมกัน และคำสั่งที่เกี่ยวข้องกับการควบคุมความปลอดภัยของข้อมูลด้วยการกำหนดสิทธิของผู้ใช้ที่แตกต่างกัน เป็นต้น

2.) ชนิดของข้อมูลที่ใช้ในภาษาเอสคิวแอล

ในภาษาเอสคิวแอล การบรรจุข้อมูลลงในสดมภ์ต่าง ๆ ของตารางจะต้องกำหนดชนิดของข้อมูล (Data Type) ให้แต่ละสดมภ์ ชนิดของข้อมูลนี้จะแสดงชนิดของค่าที่อยู่ในสดมภ์ ค่าทุกค่าในสดมภ์ที่กำหนดจะต้องเป็นชนิดเดียวกัน เช่น ในตารางลูกค้าสดมภ์ที่เป็นรายชื่อลูกค้าจะต้องเป็นตัวอักขระ ในขณะที่สดมภ์จำนวนเงินที่ลูกค้าซื้อสินค้าเป็นตัวเลข

ชนิดของข้อมูลของแต่ละสดมภ์จะขึ้นกับลักษณะของข้อมูลแต่ละสดมภ์ แบ่งชนิดข้อมูลพื้นฐานในภาษาเอสคิวแอล ได้ดังนี้

2.1 ตัวอักขระ (Character)

- ตัวอักขระแบบความยาวคงที่ (Fixed-Length Character) จะใช้ char (n) หรือ character (n) แทนประเภทของข้อมูลที่เป็นตัวอักขระใดๆที่มีความยาวของข้อมูลคงที่โดยมีความยาว n ตัวอักขระประเภทนี้จะมีการจองเนื้อที่ตามความยาวที่คงที่ตามที่กำหนดไว้ ชนิดของข้อมูลประเภทนี้จะเก็บความยาวของข้อมูลได้มากที่สุดได้ 255 ตัวอักขระ

- ตัวอักขระแบบความยาวไม่คงที่ (Variable-Length Character) จะใช้ varchar (n) แทนประเภทของข้อมูลที่เป็นตัวอักขระใดๆที่มีความยาวของข้อมูลไม่คงที่ โดยมีความยาว n ตัวอักขระประเภทนี้จะมีการจองเนื้อที่ตามความยาวของข้อมูล ชนิดของข้อมูลประเภทนี้จะเก็บความยาวของข้อมูลได้มากที่สุดได้ 4000 ตัวอักขระ

2.2 จำนวนเลข (Numeric)

- จำนวนเลขที่มีจุดทศนิยม (Decimal) จะใช้ dec(m,n) หรือ decimal(m,n) เป็นประเภทข้อมูลที่เป็นจำนวนเลขที่มีจุดทศนิยมโดย m คือจำนวนตัวเลขทั้งหมด (รวมจุดทศนิยม) และ n คือจำนวนตัวเลขหลังจุดทศนิยม

- จำนวนเลขที่ไม่มีจุดทศนิยม จะใช้ int หรือ integer เป็นเลขจำนวนเต็มบวกหรือลบขนาดใหญ่ เป็นตัวเลข 10 หลัก ที่มีค่าตั้งแต่ –2,147,483,648 ถึง +2,147,483,647 และจะใช้ smallint เป็นประเภทข้อมูลที่เป็นเลขจำนวนเต็มบวกหรือลบขนาดเล็ก เป็นตัวเลข 5 หลัก ที่มีค่าตั้งแต่ – 32,768 ถึง + 32,767 ตัวเลขจำนวนเต็มประเภทนี้จะมีการจองเนื้อที่น้อยกว่าแบบ integer

- เลขจำนวนจริง อาจใช้ number (n) แทนจำนวนเลขที่ไม่มีจุดทศนิยม และจำนวนเลขที่มีจุดทศนิยม

2.3 ข้อมูลในลักษณะอื่นๆ

- วันที่และเวลา (Date/Time) เป็นชนิดวันที่หรือเวลา จะใช้ date เป็นข้อมูลวันที่ ซึ่งจะมีหลายรูปแบบให้เลือกใช้ เช่น yyyy-mm-dd (1999-10-31) dd.mm.yyyy (31. 10.1999) หรือ dd/mm/yyyy (31/10/1999)

3.) ลักษณะการใช้งาน

ภาษาเอสคิวแอลเป็นส่วนประกอบหนึ่งของระบบการจัดการฐานข้อมูล มักพบในระบบการจัดการฐานข้อมูลเชิงสัมพันธ์หลายตัวและเป็นที่นิยมใช้ในปัจจุบัน การใช้งานในภาษาเอสคิวแอลแบ่งเป็น 2 ลักษณะ คือ ภาษาเอสคิวแอลที่โต้ตอบได้ (Interactive SQL) และภาษาเอสคิวแอลที่ฝังในโปรแกรม (Embedded SQL)

- ภาษาเอสคิวแอลที่โต้ตอบได้ ใช้เพื่อปฏิบัติงานกับฐานข้อมูลโดยตรง เป็นการใช้คำสั่งเอสคิวแอลสั่งงานบนจอภาพ โดยเรียกดูข้อมูลได้โดยตรงในขณะที่ทำงาน เพื่อให้ได้ผลลัพธ์ที่นำไปใช้ได้

- ภาษาเอสคิวแอลที่ฝังในโปรแกรม เป็นภาษาเอสคิวแอลที่ประกอบด้วยคำสั่งต่าง ๆ ของ ภาษาเอสคิวแอล ที่ใส่ไว้ในโปรแกรมที่ส่วนมากแล้วเขียนด้วยภาษาอื่น เช่น ภาษาซี/ซีพลัสพลัส (C/C++) ภาษาวิชวลเบสิก(Visual Basic) ภาษาเดลฟี (Delphi) ภาษาจาวา(Java) เป็นต้น ลักษณะของคำสั่งจะแตกต่างจากภาษาอื่นๆ ในแง่ที่ว่าภาษาเอสคิวแอลไม่มีคำสั่งที่เกี่ยวกับการควบคุม (Control Statement) เหมือนภาษาอื่น เช่น if..Then…else for…do หรือ Loop หรือ while ทำให้มีข้อจำกัดในการเขียนชุดคำสั่งงาน การใช้ภาษาเอสคิวแอลฝังในโปรแกรมอื่น จะทำให้ภาษาเอสคิวแอลมีความสามารถและมีประสิทธิภาพมากยิ่งขึ้น ผลลัพธ์ของคำสั่งที่เกิดจากภาษาเอสคิวแอลที่ฝังในโปรแกรมจะถูกส่งผ่านไปให้กับตัวแปรหรือพารามิเตอร์ที่ใช้ โดยโปรแกรมที่ภาษาเอสคิวแอลไปฝังตัวอยู่

2.1.5 **เงื่อนไขของฐานข้อมูลเชิงสัมพันธ์** (Relational Database Constraints) [6]

เงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ เป็นเงื่อนไขที่ผู้ใช้หรือผู้ออกแบบฐานข้อมูลทำการระบุหรือกำหนดไว้ในฐานข้อมูลเชิงสัมพันธ์ เพื่อเป็นการรับประกันข้อมูลจะมีความถูกต้องและไม่มีความซ้ำซ้อนในฐานข้อมูลเชิงสัมพันธ์ เงื่อนไขของฐานข้อมูลเชิงสัมพันธ์แบ่งออกได้ 3 เงื่อนไข

หลัก คือ เงื่อนไขบังคับบูรณภาพเอนทิตี เงื่อนไขบังคับบูรณภาพการอ้างอิงถึง และเงื่อนไขบังคับของโดเมน

1.) เงื่อนไขบังคับบูรณภาพของเอนทิตี คือ เงื่อนไขที่ระบุว่าตารางหนึ่งจะต้องประกอบด้วยคีย์หลัก ซึ่งในแต่ละระเบียน (Record) จะต้องเป็นค่าที่ไม่ซ้ำกัน (Unique) หรือต้องเป็นค่าที่ไม่เป็นค่าว่าง (Null)

ตัวอย่างเช่น ตาราง Product จะมีสดมภ์รหัสสินค้า (Product_id) แสดงให้เห็นว่าระเบียนจะมีค่าเป็นไม่ซ้ำกัน ตามมาตรฐานของแอนซี ได้สนับสนุนเงื่อนไขบังคับบูรณภาพของเอนทิตี ด้วยประโยคคีย์หลักในคำสั่งการสร้างตาราง (Create Table) และปรับปรุงตาราง (Alter Table) โดยปกติคีย์หลักจะมีได้เพียงหนึ่งคีย์เท่านั้นต่อหนึ่งตาราง แต่สามารถใช้คีย์เวิร์ด unique สำหรับสดมภ์ที่สองได้

2.) เงื่อนไขบังคับบูรณภาพของการอ้างอิงถึง คือ เงื่อนไขที่ระบุว่าถ้าคีย์อ้างอิงมีข้อมูลหรือค่าอยู่ค่านั้นจะต้องอ่างอิงกับค่าที่มีอยู่แล้วในตารางหลัก

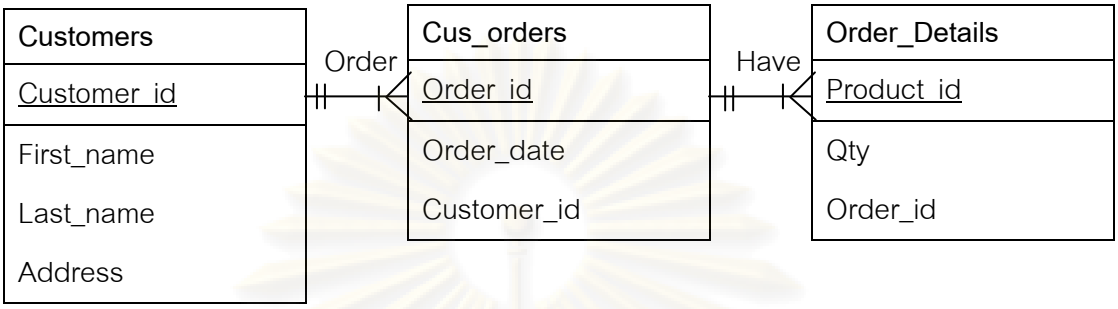
ตัวอย่างเช่น ตาราง Cus_Order จะมีคีย์อ้างอิงที่สดมภ์Customer_id เพื่อเชื่อมความสัมพันธ์กับระเบียนในตาราง Customer ตามมาตรฐานของแอนซี ได้สนับสนุนข้อกำหนดคีย์อ้างอิงด้วยประโยค References ในคำสั่งการสร้างตารางและปรับปรุงตาราง เมื่อมีการใช้คีย์อ้างอิง และใช้คำสั่งเพิ่ม หรือปรับปรุง แล้วต้องการสร้างคีย์อ้างอิงในตารางลูกโดยปราศจากการจับคู่กับคีย์หลักในตารางหลักแล้ว ระบบฐานข้อมูลจะปฏิเสธการใช้ค่าสั่งนั้น

ในกรณีที่พยายามทำการปรับปรุงหรือลบค่าคีย์หลักในตารางหลัก ซึ่งมีการจับคู่กับค่าคีย์อ้างอิงได้บางค่าในตารางลูกได้หรือไม่นั้น จะขึ้นอยู่กับการกำหนดประโยคย่อย ON UPDATE หรือ ON DELETE ในประโยคคีย์อ้างอิง ภาษาเอสคิวแอลจะสนับสนุนการกระทำการลบดังต่อไปนี้ [6]

- กรณีการลบระเบียนจากตารางหลัก จะต้องลบระเบียนที่จับคู่ได้ในตารางลูกโดยอัตโนมัติ เนื่องจากการลบระเบียนทั้งหมดจะมีคีย์หลักที่ใช้อ้างอิงคีย์อ้างอิงในตารางอื่นๆ มีผลทำให้ตารางอื่นๆที่มีระเบียนจับคู่ได้จะถูกลบออกได้โดยอัตโนมัติในลักษณะตกเป็นทอดๆ (Cascade)

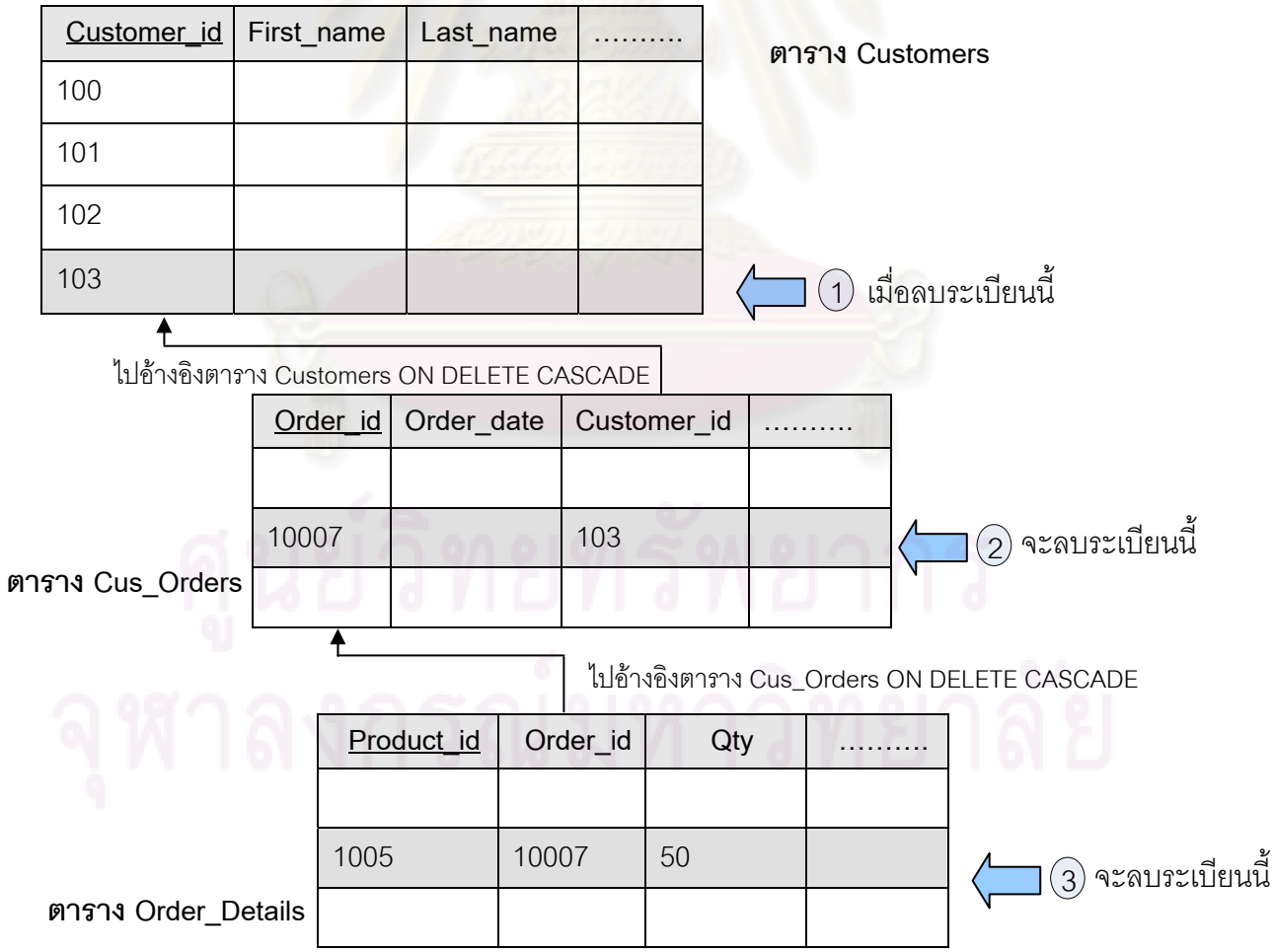
ตัวอย่างเช่น ตาราง Customers (ตารางหลัก) มี Customer_id เป็นคีย์หลัก และถูกอ้างอิงจากตาราง Cus_Orders (ตารางลูก) พร้อมมีการใช้งานคีย์เวิร์ด ON DELETE CASCADE และขณะเดียวกันตาราง Cus_Orders (ตารางหลัก) มี Order_id เป็นคีย์หลัก และถูก

อ้างอิงโดยตาราง Order_Details (ตารางลูก) พร้อมมีการใช้งานคีย์เวิร์ด ON DELETE CASCADE ดังรูปที่ 2.3



รูปที่ 2.3 ความสัมพันธ์ระหว่างตาราง Customersตาราง Cus_Orders และตารางOrder_Details

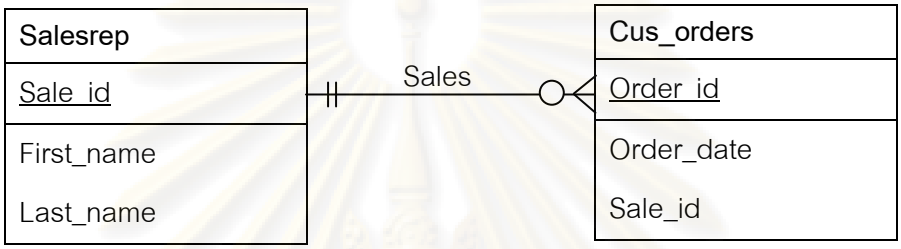
ถ้าต้องการลบเฉพาะระเบียนที่มีรหัสลูกค้าเท่ากับ 103 จากตาราง Customers ผลลัพธ์ที่ได้จะทำการลบระเบียนตาราง Cus_Order ที่ Customer_id = 103 และลบระเบียนตาราง Order_Details ที่ Order_id เท่ากับ 10007 อีกด้วยดังรูปที่ 2.4



รูปที่ 2.4 ผลการลบระเบียนตารางหลักที่ใช้คีย์เวิร์ด ON DELETE CASCADE

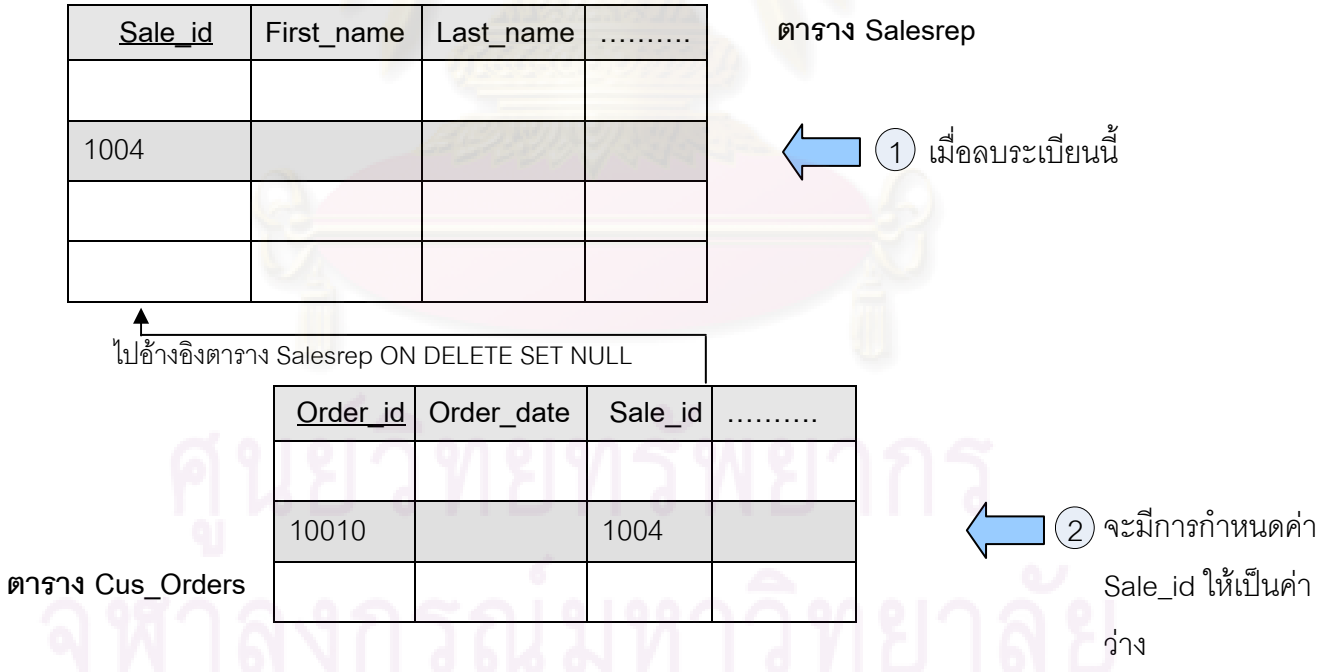
-กรณีการลบระเบียนจากตารางหลัก จะทำการลบระเบียนที่จับคู่ได้ในตารางลูก แล้วกำหนดสดมภ์ของคีย์อ้างอิงในตารางลูกให้เป็นค่าว่าง และจะใช้เมื่อสดมภ์คีย์อ้างอิงในตารางลูกจะต้องไม่เป็นค่าว่าง

ตัวอย่างเช่น ตาราง Salesrep (ตารางหลัก) มี Sale_id เป็นคีย์หลักและถูกอ้างอิงโดยตาราง Cus_Orders (ตารางลูก) พร้อมกับการใช้งานคีย์เวิร์ด ON DELETE SET NULL ดังรูปที่ 2.5



รูปที่ 2.5 ความสัมพันธ์ระหว่างตาราง Salesrep และตาราง Cus_Orders

ถ้าต้องการลบระเบียนตารางหลักที่ Sale_id= 1004 ผลลัพธ์จะทำให้ระเบียนของตารางลูกเซตค่าเป็น NULL ดังรูปที่ 2.6



รูปที่ 2.6 ผลการลบระเบียนตารางหลักที่ใช้คีย์เวิร์ด ON DELETE SET NULL

- กรณีเมื่อทำการลบระเบียนจากตารางหลัก จะลบระเบียนที่จับคู่ได้ในตารางลูก แล้วกำหนดสดมภ์ใหม่ของคีย์อ้างอิงในตารางลูกให้เป็นค่าดีฟอลต์ (Default) และจะใช้ได้เมื่อสดมภ์คีย์อ้างอิงในตารางลูกมีการระบุเป็นค่าดีฟอลต์

3.) เงื่อนไขบังคับของโดเมน (Domain Constraints) เป็นการกำหนดเงื่อนไขหรือกฎเกณฑ์ของค่าที่จะปรากฏแต่ละคุณลักษณะ (Attributes) ในความสัมพันธ์ (Relation) ซึ่งประกอบด้วย

- ชนิดของข้อมูล (Data Type) ได้แก่ ตัวอักขระ (Character) จำนวนเต็ม (Integer) จำนวนทศนิยม (Double) ค่าตรรกะ (Boolean) ค่าวันเดือนปี (Date) ค่าเวลา (Time) และค่าตราเวลา (Timestamp)

- รูปแบบของข้อมูล (Format) เป็นการกำหนดรูปแบบของข้อมูลให้แต่ละเขตของข้อมูล ตัวอย่างเช่น เขตข้อมูลวันที่ผลิตกำหนดให้มีรูปแบบของวันที่เป็น DD-MM-YYYY เป็นต้น

- ขอบเขตค่าของข้อมูล (Range) เป็นการกำหนดขอบเขตของข้อมูลให้แต่ละเขตของข้อมูล ตัวอย่างเช่น เขตข้อมูลระดับผลการเรียนมีขอบเขตค่าของข้อมูลอยู่ระหว่าง 0.00 – 4.00 เป็นต้น

2.2 งานวิจัยและเครื่องมือที่เกี่ยวข้อง

2.2.1 วิทยานิพนธ์เรื่อง “การพัฒนาเครื่องมือซอฟต์แวร์สำหรับสร้างข้อมูลทดสอบ (Development of a Software Tool for Generating Test Data)” [7]

เครื่องมือสร้างข้อมูลทดสอบชิ้นนี้ถูกพัฒนาขึ้นโดยนายภพพงศ์ สกุลพิพัฒน์ศิลป์ เครื่องมือสามารถช่วยผู้ใช้สร้างข้อมูลทดสอบ โดยผู้ใช้ต้องสร้างเขตข้อมูลที่ต้องการสร้างข้อมูลแล้วกำหนดคุณสมบัติต่างๆ ให้เขตข้อมูลนั้นเอง ไม่สามารถอ่านโครงสร้างเหล่านี้ขึ้นมาจากฐานข้อมูลได้ เครื่องมือนี้สามารถกำหนดการขึ้นต่อกันเชิงตรรกะให้กับแต่ละเขตข้อมูลที่ต้องการได้ ซึ่งเครื่องมือจะเรียกว่า เงื่อนไขของเขตข้อมูลที่ขึ้นแก่กัน (Condition of the Dependent Fields) โดยจะให้ผู้ใช้กำหนดว่าต้องการสร้างความสัมพันธ์กี่ความสัมพันธ์ แล้วจึงให้ผู้ใช้กำหนดความสัมพันธ์ระหว่างแต่ละเขตข้อมูลที่ต้องการ ข้อมูลทดสอบที่ได้จากเครื่องมือนี้จะมีความใกล้เคียงกับข้อมูลจากการใช้งานจริงในระดับหนึ่ง แต่ยังไม่สมบูรณ์นัก เนื่องจากยังไม่สามารถกำหนดความคงสภาพในการอ้างอิงระหว่างตารางข้อมูล(Referential Integrity)ได้ ทำให้ข้อมูลทดสอบในแต่ละตารางข้อมูลไม่สอดคล้องตามกฎความคงสภาพในการอ้างอิง เครื่องมือนี้สามารถสร้างข้อมูลทดสอบได้ครั้งละ 1 ตารางและสามารถบันทึกข้อมูลทดสอบได้ 2 รูปแบบคือ บันทึกข้อมูลเป็นแฟ้มข้อความที่มีตัวคั่น และบันทึกข้อมูลลงในฐานข้อมูลไมโครซอฟท์แอ็คเซส (Microsoft Access)

2.2.2 **วิทยานิพนธ์เรื่อง "การพัฒนาเครื่องมือสร้างข้อมูลทดสอบเพื่อสนับสนุนการทดสอบซอฟต์แวร์จากสคีมาของฐานข้อมูล** (DEVELOPMENT OF A TEST DATA GENERATING TOOL TO SUPPORT SOFTWARE TESTING FROM DATABASE SCHEMA)" [8]

เครื่องมือสร้างข้อมูลทดสอบชิ้นนี้ถูกพัฒนาขึ้นโดยนายกฤษณะ พิริยะกิจไพบูลย์ วิทยานิพนธ์นำเสนอเครื่องมือที่ช่วยผู้ทดสอบสร้างข้อมูลทดสอบได้ง่ายขึ้น โดยเครื่องมือจะสุ่มสร้างข้อมูลทดสอบตามเงื่อนไขที่ได้มาจากผู้ทดสอบและฐานข้อมูลต้นทาง เงื่อนไขจะประกอบด้วยสคีมาของฐานข้อมูล การขึ้นต่อกันเชิงตรรกะระหว่างเขตข้อมูลในแต่ละตารางข้อมูล ความคงสภาพในการอ้างอิงของฐานข้อมูล ปริมาณระเบียนข้อมูลที่ต้องการสร้าง วิธีการสร้างข้อมูลทดสอบ และรูปแบบการบันทึกข้อมูลทดสอบ เป้าหมายของเครื่องมือที่พัฒนาขึ้นนี้คือการสร้างข้อมูลทดสอบที่มีความใกล้เคียงกับข้อมูลที่ได้จากการใช้งานจริงของซอฟต์แวร์ที่กำลังถูกทดสอบ ซึ่งจากการทดสอบการใช้งานเครื่องมือกับระบบงานจริงพบว่าเครื่องมือสามารถสร้างข้อมูลทดสอบได้ตรงกับคุณสมบัติของข้อมูลที่มีอยู่ โดยข้อมูลทดสอบที่สร้างได้จะมีความถูกต้องตามการขึ้นต่อกันเชิงตรรกะและความคงสภาพในการอ้างอิงของสคีมาของฐานข้อมูลด้วย จากแนวคิดการกำหนดความคงสภาพในการอ้างอิงระหว่างตารางข้อมูลของวิทยานิพนธ์นี้ สามารถนำใช้ในการสร้างกรณีทดสอบ

2.2.3 DeZign for databases to create ER Diagrams [9]

เครื่องมือนี้สามารถจะจัดการในเรื่องของการออกแบบฐานข้อมูล โดยเครื่องมือสามารถจะทำการสร้างและสามารถทำการสร้างสคริปต์ของสคีมาของฐานข้อมูล โดยเฉพาะโปรแกรมเมอร์ฐานข้อมูลสามารถนำไปใช้ในการพิจารณาเพื่อสร้างฐานข้อมูลต่อไปได้ เครื่องมือมีความสามารถในการสร้างสคีมาในรูปแบบของสคริปต์ในรูปแบบของหลายฐานข้อมูล ได้แก่ ออราเคิล ดีบีทู เป็นต้น นอกจากนี้เครื่องมือยังให้ผู้ใช้เป็นผู้กำหนดเงื่อนไขข้อบังคับข้อมูล ได้แก่ กำหนดคีย์หลัก คีย์อ้างอิง และการกำหนดเงื่อนไขของโดเมน แต่เงื่อนไขข้อบังคับที่กำหนดยังไม่ครอบคลุมทุกๆ เงื่อนไขข้อบังคับบูรณภาพของฐานข้อมูลทำให้ฐานข้อมูลที่ทำการออกแบบอาจไม่เป็นไปตามเงื่อนไขที่ต้องการจริง

2.2.4 The Altova DatabaseSpy 2008 [10]

เครื่องมือสามารถช่วยให้ผู้ใช้ทำการออกแบบฐานข้อมูล โดยเครื่องมือสามารถเพิ่มความสะดวกในการใช้งานของผู้ใช้ในการออกแบบฐานข้อมูล โดยผู้ใช้สามารถสร้างตาราง

และความสัมพันธ์ของตารางได้ และสามารถทำการเพิ่ม แก้ไข หรือปรับปรุงตารางและความสัมพันธ์ได้สะดวกเพียงการลากวาง นอกจากนี้การใช้งานเครื่องมือสามารถแสดงมุมมองโครงสร้างของฐานข้อมูลที่กำลังออกแบบได้หลายมุมมอง เครื่องมือยังสามารถทำการกำหนดเงื่อนไขข้อบังคับของฐานข้อมูลได้แก่ การกำหนดคีย์หลัก การกำหนดคีย์อ้างอิง การกำหนดเงื่อนไขในการตรวจสอบข้อมูล แต่เครื่องมือยังไม่สามารถให้ผู้ใช้ทำการกำหนดเงื่อนไขข้อบังคับการอ้างอิงถึงในส่วนของตัวเลือก (Option) เช่น การลบในลักษณะ Cascade เป็นต้น ยังไม่มีระบุในเครื่องมือนี้ ทำให้การออกแบบในบางกรณีอาจจะก่อให้เกิดข้อผิดพลาดตามมาได้

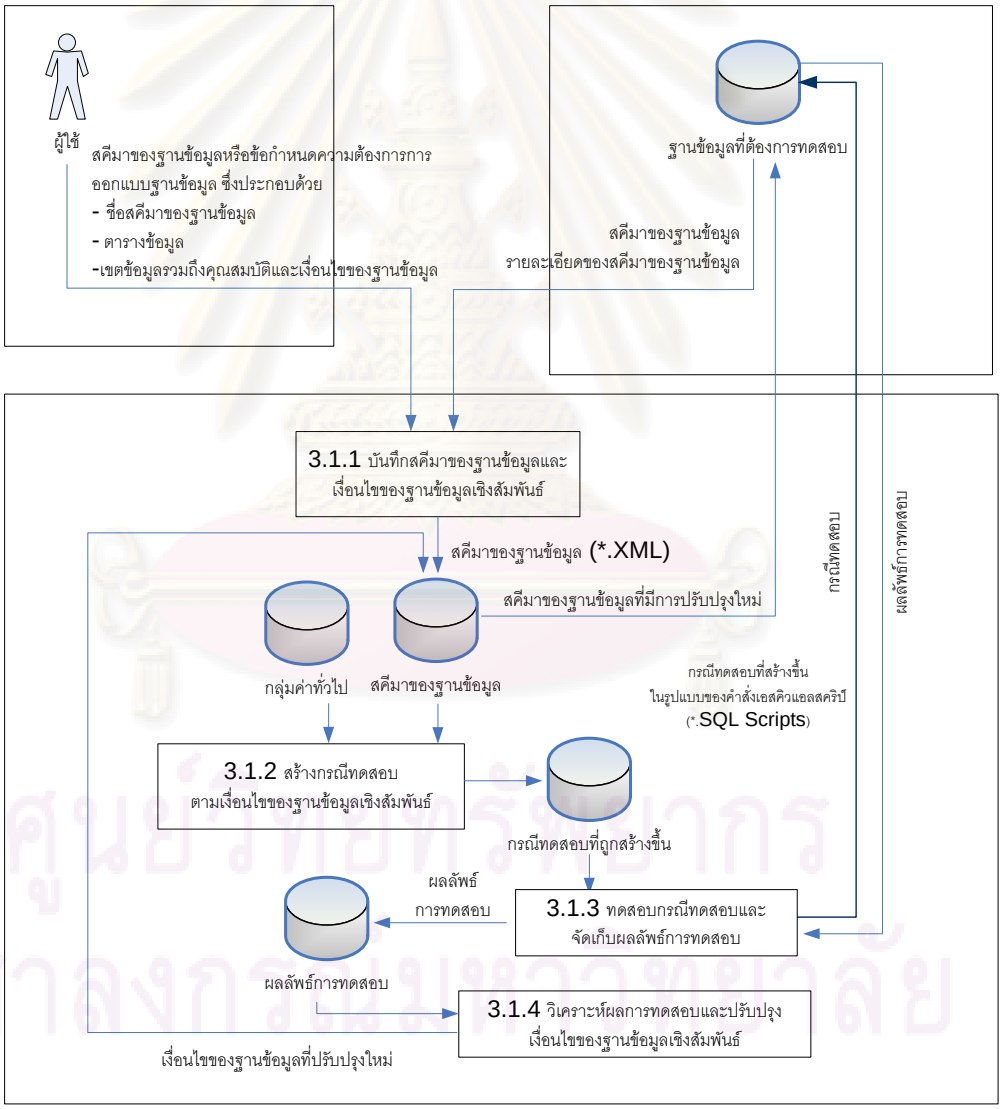
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การวิเคราะห์และออกแบบเครื่องมือ

3.1 โครงสร้างการทำงานของเครื่องมือ

วิทยานิพนธ์นี้ได้พัฒนาเครื่องมือสร้างกรณีทดสอบ เพื่อทำการทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ตามที่ผู้ออกแบบฐานข้อมูลได้กำหนดขึ้น ทั้งนี้กรณีทดสอบที่สร้างขึ้นอยู่ในรูปแบบของคำสั่งการเพิ่ม ลบ และปรับปรุงข้อมูล จากนี้จะเป็นการอธิบายโครงสร้างการทำงานของเครื่องมือ โดยทำการแบ่งออกเป็น 4 ส่วนการทำงาน ดังรูปที่ 2.3.1



รูปที่ 3.1 โครงสร้างการทำงานของเครื่องมือ

3.1.1 บันทึกสคีมาของฐานข้อมูลและเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์

