การควบคุมเวอร์ชันของข้อมูลเชิงวัตถุในระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

นางสาว จารุวรรณ สุขุมลักษณ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2543

ISBN 974-347-003-4

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

I20066582

VERSION CONTROL OF DATA OBJECTS IN AN OBJECT-RELATIONAL DATABASE MANAGEMENT SYSTEM

Miss Jaruwan Sukhumlug

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2000

ISBN 974-347-003-4

หัวข้อวิทยานิพนธ์	การควบคุมเวอร์ชันของข้อมูลเชิงวัตถุในระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์
โดย	นางสาว จารุวรรณ สุขุมลักษณ์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ จารุมาตร ปิ่นทอง

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วันชัย ริ้วไพบูลย์)

..... อาจารย์ที่ปรึกษา
(อาจารย์ จารุมาตร ปิ่นทอง)

.....ทวิตีย์ เสนีวงศ์ ณ อยุธยา..... กรรมการ
(อาจารย์ ดร. ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... กรรมการ
(อาจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จารุวรรณ สุขุมลักษณ์ : การควบคุมเวอร์ชันข้อมูลเชิงวัตถุในระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์. (VERSION CONTROL OF DATA OBJECTS IN AN OBJECT-RELATIONAL DATABASE MANAGEMENT SYSTEM) อ. ที่ปรึกษา : อ. จารุมาตร ปิ่นทอง, 107 หน้า. ISBN 974-347-003-4.

ปัจจุบันมีการนำข้อมูลหลายประเภทมาใช้ในระบบสารสนเทศ รวมทั้งเอนทิตีที่มีหลายค่าหรือหลายเวอร์ชัน โดยสามารถเลือกใช้งานเวอร์ชันใดของเอนทิตีก็ได้ตามต้องการ ถ้านำแนวคิดของฐานข้อมูลเชิงสัมพันธ์ซึ่งแพร่หลายในปัจจุบันมาใช้ จะได้ฐานข้อมูลที่ซับซ้อนยากต่อการพัฒนาและดูแลระบบ

วิทยานิพนธ์นี้มีจุดมุ่งหมายเพื่อออกแบบและพัฒนากลไกในการเก็บและใช้งานเวอร์ชันข้อมูลเชิงวัตถุบนระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยการนำแนวคิดเชิงวัตถุและแนวคิดเกี่ยวกับเวอร์ชันมาประยุกต์ใช้ สามารถกำหนดโครงสร้างความสัมพันธ์ของข้อมูลในระบบงานดังกล่าวได้ในรูปโครงสร้างลำดับชั้นของต้นไม้ที่เจาะจงเฉพาะความสัมพันธ์แบบคอมโพซิชัน งานวิจัยนี้กำหนดให้ใช้ตัวระบุออบเจคต์ในการเข้าถึงข้อมูลของออบเจคต์ และตัวระบุเวอร์ชันในการเข้าถึงข้อมูลของเวอร์ชัน รวมทั้งระบุว่าเวอร์ชันนั้นเป็นของออบเจคต์ใด และได้เสนอแบบจำลองข้อมูลเชิงวัตถุที่ประกอบด้วย 2 ส่วน คือ สกีมาออบเจคต์ของระบบงานวิจัยและสกีมาออบเจคต์ที่ผู้ใช้นิยาม การพัฒนาระบบงานทำโดยใช้ซอฟต์แวร์เจเดฟเวอลอปเปอร์ เวอร์ชัน 3.0 ร่วมกับระบบปฏิบัติการฐานข้อมูลเชิงวัตถุสัมพันธ์ออราเคิล เวอร์ชัน 8.1.6 โปรแกรมของระบบงานสนับสนุนฟังก์ชันการทำงานต่างๆ ที่เกี่ยวกับการเก็บเวอร์ชันข้อมูลเชิงวัตถุของกระบวนการออกแบบในฐานข้อมูล เช่น ฟังก์ชันในการกำหนดลำดับชั้นโครงสร้างต้นไม้ของโครงงานหนึ่งๆ ฟังก์ชันสำหรับกำหนดคุณลักษณะทางโครงสร้างของคลาสที่ผู้ใช้นิยาม ฟังก์ชันในการสร้างหรือดรอปคลาสที่ผู้ใช้นิยามในฐานข้อมูล ฟังก์ชันการแทรกแถวข้อมูล การแก้ไขค่าข้อมูลและการลบข้อมูลจากคลาสที่ผู้ใช้นิยาม ฟังก์ชันเกี่ยวกับการกำหนดเวอร์ชันของออบเจคต์ และฟังก์ชันที่แสดงโครงสร้างความสัมพันธ์แบบคอมโพซิชันของแต่ละโครงงาน เป็นต้น ระบบงานที่พัฒนาขึ้นนี้สามารถเก็บเวอร์ชันข้อมูลได้ 9,999,999,999,999 เวอร์ชัน

จากการทดสอบระบบงานวิจัยโดยใช้ตัวอย่างข้อมูลของการแต่งหนังสือ พบว่า แต่ละโครงงานจะประกอบด้วยลำดับชั้นโครงสร้างต้นไม้จำนวนเท่าไรก็ได้ และออบเจคต์รวมอาจประกอบด้วยออบเจคต์ย่อยจากต่างระดับกันหรือระดับเดียวกัน โดยออบเจคต์รวมต้องอยู่ในระดับชั้นที่สูงกว่าออบเจคต์ย่อย และออบเจคต์การออกแบบต้องอยู่ระดับที่ 1 ของโครงสร้างต้นไม้

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต จารุวรรณ สุขุมลักษณ์
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2543 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

4071415621 : MAJOR COMPUTER SCIENCE

KEY WORD: VERSION CONTROL / VERSIONS / DATA OBJECTS /
OBJECT-RELATIONAL DATABASE MANAGEMENT SYSTEM / DESIGN PROCESS

JARUWAN SUKHUMLUG : VERSION CONTROL OF DATA OBJECTS IN AN OBJECT-RELATIONAL DATABASE MANAGEMENT SYSTEM.

THESIS ADVISOR : CHARUMART PINTHONG, 107 pp. ISBN 974-347-003-4.

At present, many kinds of data are used in information systems, including data entities with multiple values or versions. A version of an entity can be selected for used by a user as required. However, maintaining versions of entities using the widely accepted relational database concept may result in a database that is both complex and difficult to maintain.

The objective of this thesis is to design and develop a mechanism for storing and accessing versions of data objects in an ORDBMS. Using concepts of objects and versions, this thesis focuses on the composition relationship between data objects which is represented as a tree. In this model, an object identification specifies a data object to be accessed and a version identification indicates the version of a particular object. The model comprises 2 schemas: schema of system objects and schema of user-defined objects. An implementation is developed using JDeveloper version 3.0 with Oracle ORDBMS release 8.1.6 and provides several functions for storing versioned objects of a design process. The functions include specifying the depth of the tree for a particular design project, defining the structure for user-defined classes, creating or dropping a user-defined class, inserting, updating, or deleting a data object, creating a new version of an object, and viewing of the composition hierarchy. This implementation can accommodate for 9,999,999,999,999 data versions.