เครื่องมือทดสอบจะรับสคีมาของฐานข้อมูล หรือข้อกำหนดการออกแบบฐานข้อมูล (Database Design Specifications) โดยเครื่องมือสามารถทำการเลือกรับสคีมาของฐานข้อมูลได้ 2 ลักษณะ คือ

(1) รับสคีมาของฐานข้อมูลจากผู้ใช้งานทั้งหมด ส่วนนี้ผู้ใช้งานจะเป็นผู้กำหนดข้อมูลเกี่ยวกับสคีมาของฐานข้อมูลทั้งหมด ดังนี้

1.1 ชื่อสคีมาของฐานข้อมูล

1.2 ตารางข้อมูลในสคีมาของฐานข้อมูล

1.3 เขตข้อมูลในแต่ละตารางข้อมูล และกำหนดคุณสมบัติของเขตข้อมูลและเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ ซึ่งคุณสมบัติของเขตข้อมูลและเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์จะประกอบด้วย 3 เงื่อนไข ดังนี้

- เงื่อนไขบังคับบูรณภาพของเอนทิตี เป็นการกำหนดเงื่อนไขเกี่ยวกับสถานะของคีย์หลักโดยแต่ละระเบียน ต้องมีการกำหนดเป็นค่าที่ไม่ซ้ำกันหรือค่าต้องไม่เป็นค่าว่าง

- เงื่อนไขบังคับบูรณภาพของการอ้างอิงถึง เป็นการกำหนดเงื่อนไขหรือกฎเกณฑ์การอ้างอิงถึงคุณลักษณะจากความสัมพันธ์หนึ่งในความสัมพันธ์ใดโดยจะต้องเป็นการอ้างอิงถึงคุณลักษณะที่มีอยู่จริงในความสัมพันธ์นั้น การอ้างอิงถึงคุณลักษณะที่ไม่มีอยู่จริงจะทำให้ไม่สามารถรักษาความถูกต้องและความสมบูรณ์ของข้อมูลไว้ได้ ซึ่งประกอบด้วย

1.) กำหนดคีย์อ้างอิง

2.) กำหนดคีย์เวิร์ดหรือตัวเลือกการอ้างอิงถึง ซึ่งแบ่งออกเป็น 4 กรณี ดังนี้

- กรณี On Delete Cascade กรณีนี้เมื่อทำการลบระเบียนจากตารางหลัก จะทำการลบระเบียนในตารางลูกที่มีค่าอ้างอิงด้วย โดยที่คีย์อ้างอิงในตารางลูกต้องจับคู่กับค่าคีย์หลักในตารางหลักจึงจะลบระเบียนโดยอัตโนมัติ

- กรณี On Delete Set Null กรณีเมื่อทำการลบระเบียนในตารางหลัก จะลบระเบียนในตารางลูกโดยอัตโนมัติก็ต่อเมื่อคีย์อ้างอิงจับคู่กับค่าคีย์หลักได้ และเมื่อลบแล้วจะทำการกำหนดค่าในสดมภ์ที่มีคีย์อ้างอิงเป็นค่าว่าง

- กรณี On Delete Default เมื่อทำการลบระเบียนในตารางหลัก จะลบระเบียนในตารางลูกโดยอัตโนมัติ ก็ต่อเมื่อคีย์อ้างอิงจับคู่กับคีย์หลักได้ และเมื่อลบแล้วจะกำหนดค่าในสดมภ์ที่มีค่าอ้างอิงเป็นค่าดีฟอลต์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- กรณี No Action ซึ่งจะเป็นค่าดีฟอลต์เมื่อไม่ได้ทำการระบุในตัวเลือกนี้ ดังนั้นเมื่อทำการลบระเบียนจึงจะต้องทำการลบระเบียนในตารางลูกก่อนที่จะทำการลบในตารางหลัก

- เงื่อนไขข้อบังคับบูรณภาพของโดเมน เป็นการกำหนดเงื่อนไขหรือกฎเกณฑ์ของค่าที่จะปรากฏแต่ละคุณลักษณะ ในความสัมพันธ์ซึ่งประกอบด้วย

1.) ชนิดของข้อมูล ได้แก่ ตัวอักขระ จำนวนเต็ม จำนวนทศนิยม ค่าตรรกะ ค่าวันเดือนปี ค่าเวลา และค่าตราเวลา (Timestamp)

2.) ขนาดของเขตข้อมูล (Size) หรือความเที่ยงตรงของจำนวน (Numeric Precision) ขึ้นอยู่กับชนิดของข้อมูล ถ้าเป็นตัวอักขระจะกำหนดเป็นขนาดของเขตข้อมูล แต่ถ้าเป็นจำนวนเต็มหรือจำนวนทศนิยม จะกำหนดเป็นความเที่ยงตรงของจำนวน

3.) ขนาดของทศนิยม (Numeric Scale) คือ จำนวนหลักของทศนิยม สามารถทำการกำหนด ได้เฉพาะเขตข้อมูลที่มีชนิดเป็นทศนิยมเท่านั้น

4.) รูปแบบของข้อมูล (Format) เป็นการกำหนดรูปแบบของข้อมูลให้แต่ละเขตของข้อมูล ตัวอย่างเช่น เขตข้อมูลวันที่ผลิตกำหนดให้มีรูปแบบของวันที่เป็น DD-MM-YYYY เป็นต้น

5.) ขอบเขตค่าของข้อมูล (Range) เป็นการกำหนดขอบเขตของข้อมูลให้แต่ละเขตของข้อมูล ตัวอย่างเช่น เขตข้อมูลระดับผลการเรียนหรือ GPA มีขอบเขตค่าของข้อมูลอยู่ระหว่าง 0.00 – 4.00 เป็นต้น

(2) รับสคีมาของฐานข้อมูลจากฐานข้อมูลที่ต้องการทดสอบ ส่วนนี้เครื่องมือจะทำการอ่านสคีมาของฐานข้อมูลผ่านทางการเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบที่กำหนดโดยผู้ใช้มาเก็บไว้ โดยเครื่องมือสามารถอ่านสคีมาของฐานข้อมูลจากฐานข้อมูล 3 ชนิด คือ ฐานข้อมูลมายเอสคิวแอล (MySQL database) ฐานข้อมูลออราเคิล (ORACLE Database) และฐานข้อมูลไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ 2000 (Microsoft SQL Server 2000) สำหรับสคีมาของฐานข้อมูลที่ได้จากฐานข้อมูลที่ต้องทดสอบ ผู้ใช้สามารถทำการกำหนดข้อมูลรายละเอียดของฐานข้อมูลเชิงสัมพันธ์เพิ่มเติมได้

จากนั้นข้อมูลเกี่ยวกับสคีมาของฐานข้อมูลทั้งหมดจะถูกบันทึกไว้ในฐานข้อมูลของเครื่องมือทดสอบ เมื่อผู้ใช้ต้องการสร้างกรณีทดสอบ ผู้ใช้สามารถเลือกข้อมูลของสคีมา(ในรูปเอกสารเอ็กซ์เอ็มแอล) มาใช้งานได้

3.1.2 **สร้างกรณีทดสอบตามเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์** เครื่องมือจะทำการสร้างกรณีทดสอบตามสคีมาของฐานข้อมูลที่จัดเก็บไว้ในฐานข้อมูลของเครื่องมือ และนำข้อมูลเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์มาใช้ประกอบการสร้างกรณีทดสอบด้วย กรณีทดสอบที่สร้างขึ้นนั้นจะต้องสามารถทำการทดสอบได้ว่าเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ที่ผู้ใช้กำหนดไว้ในขั้นตอนของการออกแบบเป็นไปตามเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ที่ระบุไว้หรือไม่

นอกจากข้อมูลต่างๆ ที่เครื่องมือทดสอบรับเข้ามาแล้ว เครื่องมือทดสอบจะอาศัยข้อมูลของกลุ่มค่าทั่วไป (Generic Value) คือ การกำหนดให้ใช้กลุ่มค่าทั่วไป โดยเครื่องมือได้จัดเตรียมไว้ให้แล้ว เช่น ชื่อ นามสกุล เป็นต้น สำหรับใช้สร้างข้อมูลทดสอบให้กับเขตข้อมูล เพื่อประกอบกับกรณีทดสอบที่จะสร้างขึ้น ข้อมูลทดสอบของเครื่องมือสามารถสร้างได้ 2 ลักษณะ คือ การสุ่มค่าทั่วไปขึ้นมาจากกลุ่มค่าทั่วไปที่เลือกไว้ และสร้างข้อมูลโดยเครื่องมือสุ่มให้อัตโนมัติ

ทั้งนี้กรณีทดสอบที่สร้างขึ้นนั้นจะครอบคลุมทุกเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ที่กำหนดไว้ รายละเอียดของเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์แบ่งออกเป็น 3 ส่วน ดังนี้

1.) เงื่อนไขข้อบังคับบูรณภาพของเอนทิตี เครื่องมือจะสร้างกรณีทดสอบเพื่อตรวจสอบเงื่อนไขข้อบังคับของคีย์หลัก ตามกฎข้อบังคับของคีย์หลักค่าในสดมภ์จะต้องไม่เป็นค่าซ้ำและค่าในสดมภ์จะต้องมีค่าอยู่และไม่เป็นค่าไม่ว่าง โดยจะอาศัยคำสั่งการเพิ่มข้อมูลประกอบการพิจารณา

2.) เงื่อนไขข้อบังคับบูรณภาพของการอ้างอิงถึง เครื่องมือจะทำการสร้างกรณีทดสอบเพื่อตรวจสอบเงื่อนไขข้อบังคับของการอ้างอิงถึง การทดสอบเงื่อนไขข้อบังคับบูรณภาพของการอ้างอิงถึงนี้จะอาศัยคำสั่งการลบข้อมูลประกอบการพิจารณาดังนี้

ก. กรณีการลบระเบียนจากตารางหลัก กรณีทดสอบที่สร้างขึ้นนั้นจะต้องสามารถตรวจสอบว่าระเบียนที่จับคู่ได้ในตารางลูกสามารถลบได้โดยอัตโนมัติหรือไม่ เนื่องจากการลบระเบียนทั้งหมดจะมีคีย์หลักที่ใช้อ้างอิงคีย์อ้างอิงในตารางอื่นๆ มีผลทำให้ตารางอื่นๆ ที่มีระเบียนจับคู่ได้จะถูกลบออกได้โดยอัตโนมัติในลักษณะตกเป็นทอดๆ (Cascade)

ข. กรณีการลบระเบียนจากตารางหลัก กรณีทดสอบที่สร้างขึ้นนั้นจะต้องสามารถตรวจสอบว่าระเบียนที่จับคู่ได้ในตารางลูกสามารถลบได้หรือไม่ นอกจากนี้ยังทำการตรวจสอบว่ามีการกำหนดสดมภ์ (Column) คีย์อ้างอิงในตารางลูกให้เป็นค่าว่างหรือไม่ (จะใช้กรณีนี้เมื่อสดมภ์คีย์อ้างอิงในตารางลูกจะต้องระบุไม่เป็นค่าว่าง)

ค. กรณีเมื่อทำการลบระเบียนจากตารางหลัก กรณีทดสอบที่สร้างขึ้นนั้นจะต้องสามารถตรวจสอบว่าระเบียนที่จับคู่ได้ในตารางลูกสามารถลบได้หรือไม่ พร้อมทั้งมีการ

ตรวจสอบการกำหนดสดมภ์ใหม่ของคีย์อ้างอิงในตารางลูกให้เป็นค่าดีฟอลต์หรือไม่ (จะใช้กรณีนี้เมื่อสดมภ์คีย์อ้างอิงในตารางลูกมีการระบุเป็นค่าดีฟอลต์)

ง. กรณีเมื่อทำการลบระเบียนแล้วมีการกำหนดเป็นค่า No Action ซึ่งเป็นค่าดีฟลอต์ เมื่อไม่ได้ทำการระบุในตัวเลือกนี้ ดังนั้นกรณีทดสอบจะทำการตรวจสอบว่ามีการลบระเบียนในตารางลูกก่อนที่จะทำการลบในตารางหลักหรือไม่

3.) เงื่อนไขข้อบังคับของโดเมน (Domain Constraints) ส่วนนี้เครื่องมือจะทำสร้างกรณีทดสอบเพื่อตรวจสอบเงื่อนไขข้อบังคับของแต่ละคุณลักษณะ ทั้งนี้อาศัยคำสั่งการเพิ่มและการปรับปรุงข้อมูลประกอบการพิจารณาสร้างกรณีทดสอบ

3.1.3 ทดสอบฐานข้อมูลและแสดงผลจากการทดสอบฐานข้อมูล เครื่องมือจะทำการทดสอบฐานข้อมูลโดยการนำกรณีทดสอบที่สร้างขึ้นในรูปแบบเอกสารเอ็กซ์เอ็มแอลตามเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ ในรูปของคำสั่งการเพิ่มข้อมูล การลบข้อมูล และการปรับปรุงข้อมูลไปทำการประมวลผลที่ฐานข้อมูลที่ต้องการทดสอบ จากนั้นเมื่อทำการทดสอบฐานข้อมูลเสร็จแล้วเครื่องมือจะแสดงผลลัพธ์ของการทดสอบที่ได้จากการทดสอบฐานข้อมูลตามกรณีทดสอบที่สร้างขึ้น โดยจะแสดงผลลัพธ์การทดสอบในรูปของเอกสารเอ็กซ์เอ็มแอลและบันทึกผลการทดสอบลงฐานข้อมูลของเครื่องมือ โดยข้อมูลที่ทำการบันทึกลงฐานข้อมูลประกอบด้วย กรณีทดสอบที่สร้างขึ้น ผลของการทดสอบ และครั้งที่ทำการทดสอบ

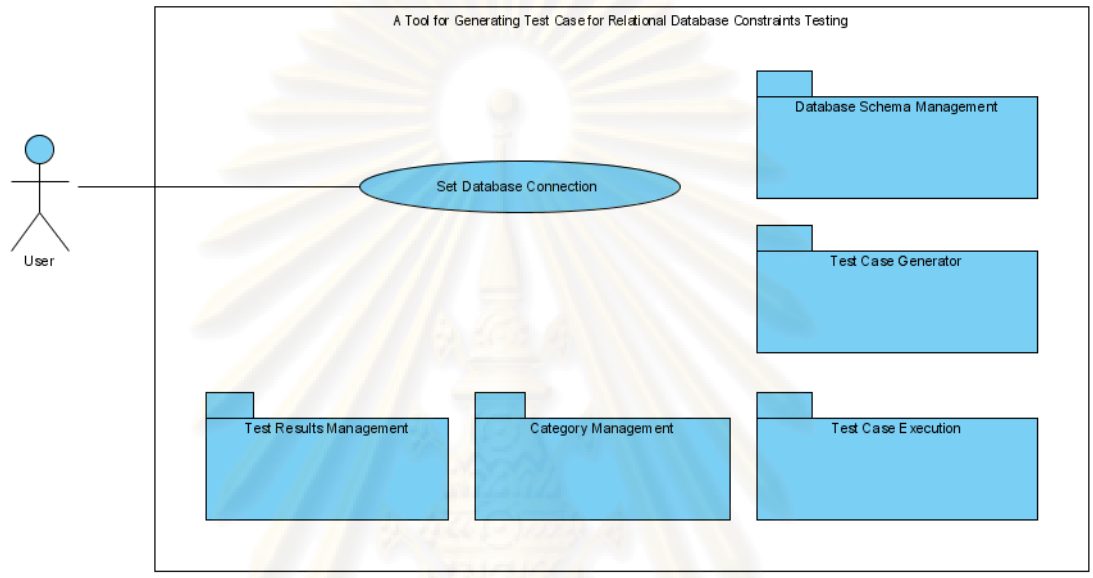
3.1.4 การวิเคราะห์ผลการทดสอบและปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ เมื่อทำการทดสอบฐานข้อมูลแล้วพบว่ามีข้อผิดพลาดเกิดขึ้น เครื่องมือจะนำผลลัพธ์ของการทดสอบมาทำการตรวจสอบถึงข้อผิดพลาดที่เกิดขึ้นและทำการปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ โดยเครื่องมือทดสอบจะทำการปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ให้เป็นตามที่ได้มีการกำหนดไว้ในขั้นตอนของการออกแบบโดยอัตโนมัติ และจัดเก็บข้อมูลเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ที่ทำการปรับปรุงลงฐานข้อมูล(ส่วนของสคีมาของฐานข้อมูล)ของเครื่องมือ จากนั้นเครื่องมือจะทำการปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ที่ฐานข้อมูลที่ทำการทดสอบ

3.2 การวิเคราะห์และออกแบบเครื่องมือ

สำหรับการวิเคราะห์และออกแบบเครื่องมือนั้นอธิบายด้วยแผนภาพต่างๆ ในภาษายูเอ็มแอล (UML: Unified Modeling Language) ซึ่งเป็นสัญลักษณ์มาตรฐานที่ใช้ในการวิเคราะห์และออกแบบเชิงวัตถุ (Object Oriented Analysis and Design) โดยแผนภาพที่เลือกใช้ได้แก่ แผนภาพยูสเคส (Use case diagram) แผนภาพคลาส (Class diagram) และแผนภาพซีเควนซ์ (Sequence diagram)

3.2.1 แผนภาพยูสเคส

แผนภาพยูสเคสแสดงว่าผู้ใช้เครื่องมือทดสอบ สามารถทำงานอะไรได้บ้าง ในที่นี้ผู้ใช้หมายถึงผู้ใช้ และระบบจะหมายถึงเครื่องมือที่ทำการพัฒนาขึ้น โดยแผนภาพยูสเคสโดยรวมของเครื่องมือเป็นดังนี้



รูปที่ 3.2 แผนภาพยูสเคสโดยรวม

จากรูปที่ 3.2 แผนภาพยูสเคสโดยรวมจะประกอบด้วยแพ็คเก็จ (Package) 5 แพ็คเกจคือ แพ็คเกจการรับสคีมาของฐานข้อมูล (Database Schema Management) แพ็คเกจการสร้างกรณีทดสอบ (Test Case Generator) แพ็คเกจการจัดการกลุ่มค่าทั่วไป (Category Management) แพ็คเกจทดสอบกรณีทดสอบ (Test Case Execution) และแพ็คเกจการจัดการผลลัพธ์การทดสอบ (Test Result Management) นอกจากนั้นในแผนภาพยังประกอบด้วยแผนภาพยูสเคสอีก 1 ยูสเคส คือ ยูสเคสกำหนดการเชื่อมต่อกับฐานข้อมูลที่ทดสอบโดยผู้ใช้

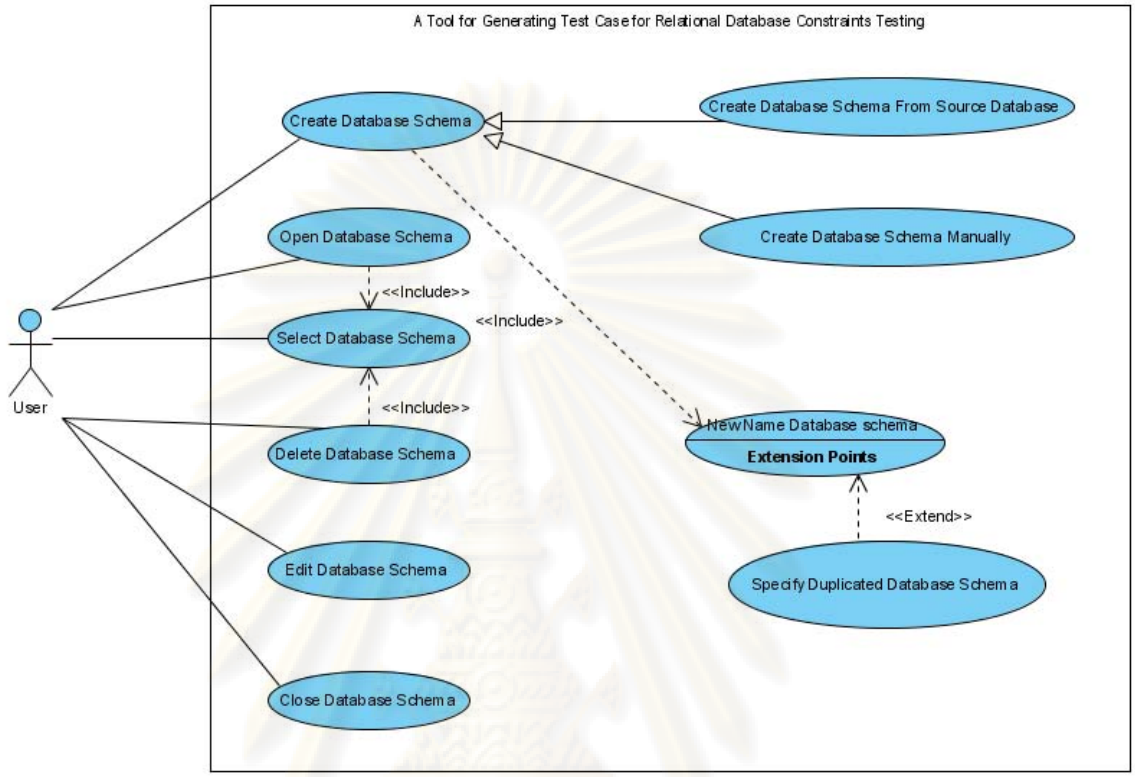
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.1 รายละเอียดของยูสเคสกำหนดการเชื่อมต่อกับฐานข้อมูลทดสอบ

Use Case:	Set Database connection
Actors:	User
Goal:	เพื่อกำหนดให้เครื่องมือเชื่อมต่อกับฐานข้อมูลที่ทดสอบ จะบันทึกกรณีทดสอบหลังจากที่สร้างกรณีทดสอบเสร็จแล้ว
Related use cases:	-
Preconditions:	ฐานข้อมูลทดสอบเชื่อมต่อนั้นต้องเปิดให้บริการอยู่
Steps:	1.ผู้ใช้เลือกกำหนดรายละเอียดการเชื่อมต่อกับฐานข้อมูลทดสอบ 2.เครื่องมือแสดงหน้าจอรับข้อมูลรายละเอียดการเชื่อมต่อกับฐานข้อมูล 3.ผู้ใช้เลือกชนิดของฐานข้อมูลที่ต้องการเชื่อมต่อ 4.ผู้ใช้กำหนดพารามิเตอร์ที่ใช้ในการเชื่อมต่อ 5.ผู้ใช้ยืนยันข้อมูลที่ใช้ในการเชื่อมต่อ 6.เครื่องมือตอบรับการเชื่อมต่อกับฐานข้อมูลทดสอบ
Postconditions:	เครื่องมือทำการเชื่อมต่อกับฐานข้อมูลทดสอบ

สำหรับแพ็คเกจทั้ง 4 แพ็คเกจที่ปรากฏในแผนภาพยูสเคสนั้นเป็นเพ็คเกจที่รวมเอายูสเคสที่มีความเกี่ยวข้องในกลุ่มงานเดียวกันไว้ด้วยกัน โดยรายละเอียดภายในแต่ละแพ็คเกจนั้นเป็นดังนี้

1.) แพ็คเกจการจัดการสคีมาของฐานข้อมูล แพ็คเกจการจัดการสคีมาของฐานข้อมูลแสดงดังรูปที่ 3.3 ประกอบด้วยยูสเคส ทั้งหมด 10 ยูสเคส ซึ่งแต่ละยูสเคสจะเกี่ยวข้องกับการจัดการสคีมาของฐานข้อมูลทั้งหมด โดยรายละเอียดของแต่ละยูสเคสเป็นดังตารางที่ 3.2 ถึงตารางที่ 3.11



รูปที่ 3.3 ยูสเคสภายในแพ็คเกจการจัดการสคีมาของฐานข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.2 รายละเอียดของยูสเคสกำหนดชื่อสคีมาของฐานข้อมูล

Use case:	Name new database schema
Actors:	User
Goal:	เพื่อกำหนดชื่อและคำอธิบายให้กับสคีมาของฐานข้อมูลใหม่ที่จะสร้าง
Related use cases:	-
Preconditions:	ผู้ใช้เลือกสร้างสคีมาของฐานข้อมูล
Steps:	1.เครื่องมือแสดงหน้าต่างเพื่อรับข้อมูลชื่อและคำอธิบายสคีมาของฐานข้อมูล 2.ผู้ใช้กรอกชื่อและคำอธิบายของสคีมาของฐานข้อมูลที่ต้องการสร้าง 3.ผู้ใช้ยืนยันการกำหนดชื่อและคำอธิบายของสคีมาของฐานข้อมูล 4. เครื่องมือตอบรับการกำหนดชื่อและคำอธิบายของสคีมาของฐานข้อมูล
Postconditions:	-

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.3 รายละเอียดของยูสเคสสคีมาของฐานข้อมูลซ้ำซ้อน

Use case:	Specify duplicated database schema
Actors:	User
Goal:	เพื่อไม่ให้มีการสร้างสคีมาของฐานข้อมูลที่ใช้ชื่อซ้ำซ้อนกัน
Related use cases:	Extensions of: Name new database schema
Preconditions:	ผู้ใช้ยืนยันการกำหนดชื่อและคำอธิบายของสคีมาของฐานข้อมูล
Steps:	เครื่องมือแจ้งว่าชื่อของสคีมาของฐานข้อมูลที่ผู้ใต้องการสร้างนั้นซ้ำซ้อนกับชื่อของสคีมาของฐานข้อมูลที่มีอยู่ในฐานข้อมูลของเครื่องมือ
Postconditions:	-

ตารางที่ 3.4 รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูล

Use case:	Create database schema
Actors:	User
Goal:	เพื่อสร้างสคีมาของฐานข้อมูลใหม่เก็บไว้ในฐานข้อมูลของเครื่องมือสำหรับใช้ในการสร้างข้อมูลทดสอบต่อไป
Related Use cases:	Generalization of: ■ Create database schema manually ■ Create database schema from source database Includes: Name new database schema
Preconditions:	-

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.4 รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูล (ต่อ)

Steps:	1.ผู้ใช้เลือกสร้างสคีมาของฐานข้อมูล 2.ผู้ใช้กำหนดชื่อ และคำอธิบายของสคีมาของฐานข้อมูล (Use case :Name new database schema) 3.ผู้ใช้กำหนดรายละเอียดของสคีมาของฐานข้อมูล 4.เครื่องมือตอบรับการสร้างสคีมาของฐานข้อมูล
Postconditons:	เครื่องมือสร้างสคีมาของฐานข้อมูลใหม่ตามรายละเอียดที่ผู้ใช้กำหนดลงในฐานข้อมูลของเครื่องมือ

ตารางที่ 3.5 รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูลโดยผู้ใช้

Use case:	Create database schema manually
Actors:	User
Goal:	เพื่อสร้างสคีมาของฐานข้อมูลใหม่เก็บไว้ในฐานข้อมูลของเครื่องมือโดยผู้ใช้เป็นผู้กรอกรายละเอียดของสคีมาของฐานข้อมูลเองทั้งหมด
Related use cases:	Specialization of: Create database schema
Preconditions:	-
Steps:	1. ผู้ใช้เลือกสร้างสคีมาของฐานข้อมูล 2. ผู้ใช้กำหนดชื่อและคำอธิบายของสคีมาของฐานข้อมูล 3. เครื่องมือแสดงหน้าต่างเพื่อรับรายละเอียดของสคีมาของฐานข้อมูล 4. ผู้ใช้กรอกชื่อตารางข้อมูลที่มีในสคีมาของฐานข้อมูล 5. ผู้ใช้กำหนดเขตข้อมูลของแต่ละตารางในสคีมาของฐานข้อมูล 6. ผู้ใช้กำหนดความคงสภาพในการอ้างอิงระหว่างตารางที่ต้องการ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.5 รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูลโดยผู้ใช้ (ต่อ)

Steps:	7. ผู้ใช้กำหนดการขึ้นต่อกันเชิงตรรกะระหว่างเขตข้อมูลที่ต้องการภายในแต่ละตาราง 8. ผู้ใช้ยืนยันการบันทึกข้อมูลรายละเอียดของสคีมาของฐานข้อมูล 9. เครื่องมือตอบรับการสร้างสคีมาของฐานข้อมูล
Postconditions:	เครื่องมือสร้างสคีมาของฐานข้อมูลใหม่ตามรายละเอียดที่ได้จากฐานข้อมูลต้นทาง ซึ่งจะมีเพียงรายการตารางข้อมูล รายการเขตข้อมูลในแต่ละตาราง และความคงสภาพในการอ้างอิงระหว่างตาราง

ตารางที่ 3.6 รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่ต้องการทดสอบ

Use case:	Create database schema from source database
Actors:	User
Goal:	เพื่อสร้างสคีมาของฐานข้อมูลใหม่เก็บไว้ในฐานข้อมูลของเครื่องมือโดยอ่านมาจากฐานข้อมูลต้นทางผ่านทางการเชื่อมต่อ
Related use cases:	Specialization of: Create database schema
Preconditions:	ฐานข้อมูลต้นทางที่ต้องการเชื่อมต่อนั้นต้องเปิดให้บริการอยู่
Steps:	1. ผู้ใช้เลือกสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลต้นทาง 2. ผู้ใช้กำหนดชื่อและคำอธิบายของสคีมาของฐานข้อมูล 3. เครื่องมือแสดงหน้าต่างรับข้อมูลรายละเอียดการเชื่อมต่อกับฐานข้อมูลต้นทาง 4. ผู้ใช้เลือกชนิดของฐานข้อมูลที่ต้องการเชื่อมต่อ 5. ผู้ใช้กำหนดพารามิเตอร์ (Parameters) ที่ใช้ในการเชื่อมต่อ 6. ผู้ใช้ยืนยันข้อมูลที่ใช้ในการเชื่อมต่อ 7. เครื่องมือเชื่อมต่อกับฐานข้อมูลต้นทาง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.6 รายละเอียดของยูสเคสสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่ต้องการทดสอบ (ต่อ)

Steps:	8. เครื่องมืออ่านรายละเอียดของสคีมาของฐานข้อมูลจากฐานข้อมูลต้นทางและแสดงผลการอ่านข้อมูล 9. เครื่องมือตอบรับการสร้างสคีมาของฐานข้อมูล
Postconditions:	เครื่องมือสร้างสคีมาของฐานข้อมูลใหม่ตามรายละเอียดที่ได้จากฐานข้อมูลต้นทาง ซึ่งจะมีเพียงรายการตารางข้อมูล รายการเขตข้อมูลในแต่ละตาราง และความคงสภาพในการอ้างอิงระหว่างตาราง

ตารางที่ 3.7 รายละเอียดของยูสเคสเปิดสคีมาของฐานข้อมูล

Use case:	Open database schema
Actors:	User
Goal:	เพื่อเปิดใช้สคีมาของฐานข้อมูลที่เคยสร้างไว้ในฐานข้อมูลของเครื่องมือ
Related use cases:	Includes: Select database schema
Preconditions:	-
Steps:	1. ผู้ใช้เลือกเปิดสคีมาของฐานข้อมูล 2. ผู้ใช้เลือกสคีมาของฐานข้อมูลที่ต้องการ (Use case: Select database schema) 3. เครื่องมือตอบรับการเปิดสคีมาของฐานข้อมูลที่ผู้ใช้เลือกมา
Postconditions:	เครื่องมือเปิดสคีมาของฐานข้อมูลที่ผู้ใช้เลือก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.8 รายละเอียดของยูสเคสลบสคีมาของฐานข้อมูล

Use cases:	Delete database schema
Actors:	User
Goal:	เพื่อลบสคีมาของฐานข้อมูลที่เคยสร้างไว้ออกจากฐานข้อมูลของเครื่องมือ
Related use cases:	Includes: Select database schema
Preconditions:	สคีมาของฐานข้อมูลที่ต้องการลบจะต้องไม่ถูกเปิดใช้อยู่
Steps:	1. ผู้ใช้เลือกลบสคีมาของฐานข้อมูล 2. ผู้ใช้เลือกสคีมาของฐานข้อมูลที่ต้องการ (Use case: Select database schema) 3. เครื่องมือตอบรับการลบสคีมาของฐานข้อมูลที่ผู้ใช้เลือกมา
Postconditions:	เครื่องมือลบสคีมาของฐานข้อมูลที่ผู้ใช้เลือกออกจากฐานข้อมูลเครื่องมือ

ตารางที่ 3.9 รายละเอียดของยูสเคสเลือกสคีมาของฐานข้อมูล

Use case:	Select database schema
Actors:	User
Goals:	เพื่อเลือกสคีมาของฐานข้อมูลที่เคยสร้างไว้ในการเปิด หรือลบสคีมาของฐานข้อมูล
Related use cases:	-
Preconditions:	ผู้ใช้เลือกเปิดสคีมาของฐานข้อมูล หรือผู้ใช้เลือกลบสคีมาของฐานข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.9 รายละเอียดของยูสเคสเลือกสคีมาของฐานข้อมูล (ต่อ)

Steps:	1. เครื่องมือแสดงหน้าต่างที่แสดงรายชื่อสคีมาของฐานข้อมูลที่มีอยู่ในฐานข้อมูลของเครื่องมือ 2. ผู้ใช้เลือกสคีมาของฐานข้อมูลที่ต้องการ 3. ผู้ใช้ยืนยันการเลือกสคีมาของฐานข้อมูล
Postconditions:	เครื่องมือแก้ไขข้อมูลของสคีมาของฐานข้อมูลตามที่ผู้ใช้กำหนด

ตารางที่ 3.10 รายละเอียดของยูสเคสแก้ไขสคีมาของฐานข้อมูล

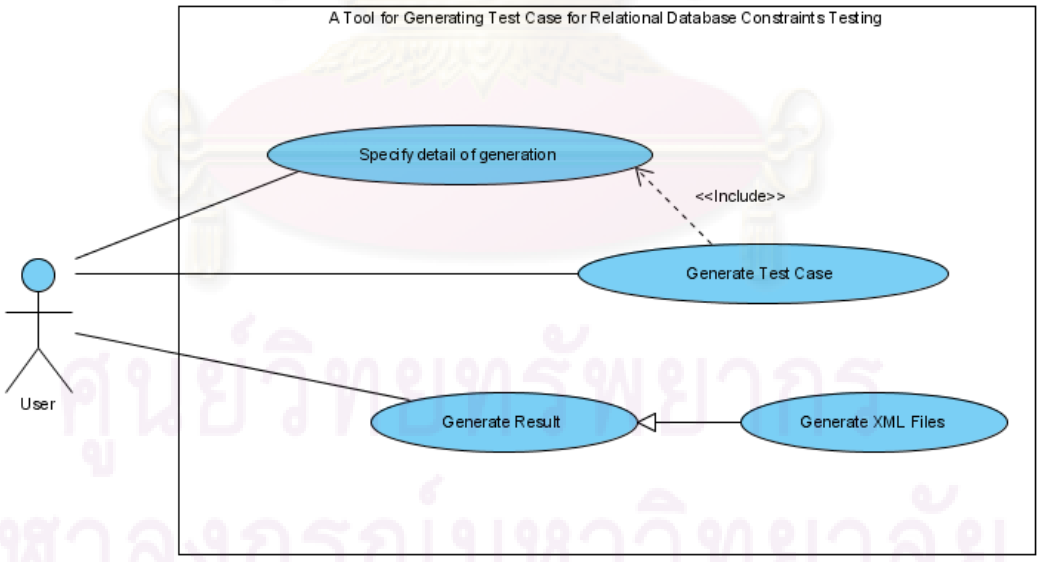
Use case:	Edit database schema
Actors:	User
Goal:	เพื่อแก้ไขสคีมาของฐานข้อมูลที่กำลังเปิดใช้อยู่
Related use cases:	-
Preconditions:	ผู้ใช้ต้องเปิดสคีมาของฐานข้อมูลที่ต้องการแก้ไขไว้แล้ว
Steps:	1. ผู้ใช้เลือกแก้ไขสคีมาของฐานข้อมูล 2. เครื่องมือแสดงหน้าต่างจัดการข้อมูลภายในสคีมาของฐานข้อมูล 3. ผู้ใช้แก้ไขข้อมูลที่ต้องการภายในสคีมาของฐานข้อมูล 4. ผู้ใช้ยืนยันการแก้ไขสคีมาของฐานข้อมูล
Postconditions:	เครื่องมือแก้ไขข้อมูลของสคีมาของฐานข้อมูลตามที่ผู้ใช้กำหนด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.11 รายละเอียดของยูสเคสปิดสคีมาของฐานข้อมูล

Use case:	Close database schema
Actors:	User
Goal:	เพื่อปิดสคีมาของฐานข้อมูลที่กำลังเปิดใช้อยู่
Related use cases:	-
Preconditions:	ผู้ใช้ต้องเปิดสคีมาของฐานข้อมูลที่จะปิดไว้แล้ว
Steps:	1.ผู้ใช้เลือกปิดสคีมาของฐานข้อมูล 2.เครื่องมือตอบรับการปิดสคีมาของฐานข้อมูล
Postconditions:	เครื่องมือปิดสคีมาของฐานข้อมูลที่กำลังเปิดไว้

2.) แพ็คเกจการสร้างกรณีทดสอบ รายละเอียดของแพ็คเกจนี้สามารถแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 ยูสเคสภายในแพ็คเกจการสร้างกรณีทดสอบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากรูปที่ 3.4 แพ็คเกจการสร้างกรณีทดสอบจะประกอบด้วยยูสเคสทั้งหมด 4 ยูสเคส ซึ่งแต่ละยูสเคสจะเกี่ยวข้องกับการสร้างกรณีทดสอบ โดยรายละเอียดของแต่ละยูสเคสจะเป็นดังตารางที่ 3.12 ถึง ตารางที่ 3.15

ตารางที่ 3.12 รายละเอียดของยูสเคสสร้างกรณีทดสอบ

Use Case:	Generate Test Case
Actors:	User
Goal:	เพื่อสร้างกรณีทดสอบตามสคีมาของฐานข้อมูลที่กำลังเปิดใช้งานอยู่
Related use cases:	Include: ■ Specify details of generation ■ Generate Result
Preconditions:	ผู้ใช้ต้องเปิดสคีมาของฐานข้อมูลที่ต้องการสร้างกรณีทดสอบไว้แล้ว
Steps:	1.ผู้ใช้เลือกสร้างกรณีทดสอบ 2.ผู้ใช้กำหนดรายละเอียดของการสร้างกรณีทดสอบ(Use Case: Speciry details of generation) 3. ผู้ใช้ยืนยันการสร้างกรณีทดสอบ 4. เครื่องมือสร้างกรณีทดสอบ 5. เครื่องมือบันทึกข้อมูลทดสอบตามรูปแบบที่ผู้ใช้กำหนด (Use Case: Generate Results) 6. เครื่องมือแสดงผลการสร้างกรณีทดสอบ
Postconditions:	เครื่องมือสร้างกรณีทดสอบตามสคีมาของฐานข้อมูลและรายละเอียดที่ผู้ใช้กำหนด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.13 รายละเอียดของยูสเคสกำหนดรายละเอียดสำหรับการสร้างกรณีทดสอบ

Use Case:	Specify details of generation
Actors:	User
Goal:	เพื่อกำหนดรายละเอียดของการสร้างกรณีทดสอบ
Related use cases:	-
Preconditions:	ผู้ใช้เลือกสร้างกรณีทดสอบ
Steps:	1.เครื่องแสดงหน้าจอการรับรายละเอียดของกรณีทดสอบ 2.ผู้ใช้กำหนดรายละเอียดของรูปแบบของกรณีทดสอบที่สร้างขึ้น 3.ผู้ใช้กำหนดชนิดของฐานข้อมูลที่จะนำไปทดสอบใช้ 4.ผู้ใช้กำหนดรูปแบบการบันทึกของกรณีทดสอบ 5. ผู้ใช้ยืนยันรายละเอียดของการสร้างกรณีทดสอบ
Postconditions:	เครื่องมือรับรายละเอียดของการสร้างกรณีทดสอบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.14 รายละเอียดของยูสเคสผลลัพธ์

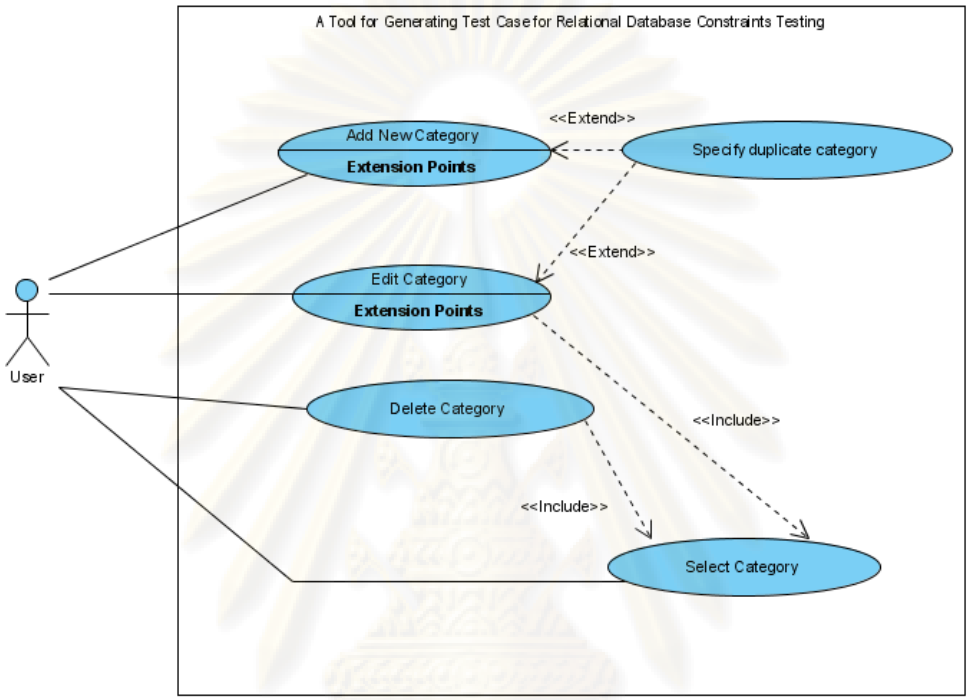
Use Case:	Generate Result
Actors:	User
Goal:	เพื่อบันทึกกรณีทดสอบที่สร้างเสร็จตามรูปแบบที่ผู้ใช้กำหนดขึ้น
Related use cases:	Generalization of: ▪ Generate XML File
Preconditions:	เครื่องมือสร้างกรณีทดสอบเสร็จแล้ว
Steps:	เครื่องมือบันทึกกรณีทดสอบที่สร้างขึ้น
Postconditions:	-

ตารางที่ 3.15 รายละเอียดของยูสเคสสร้างแฟ้มเอกสารเอ็กซ์เอ็มแอล

Use Case:	Generate XML Files
Actors:	User
Goal:	เพื่อบันทึกกรณีทดสอบเป็นแฟ้มเอกสารเอ็กซ์เอ็มแอล
Related use cases:	Specialization of: Generate Result
Preconditions:	เครื่องมือสร้างกรณีทดสอบเสร็จแล้ว
Steps:	เครื่องมือบันทึกกรณีทดสอบที่สร้างขึ้นเป็นแฟ้มเอกสารเอ็กซ์เอ็มแอล
Postconditions:	-

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.) แพ็คเกจการจัดการกลุ่มค่าทั่วไป รายละเอียดของแพ็คเกจนี้สามารถแสดงได้ดังรูปที่ 3.5 ซึ่งแพ็คเกจการจัดการกลุ่มค่าทั่วไปจะประกอบด้วยยูสเคสทั้งหมด 5 ยูสเคส ซึ่งแต่ละยูสเคสจะเกี่ยวข้องกับการจัดการกลุ่มค่าทั่วไปทั้งหมด โดยรายละเอียดของแต่ละยูสเคสจะเป็นดังตารางที่ 3.16 ถึงตารางที่ 3.19



รูปที่ 3.5 ยูสเคสภายในแพ็คเกจการจัดการกลุ่มค่าทั่วไป

ตารางที่ 3.16 รายละเอียดของยูสเคสการเพิ่มกลุ่มค่าทั่วไปใหม่

Use Case:	Add new category
Actors:	User
Goal:	เพื่อสร้างกลุ่มค่าทั่วไปกลุ่มใหม่ที่ผู้ใช้ต้องการลงในฐานข้อมูลของเครื่องมือ
Related use cases:	-
Preconditions:	-
Steps:	1.ผู้ใช้เลือกเพิ่มกลุ่มค่าทั่วไป 2.เครื่องมือแสดงหน้าจอการจัดการข้อมูลกลุ่มค่าทั่วไปใหม่

ตารางที่ 3.16 รายละเอียดของยูสเคสการเพิ่มกลุ่มค่าทั่วไปใหม่(ต่อ)

Steps:	3.ผู้ใช้กำหนดชื่อและชนิดของกลุ่มค่าทั่วไป 4.ผู้ใช้เพิ่มค่าทั่วไปให้กลุ่มค่าทั่วไป 5.ผู้ใช้ยืนยันการสร้างกลุ่มค่าทั่วไป 6.เครื่องมือตอบรับการสร้างกลุ่มค่าทั่วไป
Postconditions:	เครื่องมือสร้างกลุ่มค่าทั่วไปกลุ่มใหม่และเพิ่มเข้าสู่รายการกลุ่มค่าทั่วไปของเครื่องมือ

ตารางที่ 3.17 รายละเอียดของยูสเคสแก้ไขกลุ่มค่าทั่วไป

Use Case:	Edit category
Actors:	User
Goal:	เพื่อแก้ไขข้อมูลกลุ่มค่าทั่วไปที่ผู้ใช้เลือก
Related use cases:	Include: Select category
Preconditions:	-
Steps:	1.ผู้ใช้เลือกแก้ไขกลุ่มค่าทั่วไป 2.ผู้ใช้เลือกกลุ่มค่าทั่วไปที่ต้องการแก้ไข (Use case: Select category) 3.เครื่องมือแสดงหน้าจอการจัดการข้อมูลกลุ่มค่าทั่วไปใหม่ 4.ผู้ใช้แก้ไขค่าทั่วไปให้กลุ่มค่าทั่วไป 5.ผู้ใช้ยืนยันการแก้ไขกลุ่มค่าทั่วไป

ตารางที่ 3.17 รายละเอียดของยูสเคสแก้ไขกลุ่มค่าทั่วไป(ต่อ)

Steps:	6.เครื่องมือตอบรับการแก้ไขกลุ่มค่าทั่วไป
Postconditions:	เครื่องมือแก้ไขกลุ่มค่าทั่วไปที่ผู้ใช้กำหนด

ตารางที่ 3.18 รายละเอียดของยูสเคสลบกลุ่มค่าทั่วไป

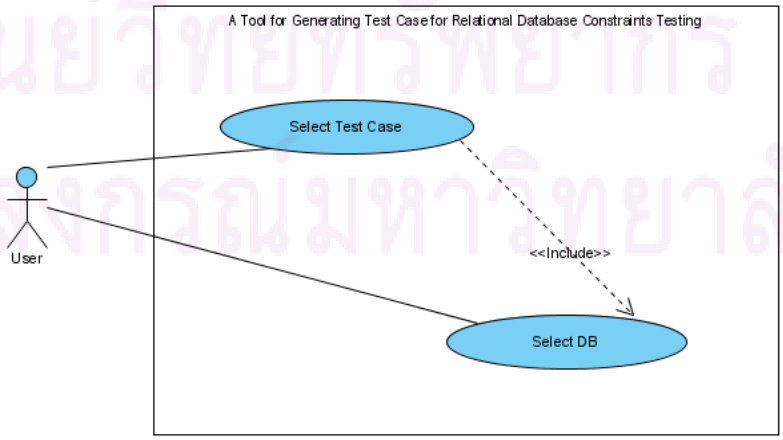
Use Case:	Delete category
Actors:	User
Goal:	เพื่อลบกลุ่มค่าทั่วไปที่ผู้ใช้เลือก
Related use cases:	Include: Select category
Preconditions:	-
Steps:	1.ผู้ใช้เลือกลบกลุ่มค่าทั่วไป 2.ผู้ใช้เลือกกลุ่มค่าทั่วไปที่ต้องการลบ (Use case: Select category) 3.เครื่องมือแสดงหน้าจอการจัดการข้อมูลกลุ่มค่าทั่วไปใหม่ 4.ผู้ใช้ลบค่าทั่วไปให้กลุ่มค่าทั่วไป 5.ผู้ใช้ยืนยันการลบกลุ่มค่าทั่วไป 6.เครื่องมือตอบรับการลบกลุ่มค่าทั่วไป
Postconditions:	เครื่องมือลบกลุ่มค่าทั่วไปที่ผู้ใช้กำหนดออกจากฐานข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.19 รายละเอียดของยูสเคสเลือกกลุ่มค่าทั่วไป

Use Case:	Select category
Actors:	User
Goal:	เพื่อเลือกกลุ่มค่าทั่วไปในการแก้ไขหรือลบกลุ่มค่าทั่วไป
Related use cases:	-
Preconditions:	ผู้ใช้เลือกแก้ไขกลุ่มค่าทั่วไป หรือผู้ใช้เลือกลบกลุ่มค่าทั่วไป
Steps:	1.เครื่องมือแสดงหน้าจอรายชื่อกลุ่มค่าทั่วไปที่มีอยู่ในฐานข้อมูลเครื่องมือ 2.ผู้ใช้เลือกกลุ่มค่าทั่วไปที่ต้องการ 3.ผู้ใช้ยืนยันการเลือกกลุ่มค่าทั่วไป
Preconditions:	เครื่องมือเลือกกลุ่มค่าทั่วไปที่ผู้ใช้ต้องการ

4.) แพ็คเกจการประมวลผลกรณีทดสอบ รายละเอียดของแพจเกจนี้สามารถแสดงดังรูปที่ 3.6 ซึ่งแพ็คเกจการประมวลผลกรณีทดสอบจะประกอบด้วยยูสเคสทั้งหมด 2 ยูสเคส ซึ่งแต่ละยูสเคสจะเกี่ยวข้องกับการประมวลผลกรณีทดสอบทั้งหมด โดยรายละเอียดของแต่ละยูสเคสจะเป็นดังตารางที่ 3.20 ถึงตารางที่ 3.21



รูปที่ 3.6 ยูสเคสภายในแพ็คเกจการประมวลผลกรณีทดสอบ

ตารางที่ 3.20 รายละเอียดของยูสเคสเลือกกรณีทดสอบ

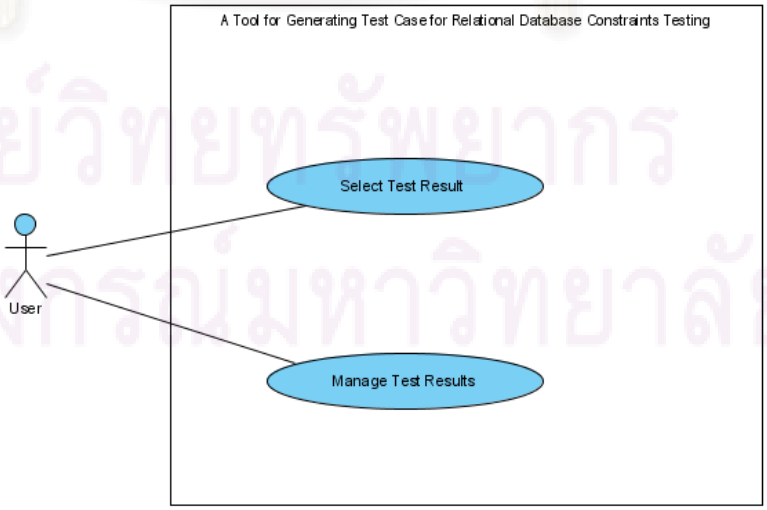
Use Case:	Select Test case
Actors:	User
Goal:	เพื่อเลือกกรณีทดสอบที่ต้องการทำการทดสอบ
Related use cases:	Include: Select DB
Preconditions:	เครื่องมือจะต้องทำการเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ
Steps:	1. เครื่องมือแสดงหน้าจอการเลือกกรณีทดสอบที่ต้องการทดสอบ 2. ผู้ใช้กำหนดกรณีทดสอบที่ต้องการทดสอบ 3. ผู้ใช้ยืนยันการเลือกกรณีทดสอบ 4. เครื่องมือทำการประมวลผลกรณีทดสอบ
Postconditions:	เครื่องมือจะทำประมวลผลกรณีทดสอบที่ฐานข้อมูลที่ระบุ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.21 รายละเอียดของยูสเคสเลือกฐานข้อมูลเชื่อมต่อ

Use Case:	Select Database
Actors:	User
Goal:	เพื่อเลือกฐานข้อมูลที่ต้องการเชื่อมต่อ
Related use cases:	-
Preconditions:	-
Steps:	1. เครื่องมือแสดงหน้าจอการเชื่อมกับฐานข้อมูล 2. ผู้ใช้กำหนดฐานข้อมูลที่ต้องการทดสอบ 3. ผู้ใช้ยืนยันการเลือกฐานข้อมูล
Postconditions:	เครื่องมือจะเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ

5.) แพ็คเกจการจัดการผลลัพธ์ของการทดสอบ รายละเอียดของแพ็คเกจนี้สามารถแสดงได้ดังรูปที่ 3.7 ซึ่งแพ็คเกจการจัดการผลลัพธ์ของการทดสอบจะประกอบด้วยยูสเคสทั้งหมด 2 ยูสเคส ซึ่งแต่ละยูสเคสจะเกี่ยวข้องกับการจัดการผลลัพธ์ทั้งหมด โดยรายละเอียดของแต่ละยูสเคสจะเป็นดังตารางที่ 3.22 ถึงตารางที่ 3.23



รูปที่ 3.7 ยูสเคสในแพ็คเกจการจัดการผลลัพธ์การทดสอบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.22 รายละเอียดของยูสเคสเรียกดูผลลัพธ์การทดสอบ

Use Case:	Show test result
Actors:	User
Goal:	เพื่อแสดงผลลัพธ์ของการทดสอบ
Related use cases:	-
Preconditions:	-
Steps:	1. เครื่องมือแสดงหน้าจอการเชื่อมกับฐานข้อมูล 2. ผู้ใช้กำหนดฐานข้อมูลที่ต้องการทดสอบ 3. ผู้ใช้ยืนยันการเลือกฐานข้อมูล
Postconditions:	เครื่องมือจะเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ

ตารางที่ 3.23 รายละเอียดของยูสเคสจัดการผลลัพธ์การทดสอบ

Use Case:	Manage test result
Actors:	User
Goal:	เพื่อจัดการผลการทดสอบ
Related use cases:	-
Preconditions:	เมื่อสร้างกรณีทดสอบและประมวลผลการทดสอบ
Steps:	1.เครื่องมือแสดงหน้าจอแสดงผลลัพธ์ของการทดสอบ 2.ผู้ใช้เลือกดูผลลัพธ์ของแต่ละการทดสอบที่ต้องการจัดการ

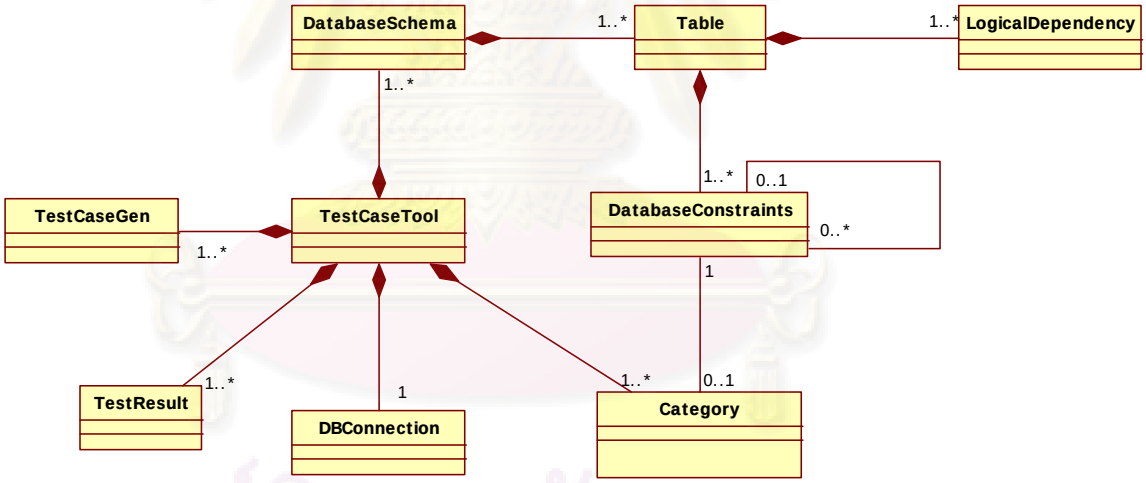
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.23 รายละเอียดของยูสเคสจัดการผลลัพธ์การทดสอบ (ต่อ)

Steps:	3. ผู้ใช้แก้ไขจัดการผลลัพธ์การทดสอบ 4. ผู้ใช้ยืนยันผลการแก้ไขเพิ่มเติมผลการทดสอบ
Postconditions:	เครื่องมือจัดเก็บข้อมูลที่ทำการแก้ไขลงฐานข้อมูล

3.2.2แผนภาพคลาส

แผนภาพคลาสจะแสดงถึงคลาสต่างๆ และความสัมพันธ์ระหว่างคลาสต่างๆ เพื่อจำลองการออกแบบโครงสร้างของระบบ ความสัมพันธ์ระหว่างวัตถุ (Object) ในระบบและองค์ประกอบของข้อมูลที่เกี่ยวข้อง สำหรับแผนภาพของคลาสของเครื่องมือสร้างกรณีทดสอบจะเป็นดังรูปที่ 3.8



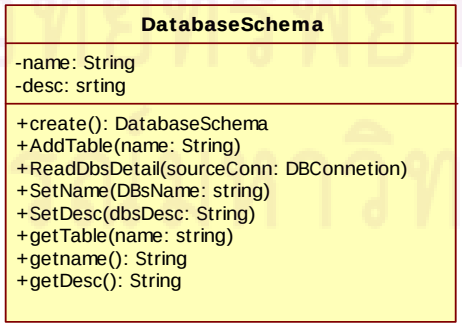
รูปที่ 3.8 แผนภาพคลาสของเครื่องมือสร้างกรณีทดสอบ

จากรูปที่ 3.8 เครื่องมือสร้างกรณีทดสอบจะประกอบด้วยคลาสทั้งหมด 9 คลาส ที่สำคัญ โดยรายละเอียดของคลาสแต่ละคลาสเป็นดังนี้

1. คลาส TestCaseTool คือคลาสหลักที่เป็นตัวแทนของเครื่องมือในการติดต่อกับผู้ใช้ จะทำหน้าที่รับคำสั่งทุกอย่างจากผู้ใช้เพื่อนำมาประมวลผลและส่งผลลัพธ์กลับไปยังผู้ใช้ รายละเอียดของคลาส จะเป็นดังรูปที่ 3.9

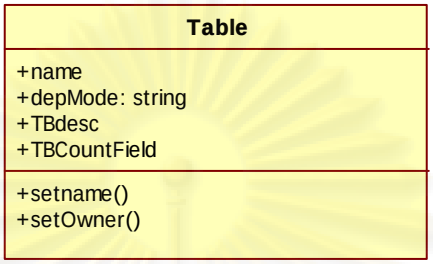
รูปที่ 3.9 คลาส TestCaseTool

2.คลาส DatabaseSchema คือ คลาสที่เป็นตัวแทนของสคีมาของฐานข้อมูลที่มีการใช้งานในเครื่องมือ คลาสนี้จะเก็บข้อมูลทั้งหมดที่มีอยู่ในแต่ละสคีมาของฐานข้อมูล รายละเอียดคลาส DatabasSchema จะเป็นดังรูปที่ 3.10



รูปที่ 3.10 คลาส DatabaseSchema

3. คลาส Table คือ คลาสที่เป็นตัวแทนของตารางข้อมูลในแต่ละสคีมาของฐานข้อมูล คลาสนี้จะเก็บข้อมูลของเขตข้อมูลทั้งหมดที่มีอยู่ในตาราง รายละเอียดของคลาส Table จะเป็นดังรูปที่ 3.1

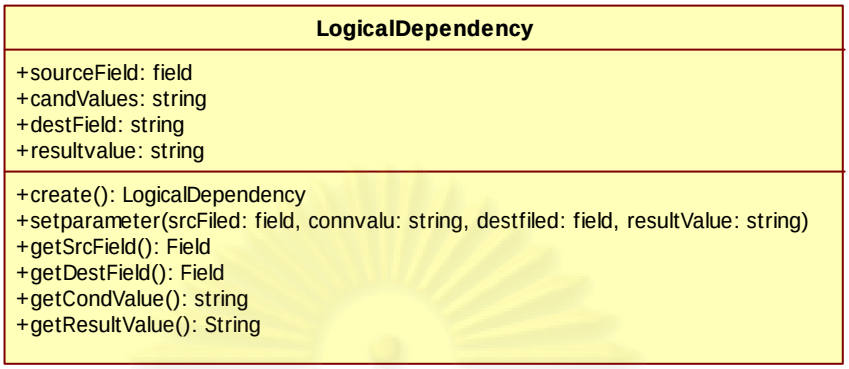


รูปที่ 3.11 คลาส Table

4. คลาส DBConstraints คือ คลาสที่เป็นตัวแทนของเขตข้อมูลในแต่ละตารางข้อมูล คลาสนี้จะทำหน้าที่หลักในการสุ่มสร้างกรณีทดสอบตามคุณสมบัติและเงื่อนไขของฐานข้อมูลรายละเอียดของคลาส DBConstraints จะเป็นดังรูปที่ 3.12

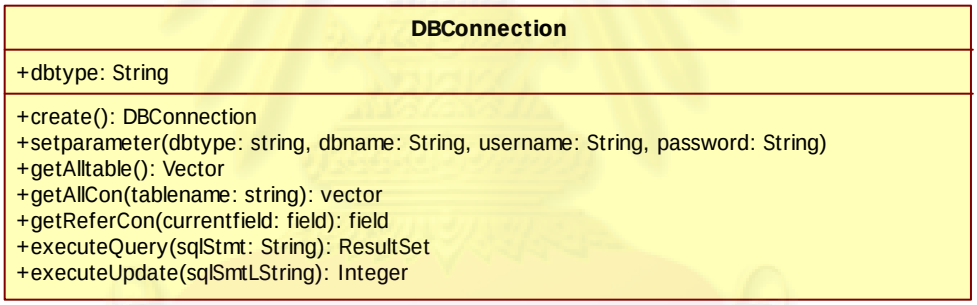
รูปที่ 3.12 คลาส DBConstriants

5. คลาส Logicaldependency คือ คลาสที่เป็นตัวแทนของการขึ้นต่อกันเชิงตรรกะแบบปกติในแต่ละตารางข้อมูล คลาสนี้จะเก็บรายละเอียดของเงื่อนไขการขึ้นต่อกันเชิงตรรกะเพื่อนำไปใช้ในการพิจารณาร่วมในการสร้างกรณีทดสอบ รายละเอียดของคลาส Logicaldependency จะเป็นดังรูปที่ 3.13



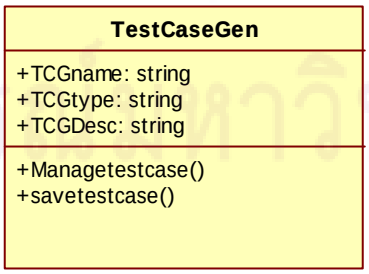
รูปที่ 3.13 คลาส Logicaldependency

6. คลาส DBConnection คือ คลาสที่เป็นตัวแทนการเชื่อมต่อกับฐานข้อมูลต่างๆ ที่จำเป็นต้องใช้ในการทำงานของเครื่องมือ เช่น ฐานข้อมูลที่ต้องการทดสอบที่ให้ข้อมูลในการสร้างสคีมาของฐานข้อมูล และฐานข้อมูลที่จะทำการบันทึกกรณีทดสอบ จะทำหน้าที่รับคำสั่งทุกอย่างที่ต้องทำกับฐานข้อมูลไปประมวลผล รายละเอียดของคลาส DBConnection จะเป็นดังรูปที่ 3.14



รูปที่ 3.14 คลาส DBConnection

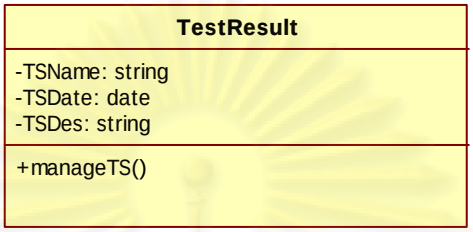
7. คลาส TestCaseGen คือ คลาสที่เป็นตัวแทนของการสร้างกรณีทดสอบ คลาสนี้จะทำการเก็บรายละเอียดจากการสร้างกรณีทดสอบ รายละเอียดของคลาส TestCaseGen จะเป็นดังรูปที่ 3.15



รูปที่ 3.15 คลาสTestCaseGen

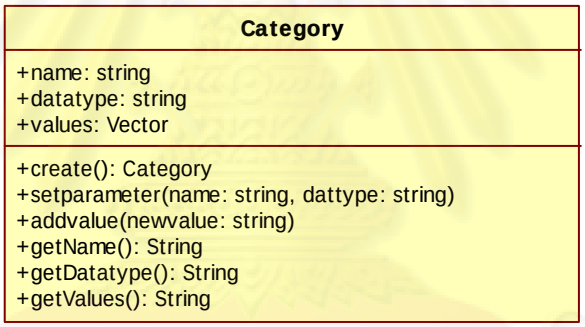
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

8. คลาส TestResult คือ คลาสที่เป็นตัวแทนของการจัดการผลลัพธ์ของการทดสอบ คลาสนี้จะทำการเก็บรายละเอียดของผลลัพธ์การประมวลผลการทดสอบ รายละเอียดของคลาส TestResult จะเป็นดังรูปที่ 3.16



รูปที่ 3.16 คลาส TestResult

9. คลาส Category คือ คลาสที่เป็นตัวแทนของกลุ่มค่าทั่วไปที่เก็บรายการข้อมูลค่าทั่วไปไว้ เพื่อนำไปใช้ในการสร้างกรณีทดสอบ รายละเอียดของคลาส Category จะเป็นดังรูปที่ 3.17



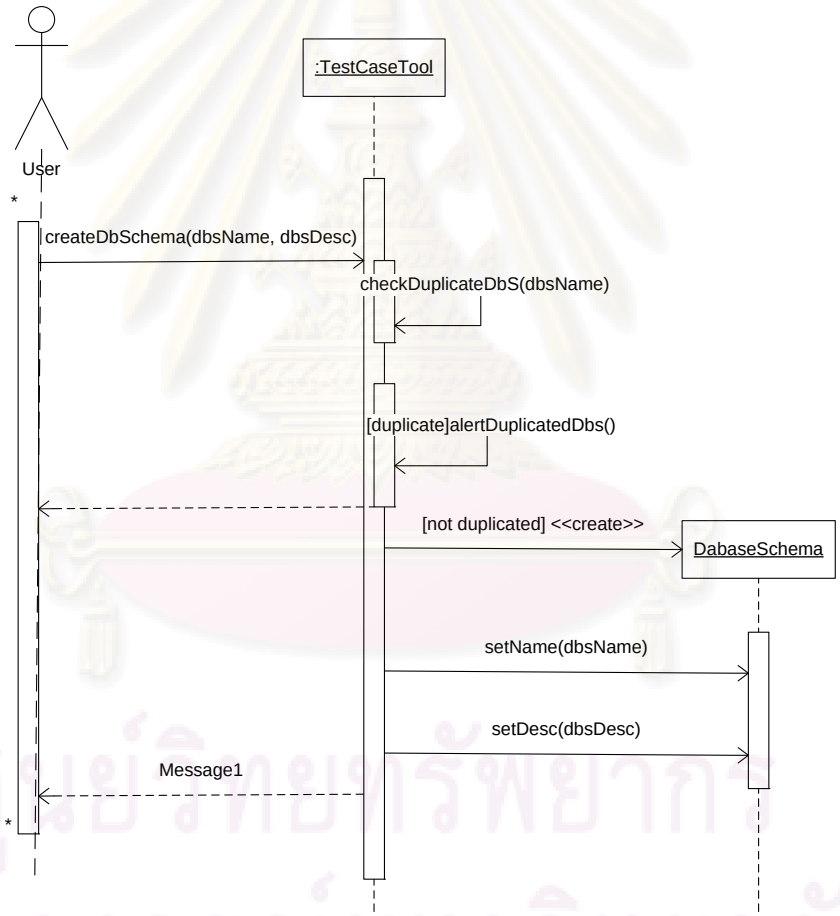
รูปที่ 3.17 คลาสCategory

3.2.3 **แผนภาพซีเควนซ์**

แผนภาพซีเควนซ์เป็นแผนภาพที่แสดงปฏิสัมพันธ์กันระหว่างวัตถุภายในระบบ ปฏิสัมพันธ์คือ การส่งสาร (Message) กันระหว่างวัตถุเพื่อให้บรรลุเป้าหมายในการทำงานหนึ่งๆ แผนภาพซีเควนซ์ของเครื่องมือสามารถอธิบายเป็นกลุ่มๆ ตามยูสเคส ได้แก่ ยูสเคสกำหนดชื่อสคีมาของฐานข้อมูล ยูสเคสการสร้างสคีมาของฐานข้อมูลโดยระบุขึ้นเอง ยูสเคสของการสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่จะทดสอบ ยูสเคสเปิดสคีมาของฐานข้อมูล ยูสเคสปิดสคีมาของฐานข้อมูล ยูสเคสลบสคีมาของฐานข้อมูล ยูสเคสกำหนดการเชื่อมต่อกับฐานข้อมูลที่ทดสอบ ยูสเคลสร้างกรณีทดสอบ ยูสเคสประมวลผลกรณีทดสอบ ยูสเคสยูสเคสเพิ่มกลุ่มค่าทั่วไปใหม่ ยูสเคสลบกลุ่มค่าทั่วไป และยูสเคสจัดการผลลัพธ์การทดสอบ ดังนี้