The system has been tested with an example of a book composition project. The project can be modeled as a tree of any depth. A composite object may contain either other composite objects or elementary objects of a lower level. The object representing the project must be at the first level of the tree.

ภาควิชา Computer Engineering ลายมือชื่อนิสิต .....

สาขาวิชา Computer Science ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2543 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สามารถสำเร็จลุล่วงไปได้ด้วยความกรุณาจากอาจารย์ จารุมาตร ปิ่นทอง และ ดร. กนกลักษณ์ วนาพิพัฒน์ ซึ่งได้ให้ความช่วยเหลือและติดตามผลการวิจัย อีกทั้งได้สละเวลาในการให้คำปรึกษาและคำแนะนำต่างๆ มากมายอันเป็นประโยชน์อย่างยิ่งแก่ผู้วิจัยมาโดยตลอด

ขอขอบคุณรองศาสตราจารย์ ดร. วันชัย ริ้วไพบูลย์ อาจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา และอาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ ที่ได้สละเวลาในการสอบวิทยานิพนธ์

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ทั้งในบริษัทออราเคิลและในภาควิชาคอมพิวเตอร์ที่ได้ให้ความช่วยเหลือทางด้านซอฟท์แวร์ หนังสือ บทความ และคำแนะนำที่เป็นประโยชน์การออกแบบและพัฒนาโปรแกรม

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่เข้าใจ ให้การสนับสนุน ช่วยเหลือและให้กำลังใจมาโดยตลอด

จารุวรรณ สุขุมลักษณ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง
บทคัดย่อภาษาอังกฤษ..... จ
กิตติกรรมประกาศ..... ฉ
สารบัญ..... ช
สารบัญภาพ..... ญ
สารบัญตาราง..... ฐ

บทที่

1. บทนำ..... 1
   1.1. ความเป็นมาและความสำคัญของปัญหา..... 1
   1.2 วัตถุประสงค์..... 4
   1.3 ขอบเขตการวิจัย..... 4
   1.4 คำจำกัดความที่ใช้ในการวิจัย..... 4
   1.5 ขั้นตอนในการดำเนินงาน..... 5
   1.6 ประโยชน์ที่คาดว่าจะได้รับ..... 5
   1.7 เนื้อหาของวิทยานิพนธ์..... 5

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 7
   2.1 ทฤษฎีที่เกี่ยวข้อง..... 7
      2.1.1 แนวคิดพื้นฐานของแบบจำลองเชิงวัตถุ..... 7
      2.1.2 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์..... 10
      2.1.3 แบบจำลองเชิงวัตถุของ Coad..... 12
      2.1.4 เวอร์ชัน..... 13
   2.2 งานวิจัยที่เกี่ยวข้อง..... 15
   2.3 แนวคิดในการวิจัย..... 18
      2.3.1 แนวคิดเกี่ยวกับออบเจคต์และเวอร์ชันของออบเจคต์ 18
      2.3.2 คุณลักษณะทั่วไปของระบบในทางทฤษฎี..... 19

บทที่ หน้า

3. การออกแบบระบบ....21
3.1 แนวคิดในการออกแบบ....21
3.2 การออกแบบกระบวนการใช้งาน....23
3.3 การออกแบบฐานข้อมูล.... 24
3.4 ออบเจคต์ต่างๆ ในสกีมาของระบบ....25
3.4.1 ตารางในระบบฐานข้อมูล.... 25
3.4.2 วิว.... 32

4. การพัฒนาโปรแกรม.... 34
4.1 ผังโครงสร้างของโปรแกรม.... 34
4.2 ผังโครงสร้างหน้าจอของระบบ.... 36
4.3 โปรแกรมของระบบควบคุมเวอร์ชันข้อมูล.... 36
4.3.1 ไฟล์ต้นฉบับ....38
4.3.2 คลาสในไฟล์ต้นฉบับ....38

5. การทดสอบเพื่อประเมินผลการวิจัย.... 49
5.1 สภาพแวดล้อมของระบบงานที่ใช้ในการทดสอบ....49
5.2 ตัวอย่างข้อมูลที่ใช้ในการทดสอบ....49
5.3 การทดสอบการทำงานของระบบงานวิจัย.... 50
5.4 คุณลักษณะของระบบจากการทดสอบระบบ.... 51

6. สรุปผลการวิจัยและข้อเสนอแนะ....52
6.1 สรุปผลการวิจัย....52
6.2 ข้อจำกัด....52
6.3 ข้อเสนอแนะ.... 53

รายการอ้างอิง....55
ภาคผนวก.... 57
ภาคผนวก ก คู่มือการใช้งานระบบควบคุมเวอร์ชันข้อมูลเชิงวัตถุ....58

หน้า

ก.1 ปฏิบัติการต่างๆ ของโมดูล PROJECT..... 59
ก.2 ปฏิบัติการต่างๆ ของโมดูล TREE LEVEL..... 62
ก.3 ปฏิบัติการต่างๆ ของโมดูล APPLICATION CLASS..... 66
ก.4 ปฏิบัติการต่างๆ ของโมดูล FACT DATA..... 73
ก.5 ปฏิบัติการต่างๆ ของโมดูล USER CLASS..... 78
ก.6 ปฏิบัติการต่างๆ ของโมดูล OBJECT VERSIONS..... 81
ก.7 ปฏิบัติการต่างๆ ของโมดูล RELATIONSHIPS..... 85
ก.8 การใช้งานโมดูล VIEW HIERARCHY..... 89

ภาคผนวก ข ตัวอย่างข้อมูลที่ใช้ในการทดสอบ..... 93
ข.1 ตัวอย่างการแต่งหนังสือ..... 93
ข.1.1 โครงสร้างต้นไม้ของการแต่งหนังสือ..... 93
ข.1.2 ตารางแสดงข้อมูลของการแต่งหนังสือ..... 95

ภาคผนวก ค สคริปท์สำหรับสร้างออบเจคต์หลักในฐานข้อมูล..... 101
ค.1 สคริปท์สำหรับสร้างตารางและข้อบังคับต่างๆ ของตาราง..... 101
ค.2 สคริปท์สำหรับสร้างวิว..... 105