1.<u>แผนภาพซีเควนซ์ของยูสเคสกำหนดชื่อสคีมาของฐานข้อมูล</u> แผนภาพซีเควนซ์จะเป็นดังรูปที่ 3.18

จากรูปที่ 3.18 การทำงานจะเริ่มจากการที่ผู้ใช้กำหนดชื่อและคำอธิบายของสคีมาของฐานข้อมูลเพื่อให้เครื่องมือ (คลาส TestCaseGenerator) สร้างเป็นสคีมาของฐานข้อมูลใหม่ จากนั้นเครื่องมือจะนำชื่อของสคีมาของฐานข้อมูลที่ผู้ใช้กำหนด มาทำการตรวจสอบกับรายการของสคีมาของที่มีอยู่ว่าซ้ำซ้อนหรือไม่ กรณีที่ซ้ำซ้อนเครื่องมือจะแจ้งให้ผู้ใช้ทราบ แต่ถ้าไม่มีความซ้ำซ้อนเครื่องมือจะทำการสร้างวัตถุของคลาส DatabaseSchema ขึ้นมาแล้วกำหนดชื่อและคำอธิบายให้กับวัตถุนั้น

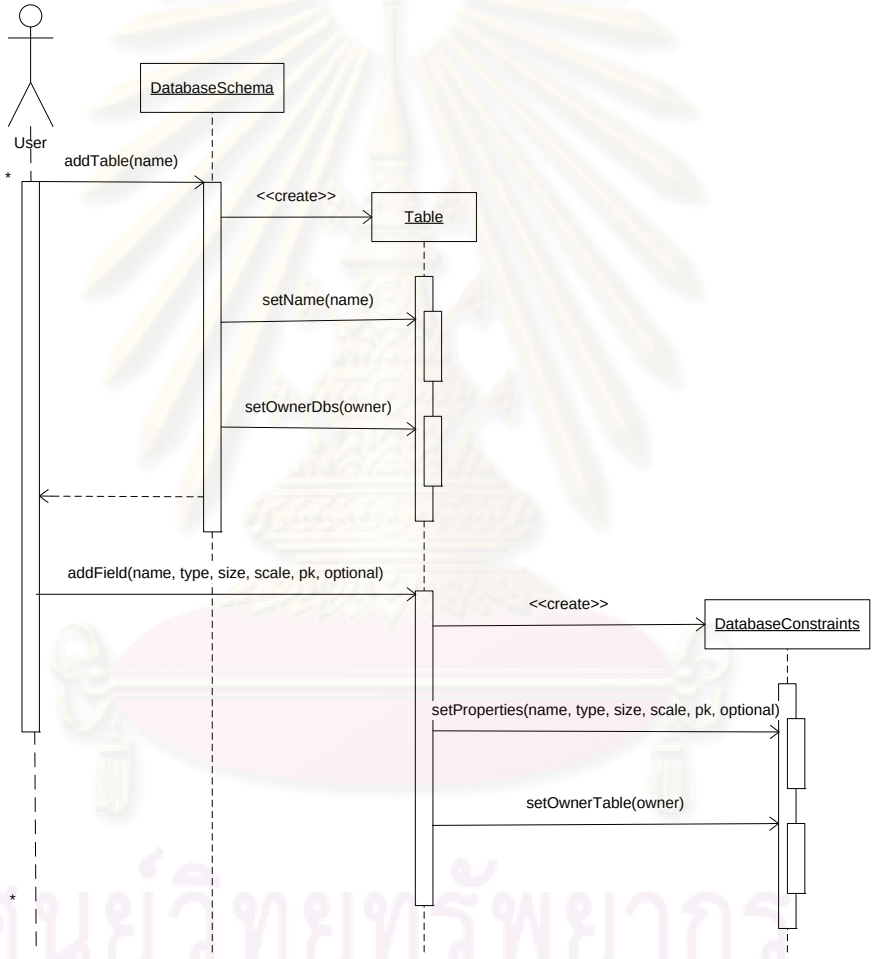


รูปที่ 3.18 แผนภาพซีเควนซ์แสดงการกำหนดชื่อสคีมาของฐานข้อมูล

2.<u>แผนภาพซีเควนซ์ของยูสเคสสร้างสคีมาของฐานข้อมูลโดยผู้ใช้ระบุ</u> สำหรับยูสเคสนี้เริ่มต้นจากการเพิ่มข้อมูลให้กับสคีมาของฐานข้อมูล ดังรูปที่ 3.19 โดยในขั้นแรกผู้ใช้จะทำการเพิ่มตารางข้อมูลให้กับสคีมาของฐานข้อมูลก่อน จากนั้นจึงทำการกำหนดรายละเอียดอื่นๆ

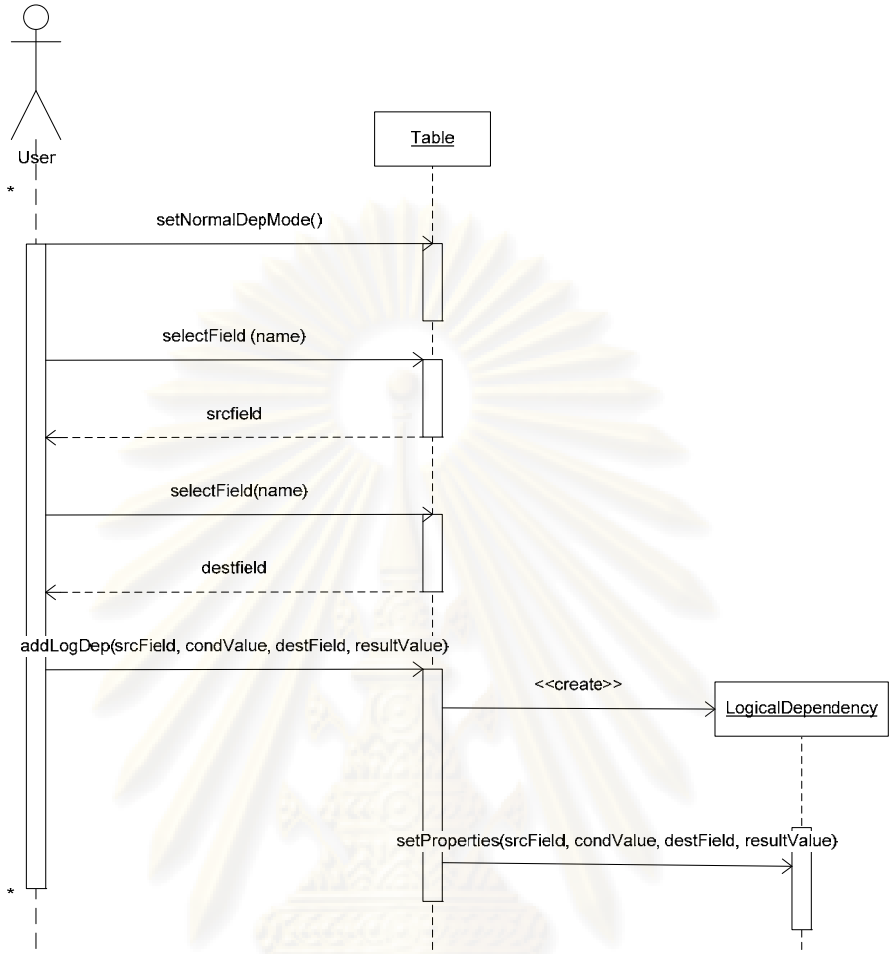
ของฐานข้อมูล เมื่อผู้ให้ทำการกำหนดตารางข้อมูลและเขตข้อมูลให้กับสคีมาของฐานข้อมูล ทั้งนี้ผู้ใช้ต้องทำการกำหนดการขึ้นต่อกันเชิงตรรกะระหว่างเขตข้อมูลภายในตาราง ดังรูปที่ 3.20

นอกจากนี้การกำหนดการขึ้นต่อกันเชิงตรรกะจะเริ่มจากการที่ผู้ใช้ทำการกำหนดให้วัตถุที่แทนตารางข้อมูล ใช้การขึ้นต่อกันเชิงตรรกะ จากนั้นผู้ใช้จะเลือกเขตข้อมูล และเขตข้อมูลปลายทางของเงื่อนไข พร้อมทั้งกำหนดค่าข้อมูลเงื่อนไข และค่าผลลัพธ์ของเงื่อนไข แล้วผู้ใช้จึงสั่งเงื่อนไขให้กับวัตถุที่แทนตารางข้อมูล ซึ่งวัตถุนี้จะนำเงื่อนไขที่กำหนดมาไปสร้างเป็นวัตถุของคลาส LogicalDependency เก็บไว้ในตัวเอง

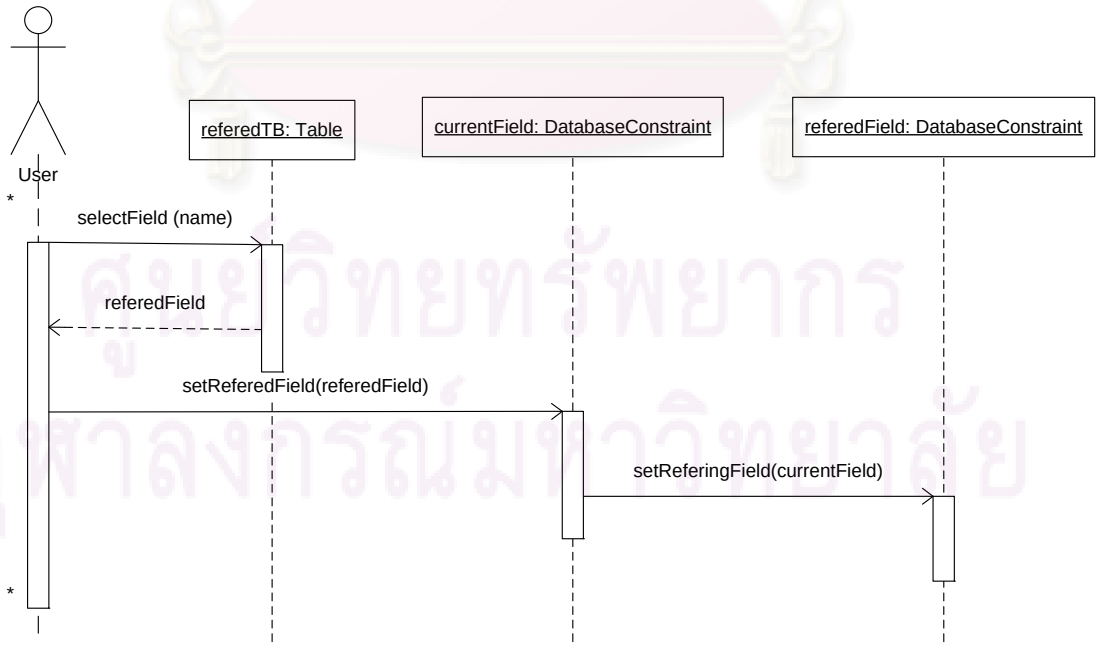


รูปที่ 3.19 แผนภาพซีเควนซ์แสดงการสร้างสคีมาของฐานข้อมูลโดยผู้ใช้ระบุ

เมื่อผู้ใช้กำหนดการขึ้นต่อกันเชิงตรรกะให้กับตารางข้อมูลแล้ว ผู้ใช้ต้องกำหนดความคงสภาพในการอ้างอิงให้กับตารางข้อมูลด้วย โดยการกำหนดความคงสภาพในการอ้างอิงจะเป็นดังรูปที่ 3.21 เริ่มต้นจากผู้ใช้เลือกเขตข้อมูลที่จะถูกอ้างอิงจากตารางข้อมูลที่เป็นเจ้าของเขตข้อมูลนั้น แล้วนำเขตข้อมูลมาทำการกำหนดให้กับเขตของข้อมูลปัจจุบัน

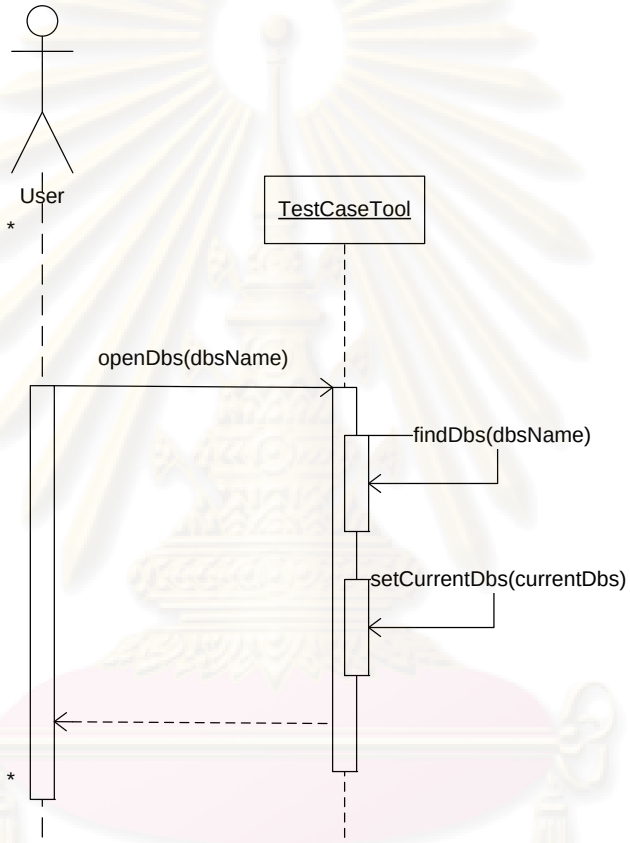


รูปที่ 3.20 แผนภาพซีเควนซ์แสดงการกำหนดการขึ้นต่อกันเชิงตรรกะ



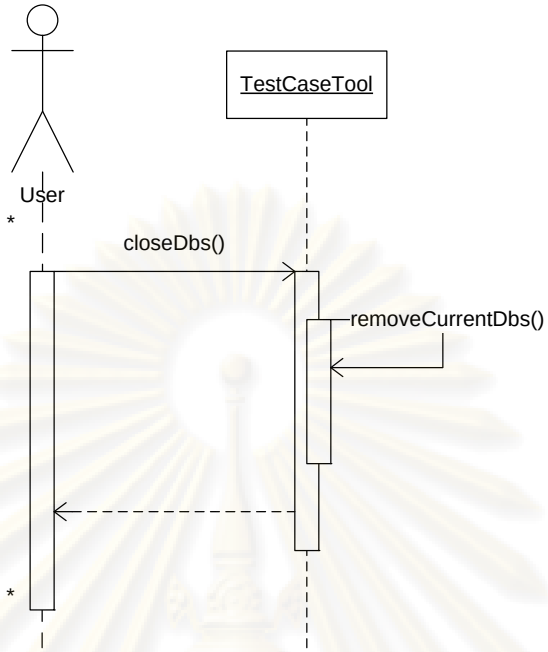
รูปที่ 3.21 แผนภาพซีเควนซ์แสดงการกำหนดความคงสภาพในการอ้างอิง

3.แผนภาพซีเควนซ์ของยูสเคสเปิดสคีมาของฐานข้อมูล แผนภาพซีเควนซ์จะเป็นดังรูปที่ 3.22 การเปิดสคีมาของฐานข้อมูลจะเริ่มจากผู้ใช้สั่งให้เครื่องมือทำการเปิดสคีมาของฐานข้อมูลที่มีชื่อตรงกับชื่อที่ผู้ใช้กำหนด จากนั้นเครื่องมือจะนำชื่อสคีมาของฐานข้อมูลไปค้นหาในรายการสคีมาของฐานข้อมูลทั้งหมด แล้วนำเอาสคีมาของฐานข้อมูลที่พบมากำหนดเป็นสคีมาของฐานข้อมูลปัจจุบันที่จะใช้งาน



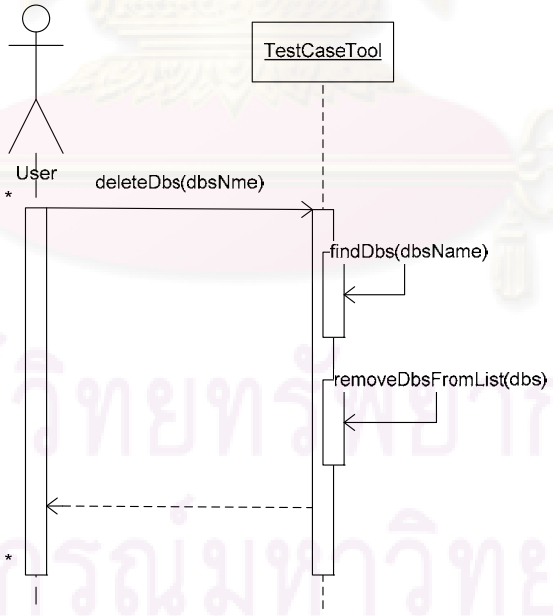
รูปที่ 3.22 แผนภาพซีเควนซ์แสดงการเปิดสคีมาของฐานข้อมูล

4.แผนภาพซีเควนซ์ของยูสเคสปิดสคีมาของฐานข้อมูล แผนภาพซีเควนซ์จะเป็นดังรูปที่ 3.22 การปิดสคีมาของฐานข้อมูลจะเริ่มจากการที่ผู้ใช้สั่งให้เครื่องมือปิดสคีมาของฐานข้อมูลปัจจุบันที่ใช้งานอยู่ จากนั้นเครื่องมือจะเอาสคีมาของฐานข้อมูลปัจจุบันออกเพื่อเป็นการปิดสคีมาของฐานข้อมูลปัจจุบัน



รูปที่ 3.22 แผนภาพซีเควนซ์แสดงการปิดสคีมาของฐานข้อมูล

5.แผนภาพซีเควนซ์ของยูสเคสลบสคีมาของฐานข้อมูล แผนภาพซีเควนซ์จะเป็นดังรูปที่ 3.23



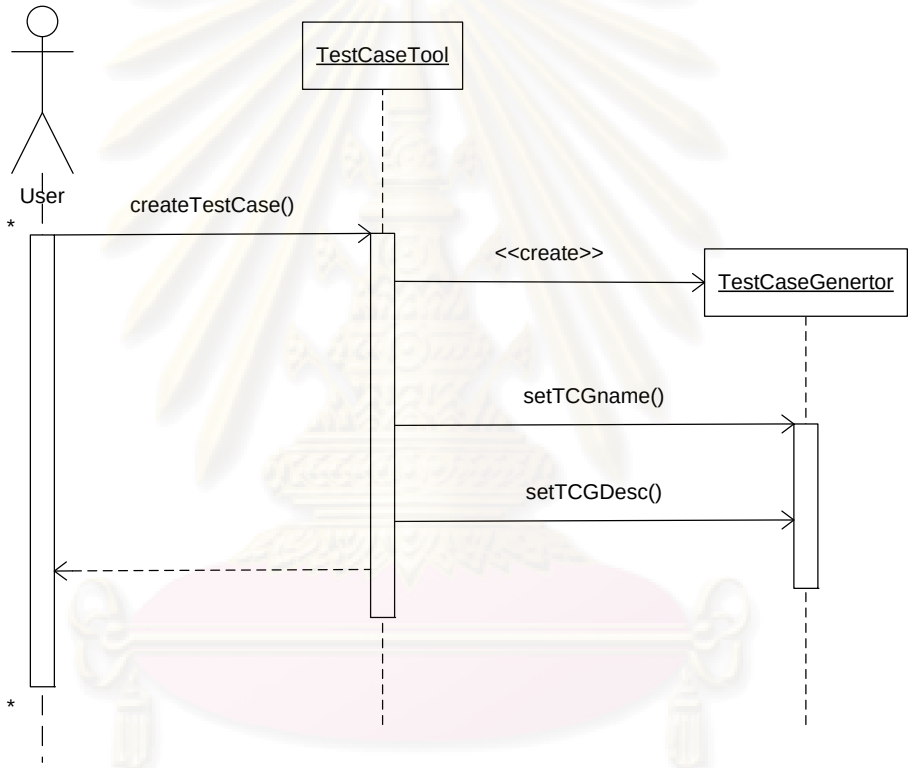
รูปที่ 3.23 แผนภาพซีเควนซ์แสดงการลบสคีมาของฐานข้อมูล

จากรูปที่ 3.23 การลบสคีมาของฐานข้อมูลจะเริ่มจากการที่ผู้ใช้สั่งให้เครื่องมือลบสคีมาของฐานข้อมูลที่มีชื่อตรงกันกับที่ผู้ใช้ได้กำหนด จากนั้นเครื่องมือจะนำชื่อสคีมาของ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

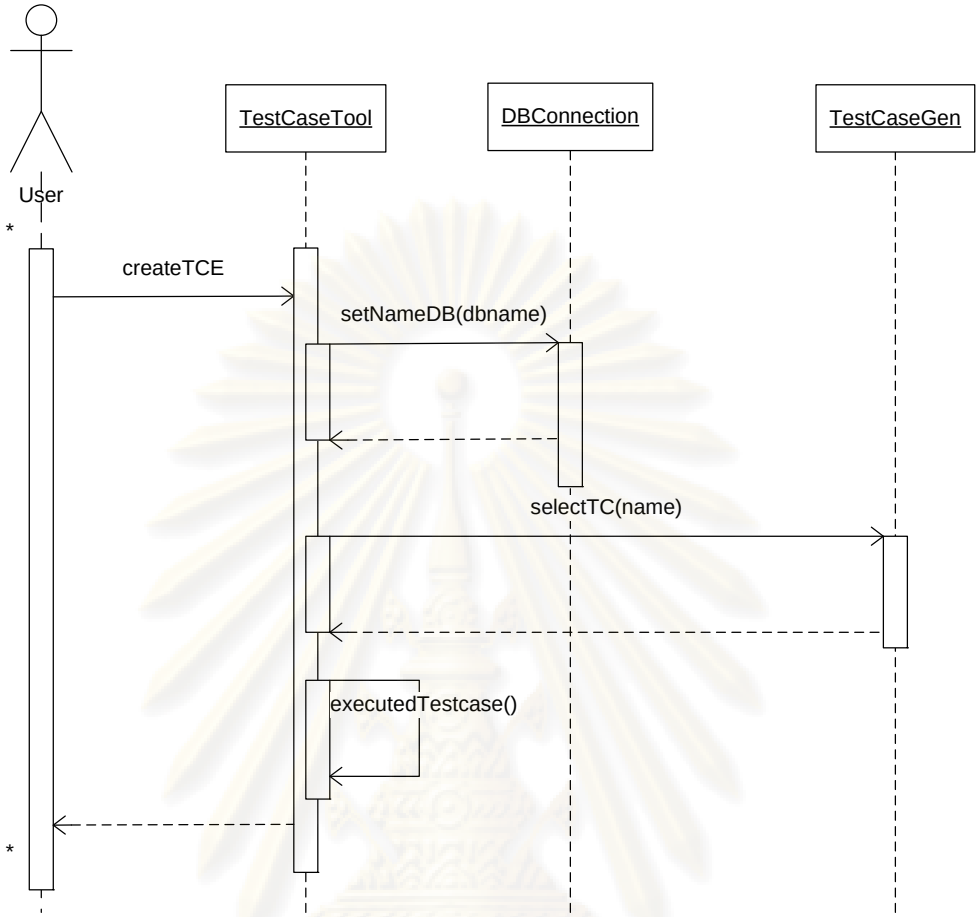
ฐานข้อมูลไปค้นหาในรายการของสคีมาของฐานข้อมูลทั้งหมด แล้วจะเอาสคีมาของฐานข้อมูลที่พบออกจากรายการสคีมาของฐานข้อมูลของฐานข้อมูล

6.แผนภาพซีเควนซ์ของยูสเคสสร้างกรณีทดสอบ แผนภาพซีเควนซ์จะเป็นดังรูป 3.24 การสร้างกรณีทดสอบจะเริ่มจากการที่ใช้ทำการระบุรูปแบบของคำสั่งเอสคิวแอลในการสร้างกรณีทดสอบ ได้แก่ การเพิ่ม การลบ และการปรับปรุงข้อมูล เครื่องมือจะสร้างวัตถุของคลาส TestCaseGen ขึ้นมาแล้วทำการกำหนดชื่อของชุดของกรณีทดสอบที่ทำการสร้างขึ้นให้กับวัตถุนั้น



รูปที่ 3.24 แผนภาพซีเควนซ์แสดงการสร้างกรณีทดสอบ

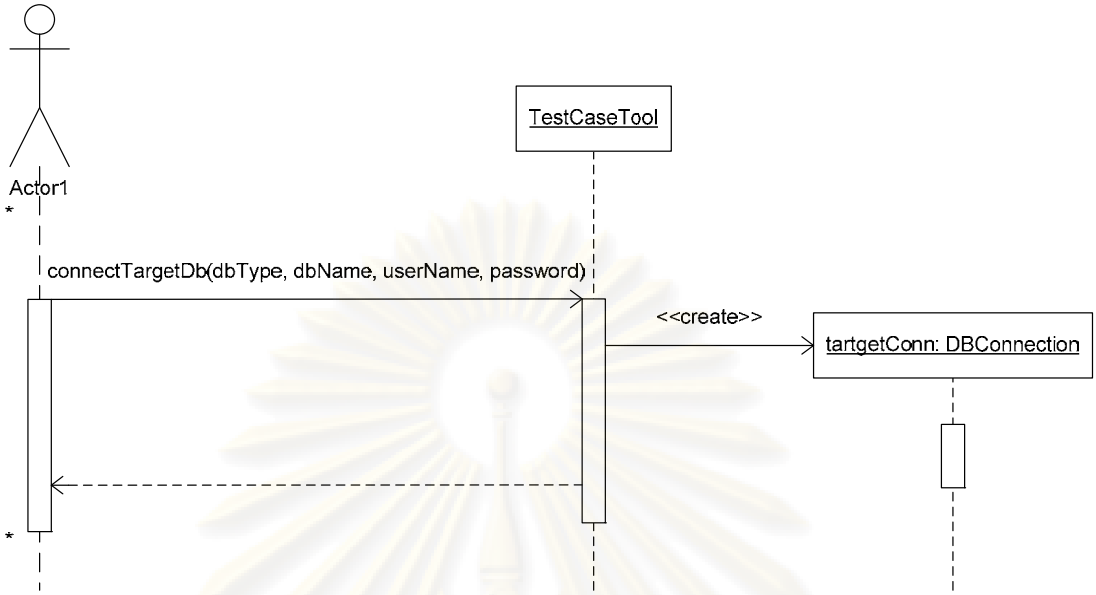
7.แผนภาพซีเควนซ์ของยูสเคสประมวลผลกรณีทดสอบ แผนภาพซีเควนซ์จะเป็นดังรูป 3.25 การประมวลผลกรณีทดสอบจะเริ่มต้นจากการที่ผู้ใช้ทำการระบุชื่อของกรณีทดสอบที่ต้องการประมวลผลและฐานข้อมูลที่ต้องการทดสอบ จากนั้นเครื่องมือจะทำการประมวลผลยังฐานข้อมูลที่มีการเชื่อมต่อไว้แล้ว



รูปที่ 3.25 แผนภาพซีเควนซ์แสดงการประมวลผลกรณีทดสอบ

8.แผนภาพซีเควนซ์ของยูสเคสกำหนดการเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ แผนภาพซีเควนซ์จะเป็นดังรูปที่ 3.26 การกำหนดการเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบจะเริ่มจากผู้ใช้สั่งเครื่องมือเชื่อมต่อกับฐานข้อมูลโดยการส่งพารามิเตอร์สำหรับการเชื่อมต่อมาให้ ซึ่งเครื่องมือจะเชื่อมต่อกับฐานข้อมูลโดยการสร้างวัตถุ DBConn จากคลาส DBConnection เพื่อใช้เป็นตัวแทนการเชื่อมต่อเพื่อใช้ต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

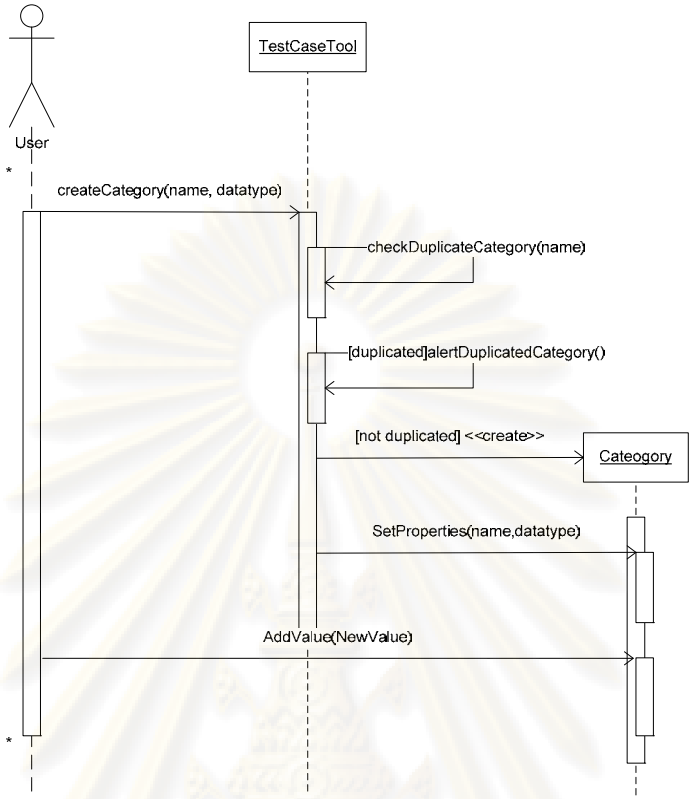


รูปที่ 3.26 แผนภาพซีเควนซ์แสดงการกำหนดการเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ

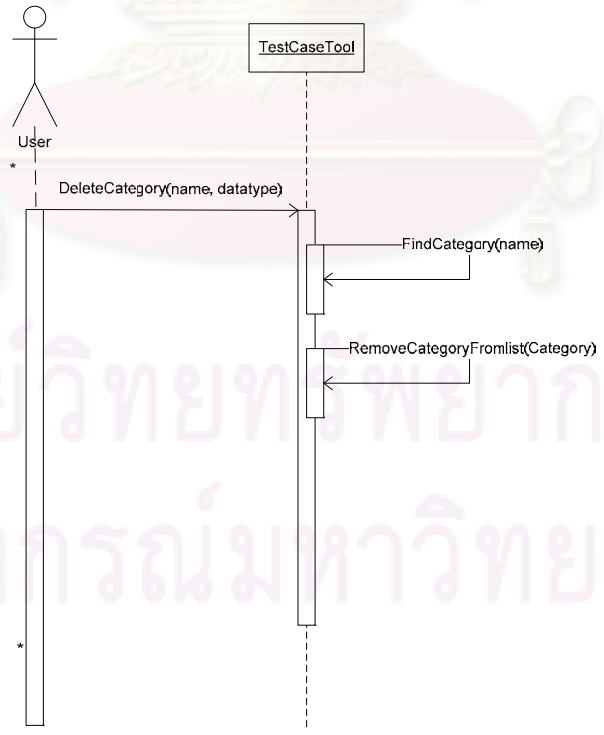
9.แผนภาพซีเควนซ์ของยูสเคสเพิ่มกลุ่มค่าทั่วไป แผนภาพซีเควนซ์จะเป็นดังรูปที่ 3.27 การเพิ่มกลุ่มค่าทั่วไปใหม่ จะเริ่มจากการที่ผู้ใช้สั่งให้เครื่องสร้างกลุ่มค่าทั่วไปโดยกำหนดชื่อและชนิดของข้อมูล จากนั้นเครื่องมือจะนำชื่อกลุ่มค่าทั่วไปมาตรวจสอบความซ้ำซ้อน หากพบจะแจ้งให้ผู้ใช้ทราบ แต่ถ้าไม่พบจะทำการสร้างวัตถุคลาส Category ขึ้นมาแล้วกำหนดชื่อและชนิดของข้อมูลให้กับวัตถุนั้น จากนั้นผู้ใช้จึงทำการเพิ่มค่าข้อมูลให้กับวัตถุที่ต้องการ

10.แผนภาพซีเควนซ์ของยูสเคสลบกลุ่มค่าทั่วไป แผนภาพซีเควนซ์จะเป็นดังรูปที่ 3.28 การลบกลุ่มค่าทั่วไปจะเริ่มจากผู้ใช้สั่งให้เครื่องมือลบกลุ่มค่าทั่วไปที่มีชื่อตรงกับที่ผู้ใช้กำหนด จากนั้นเครื่องมือจะนำชื่อของกลุ่มค่าทั่วไปไปค้นหาในรายการกลุ่มค่าทั่วไปทั้งหมด แล้วจะเอากลุ่มค่าทั่วไปที่พบออกจากรายการกลุ่มค่าทั่วไปของฐานข้อมูล

11.แผนภาพซีเควนซ์ของยูสเคสจัดการผลลัพธ์การทดสอบ แผนภาพซีเควนซ์เป็นดังรูปที่ 3.29 เริ่มจากผู้ใช้สั่งให้เครื่องมือจัดการผลลัพธ์ของการทดสอบที่มีชื่อตรงกับที่ผู้ใช้กำหนด จากนั้นเครื่องมือจะนำชื่อของผลลัพธ์การทดสอบไปค้นหาในรายการทั้งหมด แล้วทำการจัดการกับข้อมูลผลลัพธ์ของการทดสอบ

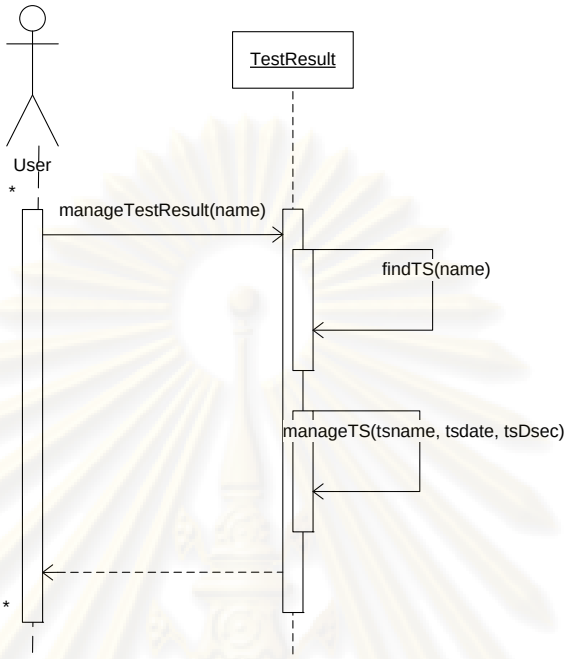


รูปที่ 3.27 แผนภาพซีเควนซ์แสดงการเพิ่มกลุ่มค่าทั่วไป



รูปที่ 3.28 แผนภาพซีเควนซ์แสดงการลบกลุ่มค่าทั่วไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.29 แผนภาพซีเควนซ์แสดงการจัดการผลลัพธ์การทดสอบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การพัฒนาเครื่องมือ

การอธิบายการพัฒนาเครื่องมือจะแบ่งเป็น 3 ส่วน ดังนี้

4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมีมีรายละเอียดดังต่อไปนี้

1.ฮาร์ดแวร์ (Hardware)

- เครื่องคอมพิวเตอร์แบบพีซี (PC) หน่วยประมวลผลอินเทลเพนเทียมโฟร์ 3.0 กิกะเฮิร์ทซ์(Intel Pentium IV 3.0c GHz)
- หน่วยความจำสำรอง (RAM) 1 กิกะไบต์ (1 GB)
- ฮาร์ดดิสก์(Hardisk) 120 กิกะไบต์ (120 GB)

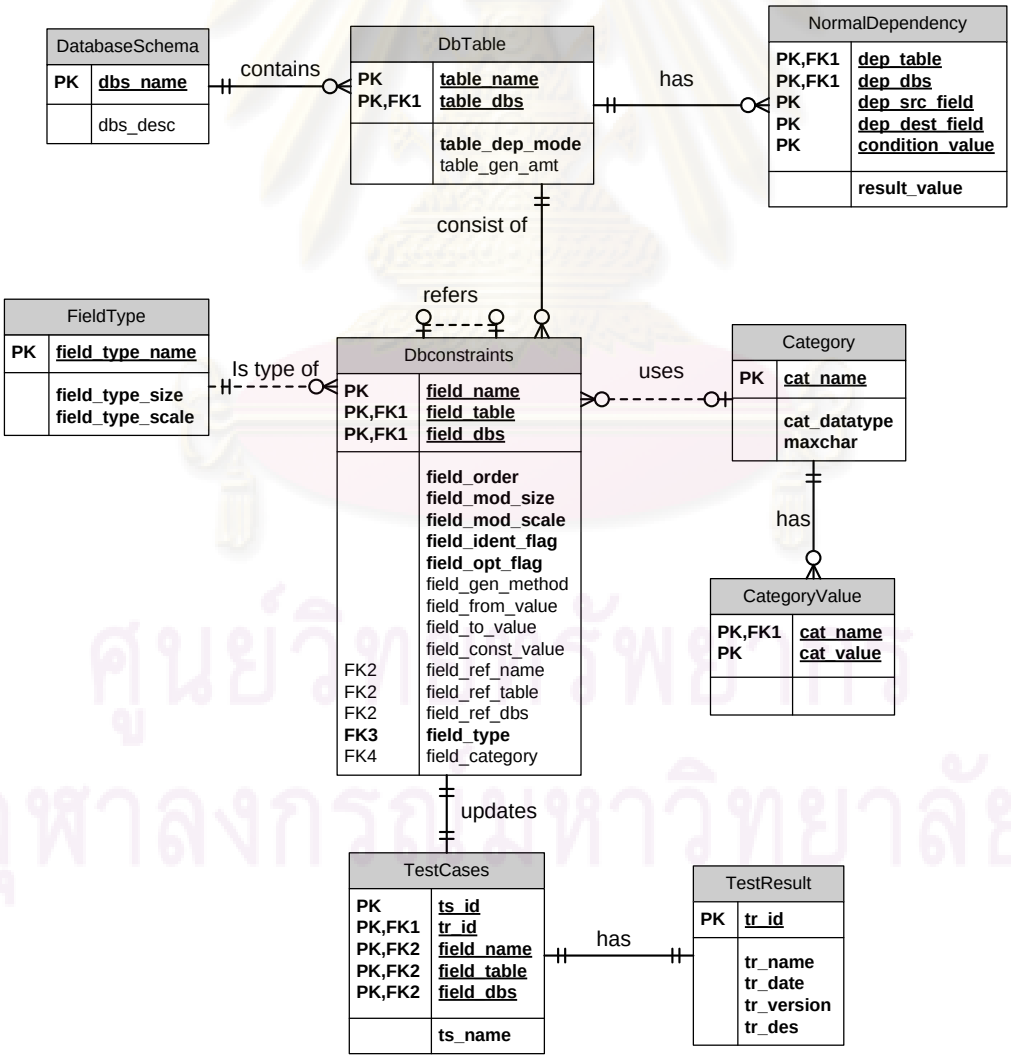
2. ซอฟต์แวร์

- ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์เอ็กซ์พี (Microsoft Windows XP)
- ไมโครซอฟท์วิชวล์สตูดิโอ 2005 (Microsoft Visual Studio 2005)
- ฐานข้อมูลมายเอสคิวแอล เวอร์ชัน 4.0.18 (MySQL database 4.0.18)
- ฐานข้อมูลออราเคิล เวอร์ชัน 9i (Oracle 9i)
- ฐานข้อมูลไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ 2000 (Microsoft SQL Server 2000)
- ฐานข้อมูลไมโครซอฟท์แอ็คเซส 2003 (Microsoft Access 2003)

4.2 โครงสร้างฐานข้อมูลของเครื่องมือ

โครงสร้างของฐานข้อมูลของเครื่องมือจะอธิบายด้วยแผนภาพความสัมพันธ์ระหว่างเอนทิตีระดับกายภาพ (Physical Entity-Relationship Diagram) ดังรูปที่ 4.1 เอนทิตีที่ปรากฏอยู่ในแผนภาพนั้นคือ ตารางข้อมูลในฐานข้อมูลของเครื่องมือ โดยรายละเอียดของแต่ละตารางเป็นดังนี้

1. ตาราง DBSchema เป็นตารางที่เก็บข้อมูลชื่อและคำอิบายสคีมาของฐานข้อมูลที่เคยสร้างไว้โดยผู้ใช้
2. ตาราง DBTable เป็นตารางที่เก็บรายละเอียดของแต่ละตารางข้อมูลที่อยู่ในสคีมาของฐานข้อมูล
3. ตาราง DBConstraints เป็นตารางที่เก็บรายละเอียดของแต่ละเขตข้อมูลที่อยู่ตารางข้อมูลต่างๆ
4. ตาราง FiledType เป็นตารางที่เก็บรายละเอียดของชนิดของข้อมูลต่างๆ ที่จะถูกใช้กำหนดให้กับเขตของข้อมูล
5. ตาราง NorDependency เป็นตารางที่เก็บรายละเอียดของการขึ้นต่อกันเชิงตรรกะแบบปกติระหว่างเขตของข้อมูลในตารางข้อมูลต่างๆ



รูปที่ 4.1 โครงสร้างฐานข้อมูลของเครื่องมือ

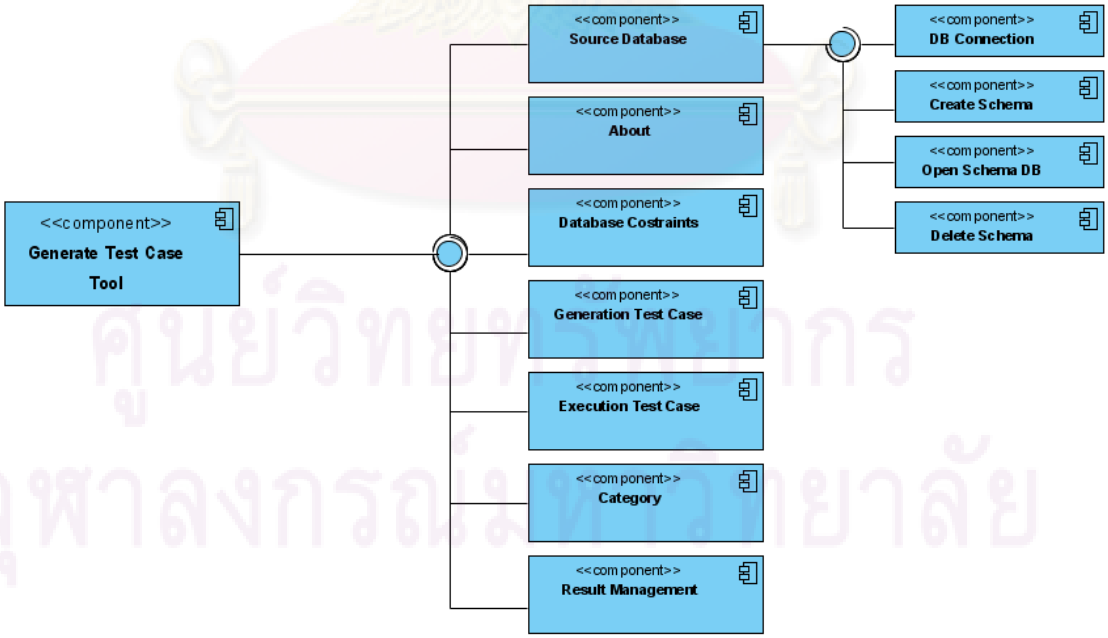
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6. ตาราง Categories เป็นตารางที่เก็บรายละเอียดของกลุ่มค่าทั่วไปที่จะใช้ในการสร้างข้อมูลทดสอบและกรณีทดสอบ
7. ตาราง CateValue เป็นตารางที่เก็บค่าข้อมูลของกลุ่มค่าทั่วไป
8. ตาราง TestCase เป็นตารางที่เก็บรายละเอียดจากการสร้างกรณีทดสอบตามเงื่อนไขของฐานข้อมูล
9. ตาราง TestResult เป็นตารางที่เก็บรายละเอียดของการทดสอบฐานข้อมูล

สำหรับพจนานุกรมข้อมูล (Data Dictionary) ของแต่ละตารางข้อมูล สามารถดูได้ที่ภาคผนวก ก

4.3 โครงสร้างของเครื่องมือ

โครงสร้างของเครื่องมือสร้างกรณีทดสอบสำหรับทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์นั้น จะอธิบายโดยใช้แผนภาพส่วนประกอบ (Component Diagram) ซึ่งเป็นแผนภาพที่ใช้แสดงความสัมพันธ์ระหว่างส่วนประกอบต่างๆ ในระบบ โดยแผนภาพส่วนประกอบของเครื่องมือสร้างกรณีทดสอบสำหรับทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ เป็นดังรูปที่ 4.2



รูปที่ 4.2 แผนภาพส่วนประกอบของเครื่องมือ

จากรูปที่ 4.2 แต่ละส่วนประกอบจะแทนแต่ละหน้าจอของเครื่องมือ ซึ่งมีรายละเอียดดังต่อไปนี้

1. หน้าแรกหรือหน้าหลักของเครื่องมือสร้างกรณีทดสอบ (GenerateTestCaseTool) เพื่อทดสอบเงื่อนไขของ ฐานข้อมูลเชิงสัมพันธ์ มีลักษณะเป็นดังรูปที่ 4.3 หน้าจอนี้ประกอบด้วยแถบรายการ (Menu Bar) ซึ่งประกอบด้วยรายการ 5 รายการ ดังนี้

1.1 Source Database เป็นรายการที่ใช้ในการจัดเก็บข้อมูลเกี่ยวกับสคีมาของฐานข้อมูลที่จะทำการสร้างกรณีทดสอบ ได้แก่ การสร้างสคีมาของฐานข้อมูล การเปิดหรือการปิดสคีมาของฐานข้อมูล การแก้ไขสคีมาของฐานข้อมูล และการลบสคีมาของฐานข้อมูล รวมทั้งการออกจากโปรแกรมด้วย

1.2 Database Constraints เป็นรายการที่ใช้กำหนดและรับเงื่อนไขบังคับของฐานข้อมูล ได้แก่ ชื่อตารางข้อมูล เขตข้อมูล และคุณสมบัติต่างๆของเขตข้อมูล รวมถึงเงื่อนไขบังคับของฐานข้อมูล (ได้แก่ เงื่อนไขบังคับบูรณภาพของเอนทิตี เงื่อนไขบังคับบูรณภาพการอ้างอิงถึง และเงื่อนไขบังคับของโดเมน)

1.3 Generator Test Case เป็นรายการที่ใช้ในการสร้างกรณีทดสอบตามสคีมาของฐานข้อมูลที่เลือกไว้

1.4 Execution เป็นรายการที่ใช้ในการประมวลผลกรณีทดสอบ ซึ่งเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ

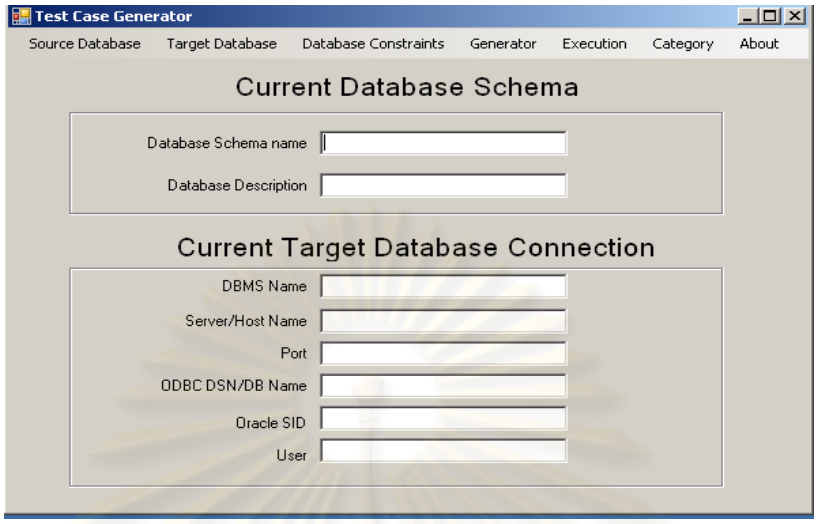
1.5 Category เป็นรายการที่ใช้ในการจัดการกลุ่มค่าทั่วไปของเครื่องมือ ได้แก่ การเพิ่มค่าทั่วไป การแก้ไขกลุ่มค่าทั่วไป และการลบกลุ่มค่าทั่วไป

1.6 Result Management เป็นรายการที่ใช้ในการจัดการผลลัพธ์จากการทดสอบกรณีทดสอบ ได้แก่ ชื่อของกรณีทดสอบที่ทดสอบ วันเวลาที่ทดสอบ และครั้งที่ทดสอบ

1.7 About เป็นรายการที่ใช้แสดงข้อมูลเกี่ยวกับเครื่องมือ

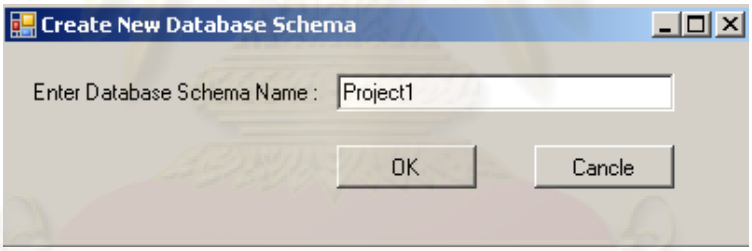
นอกจากแถบรายการแล้ว หน้าจอหลักยังประกอบด้วยส่วนการแสดงผล 2 ส่วน คือ ส่วนแสดงข้อมูลของสคีมาของฐานข้อมูลของฐานข้อมูลปัจจุบันที่ใช้งานอยู่ และส่วนแสดงรายละเอียดการเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบที่เครื่องมือกำลังเชื่อมต่ออยู่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.3 หน้าจอหลัก

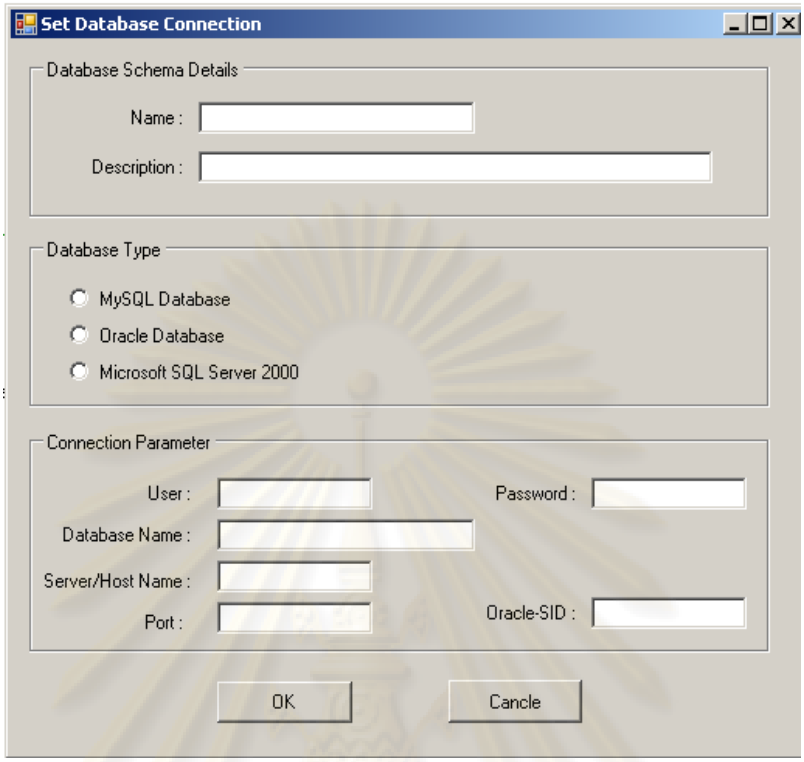
2. หน้าจอการสร้างสคีมาของฐานข้อมูล(CreateSchema) เป็นหน้าจอสร้างสคีมาของฐานข้อมูลใหม่ดังรูปที่ 4.4 หน้าจอนี้ทำการรับชื่อสคีมาของฐานข้อมูลเพื่อนำไปสร้างเป็นสคีมาของฐานข้อมูลใหม่



รูปที่ 4.4 หน้าจอสร้างสคีมาของฐานข้อมูลใหม่

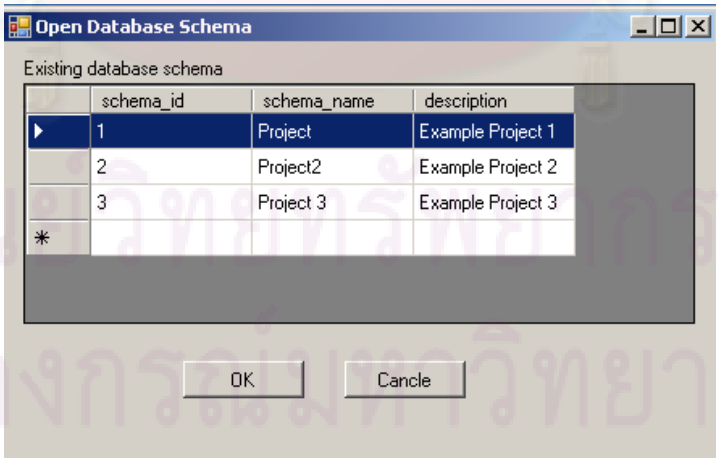
3. หน้าจอการเชื่อมต่อฐานข้อมูล (DBConnection) หน้าจอสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่ต้องการทดสอบ มีลักษณะดังรูปที่ 4.5 หน้าจอจะรับข้อมูลชื่อและคำอธิบายของสคีมาของฐานข้อมูลใหม่ พร้อมทั้งชนิดของฐานข้อมูลและพารามิเตอร์ (Parameters) ต่างๆ ที่จะใช้ในการเชื่อมต่อกับฐานข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



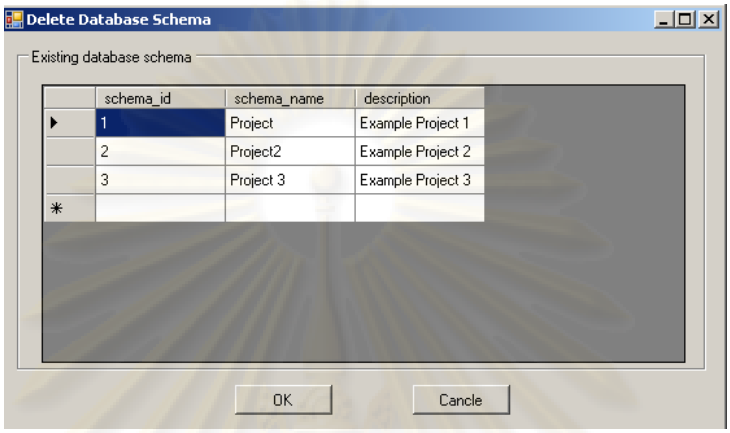
รูปที่ 4.5 หน้าจอสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่ต้องการทดสอบ

4. หน้าจอเปิดสคีมาของฐานข้อมูล (OpenSchemaDB) มีลักษณะดังรูปที่ 4.6 หน้าจอจะแสดงรายการสคีมาของฐานข้อมูลที่มีอยู่ในเครื่องมือ เพื่อให้ผู้ใช้เลือกเปิดตามต้องการ



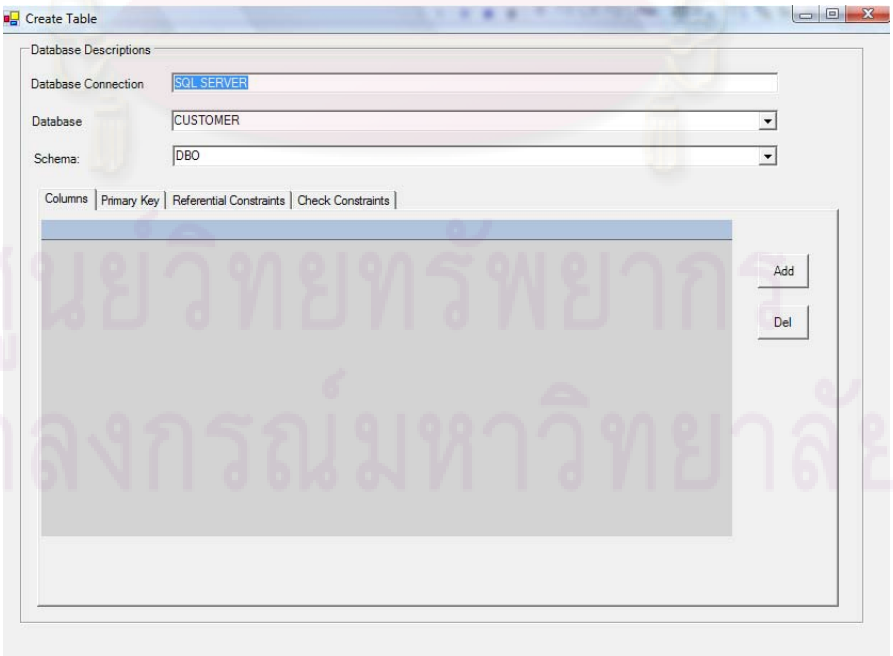
รูปที่ 4.6 หน้าจอเปิดสคีมาของฐานข้อมูล

5. หน้าจอลบสคีมาของฐานข้อมูล(DeleteSchema) มีลักษณะดังรูปที่ 4.7 หน้าจอนี้จะเหมือนกับหน้าจอเปิดสคีมาของฐานข้อมูล เพียงแต่จะเป็นการเลือกสคีมาของฐานข้อมูลเพื่อลบออกจากฐานข้อมูลของเครื่องมือ



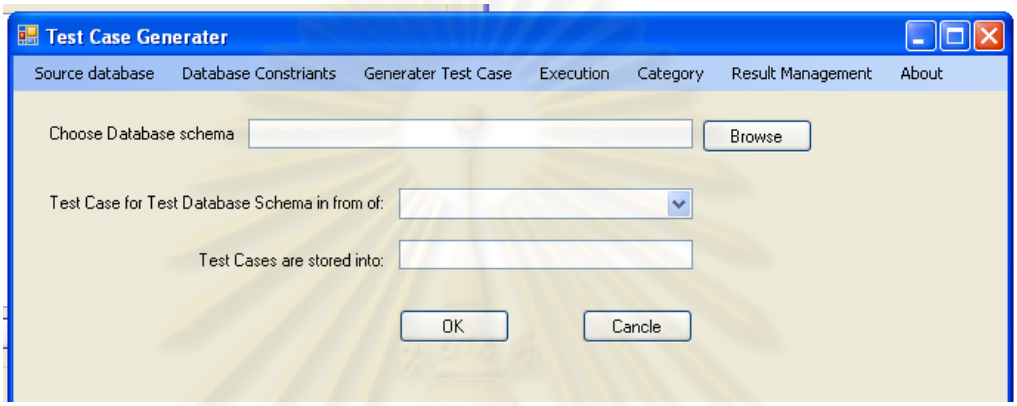
รูปที่ 4.7 หน้าจอลบสคีมาของฐานข้อมูล

6. หน้าจอรับข้อมูลเงื่อนไขบังคับของฐานข้อมูล (DatabaseConstraints) มีลักษณะดังรูปที่ 4.8 หน้าจอจะรับข้อมูลชื่อของสคีมาของฐานข้อมูล ประเภทของฐานข้อมูล และชื่อของฐานข้อมูล นอกจากนี้มีส่วนของการระบุเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ ได้แก่ ชื่อของตารางข้อมูล เขตข้อมูล คุณสมบัติต่างๆ ของเงื่อนไขบังคับของฐานข้อมูล ซึ่งประกอบด้วยเงื่อนไขบังคับบูรณภาพของเอนทิตี เงื่อนไขบังคับบูรณภาพของการอ้างอิงถึง และเงื่อนไขบังคับของโดเมน



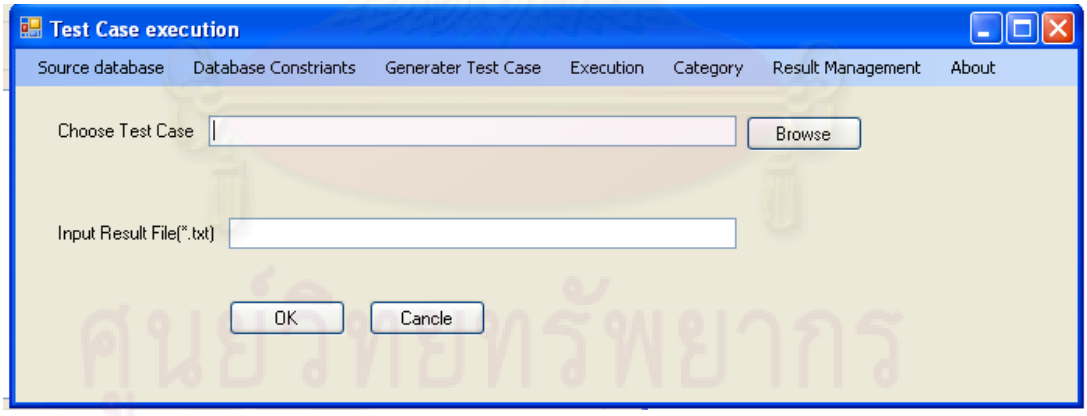
รูปที่ 4.8 หน้าจอการรับข้อมูลเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์

7. หน้าจอการสร้างกรณีทดสอบ (GenerateTestCase) มีลักษณะดังรูปที่ 4.9 หน้าจอจะรับข้อมูลของเงื่อนไขบังคับของฐานข้อมูล โดยการเลือกเอกสารที่ต้องการสร้างกรณีทดสอบ พร้อมทั้งระบุรูปแบบของคำสั่งเอสคิวแอลที่ต้องการสร้าง และระบุชื่อของที่จัดเก็บกรณีทดสอบ



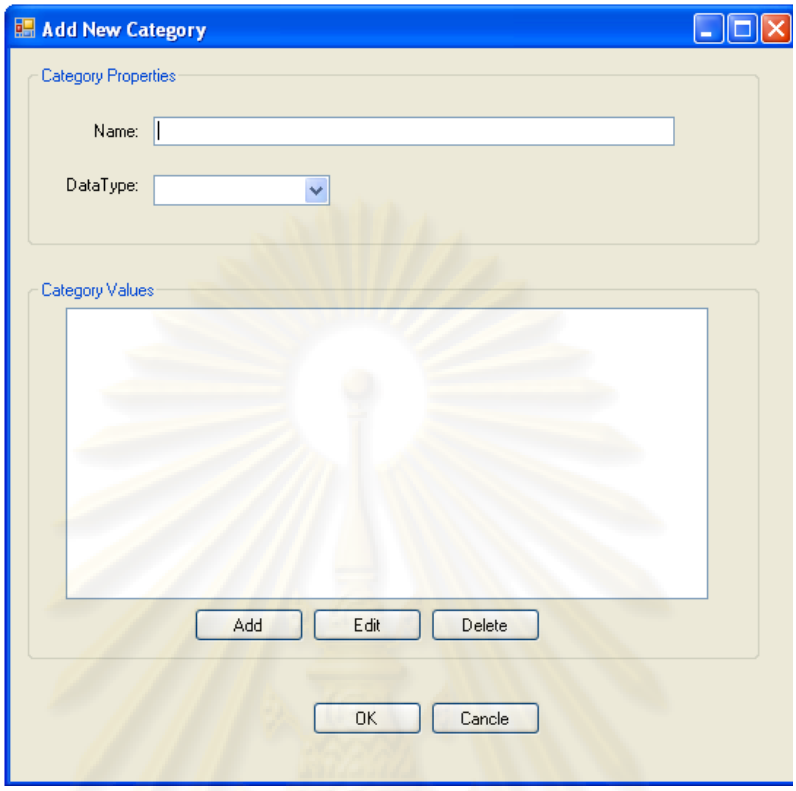
รูปที่ 4.9 หน้าจอการสร้างกรณีทดสอบ

6. หน้าจอการประมวลผลกรณีทดสอบ (ExecuteTestCase) มีลักษณะดังรูปที่ 4.10 หน้าจอจะทำการประมวลผลกรณีทดสอบที่สร้างขึ้น โดยการระบุกรณีทดสอบที่จะทำการประมวลผล และฐานข้อมูลที่ต้องการเชื่อมต่อ



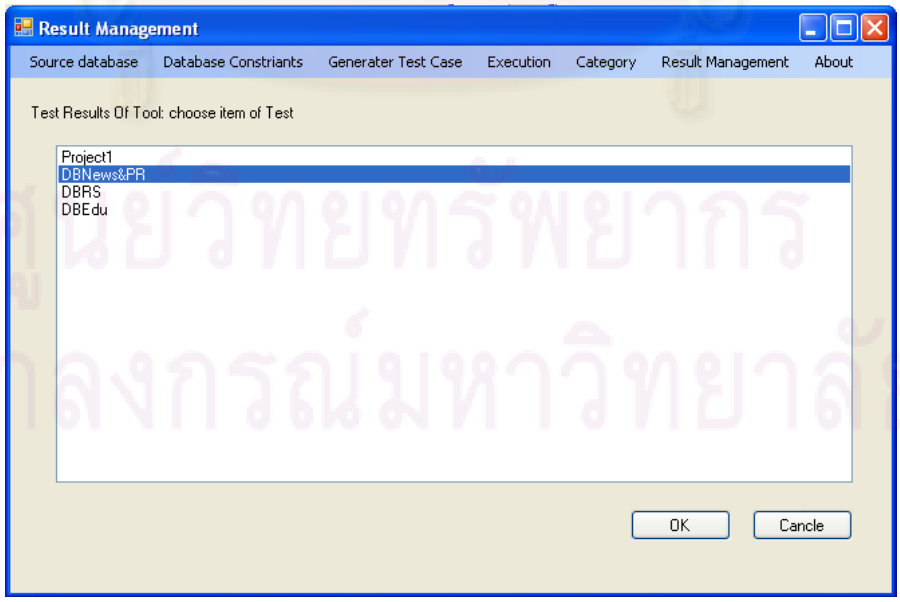
รูปที่ 4.10 หน้าจอการประมวลผลกรณีทดสอบ

7. หน้าจอกำหนดและระบุกลุ่มค่าข้อมูลทั่วไปเพิ่มเติม(Category) มีลักษณะดังรูปที่ 4.11 หน้าจอจะทำการรับข้อมูลที่ต้องการกำหนดข้อมูลกลุ่มค่าทั่วไป โดยสร้างขึ้นเพิ่มเติมและกำหนดจากเอกสารที่มีอยู่เดิม



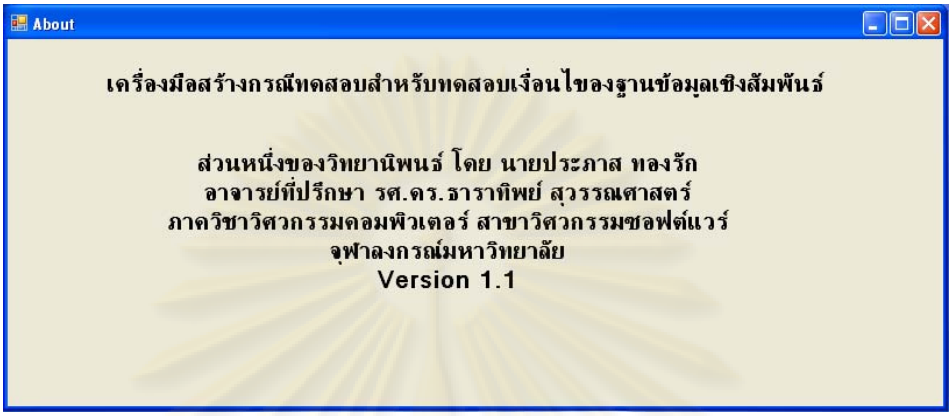
รูปที่ 4.11 หน้าจอการกำหนดหรือระบุกลุ่มค่าทั่วไป

8. หน้าจอผลลัพธ์ของการทดสอบ (ResultManage) มีลักษณะดังรูปที่ 4.12 หน้าจอจะทำการแสดงผลลัพธ์ของการทดสอบจากการประมวลผลกรณีทดสอบ โดยแสดงชื่อของกรณีทดสอบที่ประมวลผล วันเวลา และเวอร์ชันของการประมวลผลกรณีทดสอบ



รูปที่ 4.12 หน้าจอการจัดการผลลัพธ์การทดสอบ

9. หน้าจอเกี่ยวกับเครื่องมือ(About) หน้าจอจะแสดงรายละเอียดเกี่ยวกับเครื่องมือทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ แสดงดังรูปที่ 4.13



รูปที่ 4.13 หน้าจอเกี่ยวกับเครื่องมือ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การทดสอบเครื่องมือ

5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือมีรายละเอียดดังนี้

1.ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์แบบพีซี หน่วยประมวลผลอินเทลเพนเทียมโฟร์ 3.0 กิกะเฮิร์ทซ์
- หน่วยความจำสำรอง 1 กิกะไบต์ฮาร์ดดิสก์ 120 กิกะไบต์

2. ซอฟต์แวร์

- ระบบปฏิบัติการไมโครซอฟท์วินโดวส์เอ็กซ์พี
- ไมโครซอฟท์วิชวล์สตูดิโอ 2005 (Microsoft Visual Studio 2005)
- ฐานข้อมูลมายเอสคิวแอล เวอร์ชัน 4.0.18
- ฐานข้อมูลออราเคิล เวอร์ชัน 9i
- ฐานข้อมูลไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ 2000
- ฐานข้อมูลไมโครซอฟท์แอ็คเซส 2003

5.2 การทดสอบเครื่องมือ

การทดสอบเครื่องมือนี้เป็นการทดสอบ เพื่อตรวจสอบว่าเครื่องมือทดสอบสามารถสร้างกรณีทดสอบตามเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ ในระบบงานที่ใช้เป็นกรณีทดสอบได้ถูกต้องหรือไม่ ระบบงานที่ใช้ทดสอบเครื่องมือมี 3 ระบบงาน คือ คือ ระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์ [11] ระบบการจัดการการศึกษาคณะวิทยาศาสตร์ [12] และระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี [13] โดยรายละเอียดของการทดสอบจะยกตัวอย่างมาทำการอธิบายเพียง 1 ระบบ (รายะเอียดของการทดสอบแต่ละระบบงาน สามารถดูได้ที่ ภาคผนวก ข) ดังนี้