ประวัติผู้วิจัย..... 107

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ หน้า

รูปที่ 1.1 ตัวอย่างหนังสือและส่วนประกอบของหนังสือ....3
รูปที่ 1.2 เวอร์ชันในระดับส่วน ของบทที่ 1 3
รูปที่ 1.3 เวอร์ชันในระดับบท ของบทที่ 1....3
รูปที่ 2.1 ความสัมพันธ์แบบแอกกรีเกชัน.... 9
รูปที่ 2.2 ความสัมพันธ์แบบคอมโพซิชัน.... 9
รูปที่ 2.3 การใช้ แอนด์/ออร์ กราฟในการแสดงประเภทต่างๆ ของลำดับการเลือก.... 16
รูปที่ 2.4 ความสัมพันธ์ระหว่าง Design Object กับ Elementary Object.... 19
รูปที่ 2.5 ความสัมพันธ์ระหว่าง Object กับ Version.... 19
รูปที่ 3.1 ตัวอย่างหนังสือและส่วนประกอบของหนังสือ....22
รูปที่ 3.2 แบบจำลองเชิงออบเจคต์ของการออกแบบฐานข้อมูล....26
รูปที่ 4.1 ผังโครงสร้างของโปรแกรมของงานวิจัย....35
รูปที่ 4.2 ผังโครงสร้างหน้าจอของระบบ....37
รูปที่ 5.1 ตัวอย่างหนังสือ Tales และส่วนประกอบของหนังสือ....50
รูป ก1 หน้าจอหลักของระบบควบคุมเวอร์ชันข้อมูลเชิงวัตถุ....58
รูป ก2 หน้าจอสอบถามของโมดูล PROJECT.... 59
รูป ก3 หน้าจอสร้างออบเจคต์ของโมดูล PROJECT....60
รูป ก4 หน้าจอย่อยแสดงค่าข้อมูลแอตทริบิวท์ของโครงงาน Project1.... 61
รูป ก5 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล PROJECT: ก่อนการแก้ไข....62
รูป ก6 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบแถวข้อมูลจากตาราง PROJECT0....62
รูป ก7 หน้าจอสอบถามของโมดูล TREE LEVEL....63
รูป ก8 หน้าจอสร้างออบเจคต์ของโมดูล TREE LEVEL....64
รูป ก9 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล TREE LEVEL: ก่อนการแก้ไข.... 65
รูป ก10 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบแถวข้อมูลจากตาราง TREE_LEVEL0..65
รูป ก11 หน้าจอสอบถามของโมดูล APPLICATION CLASS.... 67
รูป ก12 หน้าจอย่อยแสดงรายละเอียดของระดับขั้นโครงสร้างต้นไม้ทั้งหมดในระบบ.... 67
รูป ก13 หน้าจอสร้างออบเจคต์ของโมดูล APPLICATION CLASS....68
รูป ก14 หน้าจอย่อยแสดงคุณลักษณะของแอตทริบิวท์ทั้งหมดของคลาส Tales....69

ภาพประกอบ หน้า

รูป ก15 หน้าจอแก้ไขออบเจคต์ของโมดูล APPLICATION CLASS: ก่อนการแก้ไข.....70
รูป ก16 หน้าจอย่อยแสดงข้อความเตือนในกรณีที่ผู้ใช้ต้องการแก้ไขแอตทริบิวท์ของคลาส ซึ่งถูกสร้างเป็นตารางในฐานข้อมูลแล้ว.....71
รูป ก17 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบคลาสออกจากตาราง APP_CLASS.... 72
รูป ก18 หน้าจอย่อยแสดงข้อความเตือนในกรณีที่ผู้ใช้ต้องการลบคลาสซึ่งถูกสร้างเป็นตารางในฐานข้อมูลแล้ว.....72
รูป ก19 หน้าจอย่อยแสดงข้อความเตือนในกรณีที่ผู้ใช้ต้องการดรอปตารางซึ่งยังไม่ถูกสร้างขึ้นในฐานข้อมูล.....73
รูป ก20 หน้าจอย่อยแสดงข้อความขอคำยืนยันการดรอปตาราง..... 73
รูป ก21 หน้าจอสอบถามของโมดูล FACT DATA.....74
รูป ก22 ข้อมูลดิบของออบเจคต์ข้อมูลดิบ.....75
รูป ก23 หน้าจอสร้างออบเจคต์ของโมดูล FACT DATA.....76
รูป ก24 หน้าจอย่อยสำหรับเลือกไฟล์ข้อมูลดิบให้กับออบเจคต์ข้อมูลดิบ..... 76
รูป ก25 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล FACT DATA: ก่อนการแก้ไข..... 77
รูป ก26 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบออบเจคต์ข้อมูลดิบจากตาราง FACTDATA..... 78
รูป ก27 หน้าจอสอบถามของโมดูล USER CLASS.....79
รูป ก28 หน้าจอสร้างออบเจคต์ของโมดูล USER CLASS.....79
รูป ก29 หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ในคลาส Tales: ก่อนการแก้ไข.....80
รูป ก30 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบออบเจคต์ของคลาสที่ผู้ใช้นิยาม.....81
รูป ก31 หน้าจอสอบถามของโมดูล OBJECT VERSIONS.....82
รูป ก32 หน้าจอสร้างออบเจคต์ของโมดูล OBJECT VERSIONS.....83
รูป ก33 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล OBJECT VERSIONS: ก่อนการแก้ไข..... 84
รูป ก34 หน้าจอย่อยแสดงรายละเอียดของออบเจคต์ข้อมูลดิบทั้งหมดในระบบ.....84
รูป ก35 หน้าจอย่อยแสดงข้อมูลดิบที่สัมพันธ์กับเวอร์ชันของออบเจคต์หนึ่ง.....85
รูป ก36 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบแถวข้อมูลจากตาราง OBJ_VERSION..86
รูป ก37 หน้าจอของโมดูล RELATIONSHIPS: แสดงข้อมูลของเวอร์ชันลูกทั้งหมดที่สัมพันธ์กับเวอร์ชันแม่.....86

ภาพประกอบ หน้า

รูป ก38 หน้าจอแสดงการสร้างออบเจคต์ความสัมพันธ์: ลำดับที่ของออบเจคต์นี้มีค่าเป็น 1.... 88
รูป ก39 หน้าจอแสดงการสร้างออบเจคต์ความสัมพันธ์: ลำดับที่ของออบเจคต์นี้มีค่าเป็น 2.... 89
รูป ก40 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบออบเจคต์ความสัมพันธ์..... 89
รูป ก41 ระบบแสดงเวอร์ชันของออบเจคต์ทั้งหมดที่อยู่ในระดับโครงสร้างต้นไม้ซึ่งผู้ใช้เลือก.... 90
รูป ก42 ระบบแสดงเวอร์ชันลูกทั้งหมดของเวอร์ชันแม่ VID: 2..... 91
รูป ก43 ระบบแสดงเวอร์ชันลูกทั้งหมดของเวอร์ชันแม่ VID: 9..... 92
รูป ข1 โครงสร้างต้นไม้ของตัวอย่างข้อมูลการเขียนหนังสือ..... 94