1.<u>กำหนดระบบงานที่ใช้ทดสอบเครื่องมือ</u> ระบบการจัดการการศึกษาคณะวิทยาศาสตร์ เป็นระบบที่เกี่ยวข้องกับการบริหารการการเรียนการสอนภายในคณะ ได้แก่ งาน

ทะเบียนบุคลากร งานจัดการตารางเรียนตารางสอน เป็นต้น ดังนั้นข้อมูลที่ใช้งานในระบบจะเป็นข้อมูลเกี่ยวกับนักศึกษา ข้อมูลคณาจารย์ ข้อมูลห้องเรียน ข้อมูลการสอน

2. การดำเนินการทดสอบ จะประกอบด้วยขั้นตอนดังนี้

2.1 ทำการศึกษาระบบที่จะทำการทดสอบเกี่ยวกับเงื่อนไขบังคับของฐานข้อมูลของระบบงานที่เป็นกรณีทดสอบ จากเอกสารข้อกำหนดความต้องการการออกแบบฐานข้อมูลของระบบและจากสคีมาของฐานข้อมูล โดยสคีมาของฐานข้อมูลของระบบดังกล่าวแสดงในตารางที่ 5.1

ตารางที่ 5.1 รายละเอียดของสคีมาของฐานข้อมูลระบบการจัดการการศึกษาคณะวิทยาศาสตร์

ชื่อตาราง	จำนวนของเซตข้อมูล	จำนวนเงื่อนไขบังคับของเอนทิตี	จำนวนเงื่อนไขบังคับของการอ้างอิงถึง	จำนวนเงื่อนไขบังคับของโดเมน
ข้อมูลคณาจารย์ (Teacher)	10	6	2	10
ตำแหน่ง (Position)	2	4	-	2
คณะ (Faculty)	3	4	-	3
สาขาวิชา (Major)	3	4	1	3
ระดับ (Level)	3	4	-	3
วิชาเรียน (Subject)	5	4	-	5
กลุ่มเรียน(Group)	2	4	-	2
หลักสูตร (Course)	3	4	-	3
วันหยุด (Holiday)	3	4	-	3
กำหนดการ(Schedule)	15	4	4	15

2.2 เปิดใช้เครื่องสร้างกรณีทดสอบ

2.3 สร้างสคีมาของฐานข้อมูลตามข้อมูลเงื่อนไขของแต่ละตารางข้อมูลที่ได้ทำการศึกษาจากระบบงาน ดังรูปที่ 5.1 ซึ่งเป็นตัวอย่างหนึ่งของการสร้างสคีมา

จากตัวอย่างสคีมาของฐานข้อมูล มีรายละเอียดของชื่อตารางคณาจารย์ (teacher) ประกอบด้วยเขตข้อมูลและรายละเอียดเงื่อนไข ดังนี้

- Id คือ รหัสของอาจารย์ (ต้องมีค่าไม่ซ้ำ และไม่เป็นค่าว่าง)

```
SQL> create table teacher
        (id  varchar2(10), prefix varchar2(10),name varchar2(30) not null,
        surname varchar2(30) not null, engprefix varchar2(10),engname varchar2(30) not null,
        engsurname varchar2(30) not null, address vachar2(100), gender varchar2(5),salary number
            Constraint  TechPk    primary key (id))
            Constraint  cv_sal CHECK  (5000<salary < 30000)
```

รูปที่ 5.1 ตัวอย่างของสคีมาของฐานข้อมูล

- Prefix คือ คำนำหน้าชื่ออาจารย์
- Name คือ ชื่ออาจารย์ (ไม่เป็นค่าว่าง)
- Surname คือนามสกุลของอาจารย์ (ไม่เป็นค่าว่าง)
- Engprefix คือคำนำหน้าชื่อภาษาอังกฤษ
- Engname คือ ชื่อภาษาอังกฤษของอาจารย์ (ไม่เป็นค่าว่าง)
- Engsurname คือ นามสกุลภาษาอังกฤษของอาจารย์ (ไม่เป็นค่าว่าง)
- Address คือที่อยู่ของอาจารย์
- Gender คือ เพศ
- Salary คือเงินเดือน (เงินเดือนอยู่ในช่วงระหว่าง 5,000 ถึง 30,000 บาท)

2.4 สร้างกลุ่มค่าทั่วไปเพิ่มเติมเพื่อให้ข้อมูลทดสอบที่สร้างออกมามีความใกล้เคียงกับฐานข้อมูลที่ทำการทดสอบ เช่น ตัวอย่างของชื่อของคณาจารย์ ส่วนนี้สุ่มจากฐานข้อมูลของเครื่องมือ

2.5 สร้างกรณีทดสอบตามเงื่อนไขบังคับบังคับฐานข้อมูล ซึ่งประกอบด้วยเงื่อนไขบังคับบูรณภาพของเอนทิตี เงื่อนไขบังคับบูรณภาพของการอ้างอิงถึง และเงื่อนไขบังคับของโดเมน จากตัวอย่าง(รูปที่ 5.2) จะแสดงให้เห็นกรณีทดสอบที่ถูกสร้างขึ้นเพื่อทำการทดสอบเงื่อนไขบังคับ

ของโดเมน โดยตรวจสอบว่าค่าของเงินเดือนอยู่ในช่วงที่มีการกำหนดขึ้นหรือไม่ กรณีทดสอบที่สร้างขึ้น มี 3 ลักษณะ คือ

- ช่วงของเงินเดือนอยู่ในช่วงต่ำกว่าเงินเดือน (Salary<5,000) ผลที่คาดหวังคือไม่สามารถเพิ่มข้อมูลได้
- ช่วงของเงินเดือนอยู่ในช่วงของเงินเดือน (5,000<Salary<30,000) ผลที่คาดหวังคือสามารถเพิ่มข้อมูลได้
- ช่วงของเงินเดือนอยู่นอกของช่วงเงินเดือนที่กำหนด (Salary>30,000) ผลที่คาดหวังคือไม่สามารถเพิ่มข้อมูลได้

```
SQL>  insert into teacher  values ('T001','นาย','วัชรชัย','เจริญชัย','Mr.','Wacharachai',
          'Charoenchai','bangkok','Male',4999);
      insert into teacher  values ('T002','นาย','วัชรกร','แสนศักดิ์','Mr.','Wacharakorn','Sansak','bangkok',
          'Male',5001);
      insert into teacher  values ('T003','นาย','ธีระ','เจริญยิ่ง','Mr.','Teera','Charoenying','bangkok',
          'Male',30001);
```

รูปที่ 5.2 ตัวอย่างของกรณีทดสอบที่สร้างขึ้น

2.6 แก้ไขสคีมาของฐานข้อมูล โดยการเพิ่มข้อผิดพลาดเข้าไปยังสคีมาของฐานข้อมูลของระบบที่ต้องการทดสอบ เพื่อตรวจสอบว่ากรณีทดสอบที่สร้างขึ้นนั้นสามารถตรวจจับข้อผิดพลาดที่จะเกิดขึ้นได้หรือไม่ ในกรณีนี้ทำการแก้ไขสคีมาในส่วนของเงื่อนไขบังคับของโดเมน เมื่อทำการประมวลผลกรณีทดสอบแล้ว ผลการทดสอบไม่เป็นไปตามค่าคาดหวัง แสดงว่ากรณีทดสอบนั้นไม่เป็นไปตามเงื่อนไขที่กำหนดไว้ตอนต้น สคีมาของฐานข้อมูลจากรูปที่ 5.1 ทำการแก้ไขเงื่อนไขบังคับของโดเมน ดังรูปที่ 5.3

```
SQL> create table teacher
      (id  varchar2(10), prefix varchar2(10),name varchar2(30) not null,
      surname varchar2(30) not null, engprefix varchar2(10),engname varchar2(30) not null,
      engsurname varchar2(30) not null, address vachar2(100), gender varchar2(5),salary number
          Constraint  TechPk    primary key (id))
```

รูปที่ 5.3 ตัวอย่างของสคีมาของฐานข้อมูลที่มีการแก้ไขแล้ว

2.7 วิเคราะห์และสรุปผลการทดลอง เมื่อทดสอบกรณีทดสอบกับฐานข้อมูลที่ต้องการทดสอบแล้ว สำหรับกรณีที่ผลการทดสอบทีไม่เป็นไปตามค่าคาดหวัง จะมีการทำการวิเคราะห์หาสาเหตุของข้อผิดพลาดที่เกิดขึ้นจากการทดสอบ แล้วสรุปผลการทดสอบของแต่ละระบบ

3. ผลการทดสอบ เครื่องมือทดสอบสามารถตรวจจับข้อผิดพลาดที่เพิ่มเข้าไปในระบบงานได้ และสามารถทำการปรับปรุงเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ให้เป็นไปตามที่ระบุได้

4. สรุปผลการทดสอบ จากผลการทดสอบเครื่องมือสร้างกรณีทดสอบเพื่อทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ให้กับระบบดังกล่าวมา สามารถสรุปได้ว่าเครื่องมือสามารถสร้างกรณีทดสอบเพื่อตรวจสอบความถูกต้องของเงื่อนไขบังคับของฐานข้อมูล พร้อมทั้งมีการปรับปรุงเงื่อนไขของฐานข้อมูลที่ทำการทดสอบด้วย

5.3 ผลการทดสอบเครื่องมือกับระบบงานจริง

การทดสอบนี้เป็นการทดสอบ เพื่อตรวจสอบว่าเครื่องมือสามารถสร้างกรณีทดสอบตามเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ โดยผลการทดสอบระบบดังตารางที่ 5.2 ทั้งนี้จากการทดสอบแสดงให้เห็นว่าเครื่องมือสามารถสร้างกรณีทดสอบที่ครอบคลุมเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์

สำหรับข้อผิดพลาดที่พบหลังจากการแก้ไขสคีมาของฐานข้อมูล จากการทดลองพบว่าจำนวนของข้อผิดพลาดที่เพิ่มเข้าไปในระหว่างการทดลอง และจำนวนข้อผิดพลาดที่พบหลังจากการทดสอบไม่เท่ากัน ทั้งนี้จากการวิเคราะห์ผลการทดลองพบว่าสำหรับกรณีทดสอบบางกรณียอมให้สามารถดำเนินการเพิ่ม ลบ และแก้ไขข้อมูลได้ ตัวอย่างเช่น กรณีทดสอบที่สร้างขึ้นเพื่อตรวจสอบเงื่อนไขบังคับของโดเมนของตารางกลุ่มเรียน ในเขตข้อมูลของจำนวนกลุ่มเรียนที่มีชนิดข้อมูลเป็นตัวเลข จากนั้นแก้ไขสคีมาของฐานข้อมูลเป็นตัวอักขระ เมื่อทำการประมวลผลการทดสอบอีกครั้งพบว่าฐานข้อมูลสามารถทำการเพิ่มข้อมูลลงในฐานข้อมูลได้ เป็นต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5.2 สรุปผลการสอบเครื่องมือทดสอบโดยการเพิ่มข้อผิดพลาดในสคีมาของระบบที่นำมาทดสอบ

ระบบ	จำนวนกรณีทดสอบ	ประเภทของระบบฐานข้อมูล	จำนวนข้อผิดพลาดที่เพิ่ม	จำนวนข้อผิดพลาดที่พบ
1	170	ออราเคิล	220	215
2	111	มายเอสคิวแอล	132	129
3	182	ไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ 2000	315	315

หมายเหตุ ระบบที่ 1 ระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์
ระบบที่ 2 ระบบการจัดการการศึกษาคณะวิทยาศาสตร์
ระบบที่ 3 ระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

สรุปผลการวิจัย

จากการศึกษา ออกแบบและพัฒนาเครื่องสร้างกรณีทดสอบ สามารถสรุปผลการวิจัย ประโยชน์ที่ได้รับ ข้อจำกัดของเครื่องมือ และแนวทางในการพัฒนาต่อไปได้ดังนี้

6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้พัฒนาเครื่องมือสร้างกรณีทดสอบเพื่อทดสอบเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ โดยเงื่อนไขบังคับของของฐานข้อมูลเชิงสัมพันธ์จะต้องเป็นไปตามข้อกำหนดการออกแบบฐานข้อมูลที่ผู้ออกแบบฐานข้อมูลเป็นผู้กำหนดขึ้น เครื่องมือสร้างกรณีทดสอบสามารถรับข้อมูลเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ได้โดยตรงจากผู้ใช้เป็นผู้กำหนด โดยจะต้องทำการระบุตารางข้อมูล เขตของข้อมูล และเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ ได้แก่ เงื่อนไขบังคับบูรณภาพของเอนทิตี เงื่อนไขบังคับบูรณภาพของการอ้างอิงถึง และเงื่อนไขบังคับของโดเมน นอกจากนี้ยังสามารถกำหนดให้เครื่องมืออ่านสคีมาของฐานข้อมูลที่มีอยู่แล้วจากฐานข้อมูลที่ต้องการทดสอบ ในการสร้างกรณีทดสอบผู้ใช้สามารถกำหนดรูปแบบของคำสั่งเอสคิวแอล ได้แก่ คำสั่งการเพิ่มข้อมูล การลบข้อมูล และการปรับปรุงข้อมูลได้ ทั้งนี้กรณีทดสอบจะประกอบด้วยรูปแบบของคำสั่งเอสคิวแอลที่เป็นมาตรฐาน สำหรับวิทยานิพนธ์นี้ได้พัฒนาเครื่องมือซึ่งสนับสนุนระบบการจัดการฐานข้อมูล 3 ชนิด ได้แก่ ฐานข้อมูลออราเคิล ฐานข้อมูลมายเอสคิวแอล และฐานข้อมูลไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ 2000 นอกจากกรณีทดสอบที่สร้างขึ้นแล้วยังสร้างข้อมูลทดสอบ โดยเครื่องมือจะทำการสุ่มจากกลุ่มค่าทั่วไปโดยอัตโนมัติ ทั้งนี้กรณีทดสอบจะถูกทดสอบ เพื่อตรวจสอบเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ที่ต้องการทดสอบ เครื่องมือทดสอบสามารถบันทึกกรณีทดสอบที่สร้างขึ้น รวมไปถึงผลลัพธ์ที่ได้จากการประมวลผลการทดสอบ สำหรับกรณีที่เงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ไม่เป็นไปตามข้อบังคับการออกแบบฐานข้อมูล เครื่องมือจะทำการปรับปรุงเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ให้โดยอัตโนมัติ

ในส่วนของการทดสอบเครื่องสร้างกรณีทดสอบที่พัฒนาขึ้นนี้ ผู้ทดสอบได้ทำการทดสอบสร้างกรณีทดสอบให้กับระบบงานจริง 3 ระบบงาน คือ ระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์ ระบบการจัดการการศึกษาคณะวิทยาศาสตร์ และระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี โดยผู้ทดสอบได้ทำการสร้างกรณีทดสอบจากระบบจริงขึ้น จากนั้นทำการเพิ่มข้อผิดพลาดไปยังสคีมาของฐานข้อมูลที่ต้องการ

ทดสอบ เพื่อตรวจสอบว่ากรณีทดสอบที่สร้างขึ้นนั้นสามารถตรวจจับข้อผิดพลาดซึ่งเกิดจากการเพิ่มข้อผิดพลาดได้หรือไม่ ซึ่งผลการทดสอบแสดงให้เห็นว่าเครื่องมือสร้างกรณีทดสอบสามารถสร้างกรณีทดสอบที่ครอบคลุมเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ได้ นอกจากนี้เครื่องมือทดสอบยังสามารถทำการปรับปรุงแก้ไขเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ให้กับฐานข้อมูลที่นำมาทดสอบ โดยไม่ได้ทำการเพิ่มเติมสคีมาของฐานข้อมูล แต่ปรับปรุงจากสคีมาเดิมให้เป็นไปตามข้อบังคับการออกแบบฐานข้อมูล

6.2 ข้อจำกัดของเครื่องมือสร้างกรณีทดสอบ

1. เครื่องมือสร้างกรณีทดสอบเพื่อทดสอบเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์สามารถติดต่อกับฐานข้อมูลได้เพียง 3 ชนิดเท่านั้น คือ ฐานข้อมูลมายเอสคิวแอล ฐานข้อมูลออราเคิล และฐานข้อมูลไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ 2000
2. เครื่องมือทดสอบสามารถสร้างกรณีทดสอบครอบคลุมเงื่อนไขบังคับของฐานข้อมูลเชิงสัมพันธ์ เพียง 3 เงื่อนไขเท่านั้น คือ เงื่อนไขบังคับบูรณภาพของเอนทิตี เงื่อนไขบังคับบูรณภาพการอ้างอิงถึง และเงื่อนไขบังคับของโดเมน
3. เครื่องมือทดสอบนี้สามารถสร้างกรณีทดสอบที่ครอบคลุมเงื่อนไขบังคับบูรณภาพการอ้างอิงของข้อมูลในลักษณะการตกเป็นทอดๆ เพียง 2 ระดับชั้นเท่านั้น
4. เครื่องมือทดสอบไม่ครอบคลุมการสร้างกรณีทดสอบที่มีการเพิ่มเติมภายหลังจากการสร้างกรณีทดสอบ รวมถึงทริกเกอร์ (Trigger) ของตารางข้อมูล

6.3 แนวทางในการพัฒนาต่อ

1. พัฒนาเครื่องสร้างกรณีทดสอบให้สามารถติดต่อกับฐานข้อมูลเชิงสัมพันธ์ได้มากกว่าที่เดิมมีอยู่ เช่น ดีบีทูยูนิเวอร์แซล หรือ ไซเบส เป็นต้น รวมไปถึงความสามารถในการติดต่อกับฐานข้อมูลเชิงวัตถุได้
2. เพิ่มความสามารถให้เครื่องมือสร้างกรณีทดสอบ สามารถติดต่อกับฐานข้อมูลที่ต้องการทดสอบได้โดยตรง เพื่อไม่ให้ผู้ใช้เครื่องมือต้องเสียเวลานำออกสคีมาของฐานข้อมูลผ่านเครื่องมืออื่น โดยข้อมูลที่นำเข้าเป็นเอกสารเอ็กซ์เอ็มแอล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. ปรับปรุงเครื่องมือสร้างกรณีทดสอบให้พิจารณาเงื่อนไขบังคับอื่นๆ ที่มีอยู่ในฐานข้อมูลเชิงสัมพันธ์ รวมไปถึงฐานข้อมูลเชิงวัตถุ เช่น เงื่อนไขบังคับบอกความหมาย (Semantic Constraints)

6.4 ผลงานที่เกี่ยวข้องกับการวิจัย

งานวิจัยนี้ได้รับคัดเลือกให้นำเสนอในงานประชุมวิชาการและตีพิมพ์ในเอกสาร “Proceedings of The second IEEE International Conference on Computer Science and Information Technology (ICCSIT2009)” ในระหว่างวันที่ 8-11 สิงหาคม 2552 ณ กรุงปักกิ่ง สาธารณรัฐประชาชนจีน โดยมีหัวข้อวิจัยชื่อ “A Tool for Generating Test Case from Relational Database Constraints Testing ” สำหรับผลงานที่ตีพิมพ์ สามารถดูได้ที่ภาคผนวก ง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] IEEE Computer Society. IEEE Standard Glossary of Software Engineering Terminology.1983. ANSI/IEEE Std 729-1983.

[2] Boris Beizer. Software Testing Techniques Second Edition. New York: Van Nostrand Reinhold, 1990.

[3] ชวลิต จีรทีปติสุนทร. XML Step by Step ฉบับภาษาไทย. กรุงเทพมหานคร: DSL, 2543.

[4] Roger L. Costello. XML Schema Tutorial [Online]. Avalilable from : http://www.xfront.com. [2007,September]

[5] Dan A. Simonvici and Richard L. Tenney. Relational Database Systems. USA: Academic Press, 1995.

[6] ยุทธนา ลีลาศวัฒนกุลม. สร้างระบบงานฐานข้อมูลด้วย PL/SQLและ ORACLE DEVELOPER. นนทบุรี: อินโพเพรสม, 2545.

[7] ภพพงศ์ สกุลพิพัฒน์ศิลป์. การพัฒนาเครื่องมือซอฟต์แวร์สำหรับสร้างข้อมูลทดสอบ (Development of a Software Tool for Generating Test Data). วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2544.

[8] กฤษณะ พิริยะกิจไพบูลย์. การพัฒนาเครื่องมือสร้างข้อมูลทดสอบเพื่อสนับสนุนการทดสอบซอฟต์แวร์จากสคีมาของฐานข้อมูล (DEVELOPMENT OF A TEST DATA GENERATING TOOL TO SUPPORT SOFTWARE TESTING FROM DATABASE SCHEMA. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2547.

[9] Ashutosh Gaur. DeZign for databases to create ER diagrams[Online]. Avalilable from: http://www.orafaq.com/tools/heraut/dezign.htm. [Feburary,2008]

[10] Altova T. Genert. The Altova DatabaseSpy 2008[Online], Avalilable from: http://www.altova.com. [March,2008]

[11] นราศักดิ์ สระทองแก่น และคณะ. ระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์. โครงงานวิจัยระดับปริญญาตรี, ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี, 2551.

[12] กาญจนา ศรีนาค และคณะ. ระบบการจัดการการศึกษาคณะวิทยาศาสตร์. โครงงานวิจัยระดับปริญญาตรี, ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี .

[13] ธวัชชัย มัทนัง และคณะ. ระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี. ปทุมธานี: แผนกเทคโนโลยีและสารสนเทศ, 2550.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

พจนานุกรมข้อมูลของฐานข้อมูลของเครื่องมือ

ภาคผนวกนี้แสดงพจนานุกรมข้อมูลของแต่ละตารางฐานข้อมูลของเครื่องมือ

ตารางที่ ก-1 พจนานุกรมของตาราง DatabaseSchema

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
DBS_Name	ชื่อสคีมาของฐานข้อมูล	Varchar	100	ไม่ว่าง	PK	Education
DBS_Desc	คำอธิบายสคีมาของฐานข้อมูล	Varchar	255	ค่าว่าง		The database schema of education afire of faculty of Science

ตารางที่ ก-2 พจนานุกรมของตาราง Table

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
Table_name	ชื่อตารางข้อมูล	Varchar	100	ไม่ว่าง	PK	Subject
Table_DBS	ชื่อสคีมาของฐานข้อมูล	Varchar	100	ไม่ว่าง	PK,FK	Education
Table_dep_mode	รูปแบบของการขึ้นต่อกัน	Varchar	10	ไม่ว่าง		Normal

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ก-3 พจนานุกรมของตาราง DatabaseConstraints

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
Field_name	ชื่อของเขตข้อมูล	Varchar	100	ไม่ว่าง	PK	Salary
Field_table	ชื่อตารางข้อมูล	Varchar	100	ไม่ว่าง	PK,FK	Teacher
Field_DBS	ชื่อสคีมาของฐานข้อมูล	Varchar	100	ไม่ว่าง	PK,FK	Education
Field_order	ลำดับของเขตข้อมูลในตาราง	int		ไม่ว่าง		10
Field_Mod_size	ขนาดของเขตข้อมูล	int		ค่าว่าง		4
Field_Mod_Scale	ขนาดของทศนิยมของเขตข้อมูล	int		ค่าว่าง		0
Field_Opt_flag	ค่าชี้ว่าเป็นคีย์หลักของตารางหรือไม่	int		ไม่ว่าง		
Field_Gen	วิธีการสร้างกรณีทดสอบ	int		ค่าว่าง		1
Field_from_value	ค่าเริ่มต้นของช่วงที่ใช้	Varchar	30	ไม่ว่าง		500
Field_to_value	ค่าสุดท้ายของช่วงที่ใช้	Varchar	30	ไม่ว่าง		30000
Field__constValle	ค่าคงที่ที่ใช้เป็นค่าข้อมูลทดสอบ	Varchar	255	ไม่ว่าง		1000
Field_category	กลุ่มค่าทั่วไปที่ใช้สร้าง	Varchar	200	ไม่ว่าง		
Field_ref_name	ชื่อเขตข้อมูลที่อ้างอิง	Varchar	100	ไม่ว่าง	FK	Salary

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ก-3 พจนานุกรมของตาราง DatabaseConstraints (ต่อ)

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
Field_ref_table	ชื่อตารางข้อมูล	Varchar	100	ไม่ว่าง	FK	Teacher
Field_ref_DBS	ชื่อของสคีมาของฐานข้อมูล	Varchar	100	ไม่ว่าง	FK	Education
Field_type	ชนิดข้อมูล	Varchar	20		FK	Integer

ตารางที่ ก-4 พจนานุกรมของตาราง FieldType

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
Field_type_name	ชื่อชนิดของเขตข้อมูล	Varchar	20	ไม่ว่าง	PK	Integer
Field_type_size	ขนาดของเขตข้อมูล	Int		ไม่ว่าง		1
Field_type_scale	ขนาดทศนิยมของเขตข้อมูล	Int		ไม่ว่าง		0

ตารางที่ ก-5 พจนานุกรมของตาราง NormalDependency

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
Dep_table	ชื่อตารางข้อมูล	varchar	100	ไม่ว่าง	PK,FK	Subject
Dep_DBS	ชื่อสคีมาของฐานข้อมูล	varchar	100	ไม่ว่าง	PK,FK	Education
Dep_src_field	เขตต้นทางปลายทาง	varchar	100	ไม่ว่าง	PK	Prefix
Dep_dest_field	เขตข้อมูลปลายทาง	varchar	100	ไม่ว่าง	PK	Gender
Condition_value	ค่าเงื่อนไข	varchar	100	ไม่ว่าง	PK	Mr.
Result_value	ค่าผลลัพธ์	varchar	100	ไม่ว่าง		Male

ตารางที่ ก-6 พจนานุกรมของตาราง Category

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
Catename	ชื่อกลุ่มค่าทั่วไป	Varchar	200	ไม่ว่าง	PK	FirstName
Catedatatype	ชนิดของข้อมูล	Varchar	10	ไม่ว่าง		Character
maxchar	ขนาดของค่าข้อมูลที่ยาวที่สุดในกลุ่มค่าทั่วไป	int		ไม่ว่าง		9

ตารางที่ ก-7 พจนานุกรมของตาราง CategoryValue

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
Catename	ชื่อกลุ่มค่าทั่วไป	Varchar	200	ไม่ว่าง	PK,FK	FirstName
CateValue	ค่าข้อมูล	Varchar	255	ไม่ว่าง	PK	Somchai

ตารางที่ ก-8 พจนานุกรมของตาราง TestCase

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
TCGID	เลขที่กรณีทดสอบ	Int		ไม่ว่าง	PK	100
TCGNAME	ชื่อกรณีทดสอบ	Varchar	100	ไม่ว่าง		TestProject1
TCGDesc	รายละเอียด	Varchar	255	ไม่ว่าง		Project Test of Education Database

ตารางที่ ก-9 พจนานุกรมของตาราง TestResult

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ	ตัวอย่างข้อมูล
TSID	เลขที่ของผลลัพธ์	Int		ไม่ว่าง	PK,FK	101
TSNAME	ชื่อของผลการทดสอบ	Varchar	100	ไม่ว่าง		TSDBNews
TSDATE	วันที่ทำการทดสอบ	Date/time		ไม่ว่าง		1/12/2009
TSVersion	เวอร์ชันการทดสอบ	Varchar	100	ไม่ว่าง		1.1
TSDesc	รายละเอียด	Varchar	255	ค่าว่าง		Test Result of DBNews

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

ตัวอย่างผลการทดสอบเครื่องมือสร้างกรณีทดสอบกับระบบงานจริง

ภาคผนวกนี้แสดงตัวอย่างผลการทดสอบเครื่องมือสร้างกรณีทดสอบกับระบบงานจริง ได้แก่ ระบบการจัดการการศึกษาคณะวิทยาศาสตร์ ระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์ และระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี โดยรายละเอียดของการทดสอบเครื่องมือเป็นดังนี้

ระบบการจัดการการศึกษาคณะวิทยาศาสตร์

ระบบการจัดการการศึกษาคณะวิทยาศาสตร์เป็นระบบที่เกี่ยวข้องกับการบริหารจัดการข้อมูลทางด้านการศึกษาการเรียนการสอนของคณะวิทยาศาสตร์ ทั้งนี้ข้อมูลที่เกี่ยวข้องกับการทดสอบเป็นข้อมูลคณาจารย์ ข้อมูลตารางสอน ข้อมูลห้องเรียน และข้อมูลของรายวิชาที่สอน สคีมาของระบบฐานข้อมูลประกอบด้วยตารางข้อมูลทั้งหมด 10 ตารางข้อมูล ดังนี้

ตารางที่ ข-1 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางคณาจารย์

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
TID	รหัสคณาจารย์	Int		ไม่ว่าง	PK
T_Prefix	คำนำหน้าชื่อ	Varchar	10		
T_Name	ชื่อคณาจารย์	Varchar	100		
T_Surname	นามสกุลคณาจารย์	Varchar	100		
EngPrefix	คำนำหน้าชื่อภาษาอังกฤษ	Varchar	10		
Engname	ชื่อภาษาอังกฤษ	Varchar	100		
Engsurname	นามสกุลภาษาอังกฤษ	Varchar	100		
Address	ที่อยู่	Varchar	255		
Gender	เพศ	Varchar	50		Male or Female
Salary	เงินเดือน	Int			7000<Salary<50000
F_ID	รหัสอ้างอิงคณะ	Int		ไม่ว่าง	FK
P_ID	รหัสอ้างอิงตำแหน่ง	Int		ไม่ว่าง	FK

ตารางที่ ข-2 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางตำแหน่ง

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
P_ID	รหัสตำแหน่ง	Int		ไม่ว่าง	PK
P_Name	ชื่อตำแหน่ง	Varchar	10		

ตารางที่ ข-3 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางคณะ

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
F_ID	รหัสคณะ	Int		ไม่ว่าง	PK
F_Name	ชื่อคณะภาษาไทย	Varchar	100		
F_EngName	ชื่อคณะภาษาอังกฤษ	Varchar	100		

ตารางที่ ข-4 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางสาขาวิชา

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
M_ID	รหัสสาขาวิชา	Int		ไม่ว่าง	PK
M_Name	ชื่อสาขาภาษาไทย	Varchar	100		
M_EngName	ชื่อสาขาภาษาอังกฤษ	Varchar	100		
F_ID	รหัสอ้างอิงคณะ	Int		ไม่ว่าง	FK

ตารางที่ ข-5 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางระดับ

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
L_ID	รหัสระดับการศึกษา	Int		ไม่ว่าง	PK
L_Name	ชื่อระดับการศึกษา	Varchar	100		
L_Desc	หมายเหตุเพิ่มเติม	Varchar	100		

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-6 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางวิชาเรียน

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
S_ID	รหัสสาขาวิชา	Int		ไม่ว่าง	PK
S_Name	ชื่อสาขาภาษาไทย	Varchar	100		
S_EngName	ชื่อสาขาภาษาอังกฤษ	Varchar	100		
S_Desc	รายละเอียดวิชาเรียน	Varchar	255		

ตารางที่ ข-7 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางกลุ่ม

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
G_ID	รหัสกลุ่มเรียน	Int		ไม่ว่าง	PK
G_Count	จำนวนของกลุ่มเรียน	Int			

ตารางที่ ข-8 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางหลักสูตร

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
C_ID	รหัสหลักสูตร	Int		ไม่ว่าง	PK
C_Name	ชื่อหลักสูตร	Varchar	100		
C_Desc	รายละเอียดหลักสูตร ภาษาอังกฤษ	Varchar	255		

ตารางที่ ข-9 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางวันหยุด

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
H_ID	รหัสวันหยุด	Int		ไม่ว่าง	PK
H_Name	ชื่อช่วงวันหยุด	Varchar	100		
H_Count	จำนวนวันลา	Int			

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-10 กรณีศึกษาระบบการจัดการการศึกษาคณะวิทยาศาสตร์: ตารางกำหนดการ

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
SC_ID	รหัสกำหนดการ	Int		ไม่ว่าง	PK
SC_Name	รายการกำหนดการ	Varchar	100	ไม่ว่าง	
SC_Desc	รายละเอียดเพิ่มเติม	Varchar	255		
T_ID	รหัสอ้างอิงคณาจารย์	Int		ไม่ว่าง	FK
S_ID	รหัสอ้างอิงวิชาเรียน	Int		ไม่ว่าง	FK
G_ID	รหัสอ้างอิงกลุ่มเรียน	Int		ไม่ว่าง	FK
C_ID	รหัสอ้างอิงหลักสูตร	Int		ไม่ว่าง	FK

ตัวอย่างของกรณีทดสอบ

```
SQL>    insert into position values(10,'ASSIT');
        insert into position values(10,'DR.');
        insert into position values(11,DR.);
        insert into position values(12,'PROFESSIONAL');
        insert into position values(13,'PROFESSIONAL1');

        insert into faculty values(20,'วิทยาศาสตร์','SC');
        insert into faculty values(20,'วิศวกรรมศาสตร์','ENG');
        insert into faculty values(21,'วิศวกรรมศาสตร์','ENG');
        insert into faculty values(22,'วิศวกรรมศาสตร์1','ENG1');
        insert into faculty values(23,'วิศวกรรมศาสตร์2','ENG2');
        insert into faculty values(24,'วิศวกรรมศาสตร์3','ENG3');

        insert into major values(30,'เคมี','CHEM',1);
        insert into major values(30,'คอมพิวเตอร์','COM',1);
        insert into major values(31,'คณิตศาสตร์','MATH',1);
        insert into major values(32,'คณิตศาสตร์1','MATH1',1);
        insert into major values(33,'คณิตศาสตร์2','MATH2',1);
```

รูปที่ ข-1 แสดงตัวอย่างของกรณีทดสอบของระบบการจัดการศึกษาคณะวิทยาศาสตร์

ตัวอย่างของกรณีทดสอบ(ต่อ)

```
SQL>    insert into faculty values(300,'เทคโนโลยีอุตสาหกรรม','INDUSTRIAL
        TECHNOLOGY');
        insert into major values(32,'ชีววิทยา','Biology',300);
        delete from faculty where F_ID='300';

        insert into level values(40,'MasterDegree','มหาบัณฑิต');
        insert into level values(40,'Ph.d','ดุษฎีบัณฑิต');
        insert into level values(41,'Ph.d','ดุษฎีบัณฑิต');
        insert into level values(42,'MasterDegree','มหาบัณฑิต');
        insert into level values(43,'MasterDegree1','มหาบัณฑิต1');

        insert into subject values(50,'แคลคลูลัส1','Calculas1','-');
        insert into subject values(50,'แคลคลูลัส2','Calculas2','-');
        insert into subject values(51,'แคลคลูลัส2','Calculas2','-');
        insert into subject values(52,'แคลคลูลัส3','Calculas3','-');
        insert into subject values(53,'แคลคลูลัส4','Calculas4','-');
        insert into subject values(54,'แคลคลูลัส5','Calculas5','-');

        insert into group values(60,30);
        insert into group values(60,31);
        insert into group values(61,31);
        insert into group values(62,32);

        insert into teacher  values (‘T001’,’นาย’,’วัชรชัย’,’เจริญ
        ชัย’,’Mr.’,’Wacharachai’, ’Charoenchai’,’bangkok’,‘Male’,4999);
        insert into teacher  values (‘T002’,’นาย’,’วัชรกร’,’แสน
        ศักดิ์’,’Mr.’,’Wacharakorn’, ’Sansak’,’bangkok’,‘Male’,5001);
        insert into teacher  values (‘T003’,’นาย’,’ธีระ’,’เจริญยิ่ง’,’Mr.’,’Teera’,
        ’Charoenying’,’bangkok’,‘Male’,30001);
```

รูปที่ ข-2 แสดงตัวอย่างของกรณีทดสอบของระบบการจัดการศึกษาคณะวิทยาศาสตร์(ต่อ)

ระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์

ระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์ เป็นระบบที่เกี่ยวข้องกับการบริหารจัดการข้อมูลการจองการใช้ห้องปฏิบัติการคณะวิทยาศาสตร์ ทั้งนี้ข้อมูลที่เกี่ยวข้องกับการทดสอบเป็นข้อมูลนักศึกษา ข้อมูลคณาจารย์ ข้อมูลตารางการใช้ห้องปฏิบัติคณะวิทยาศาสตร์ ข้อมูลห้องเรียน ตัวอย่างสคีมาของระบบฐานข้อมูลประกอบด้วยตารางข้อมูลทั้งหมด 20 ตารางข้อมูล ดังนี้

ตารางที่ ข-11 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางคณาจารย์

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
TechID	เลขที่คณาจารย์	Varchar	20	ไม่ว่าง	PK
Tech_Prefix	คำนำหน้าชื่อ	Varchar	10		
Tech_Name	ชื่อคณาจารย์	Varchar	100		
Tech_Surname	นามสกุลคณาจารย์	Varchar	100		
EngPrefix	คำนำหน้าชื่อ ภาษาอังกฤษ	Varchar	10		
Engname	ชื่อภาษาอังกฤษ	Varchar	100		
Engsurname	นามสกุลภาษาอังกฤษ	Varchar	100		
Tech_Address	ที่อยู่	Varchar	255		
Tech_Gender	เพศ	Varchar	50		Male or Female
P_ID	รหัสอ้างอิงตำแหน่ง	Int		ไม่ว่าง	FK
PF_ID	รหัสคำนำหน้าชื่อ	Int		ไม่ว่าง	FK
Faculty_ID	รหัสคณะ	Int		ไม่ว่าง	FK
Major_ID	รหัสสาขา	Int		ไม่ว่าง	FK

ตารางที่ ข-12 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางคำนำหน้าชื่อ

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
PF_ID	รหัสคำนำหน้าชื่อ	Int		ไม่ว่าง	PK
Prefix_Name	ชื่อคำนำหน้าชื่อ	Varchar	100		

ตารางที่ ข-13 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางวิชาเรียน

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
Course_ID	รหัสสาขาวิชา	Int		ไม่ว่าง	PK
Course_Name	ชื่อสาขาภาษาไทย	Varchar	100		
Course_EngName	ชื่อสาขาภาษาอังกฤษ	Varchar	100		
Course_Desc	รายละเอียดวิชาเรียน	Varchar	255		

ตารางที่ ข-14 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางคณะ

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
Faculty_ID	รหัสคณะ	Int		ไม่ว่าง	PK
Faculty_Name	ชื่อคณะภาษาไทย	Varchar	100		
Faculty_EngName	ชื่อคณะภาษาอังกฤษ	Varchar	100		

ตารางที่ ข-15 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางห้องเรียน

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
Class_ID	รหัสห้องเรียน	Int		ไม่ว่าง	PK
Class_Name	ชื่อห้องเรียน	Varchar	100		
Class_Desc	รายละเอียดห้องเรียน	Varchar	100		

ตารางที่ ข-16 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางสาขาวิชา

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
MajorID	รหัสสาขาวิชา	Int		ไม่ว่าง	PK
MajorName	ชื่อสาขาภาษาไทย	Varchar	100		
MajorEngName	ชื่อสาขาภาษาอังกฤษ	Varchar	100		
Faculty_ID	รหัสอ้างอิงคณะ	Int		ไม่ว่าง	FK

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข- 17 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางอุปกรณ์อำนวยความสะดวก

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
Facility_ID	รหัสอุปกรณ์	Int		ไมว่าง	PK
Facility_Name	ชื่ออุปกรณ์	Varchar	100		
Facility_Desc	รายละเอียดอุปกรณ์	Varchar	100		

ตารางที่ ข-18 กรณีศึกษาระบบระบบการจองการใช้ห้องปฏิบัติคณะวิทยาศาสตร์: ตารางการใช้ห้องเรียน

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
UID	รหัสการใช้งาน	Int		ไมว่าง	PK
UDesc	หมายเหตุ	Varchar	100	ไม่ว่าง	FK
Teach_ID	รหัสคณาจารย์	Int		ไม่ว่าง	FK
Course_ID	รหัสของรายวิชา	Int		ไม่ว่าง	FK
Class_ID	รหัสของห้องเรียน	Int		ไม่ว่าง	FK
Facilit _ID	รหัสของอุปกรณ์อำนวยความสะดวก	Int		ไม่ว่าง	FK

<u>ตัวอย่างของกรณีทดสอบ</u>

```
SQL>    insert into prefix values(10,'Mr.');
        insert into prefix values(10,'Mrs.');
        insert into prefix values(11,'Mrs.');
        insert into prefix values(12,'ว่าที่ร้อยตรี');
        insert into prefix values(13,'ว่าที่ร้อยตรี1');

        insert into faculty values(20,'วิทยาศาสตร์','SC');
        insert into faculty values(20,'วิศวกรรมศาสตร์','ENG');
```

รูปที่ ข-3 แสดงตัวอย่างของกรณีทดสอบของระบบการจองการใช้ห้องปฏิบัติการ

ตัวอย่างของกรณีทดสอบ (ต่อ)

```
SQL>    insert into faculty values(21,'วิศวกรรมศาสตร์','ENG');
        insert into faculty values(22,'วิศวกรรมศาสตร์1','ENG1');
        insert into faculty values(23,'วิศวกรรมศาสตร์2','ENG2');
        insert into faculty values(24,'วิศวกรรมศาสตร์3','ENG3');

        insert into subject values(30,'คณิตศาสตร์1','MATH1','-');
        insert into subject values(30,'คณิตศาสตร์2','MATH2','-');
        insert into subject values(31,'คณิตศาสตร์2','MATH2','-');
        insert into subject values(32,'คณิตศาสตร์3','MATH3','-');
        insert into subject values(33,'คณิตศาสตร์4','MATH4','-');
        insert into subject values(34,'คณิตศาสตร์5','MATH5','-');

        insert into major values(40,'เคมี','CHEM',1);
        insert into major values(40,'คอมพิวเตอร์','COM',1);
        insert into major values(41,'คณิตศาสตร์','MATH',1);
        insert into major values(42,'คณิตศาสตร์1','MATH1',1);
        insert into major values(43,'คณิตศาสตร์2','MATH2',1);

        insert into faculty values(400,'เทคโนโลยีอุตสาหกรรม','INDUSTRIAL
        TECHNOLOGY');
        insert into major values(42,'ชีววิทยา','Biology',300);
        delete from faculty where Faculty_ID='300';
```

รูปที่ ข-4 แสดงตัวอย่างของกรณีทดสอบของระบบการจองการใช้ห้องปฏิบัติการ (ต่อ)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

ระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี เป็นระบบที่เกี่ยวข้องกับการจัดการฐานข้อมูลข่าวและประชาสัมพันธ์ต่างๆ ของมหาวิทยาลัย ทั้งนี้ข้อมูลที่เกี่ยวข้องกับการทดสอบเป็นข้อมูลข่าว ข้อมูลประเภทข่าว ข้อมูลงานประสัมพันธ์ ข้อมูลบุคลากรของระบบ ข้อมูลสื่อทางด้านงานประชาสัมพันธ์ ตัวอย่างสคีมาของระบบฐานข้อมูลประกอบด้วยตารางข้อมูลทั้งหมด 36 ตารางข้อมูล ดังนี้

ตารางที่ ข-19 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางข่าว

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
N_ID	รหัสข่าว	Int		ไม่ว่าง	PK
N_Desc	รายละเอียดข่าว	Varchar	255		
N_	สื่อที่จัดเก็บ	Varchar	255		
NDate	วันที่ประกาศข่าว	Date			
Cate_News	รหัสประเภทข่าว	Int			FK

ตารางที่ ข-20 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางคณะ

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
Faculty_ID	รหัสคณะ	Int		ไม่ว่าง	PK
Faculty_Name	ชื่อคณะภาษาไทย	Varchar	100		
Faculty_EngName	ชื่อคณะภาษาอังกฤษ	Varchar	100		

ตารางที่ ข-21 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางคำนำหน้าชื่อ

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
PF_ID	รหัสคำนำหน้าชื่อ	Int		ไม่ว่าง	PK
Prefix_Name	ชื่อคำนำหน้าชื่อ	Varchar	100		

ตารางที่ ข-22 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางบุคคลากร

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
EmpID	เลขที่คณาจารย์	Varchar	20	ไม่ว่าง	PK
Emp_Prefix	คำนำหน้าชื่อ	Varchar	10		
Emp_Name	ชื่อคณาจารย์	Varchar	100		
Emp_Surname	นามสกุลคณาจารย์	Varchar	100		
EngPrefix	คำนำหน้าชื่อภาษาอังกฤษ	Varchar	10		
Engname	ชื่อภาษาอังกฤษ	Varchar	100		
Engsurname	นามสกุลภาษาอังกฤษ	Varchar	100		
Address	ที่อยู่	Varchar	255		
Gender	เพศ	Varchar	50		Male or Female
P_ID	รหัสอ้างอิงตำแหน่ง	Int		ไม่ว่าง	FK
PF_ID	รหัสคำนำหน้าชื่อ	Int		ไม่ว่าง	FK
Faculty_ID	รหัสคณะ	Int		ไม่ว่าง	FK
Dep_ID	รหัสแผนก	Int		ไม่ว่าง	FK

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-23 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางตำแหน่ง

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
P_ID	รหัสตำแหน่ง	Int		ไม่ว่าง	PK
P_Name	ชื่อตำแหน่ง	Varchar	10		

ตารางที่ ข-24 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางสาขาวิชา

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
DepID	รหัสแผนก	Int		ไม่ว่าง	PK
DepName	ชื่อแผนกภาษาไทย	Varchar	100		
DepEngName	ชื่อแผนกภาษาอังกฤษ	Varchar	100		
Faculty_ID	รหัสอ้างอิงคณะ	Int		ไม่ว่าง	FK

ตารางที่ ข-25 กรณีศึกษาระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี: ตารางประเภทของข่าว

ชื่อเขตข้อมูล	คำอธิบาย	ชนิดข้อมูล	ความยาว	ค่าว่าง	หมายเหตุ
Cate_NewsID	รหัสแผนก	Int		ไม่ว่าง	PK
Cate_Name	ชื่อประเภทข่าว	Varchar	100		
DepEngName	รายละเอียด	Varchar	100		

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างของกรณีทดสอบ

```
SQL>    insert into news values(10,'ข้อมูลทดสอบ1','อิเล็กทรอนิกส์','10/2/2009',20);
        insert into news values(10,'ข้อมูลทดสอบ2','อิเล็กทรอนิกส์','10/3/2009',20);

        insert into news values(11,'ข้อมูลทดสอบ3','อิเล็กทรอนิกส์','10/2/2009',20);

        insert into category vlaues(200,'การศึกษา','-');
        insert into news values(12,'ข้อมูลทดสอบ4','อิเล็กทรอนิกส์','10/2/2009',200);
        delete from category where Cate_news='200';

        insert into prefix values(10,'Mr.');
        insert into prefix values(10,'Mrs.');
        insert into prefix values(11,'Mrs.');
        insert into prefix values(12,'ว่าที่ร้อยตรี');
        insert into prefix values(13,'ว่าที่ร้อยตรี1');

        insert into faculty values(20,'วิทยาศาสตร์','SC');
        insert into faculty values(20,'วิศวกรรมศาสตร์','ENG');
        insert into faculty values(21,'วิศวกรรมศาสตร์','ENG');
        insert into faculty values(22,'วิศวกรรมศาสตร์1','ENG1');
        insert into faculty values(23,'วิศวกรรมศาสตร์2','ENG2');
        insert into faculty values(24,'วิศวกรรมศาสตร์3','ENG3');
```

รูปที่ ข-5 แสดงตัวอย่างของกรณีทดสอบของระบบระบบการจัดฐานข้อมูลข่าวและประชาสัมพันธ์มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

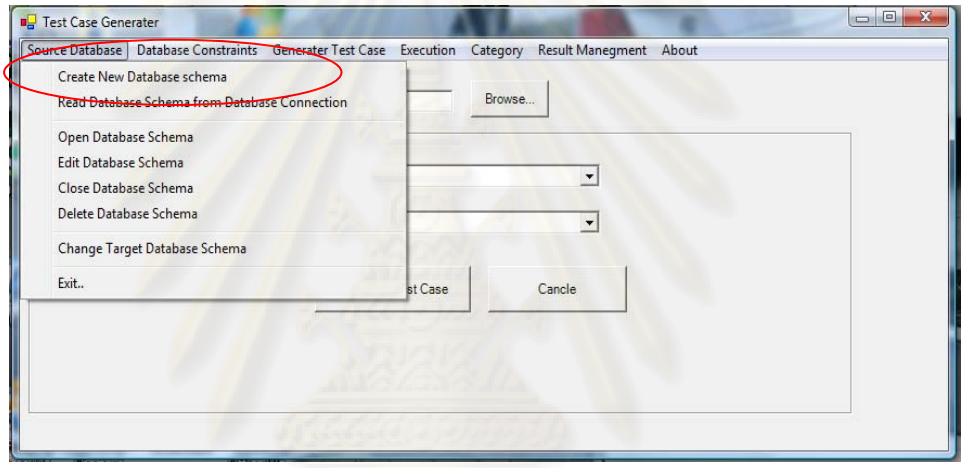
คู่มือการใช้งานเครื่องมือสร้างกรณีทดสอบ

การใช้งานเครื่องมือสร้างกรณีทดสอบแบ่งออกเป็น 4 ส่วน ดังนี้

1. การจัดการสคีมาของฐานข้อมูล

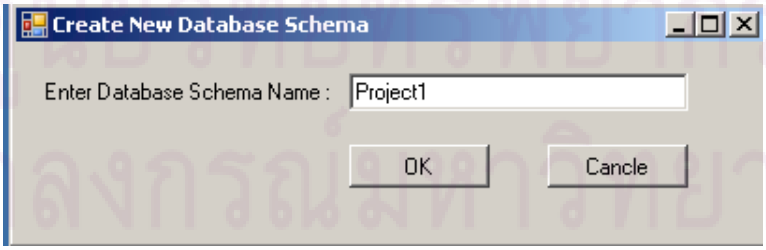
1.1 การสร้างสคีมาของฐานข้อมูลโดยผู้ใช้ทำการระบุ จะมีขั้นตอนดังนี้

1.1.1 เลือกที่เมนู Source Database และเลือกที่เมนูย่อย Create New Database Schema ดังรูปที่ ค-1



รูปที่ ค-1 หน้าจอเมนูการสร้างสคีมาของฐานข้อมูล

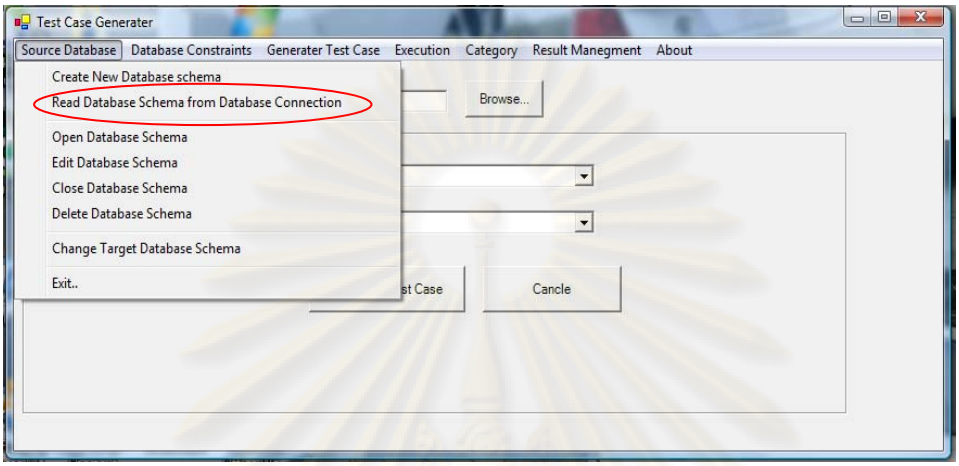
1.1.2 กำหนดชื่อของสคีมาของฐานข้อมูลที่กำลังจะสร้าง แล้วกดตกลง เพื่อยืนยันการสร้างสคีมาของฐานข้อมูล ดังรูปที่ ค-2



รูปที่ ค-2 หน้าจอเมนูการกำหนดชื่อสคีมาของฐานข้อมูล

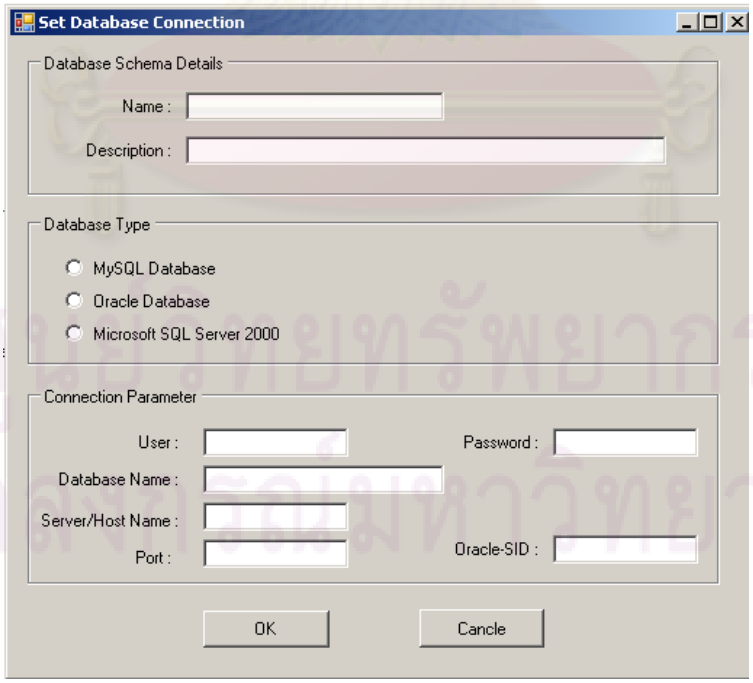
1.2 การสร้างสคีมาของฐานข้อมูลจากฐานข้อมูลที่มีต้องการทดสอบ

1.2.1 เลือกที่เมนู Source Database และเลือกที่เมนูย่อย Read Database Schema ดังรูปที่ ค-3



รูปที่ ค-3 การอ่านสคีมาของฐานข้อมูล

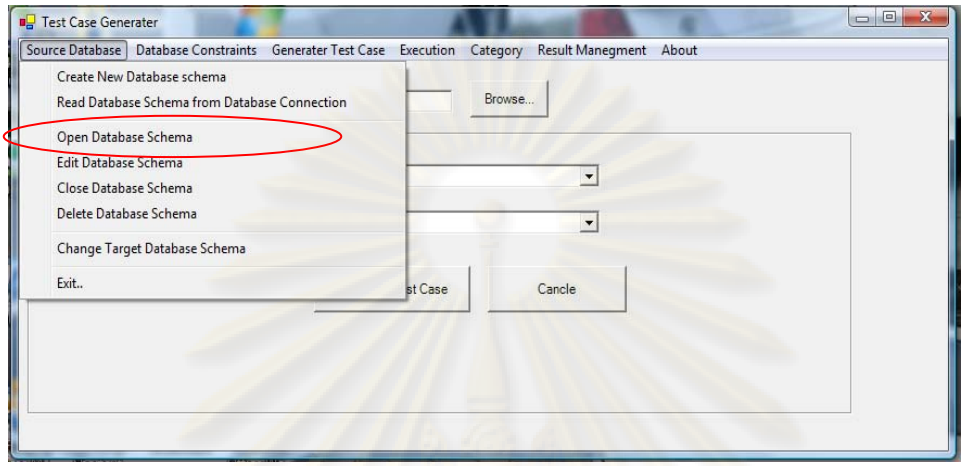
1.2.2 กำหนดชื่อและคำอธิบายให้กับสคีมาของฐานข้อมูลที่ต้องการสร้าง จากนั้นเลือกชนิดของฐานข้อมูลที่จะทำการเชื่อมต่อพร้อมทั้งกำหนดพารามิเตอร์ที่ใช้ในการเชื่อมต่อ และกดปุ่มตกลง เพื่อทำการสร้างสคีมาของฐานข้อมูล ดังรูปที่ ค-4



รูปที่ ค-4 หน้าการกำหนดการเชื่อมต่อกับฐานข้อมูลเพื่อสร้างสคีมาของฐานข้อมูล

1.3 การเปิดใช้งานสคีมาของฐานข้อมูล จะมีขั้นตอนดังนี้

1.3.1 เลือกที่เมนู Source Database และเลือกที่เมนูย่อย Open Database Schema ดังรูปที่ ค-5



รูปที่ ค-5 หน้าจอเมนูการเปิดสคีมาของฐานข้อมูล

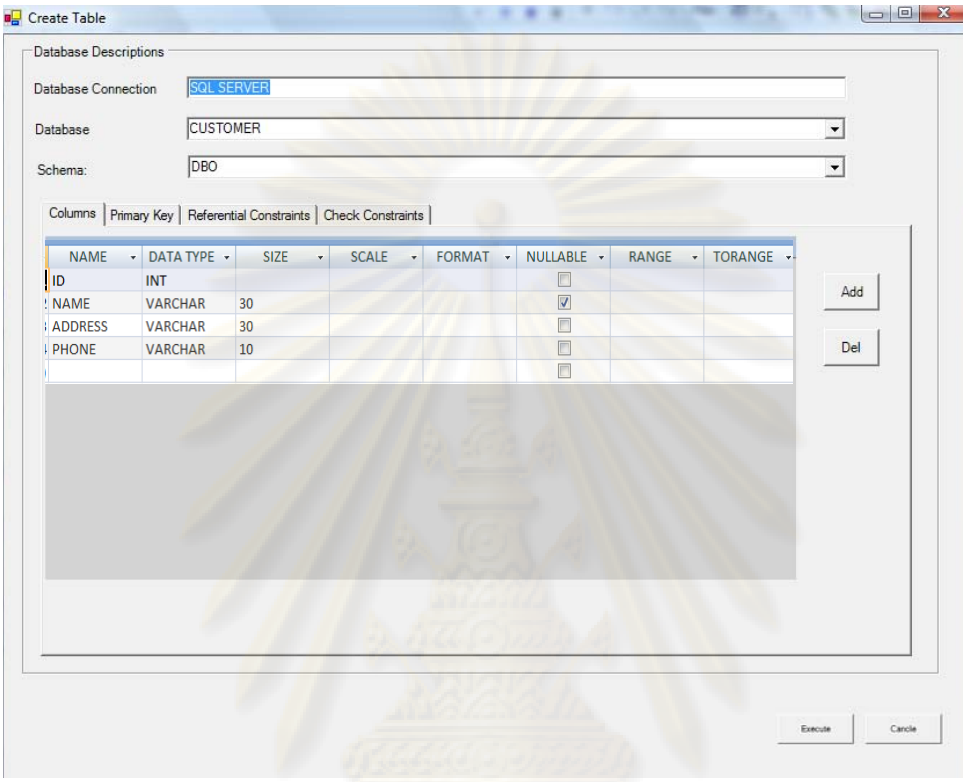
1.4 การแก้ไขสคีมาของฐานข้อมูล จะมีขั้นตอนคือ เลือกทีเมนู Source Database และเลือกที่เมนูย่อย Edit Database Schema จากนั้นเครื่องมือจะแสดงหน้าจอแก้ไขสคีมาของฐานข้อมูล ซึ่งผู้ใช้จะสามารถแก้ไขสคีมาของฐานข้อมูลได้ตามต้องการไม่ว่าจะเป็นการเพิ่ม แก้ไข หรือลบตารางข้อมูล การแก้ไขการขึ้นต่อกันเชิงตรรกะ และการกำหนดความคงสภาพของการอ้างอิง

1.5 การลบสคีมาของฐานข้อมูล จะมีขั้นตอน คือ เลือกที่เมนู Source Database และเลือกที่เมนูย่อย Delete Database Schema เครื่องมือจะแสดงรายระเอียดของสคีมาเพื่อให้ผู้ใช้ทำการเลือกสคีมาขึ้นมาทำการลบสคีมาของฐานข้อมูล จากนั้นเครื่องมือจะให้ผู้ใช้ทำการยืนยันการลบสคีมาของฐานข้อมูลอีกครั้ง

1.6 การปิดสคีมาของฐานข้อมูล จะมีขั้นตอน คือ เลือกที่เมนู Source Database และเลือกที่เมนูย่อย Close Database Schema เพื่อการปิดสคีมาของฐานข้อมูล

1.7 การเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบ จะมีขั้นตอน คือ เลือกที่เมนู Source Database และเลือกที่เมนูย่อย Change Target Database Schema เพื่อทำการแก้ไขโดยหน้าจอจะมีลักษณะคล้ายกับหน้าการสร้างสคีมาของฐานข้อมูล เครื่องมือจะทำการเรียกดูข้อมูลสคีมาแล้วแสดงผลให้ทราบ

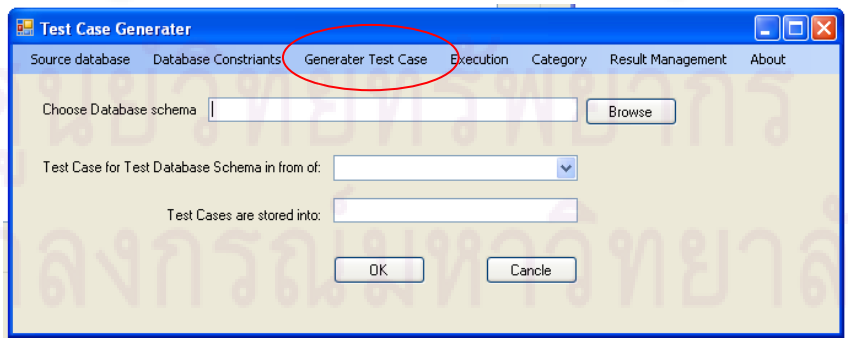
2. การจัดการข้อมูลเงื่อนไขบังคับของฐานข้อมูล ขั้นตอนจะเริ่มจากที่เมนู Database Constraints จากนั้นเครื่องมือจะแสดงหน้าจอการจัดการรับข้อมูลที่เกี่ยวกับเงื่อนไขบังคับของฐานข้อมูล ดังรูปที่ ค-6



รูปที่ ค-6 หน้าจอการกำหนดรายละเอียดเงื่อนไขบังคับของเขตข้อมูล

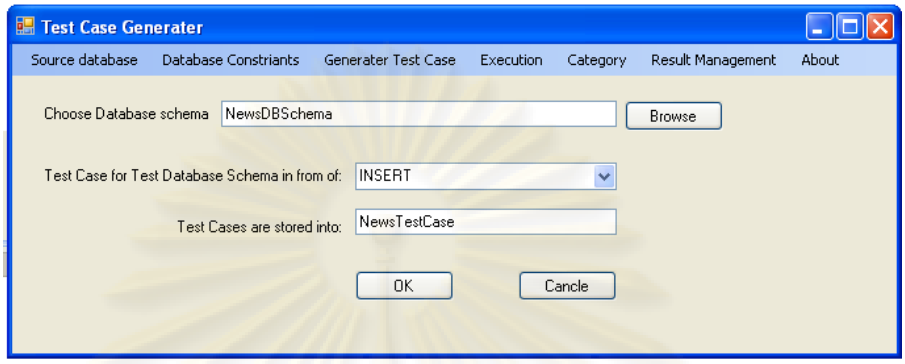
3. การสร้างกรณีทดสอบ จะมีขั้นตอนดังนี้

3.1 เลือกที่เมนู Generater Test Case ดังรูปที่ ค-7



รูปที่ ค-7 หน้าจอเมนูการสร้างกรณีทดสอบ

3.2 ทำการระบุกรณีทดสอบที่ต้องการสร้างโดยทำการเลือกจากสคีมาจากฐานข้อมูลของเครื่องมือ จากนั้นทำการระบุรูปแบบของกรณีทดสอบที่จะทำการสร้าง ได้แก่ การเพิ่ม การลบ และการแก้ไขข้อมูล ดังรูป ค-8

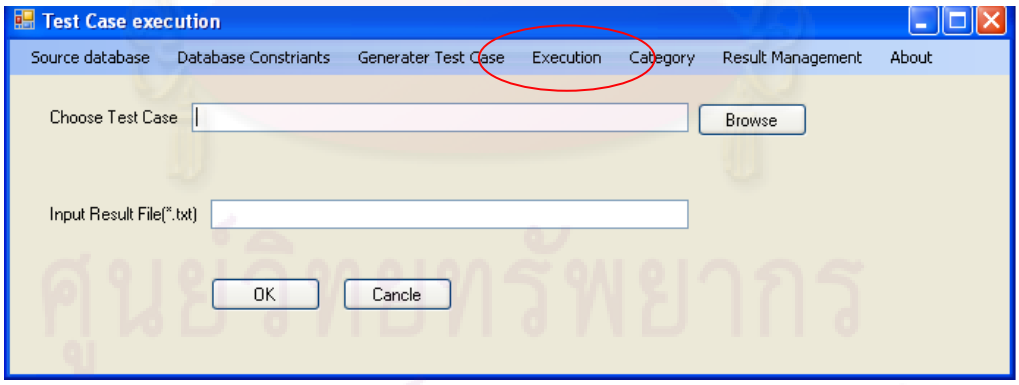


รูปที่ ค-8 หน้าจอการกำหนดรายละเอียดการสร้างกรณีทดสอบ

3.3 ทำการกำหนดชื่อชุดของกรณีทดสอบที่จะทำการสร้างขึ้นใหม่ เพื่อใช้ในการจัดเก็บลงฐานข้อมูลและเรียกดูเพื่อปรับปรุงแก้ไขกรณีทดสอบที่สร้างขึ้น จากนั้นให้ผู้ใช้ทำการเลือกปุ่มตกลง เพื่อสร้างกรณีทดสอบ หรือปุ่มยกเลิกเพื่อยกเลิกการทำงาน เครื่องมือทดสอบจะทำการสร้างกรณีทดสอบให้อัตโนมัติ และจัดเก็บเอกสารให้อยู่ในรูปแบบเอกสารเอ็กซ์เอ็มแอล

4. การประมวลผลกรณีทดสอบ จะมีขั้นตอนดังนี้

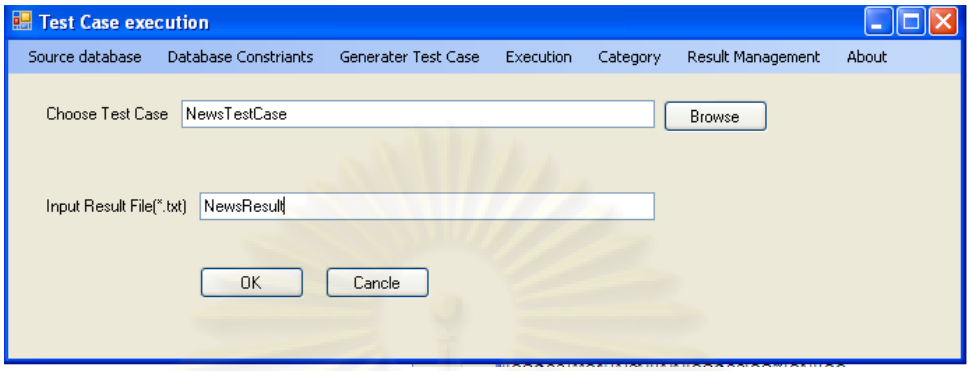
4.1 เลือกที่เมนู Execution ดังรูปที่ ค-9



รูปที่ ค-9 หน้าจอเมนูการประมวลกรณีทดสอบ

4.2 ทำการระบุกรณีทดสอบที่ต้องการสร้างโดยทำการเลือกจากกรณีทดสอบจากฐานข้อมูลของเครื่องมือ จากนั้นทำการกำหนดชื่อของผลลัพธ์การทดสอบ (โดยเอกสารที่สร้างขึ้นจะถูกจัดเก็บลงโฟลเดอร์ของเครื่องมือ) เครื่องมือทดสอบจะทำการประมวลผลกรณีทดสอบให้

อัตโนมัติ และจัดเก็บเอกสารให้อยู่ในรูปแบบเอกสารเอ็กซ์เอ็มแอล ทั้งนี้การเชื่อมต่อกับฐานข้อมูลที่ต้องการทดสอบได้มีการระบุไว้ก่อนหน้านี้แล้ว ดังรูป ค-10

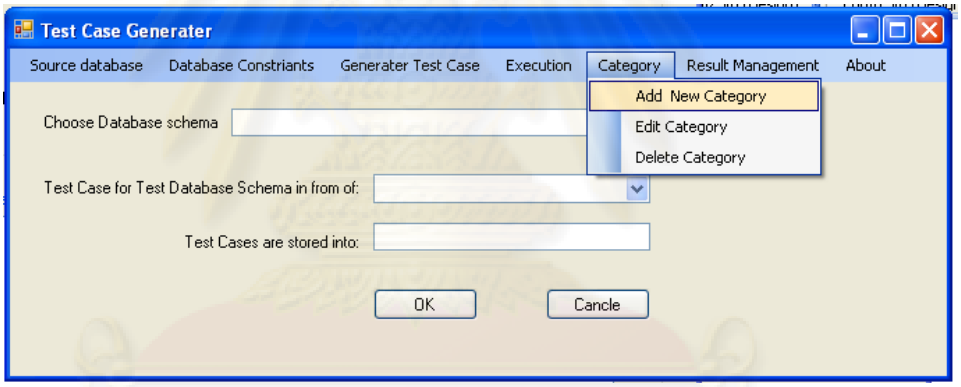


รูปที่ ค-10 หน้าจอการกำหนดรายละเอียดการประมวลผลกรณีทดสอบ

5. การจัดการกลุ่มค่าทั่วไป

5.1 การเพิ่มกลุ่มค่าทั่วไป จะมีขั้นตอนดังนี้

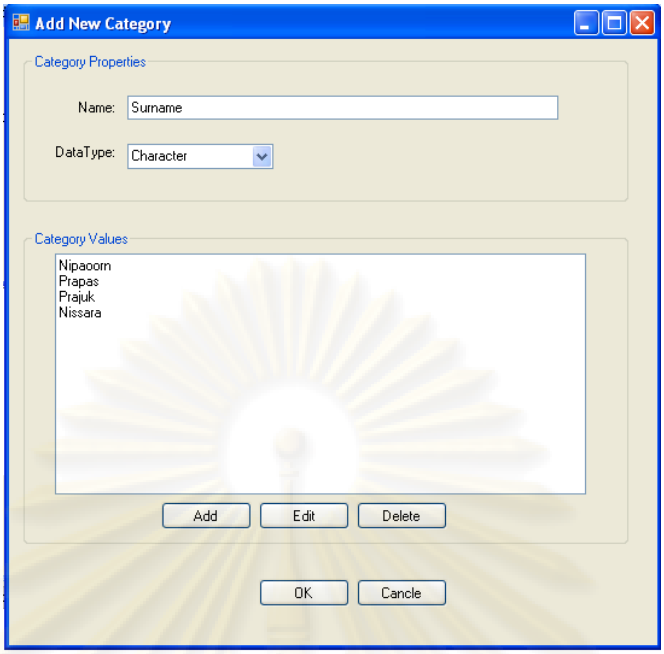
5.1.1 เลือกที่เมนู Category และเลือกที่เมนูย่อย Add New Category ดังรูปที่ ค-11