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง หน้า

ตารางที่ 2.1 สัญกรณ์และคำอธิบายในแบบจำลองเชิงวัตถุของ Coad..... 12

ตารางที่ 3.1 การใช้งานของคลาส ในแผนภาพแบบจำลองข้อมูลเชิงวัตถุ..... 27

ตารางที่ 3.2 แอตทริบิวท์และข้อบังคับของตาราง PROJECT0..... 28

ตารางที่ 3.3 แอตทริบิวท์และข้อบังคับของตาราง TREE_LEVEL0.....29

ตารางที่ 3.4 แอตทริบิวท์และข้อบังคับของตาราง ATTRLIST_TAB.....29

ตารางที่ 3.5 แอตทริบิวท์และข้อบังคับของตาราง APP_CLASS..... 30

ตารางที่ 3.6 แอตทริบิวท์และข้อบังคับของตาราง FACTDATA.....30

ตารางที่ 3.7 แอตทริบิวท์และข้อบังคับของตาราง OBJ_INSTANCE..... 31

ตารางที่ 3.8 แอตทริบิวท์และข้อบังคับของตาราง OBJ_VERSION.....31

ตารางที่ 3.9 แอตทริบิวท์และข้อบังคับของตาราง ISPARTOF.....32

ตารางที่ 3.10 คุณลักษณะของวิว และตารางฐาน.....33

ตารางที่ 4.1 ความสัมพันธ์ระหว่างไฟล์ต้นฉบับกับโมดูลและตารางในระบบฐานข้อมูล..... 39

ตารางที่ 4.2 คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ.....42

ตาราง ข1 ตัวอย่างค่าข้อมูลของตาราง PROJECT0 ในระบบฐานข้อมูล.....95

ตาราง ข2 ตัวอย่างค่าข้อมูลของตาราง TREE_LEVEL0 ในระบบฐานข้อมูล..... 95

ตาราง ข3 ตัวอย่างค่าข้อมูลของตาราง APP_CLASS ในระบบฐานข้อมูล.....95

ตาราง ข4 ตัวอย่างค่าข้อมูลของตาราง FACTDATA ในระบบฐานข้อมูล..... 96

ตาราง ข5 ตัวอย่างค่าข้อมูลของตาราง SECTION ในระบบฐานข้อมูล.....97

ตาราง ข6 ตัวอย่างค่าข้อมูลของตาราง CHAPTER ในระบบฐานข้อมูล.....97

ตาราง ข7 ตัวอย่างค่าข้อมูลของตาราง TALES ในระบบฐานข้อมูล.....97

ตาราง ข8 ตัวอย่างค่าข้อมูลของตาราง OBJ_VERSION ในระบบฐานข้อมูล..... 97

ตาราง ข9 ตัวอย่างค่าข้อมูลของตาราง ISPARTOF ในระบบฐานข้อมูล..... 99

ตาราง ค1 สคริปท์สำหรับสร้างตาราง PROJECT0.....101

ตาราง ค2 สคริปท์สำหรับสร้างตาราง TREE_LEVEL0..... 102

ตาราง ค3 สคริปท์สำหรับสร้างตาราง FACTDATA..... 102

ตาราง ค4 สคริปท์สำหรับสร้างตาราง APP_CLASS.....103

ตาราง ค5 สคริปท์สำหรับสร้างตาราง OBJ_INSTANCE.....103

ตาราง ค6 สคริปท์สำหรับสร้างตาราง OBJ_VERSION..... 104

ตาราง หน้า

ตาราง ค7 สคริปท์สำหรับสร้างตาราง ISPARTOF..... 104
ตาราง ค8 สคริปท์สำหรับสร้างวิว TLEV_VIEW..... 105
ตาราง ค9 สคริปท์สำหรับสร้างวิว FACTDATA_VIEW..... 105
ตาราง ค10 สคริปท์สำหรับสร้างวิว APPCLASS_VIEW..... 106
ตาราง ค11 สคริปท์สำหรับสร้างวิว VERSION_VIEW..... 106

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

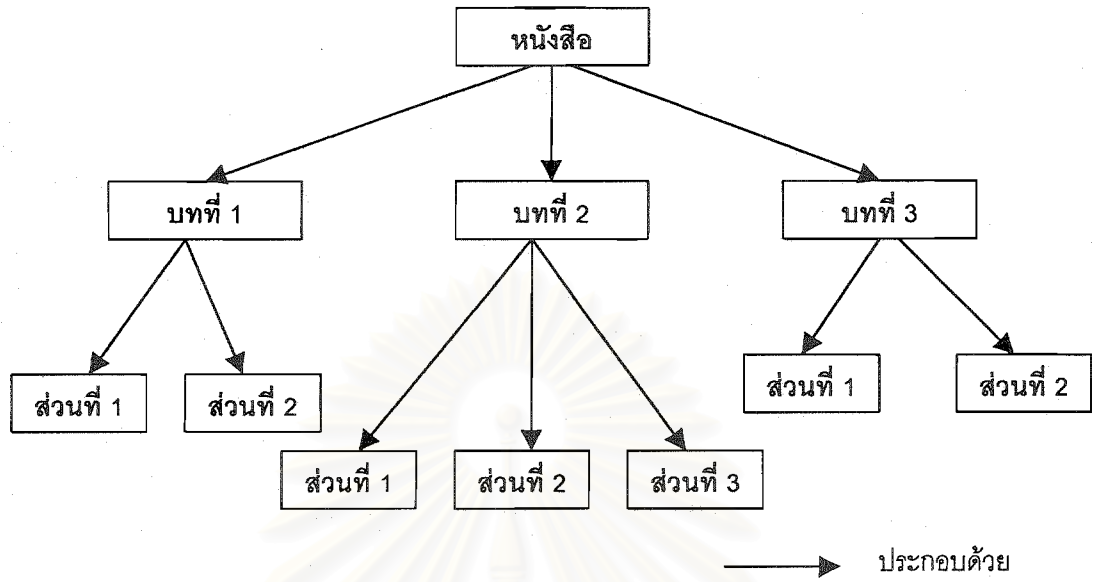
ระบบฐานข้อมูล (Traditional Database System) ที่มีอยู่ส่วนใหญ่ใช้ในการเก็บข้อมูลของระบบงานทางการค้า (Commercial Application) ซึ่งเป็นข้อมูลที่มีโครงสร้างที่แน่นอน (Structured Data) และมีการเปลี่ยนแปลงไม่มากนัก โดยฐานข้อมูลจะเก็บค่าข้อมูลเพียงเวอร์ชันเดียวเท่านั้น เมื่อมีรายการปรับปรุง (Update Transaction) เกิดขึ้นระบบจัดการฐานข้อมูล (Database Management System) จะทำการเปลี่ยนค่าของข้อมูลจากค่าหนึ่งไปเป็นอีกค่าหนึ่ง และทันทีที่เกิดรายการความสำเร็จสมบูรณ์ (Commit Transaction) ขึ้น ระบบจัดการฐานข้อมูลก็จะเขียนทับสถานะความคงตัวเก่า (Consistent State) ของฐานข้อมูลด้วยสถานะความคงตัวใหม่ โดยไม่มีการเก็บค่าข้อมูลเก่าเอาไว้ ถ้าหากระบบงานต้องการใช้ทั้งข้อมูลค่าเก่าและค่าใหม่มาประมวลผลรวมกัน จะทำให้ไม่สามารถนำค่าข้อมูลเก่าก่อนการเปลี่ยนแปลงมาใช้งานได้ ผู้ใช้จะใช้ได้เฉพาะค่าข้อมูลล่าสุดเท่านั้น เป็นเหตุให้มีการพัฒนาระบบฐานข้อมูลให้สามารถทำการเก็บข้อมูลหลายเวอร์ชัน (Version) ได้ ตัวอย่างต่อไปนี้เป็นวัตถุประสงค์ในการใช้เวอร์ชันของข้อมูล[1]

- ใช้เวอร์ชันในการควบคุมการทำงานพร้อมกัน (Concurrency Control) เพื่อเพิ่มความน่าเชื่อถือ (Reliability)ของระบบ
- ใช้เวอร์ชันของข้อมูลในลักษณะค่าข้อมูล ณ. จุดหนึ่ง (Snapshot) เพื่อเพิ่มประสิทธิผล (Performance) ของระบบแบบกระจาย (Distributed Systems) เวอร์ชันของข้อมูลนี้มีลักษณะเป็นการทำสำเนาของข้อมูลชุดเดียวกันแต่ถูกเก็บอยู่ในสถานที่ต่างๆ กัน โดยบางสถานที่ยังคงเก็บเวอร์ชันเก่าของข้อมูลไว้ใช้งานเป็นระยะเวลาสั้นๆ เพื่อลดปริมาณการส่งผ่านข้อมูลบนระบบเครือข่าย (Network System) ถึงแม้จะทำให้เกิดความขัดแย้งกันของข้อมูล (Data Inconsistency) แต่ก็เป็นการเกิดขึ้นในช่วงเวลาสั้นๆ เท่านั้น
- ใช้เวอร์ชันในการเก็บข้อมูลทะเบียนประวัติ (Historical Data) [2] ต่างๆ เช่น ทะเบียนประวัติเกี่ยวกับการเลื่อน/ดำรงตำแหน่งของข้าราชการ ทะเบียนประวัติเงินเดือนของพนักงาน เป็นต้น โดยที่แต่ละเวอร์ชันของข้อมูลมีความสัมพันธ์กันตามลำดับเวลาการเกิดขึ้นของข้อมูล
- ใช้เวอร์ชันในกระบวนการออกแบบของระบบงาน โดยเฉพาะกระบวนการออกแบบระบบงานทางวิศวกรรม มักจะมีเครื่องมือช่วยในการออกแบบ เช่น การออกแบบวงจร

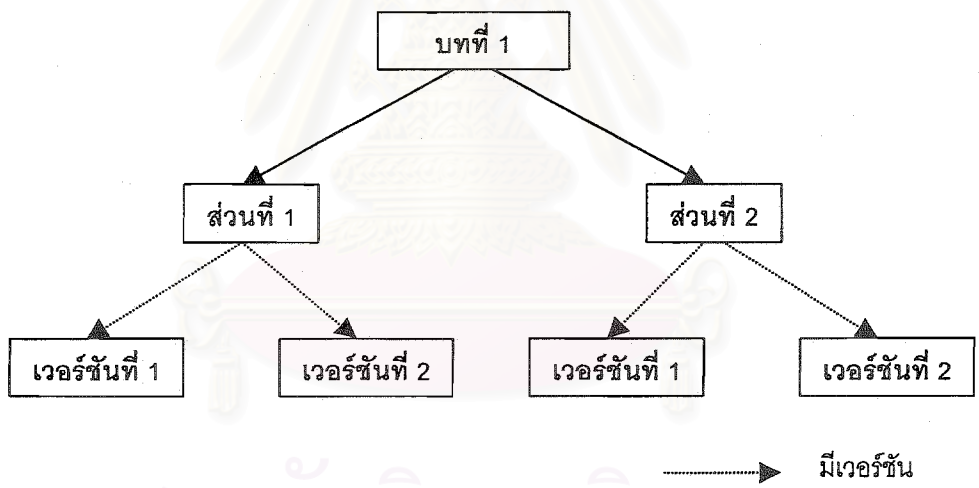
วีแอลเอสไอ (VLSI) จะใช้แคด (Computer Aided Design) การออกแบบระบบทางคอมพิวเตอร์จะใช้เคส (Computer Assisted Software Engineering) เป็นต้น นอกจากนี้ยังมีการใช้เวอร์ชันในโปรแกรมประยุกต์ทางด้านการจัดการเอกสารอีกด้วย เนื่องมาจาก การสร้างและจัดการกับระบบสารสนเทศของสำนักงาน (Office Information System) มีรูปแบบคล้ายคลึงกับกระบวนการออกแบบ

งานวิจัยนี้จะขอพิจารณาเฉพาะการเก็บข้อมูลของกระบวนการออกแบบในระบบฐานข้อมูลเท่านั้น จากเหตุผลที่ได้กล่าวมาแล้ว ประกอบกับแต่ละเวอร์ชันของข้อมูลในขั้นตอนการออกแบบหนึ่งๆ เป็นข้อมูลที่การเปลี่ยนแปลงเกิดขึ้นทีละน้อย โดยมีการเลือกใช้ข้อมูลเพียงหนึ่งเวอร์ชันในแต่ละขั้นตอนการออกแบบเท่านั้น และข้อมูลเหล่านี้จะถูกนำมาประกอบกันตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้ายเพื่อให้ได้ผลลัพธ์ที่ตรงตามวัตถุประสงค์ของการออกแบบ จะเห็นได้ว่าข้อมูลในกระบวนการออกแบบนั้นมีโครงสร้างที่ซับซ้อนและไม่แน่นอน [3] ในที่นี้จะใช้การแต่งหนังสือเป็นตัวอย่าง เพื่อให้ง่ายต่อการทำความเข้าใจ กำหนดให้หนังสือเล่มนี้ประกอบด้วย 3 บท บทที่ 1 ประกอบด้วย 2 ส่วน บทที่ 2 ประกอบด้วย 3 ส่วน และบทที่ 3 ประกอบด้วย 2 ส่วน (ดังรูปที่ 1.1) ในกระบวนการแต่งหนังสือนั้น หากผู้แต่งทำการเพิ่มเติมหรือปรับปรุงข้อความเก่า แล้วทำการเก็บทับไปที่เดิม จะทำให้ข้อความเก่าก่อนการแก้ไขปรับปรุงหายไป ต่อมาผู้แต่งอาจเปลี่ยนใจต้องการข้อความเก่ามาแก้ไขใหม่โดยไม่สนใจข้อความล่าสุดที่เพิ่งจะเก็บไป จะทำให้ไม่สามารถนำข้อความเก่าก่อนหน้ามาใช้งานได้ ดังนั้นแทนการเก็บข้อความใหม่หลังการปรับปรุงทับข้อความเก่าก่อนการปรับปรุง ให้ผู้แต่งเก็บข้อความใหม่ด้วยการเพิ่มเข้าไปในระบบพร้อมกับยังคงเก็บข้อความเดิมเอาไว้ การเก็บข้อความในลักษณะนี้เป็นการเก็บแบบเวอร์ชัน ดังนั้น แต่ละส่วนของหนึ่งบทอาจจะประกอบด้วยหลายๆ เวอร์ชัน (ดังรูปที่ 1.2) ในระดับของบทและหนังสือ (ดังรูปที่ 1.3) ก็จะมีลักษณะเช่นเดียวกัน

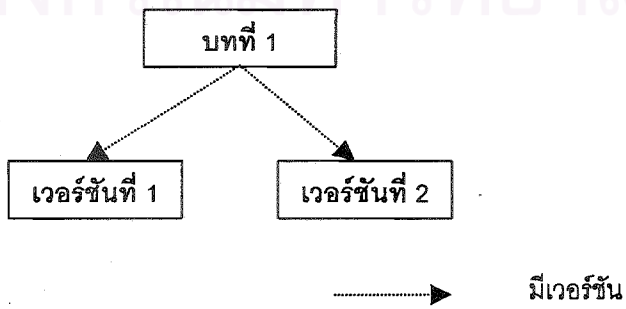
เมื่อนำแนวความคิดเกี่ยวกับออบเจคต์ (Object) มาประยุกต์ใช้กับข้อมูลกระบวนการออกแบบ ออบเจคต์จะสามารถแทนข้อมูลและความสัมพันธ์ระหว่างข้อมูลได้อย่างถูกต้องครบถ้วน จากตัวอย่างการแต่งหนังสือ เราอาจมองได้ว่าหนังสือ บท และส่วนเป็นออบเจคต์ ซึ่งหากใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: DBMS) จะไม่สามารถจัดการกับข้อมูลนี้ได้อย่างมีประสิทธิภาพ เนื่องจากมีชนิดข้อมูลพื้นฐาน (Primitive Data Type) เท่านั้น ประกอบกับบริษัทผู้ค้าผลิตภัณฑ์ทางด้านระบบจัดการฐานข้อมูลเชิงสัมพันธ์ได้เพิ่มคุณสมบัติบางประการของแบบจำลองเชิงออบเจคต์เข้าไปในผลิตภัณฑ์เวอร์ชันเก่าของตน กลายเป็นระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object-Relational Database Management System) เพื่อความสะดวกในการปรับปรุงผลิตภัณฑ์เวอร์ชันเก่ามาเป็นเวอร์ชันใหม่ ที่มีความ



รูปที่ 1.1 ตัวอย่างหนังสือ และส่วนประกอบของหนังสือ



รูปที่ 1.2 เวอร์ชันในระดับส่วน ของบทที่ 1



รูปที่ 1.3 เวอร์ชันในระดับบท ของบทที่ 1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.2 วัตถุประสงค์

เพื่อออกแบบและพัฒนากลไกในการควบคุมการเก็บและใช้งานเวอร์ชันข้อมูลเชิงวัตถุ ในระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

1.3 ขอบเขตการวิจัย

1. ใช้ออราเคิล (Oracle) เวอร์ชัน 8.1.6 เป็นระบบจัดการฐานข้อมูล
2. ใช้ซอฟต์แวร์ช่วยพัฒนา (Development Tools) ชื่อ JDeveloper เวอร์ชัน 3.0 ในการพัฒนาโปรแกรม
3. ระบบควบคุมการเก็บและใช้งานเวอร์ชันข้อมูลนี้จะเป็นการควบคุมเวอร์ชันของออบเจคต์ในระดับโปรแกรมประยุกต์ โดยจะพิจารณาเฉพาะข้อมูลของกระบวนการออกแบบ
4. ระบบควบคุมการเก็บและใช้งานเวอร์ชันข้อมูลจะทำงานบนระบบฐานข้อมูลเชิงวัตถุสัมพันธ์
5. แต่ละเวอร์ชันของออบเจคต์การออกแบบ (Design Object) ประกอบด้วยจำนวนของออบเจคต์ย่อย (Elementary Object) ที่เท่ากัน และต้องเป็นเวอร์ชันที่มีอยู่แล้วของออบเจคต์ย่อยเท่านั้น
6. ใช้การแต่งหนังสือหรือเขียนบทความเป็นกรณีศึกษา โดยกำหนดโครงสร้างลำดับชั้นแบบรวมของคลาสมี 3 ระดับ ดังรูปที่ 1.1

1.4 คำจำกัดความที่ใช้ในการวิจัย

เพื่อให้ผู้ศึกษางานวิจัยสามารถทำความเข้าใจเนื้อหาของงานวิจัยได้มากขึ้น จึงกำหนดความหมายของคำศัพท์บางคำที่ปรากฏในงานวิจัย ดังต่อไปนี้

ออบเจคต์การออกแบบ (Design Object) หมายถึง สิ่งที่ผู้ใช้ต้องการออกแบบ เช่น หนังสือ เป็นต้น

ออบเจคต์ย่อย (Elementary Object) หมายถึง สิ่งที่เป็นส่วนประกอบของออบเจคต์การออกแบบ เช่นแต่ละบทของหนังสือหนึ่งเล่ม แต่ละส่วนในหนึ่งบท เป็นต้น

ข้อมูลดิบ (Fact Data) หมายถึง ข้อมูลดิบที่ผู้ใช้สร้างขึ้น เช่น ข้อความในส่วนย่อย รูปของเอนทิตีย่อย เป็นต้น

พลตำรวจตรี ดร.วิทยา วิทยาการ
รองอธิการบดีฝ่ายวิทยาการ

ฟรีซ (Freeze) เป็นสถานะของออบเจคต์หรือเวอร์ชันของออบเจคต์ที่ไม่มีการเปลี่ยนแปลงและพร้อมที่จะนำไปใช้งานได้ หรือคือข้อมูลที่ทำปฏิบัติการความสำเร็จสมบูรณ์ (commit) แล้ว

1.5 ขั้นตอนในการดำเนินงาน

1. ศึกษาทฤษฎีที่เกี่ยวข้อง ได้แก่
   1.1. ทฤษฎีและแนวความคิดในการออกแบบโมเดลของระบบควบคุมเวอร์ชันของข้อมูล
   1.2. ทฤษฎีเกี่ยวกับการสร้างแบบจำลองข้อมูลทางความหมาย (Semantic Data Modeling: SDM) ของระบบฐานข้อมูลเชิงวัตถุ
   1.3. ทฤษฎีเกี่ยวกับวิธีในการกำหนดขีดแบ่ง (Threshold) และวิธีในการจัดการกับการแพร่กระจายการเปลี่ยนแปลง (Propagation Change)
2. ศึกษาการใช้งานของระบบจัดการฐานข้อมูลออราเคิล เวอร์ชัน 8
3. ศึกษาซอฟต์แวร์ช่วยพัฒนาที่ใช้ในการพัฒนาระบบ
4. ออกแบบและพัฒนาระบบควบคุมเวอร์ชันของข้อมูล
5. ทดสอบและประเมินผลการทำงานของระบบ
6. สรุปผลการวิจัย เสนอแนะ และจัดทำรูปเล่มวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อเพิ่มความสามารถในการจัดเก็บและจัดการกับเวอร์ชันของข้อมูลการออกแบบให้กับระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์
2. สามารถนำแนวคิดของงานวิจัยนี้ไปประยุกต์ใช้กับ ระบบงานที่เกี่ยวกับข้อมูลทะเบียนประวัติและระบบงานการออกแบบประเภทอื่นๆ

1.7 เนื้อหาของวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์เล่มนี้แบ่งออกเป็น 6 บทกับ 3 ภาคผนวก โดยบทที่ 1 เป็นบทนำ บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่ผู้วิจัยนำมาใช้ในการทำวิทยานิพนธ์ นอกจากนี้ยังได้กล่าวถึงแนวคิดในการออกแบบและพัฒนาระบบงานวิจัยอย่างคร่าวๆ ซึ่งเป็นแนวคิดทางทฤษฎี บทที่ 3 กล่าวถึงการออกแบบระบบในขั้นตอนต่างๆ ของการสร้างระบบงานวิจัย เช่น การออกแบบกระบวนการใช้งานระบบ การออกแบบฐานข้อมูล และการออกแบบส่วนต่อประสานผู้ใช้ เป็นต้น นอกจากนี้จะอธิบายสกีมาออบเจคต์ที่ถูกสร้างในฐานข้อมูล คุณลักษณะของออบเจคต์ และการใช้งานออบเจคต์เหล่านั้นในระบบ หลังจากขั้นตอนการออกแบบระบบและสร้างออบเจคต์ในฐาน

ข้อมูลเพื่อเก็บข้อมูลต่างๆ ของระบบแล้ว ขั้นต่อไปจะเป็นการพัฒนาโปรแกรม ซึ่งได้กล่าวไว้ในบทที่ 4 บทนี้เริ่มด้วยการอธิบายรายละเอียดของการออกแบบโปรแกรม ประกอบกับแสดงผังโครงสร้างหน้าจอของโปรแกรม จากนั้นจะกล่าวถึงหน้าที่การทำงานของส่วนต่างๆ ในโปรแกรม บทที่ 5 เป็นการทดสอบระบบงานวิจัย โดยกำหนดว่าระบบจะต้องถูกติดตั้งบนสภาพแวดล้อมทางฮาร์ดแวร์และซอฟต์แวร์อย่างไร จึงจะสามารถทำงานได้ และแสดงผลการทดสอบการทำงานของระบบ บทที่ 6 เป็นการสรุปผลการวิจัย ข้อจำกัดของระบบและข้อเสนอแนะต่างๆ สำหรับปรับปรุงระบบงานวิจัยนี้ให้ดียิ่งขึ้น เนื้อหาในส่วนของภาคผนวกนั้นจะเกี่ยวกับวิธีการใช้งานระบบงานวิจัย ตัวอย่างข้อมูลที่ใช้ในการทดสอบงานวิจัย และสคริปท์สำหรับสร้างสกีมาออบเจคต์ในฐานข้อมูล ดังได้กล่าวไว้ใน ภาคผนวก ก ข และ ค ตามลำดับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องในการออกแบบและพัฒนาระบบการควบคุมเวอร์ชันของข้อมูลเชิงวัตถุในระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องและใช้ในการวิจัย ได้แก่

2.1.1 แนวคิดพื้นฐานของแบบจำลองเชิงวัตถุ (Basic Concept of Object Model)

2.1.2 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

2.1.3 แบบจำลองเชิงวัตถุของ Coad (Coad's Object Model)

2.1.4 เวอร์ชัน

2.1.1 แนวคิดพื้นฐานของแบบจำลองเชิงวัตถุ [4,5]

ออบเจคต์และคลาส (Classes)

ออบเจคต์ เป็นเอนทิตีที่เอนแคปซูเลต (Encapsulate) คุณลักษณะทางโครงสร้าง (Structural Characteristics) และพฤติกรรม (Behavior Characteristics) เข้าไว้ด้วยกัน โดยเรียกคุณลักษณะทางโครงสร้างว่าแอตทริบิวท์ (Attribute) และเรียกพฤติกรรมว่าเมธทอด (Method) ออบเจคต์เป็นอินสแตนซ์ (Instance) ของคลาส และมีตัวระบุออบเจคต์ (Object Identifier: OID) ซึ่งมีค่าเฉพาะ (Unique) ที่ทำให้ออบเจคต์หนึ่งแตกต่างจากออบเจคต์อื่นๆ

คลาส นิยามคุณลักษณะที่เหมือนกันของหลายๆ ออบเจคต์ เช่นเดียวกับออบเจคต์ คลาสจะเอนแคปซูเลตแอตทริบิวท์และเมธทอดไว้ โดยแอตทริบิวท์เป็นคุณลักษณะที่เกี่ยวกับสถานะและคุณสมบัติของคลาสหรือของออบเจคต์ภายในคลาส เมธทอดเป็นปฏิบัติการ (Operations) ที่กระทำกับออบเจคต์ภายในคลาสหรือกระทำกับคลาสเอง แบ่งปฏิบัติการออกเป็นประเภทต่างๆ ตามวงจรชีวิตของคลาส ได้ดังต่อไปนี้ [6]

- ปฏิบัติการ Initializer เป็นปฏิบัติการของคลาสที่กระทำโดยปริยายเพื่อเริ่มต้นคลาส
- ปฏิบัติการ Constructor เป็นปฏิบัติการของคลาสที่ใช้ในการสร้างออบเจคต์ภายในคลาส

- ปฏิบัติการ Accessors หรือ Selector เป็นปฏิบัติการที่ใช้ในการเรียกดูข้อมูลของออบเจคต์หรือของคลาส

- ปฏิบัติการ Mutator เป็นปฏิบัติการที่ใช้ในการกำหนดค่าให้กับออบเจคต์หรือคลาส

- ปฏิบัติการ Destructor เป็นปฏิบัติการของคลาสที่ใช้ในการลบออบเจคต์ออกจากคลาส

- ปฏิบัติการ Finalizer เป็นปฏิบัติการของคลาสที่กระทำกับคลาสในลักษณะตรงกันข้ามกับปฏิบัติการ Initializer โดยจะถูกเรียกกระทำหลังจากการทำลายออบเจคต์สุดท้ายของคลาส

คุณสมบัติของแบบจำลองเชิงวัตถุ [7]

หลักนามธรรมข้อมูล (Data Abstraction) เป็นการเพิ่มความสามารถให้กับการนิยามชนิดของข้อมูล ซึ่งทำให้ผู้ใช้สามารถนิยามชนิดของข้อมูลที่มีความซับซ้อนและตรงตามลักษณะข้อมูลในระบบงานจริงได้มากขึ้น เราเรียกชนิดของข้อมูลนี้ว่า ชนิดข้อมูลนามธรรม (Abstract Data Type: ADT)

เอนแคปซูเลชัน (Encapsulation) เป็นการรวมเอาส่วนย่อยที่เป็นข้อมูลและส่วนย่อยที่เป็นปฏิบัติการของออบเจคต์เข้าไว้ด้วยกัน ตามแนวคิดของระบบงานจริง เอนแคปซูเลชันก่อให้เกิดการซ่อนข้อมูล (Information Hiding) กล่าวคือ ออบเจคต์จะส่งคำร้องขอ (Request) ถึงกันผ่านทางส่วนต่อประสาน (Interface) ของคลาสโดยใช้ข้อความ (Message) ทำให้ผู้ใช้ไม่ต้องสนใจรายละเอียดในการพัฒนาโปรแกรมของระบบ

การสืบทอด (Inheritance) เกี่ยวข้องกับการสร้างสกีมา (Schema) โดยใช้ความสัมพันธ์ของสิ่งต่างๆ ในระบบงานจริง ร่วมกับการใช้ซ้ำคุณลักษณะของคลาสแม่ (Parent Class) โดยคลาสลูก (Child Class)

ลิงค์และแอสโซซิเอชั่น (Links and Associations) [6]

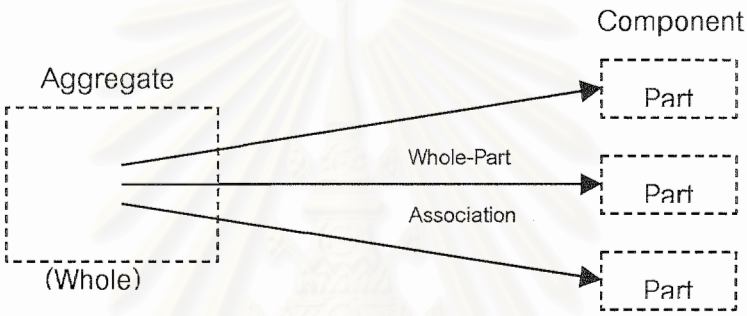
ลิงค์เป็นออบเจคต์ที่แสดงความสัมพันธ์ระหว่างเอนทิตีหนึ่งกับเอนทิตีอื่นๆ แอสโซซิเอชัน เป็นคลาสที่แสดงความสัมพันธ์ระหว่างออบเจคต์กับออบเจคต์ หรือคลาสกับคลาส อินสแตนซ์ของแอสโซซิเอชันเป็นเซตของลิงค์ซึ่งมีคุณลักษณะบางประการเหมือนกัน ประเภทของลิงค์และแอสโซซิเอชัน ได้แก่

1. ความสัมพันธ์แบบเจนเนอรัลไลท์เซชัน (Generalizations) หรือเรียกอีกอย่างว่าความสัมพันธ์แบบอีส-อะ (IS-A)

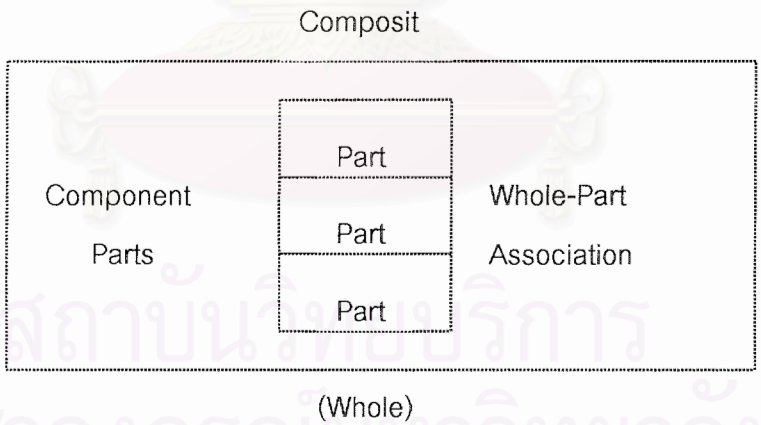
เป็นความสัมพันธ์ระหว่างซูเปอร์คลาส (Superclass) กับสับคลาส (Subclass) โดยซูเปอร์คลาสจะมีแอตทริบิวท์และเมธทอดที่แสดงคุณลักษณะร่วมกันของสับคลาสทั้งหมด (Generalization) สับคลาสจะสืบทอดแอตทริบิวท์และเมธทอดของซูเปอร์คลาส สับคลาสอาจมีคุณลักษณะเฉพาะบางประการที่เพิ่มเติมจากซูเปอร์คลาส (Specialization)

2. ความสัมพันธ์แบบแอกกรีเกชัน (Aggregations) หรือเรียกอีกอย่างว่าความสัมพันธ์แบบแฮส-อะ (HAS-A)

เป็นความสัมพันธ์ระหว่างเอนทิตีหนึ่งที่ประกอบด้วยหลายๆ เอนทิตีย่อย โดยเอนทิตีย่อยเหล่านี้เป็นอิสระจากเอนทิตีรวม ความสัมพันธ์ในลักษณะนี้เป็นความสัมพันธ์อย่างหลวม (Loosely-Coupled) ดังรูปที่ 2.1[1]



รูปที่ 2.1 ความสัมพันธ์แบบแอกกรีเกชัน



รูปที่ 2.2 ความสัมพันธ์แบบคอมโพซิชัน

3. ความสัมพันธ์แบบคอมโพซิชัน (Compositions) หรือเรียกอีกอย่างว่าความสัมพันธ์แบบคอนเทน-อะ *(CONTAINS-A)*

---

[1] อ้างอิงจาก Figure 3-15 : Aggregations หน้าที่ 58 ของรายการอ้างอิง [6]

ความสัมพันธ์ประเภทนี้คล้ายกับความสัมพันธ์แบบแอกกรีเกชัน แต่มีคุณลักษณะของความเป็นเจ้าของมากกว่า เนื่องจากเอนทิตีย่อยอาจเป็นส่วนประกอบของเอนทิตีรวมเพียงหนึ่งเอนทิตีเท่านั้น หากเอนทิตีรวมถูกลบเอนทิตีย่อยก็จะถูกลบด้วย ความสัมพันธ์ในลักษณะนี้เป็นความสัมพันธ์อย่างแน่น (Tightly-Coupled) ดังรูปที่ 2.2 [2]

4. ความสัมพันธ์แบบซิปปลิง (Sibling)

เป็นความสัมพันธ์ระหว่างสับคลาสของซูเปอร์คลาสเดียวกัน โดยสับคลาสเหล่าอยู่ในโครงสร้างต้นไม้ระดับ (Level) เดียวกัน

2.1.2 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์เป็นระบบฐานข้อมูลที่ใช้แบบจำลองผสมระหว่างเชิงวัตถุกับเชิงสัมพันธ์หรืออาจมองว่าเป็นฐานข้อมูลเชิงวัตถุที่ถูกสร้างอยู่บนแบบจำลองเชิงสัมพันธ์