รูปที่ ค-11 หน้าจอการเมนูการเพิ่มกลุ่มค่าทั่วไป

5.1.2 กำหนดข้อมูลให้กลุ่มค่าทั่วไป ได้แก่ ชื่อกลุ่มค่าทั่วไป ชนิดของข้อมูล และรายการทั่วไป ดังรูป ค-12

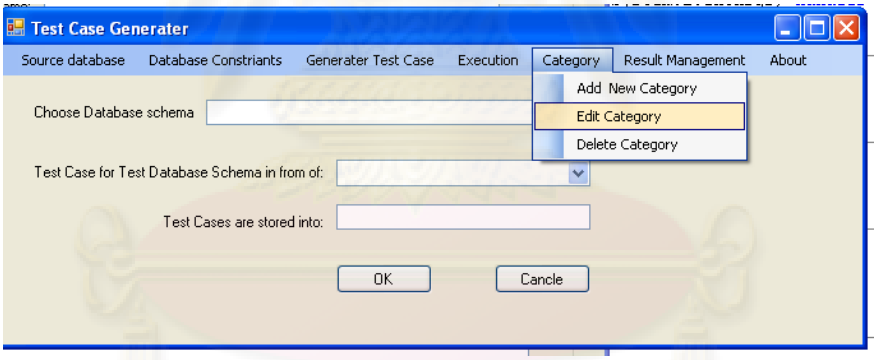
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ค-12 หน้าจอการกำหนดรายละเอียดของการเพิ่มกลุ่มค่าทั่วไป

5.2 การแก้ไขกลุ่มค่าทั่วไป จะมีขั้นตอนดังนี้

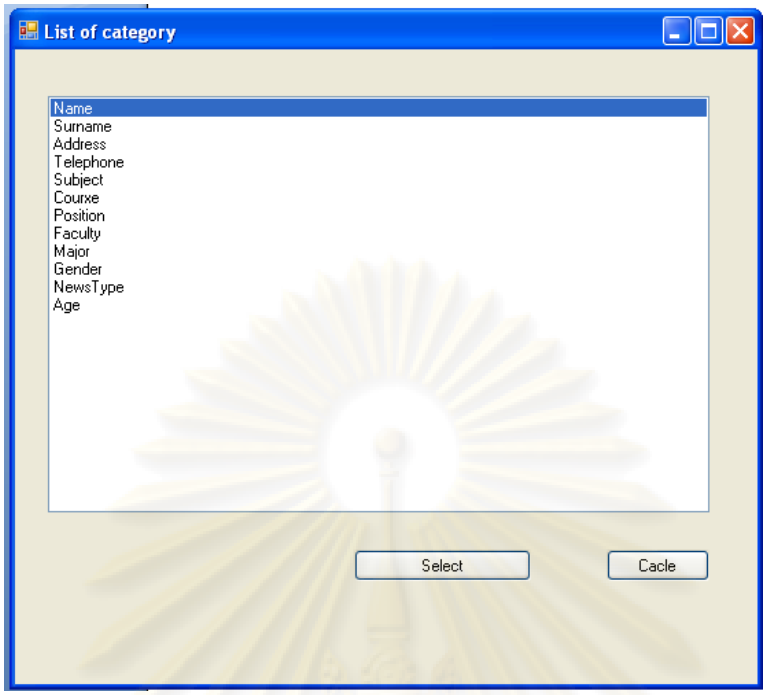
5.2.1 เลือกที่เมนู Category และเลือกที่เมนูย่อย Edit Category ดังรูปที่ ค-13



รูปที่ ค-13 หน้าจอการเมนูการแก้ไขกลุ่มค่าทั่วไป

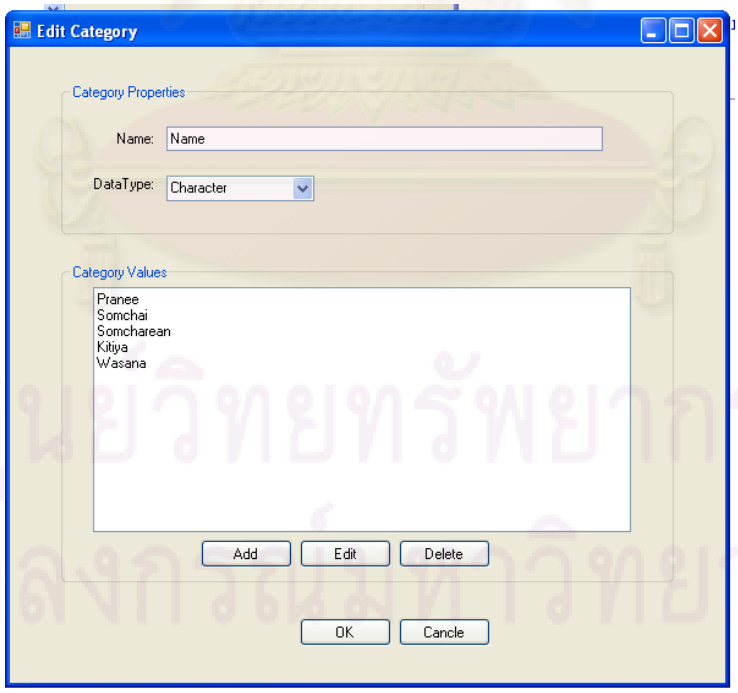
5.2.2 เลือกกลุ่มค่าทั่วไปที่ต้องการแก้ไข แล้วกดปุ่ม Select ดังรูปที่ ค-14

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ค-14 การเลือกกลุ่มค่าทั่วไปที่ต้องการแก้ไข

5.2.3 แก้ไขข้อมูลของกลุ่มค่าทั่วไปตามที่ต้องการ ได้แก่ ชื่อกลุ่มค่าทั่วไป ชนิดข้อมูล และรายการค่าทั่วไป ดังรูป ค-15

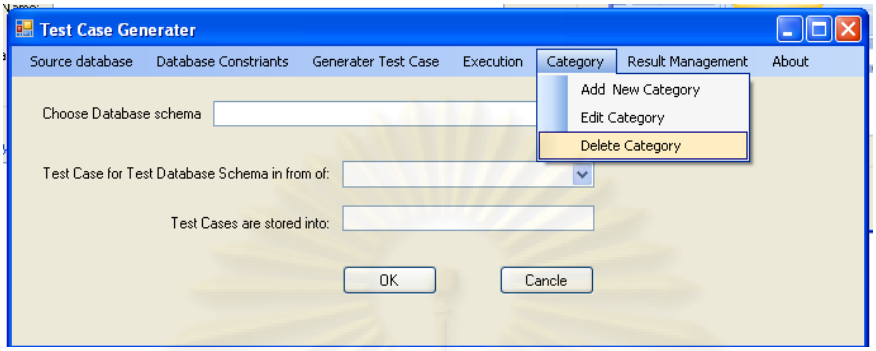


รูปที่ ค-15 หน้าจอการแก้ไขกลุ่มค่าทั่วไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

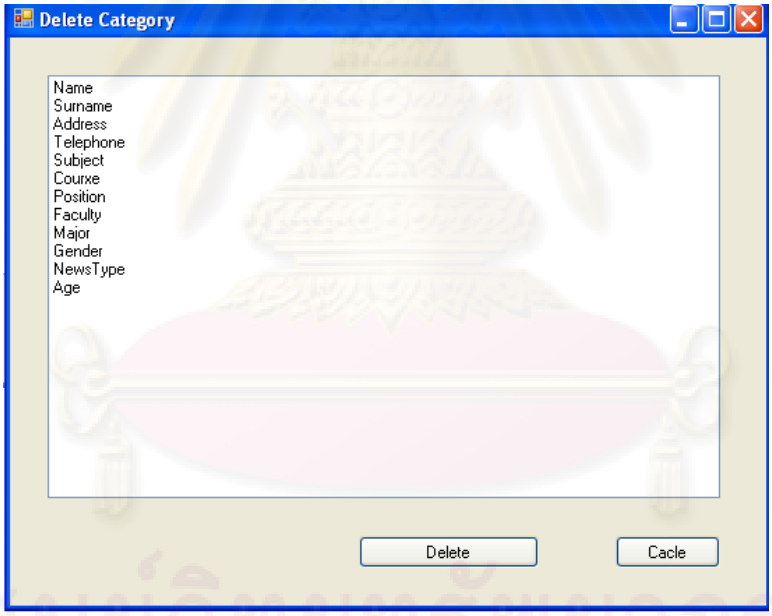
5.3 การลบกลุ่มค่าทั่วไป จะมีขั้นตอนดังนี้

5.3.1 เลือกที่เมนู Category และเลือกที่เมนูย่อย Delete Category ดังรูปที่ ค-16



รูปที่ ค-16 หน้าจอการเมนูการลบกลุ่มค่าทั่วไป

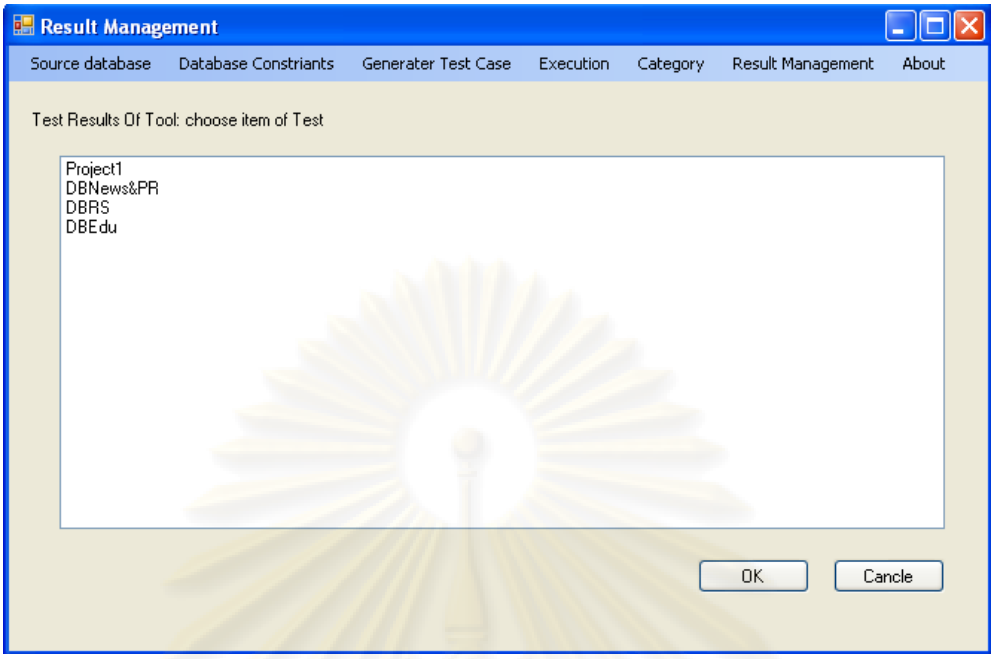
5.3.2 เลือกกลุ่มค่าทั่วไปที่ต้องการลบ จากนั้นเครื่องมือจะทำการแสดงหน้าจอยืนยันการลบอีกครั้งหนึ่ง ดังรูปที่ ค-17



รูปที่ ค-17 การเลือกกลุ่มค่าทั่วไปที่ต้องการลบ

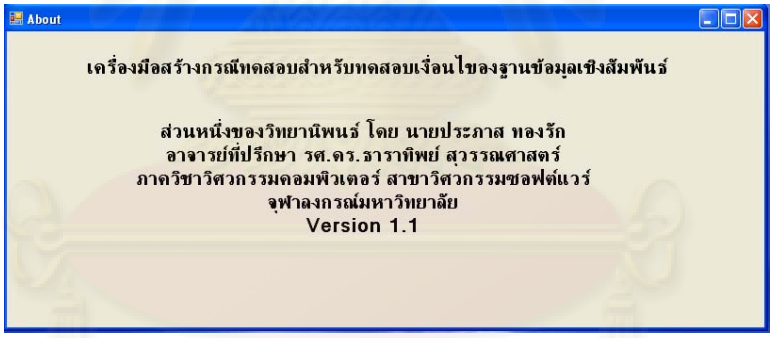
6. การจัดการผลการทดสอบ จะมีขั้นตอนดังนี้

6.1 เลือกที่เมนู Result Management จากนั้นเครื่องมือจะแสดงดังรูปที่ ค-18 โดยสามารถทำการเรียกดูและทำการแก้ไขส่วนของจัดการข้อมูลผลลัพธ์ โดยการเลือกผลลัพธ์ที่ต้องการแก้ไข



รูปที่ ค-18 หน้าจอเมนูการจัดการผลลัพธ์ของการทดสอบ

7. ข้อมูลของเครื่องมือ เลือกจากเมนู About เครื่องมือจะแสดงรายละเอียดของเครื่องมือสร้างกรณีทดสอบ ดังรูปที่ ค-19



รูปที่ ค-19 หน้าจอเกี่ยวกับข้อมูลของเครื่องมือ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

ผลงานที่ตีพิมพ์

งานวิจัยนี้ได้รับคัดเลือกให้นำเสนอในงานประชุมวิชาการและตีพิมพ์ในเอกสาร “Proceedings of The second IEEE International Conference on Computer Science and Information Technology (ICCSIT2009)” ในระหว่างวันที่ 8-11 สิงหาคม 2552 ณ กรุงปักกิ่ง สาธารณรัฐประชาชนจีน โดยมีหัวข้อวิจัยชื่อ “A Tool for Generating Test Case from Relational Database Constraints Testing ”

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

A Tool for Generating Test Case from Relational Database Constraints Testing

Prapas Tongrak
Department of Computer Engineering
Chulalongkorn University, Bangkok THAILAND
prapas.t@student.chula.ac.th

Taratip Suwannasart
Department of Computer Engineering
Chulalongkorn University, Bangkok THAILAND
taratip.s@chula.ac.th

***Abstract*— Database applications become increasingly complex. Consequently, database configuration must be changed. (For instance, business process change, authorization and role change) Testing database applications is much more challenging because most methods and tools developed for application testing have not focused on database constraints. In this paper, the tool for relational database constraints testing is proposed to support database testing and is constructed to automate test case generation according to criteria, supplied by tester and target database. The criteria consist of database schema, entity integrity constraints, referential integrity constraints, and domain constraints. The goal of the tool is to generate test cases that cover database constraints of the software under test and to assist the database application developer or tester in a usable and useful way.**

Keywords - Software testing, Test case, Relational database constraints

I. INTRODUCTION

Software testing is one of the essential processes in the software development lifecycle and it takes a lot of cost and effort to complete because software testing is the process of executing software with the intent of finding errors. Software testing requires test case to test the software.

Nowadays, data is an important corporate asset. Doesn't it make sense to invest the effort required to validate the quality of data via effective testing [1]? Database applications are becoming increasingly complex. They are composed of many components and stacked in several layers. Furthermore, most database applications are subject to constant change; for instance, business processes are re-engineered, authorization rules are changed, components are replaced by other more powerful components, or optimizations are added in order to achieve better performance for a growing number of users and data. The more complex an application becomes, the more frequently the application and its configuration must be changed.

Unfortunately, the mostly methods and testing tools for database application focus on functions of database applications but most of them have not concerned the constraints of relational database. Relational database constraints include entity integrity constraints, referential integrity constraints, and domain constraints. Thus, the generating test cases do not cover all possible conditions. Besides there are various testing tools that manage test result. But these tools require a great deal of manual work.

Therefore, this paper presents the tool for generating test case of relational database constraints testing. The paper focuses relational database constraints. The paper is organized as follows. In section 2 reviews related works. Section 3 describes relational database constraints, while section 4 explains the constructing of this tool and its implementation. Empirical studies are discussed in section 5. Section 6 suggests the future work and the conclusion.

II. RELATED WORKS

K. Piriyakitpaiboon [2] presents a tool to automate the test data generation process, named "RealGen". The tool randomly generates test data according to the criteria, supplied by testers and target database. The criteria consist of database schema, logical dependencies between fields in each table, referential integrity of database, quantity of generated records, and etc. The main goal of RealGen is to generate test data that is meaningful as the actual data of the software under test.

DeZign[3] is an intuitive database design tool for developers and DBA's that can help you model, create and maintain databases. DeZign for Database uses entity relationship diagrams (ERDs) to graphically design databases and automatically generates the most popular SQL and desktop databases. In step of input constraints, DeZign only focuses on domain constraints but lack of focusing on all constraints of relational database. Consequently, Database is not correct and complete as define in database design specification.

III. RELATIONAL DATABASE CONSTRAINTS

Relational Database constraints are identified on database by user. A part of constraints is integrity constraint that guarantees data correction and data redundancy. Data integrity is correction constraint of data in database, according to referential integrity constraints. Integrity constraints consist of 3 parts entity integrity constraints, referential integrity constraints, and domain constraints [4].

In the relational data model, entity integrity is one of the three inherent integrity rules. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be

the primary key should be unique and not null. A direct consequence of this integrity rule is that duplicate rows are forbidden in a table. Each value of a primary key must be unique. Thus, no duplicate rows can logically appear in a table. The NOT NULL characteristic of a primary key ensures that a value can be used to identify all rows in a table. Within relational databases using SQL, entity integrity is enforced by adding a primary key clause to a schema definition. The system enforces Entity Integrity by not allowing operations (INSERT, UPDATE) to produce an invalid primary key. Any operation that is likely to create a duplicate primary key or one containing null is rejected.

Referential integrity in a relational database is consistency between coupled tables. Referential integrity is usually enforced by the combination of a primary key or candidate key (alternate key) and a foreign key. To hold referential integrity, any field in a table that is declared a foreign key can contain only values from a parent table's primary key or a candidate key. For instance, deleting a record that contains a value referred by a foreign key in another table would break referential integrity. The relational database management system (RDBMS) enforces referential integrity, normally either by deleting the foreign key rows as well to maintain integrity, or by returning an error and not performing the delete. Which method is used would be determined by the referential integrity constraint, as defined in the data dictionary.

Referential integrity is the relational property that each foreign key value in a table exists as a primary key in the referenced table [5]. Referential integrity relationships are defined with SQL FOREIGN KEY and PRIMARY KEY clauses in the CREATE TABLE statement and are automatically maintained both during load, update, and insert operations to a referencing table and during delete operations from a referenced table.

During a delete operation, if a row to be deleted contains a value that is referenced by a foreign key in another table, a referential integrity violation is avoided by either:

- Deleting the original row and also deleting the referencing row from the other table. This action is called a cascaded delete and can cascade through a series of referencing tables.
- Deleting neither row--that is, taking no action. This lack of action is called a restricted delete.

The course of action to be taken--a cascaded or restricted delete--is specified at the time the table is created by the values CASCADE or NO ACTION in the ON DELETE clause of the CREATE TABLE statement.

Domain Constraint is an integrity rule. A domain of possible values should be associated with every attribute. These domain constraints are the most basic form of integrity constraint. They are easy to test for when data is entered.

Examples of domain types are as follows:

- Attributes may have the same domain, e.g. cname and employee-name.
- It is not as clear whether bname and cname domains ought to be distinct.
- At the implementation level, they are both character strings.
- At the conceptual level, we do not expect customers to have the same names as branches, in general.
- Strong typing of domains allows us to test for values inserted, and whether queries make sense. Newer systems, particularly object-oriented database systems, offer a rich set of domain types that can be extended easily.

IV. CONSTRUCTING RELATIONAL DATABASE CONSTRAINTS TESTING TOOL

The tool is developed to automate test case generation process in software testing. The tool generates test case according to database design specifications that are identified by testers. The process of the tool is shown in Figure 1.

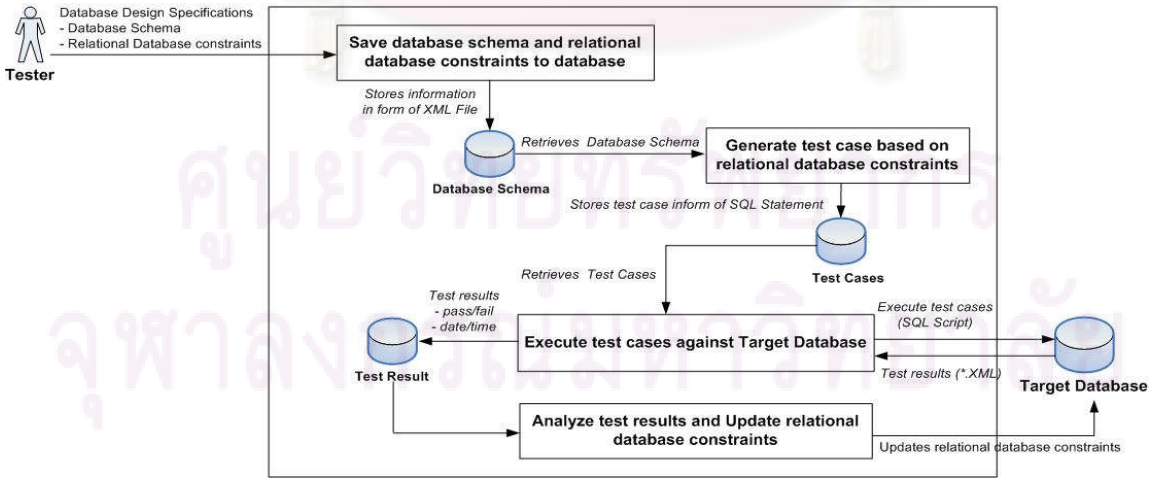


Figure 1. Relational database constraints testing tool.

Relational database constraints are entity integrity constraints, referential integrity constraints, and domain constraints. The tool uses the database schema along with relational database constraints to generate test case in form of SQL statement and save into the tool's database. Test cases are executed against target database which we want to test and stored test results. Finally, the tool analyzes faults and updates relational database constraints in accordance with constraints identified by testers. The next sections describe the process of the tool.

A. Preparing Database Schema from Database Design Specifications

The tool prepares database schema to generating test cases. Database schema is identified by testers and follows database design specifications. Database schema includes tables, attributes of tables, and relational database constraints. The tool supports relational database constraints that include entity integrity constraints, referential integrity constraints, and domain constraints. Testers could define at must ten properties for each field in each table. The properties of a field are filed name, data type, and etc. After that, the tool saves information into the tool's database. Consequently, database schema is stored into the tool's database as an XML file.

Testers identify constraints as follows:

- Entity integrity constraints; for instance, primary key is not null and unique.
- Referential integrity constraints; for instance, foreign key keyword is "on delete cascade" or "on delete set null" or "on delete default" or "no action".
- Domain constraints; for instance, identify range of field or data type or format of data.

Figure 2 shows the relational database constraints editor screen which testers can create, edit, and delete fields of each table.

B. Generating Test Case for Relational Database Constraints

The tool retrieves database schema to generate test cases. These test cases are generated as SQL statements to validate relational database constraints. The SQL statement includes insert, update, and delete statements. The tool prepares input data to generate test cases that include random data and generic data. Finally, the test cases are stored into the tool's database for later execution.

The tool generates test case according to relational database constraints that as follows:

1) Entity integrity constraint

Entity integrity is an integrity rule which states that every table must have a primary key. This key is a unique value and not null. A test case is generated as an SQL insert statement.

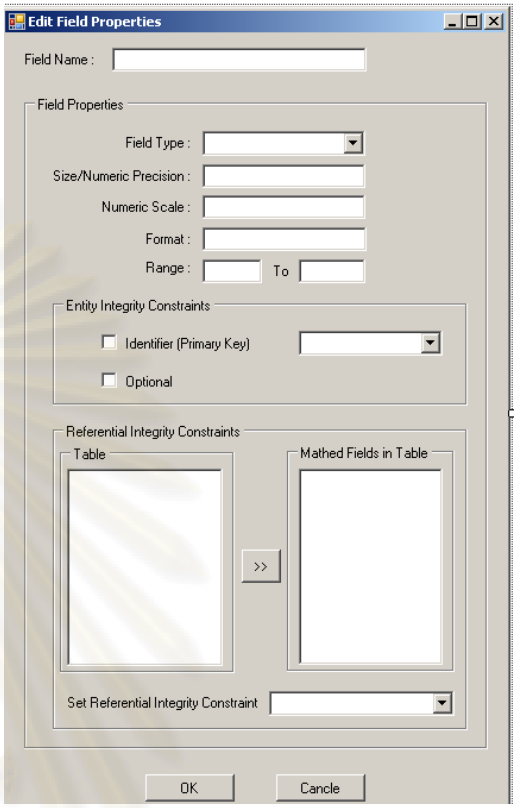


Figure 2. Specifying relational database constraints.

For example, the primary key of "customer" table is "customer_id". Thus, the "customer_id" for each row in the table must be unique. A record of the table is shown in TableI.

TABLE I. A RECORD OF CUTOMER TABLE

Customer_id	Name	City	POSTCODE
101	LYDIA	CARIFORNIA	12345

This tool generates test cases to validate entity integrity constraint. Consequently, a test case is generated as an SQL insert statement as shown in Figure 3. The "customer_id" value is "101" that shows duplicate value. The expected result is "*Duplicate Data*".

After that, test case is executed. If the new record cannot be inserted into the table, the test case will be passed. But if the new record can be inserted into the table, the test case is failed which means that the target database does not follow the entity integrity constraint. Consequently, the tool updates entity integrity constraint according to the database design specifications that are identified by testers.

```
INSERT INTO CUSTOMER VALUES
                (101, "SARAH", "SINGAPORE", 4567);
```

Figure 3. Test case is generated to validate entity integrity constraint.

2) Referential Integrity constraint

A test case is designed to validate consistency between related tables. An integrity rule is any filed in a table that is declared as a foreign key. It can contain only values from a parent table's primary key or a candidate key. The tool generates test cases for experiment of data deletion from parent table. After that, if the data of parent table can be removed, it shows that data of parent table is not referred. Therefore, this deletion is violated the referential integrity constraints. This tool also focuses delete options that includes "on delete cascade", "on delete set null", "on delete set default", and "on delete set no action". The tool is designed to support only four levels of referential integrity constraint. A test case is generated as an SQL delete statement.

An example illustrates referential integrity rule in which all table references cascade delete. The "Brand" table is referred by the "Product" table. The "Product" table is referred from the "Fact" table. The database schema is shown in Figure 4.

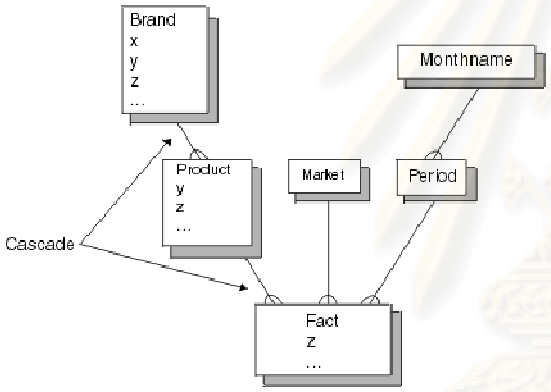


Figure 4. Example of Database Schema.

This tool generates test cases to validate referential integrity constraints. Consequently, a test case is generated as a SQL delete statement as shown in Figure 5. A test case is designed to validate consistency among "Brand" table, "Product" table, and "Fact" table. The purpose for generating test case is to delete a record ("Brand_id"="z") from "Fact" table and the expected result is "*Could not be deletion*". If the attempting to delete a row from the "Brand" table is performed before the "Fact" table is deleted, it will be violated referential integrity. Then, the error message is shown.

After that, the test case is executed. If the record can not be deleted from the table, the test case will be passed. But if the record can be deleted from the table, the test case is failed which means that the target database does not follow the referential integrity constraint. Consequently, the tool updates referential integrity constraint according to the database design specifications that are identified by testers.

```
DELETE FROM Fact WHERE Brand_id='z';
```

Figure 5. Test case is generated to validate referential integrity constraint

3) Domain constraint

A test case is designed to validate domain constraint. An integrity rule considers association of every attribute. Domain constraints are follows:

- Data type; for instance, Grade point should be float data.
- Data format for instance, Date format is "DD-MM-YYYY".
- Range of data; for instance, Range of grade point is between 0.00 and 4.00.

For example, the "city_id" is a column of the "Customer" table which boundary of city column is between 1 and 76.

This tool generates test case as SQL insert statements as shown in Figure 6. The test cases are generated to validate boundary of city column that consists of three test cases as follows:

- A test case validates lower boundary values (city_id < 1) and the expected result is "Out of Range".
- A test case validates boundary value (1 <= city_id <= 76) and the expected result is "Could be insert statement".
- A test case validates upper boundary values (city_id > 76) and the expected result is "Out of Range".

The test cases are executed. If test results pass, test cases will follow constraints. But if test results fail, target database will not follow constraints. Consequently, the tool updates domain constraints and follows database design specifications that are identified by testers.

```
INSERT INTO CUSTOMER VALUES
                (301,'MIKE','123-234-3457', 0);
INSERT INTO CUSTOMER VALUES
                (302,'POOH','234-900-8795', 36);
INSERT INTO CUSTOMER VALUES
                (303,'LYDIA','258-901-8550'77);
```

Figure 6. Test cases are generated to validate domain constraint.

C. Executing Test Case and Updating Relational Database Constraints

In this tool, all test cases are automatically generated for execution. A test case is retrieved from the tool's database. The test cases are organized in as SQL script (*.SQL) to executed the target database. Then, the tool stores test results of execution into the tool's database for analyzing in order to correct the target database and update relational database constraints. The test result is saved as XML file which shows pass or fail of each test case, as well as running time, and date.

The tool for generating test cases automatically updates relational database constraints of the target database. Relational database constraints must be implemented according to the database design specifications that are identified by testers. . If the actual results are equal to the expected results, the target database follows the constraints. Otherwise, the test cases are failed then this tool will only update relational database constraints.

V. EMPIRICAL STUDIES

The tool is applied to generate test cases for two real software systems: "Academic Affair Information System for Faculty of Science" [6] and "Reservation laboratory Online System" [7]. Both systems have entity integrity constraints, referential integrity constraints, and domain constraints in the target database. The result indicates that the tool could generate test cases which focus the constraints of both software systems.

Another empirical study of tool is the measurement of reliability of database design specifications for generating test cases. The experiment is sated up to measure the reliability of database design specifications. We use database schemas of real software [6] to generate test cases. This database schema of real software has eleven tables and all tables have relational database constraints (Relational database constraints of real software are entity integrity constraints, referential integrity constraints, and domain constraints).A database schema of this experiment is shown in Table II. The relational database constraints of real software have conditions that comply with the database design specifications which are identified by testers.

TABLE II. DATABASE SCHEMA OF EXPERIMENT

Table Name	Number of Field	Entity Integrity constraints	Referential Integrity constraints	Level of referential	Domain Constraints
Faculty	3	4	None	None	3
Holiday	3	4	None	None	3
Level	3	4	None	None	3
Major	3	4	1	1	3
Part	3	4	None	None	3
Position	2	4	None	None	2
Group	2	4	None	None	2
Subject	5	4	None	None	5
Teacher	8	4	2	1	8
Course	3	4	None	None	3
Schedule	15	4	4	2	15

The tool generates test cases and follows database schema as shown in Table II. The numbers of test case are 111 test cases. The examples of test cases are shown in Figure 7. After that, we change a database schema by seeding the faults to database schema of software under test. The numbers of fault are 132 faults. The purpose of this experiment is for executing of test cases again and detecting faults. The results of experiment are shown in Table III.

```
1 INSERT INTO Faculty VALUES
                    (10,"SCEINCE","Sci",);
2 INSERT INTO Faculty VALUES
                    (10,"ENGINEERING","Eng",);
3 INSERT INTO Faculty VALUES
                    ("a","SCIENCE","Sci",);
4 INSERT INTO Faculty VALUES
                    (11,1000,"Sci",);
5 INSERT INTO Faculty VALUES
                    (12,"SCIENCE",1000,);
```

Figure 7. Examples of test cases are generated

After that, the test cases are executed again. The tool verifies the test results. The results show that all faults can be detected. Finally, the tool could generate test cases according to relational database constraints and the database design specifications.

TABLE III. THE RESULTS OF EXPERIMENT

System No.	Number of generated test cases	Database Type	Number of Seeded faults	Number of detected faults
1	111	MySQL	132	132
2	170	Oracle	220	220

REFERENCES

[1] Scott W.Ambler and Pramod J.Sadalage, "Refactoring databases:Evolutionary Database Design",Canada,2007H.

[2] K. Piriyakitpaiboon and T. Suwannasart, "RealGen: A Test Data Generation Tool to Support Software Testing",in the proceeding of the second international conference on information and communication technologies(ICT2004), Thailand,November 28-19, 2004.

[3] Ashutosh Gaur,"DeZign for databases to create ER diagrams", February 2007.Internet:http://www.orafaq.com /tools /heraut dezign.htm

[4] Dan A. Simonvici and Richard L. Tenney, "Relational Database Systems", USA, 1995.

[5] IBM Red Brick Warehouse 6.3 Information Center, "Referential Integrity Constraints", USA, 2007. Internet:
http://publib.boulder.ibm.com/infocenter

[6] P. Rojnavipad and T. Pensri, "Academic Affair Information System for Faculty of Science", Senior Project, Department of Computer Science,Faculty of Science, Rajmangala University of Technology Thanyaburi,2007.

[7] N. Satonggan, J.Chanpolrop, and S Rompon, "Reservation laboratory Online System", Senior Project,Department of Science,Faculty of Science, Rajmangala University of Technology Thanyaburi,2006.

ประวัติผู้เขียนวิทยานิพนธ์

นาย ประภาส ทองรัก เกิดเมื่อวันที่ 15 สิงหาคม พ.ศ. 2527 ที่อำเภอพระนครศรีอยุธยา จังหวัดพระนครศรีอยุธยา สำเร็จการศึกษาระดับประถมศึกษาจากโรงเรียนประตูชัย เมื่อปีการศึกษา 2539 ระดับมัธยมศึกษาจากโรงเรียนอยุธยาวิทยาลัย เมื่อปีการศึกษา 2544 สำเร็จการศึกษาวิทยาศาสตรบัณฑิต (เกียรตินิยมอันดับที่ 1) สาขาวิชาเทคโนโลยีคอมพิวเตอร์ จากภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี จังหวัดปทุมธานี ในปีการศึกษา 2548 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2549 ที่อยู่ที่สามารถติดต่อได้ 25/2 หมู่ที่ 7 ตำบลเกาะเรียน อำเภอพระนครศรีอยุธยา จังหวัดพระนครศรีอยุธยา 13000 โทรศัพท์ (087)-0784271 อีเมล์ prapas_mac@hotmail.com

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย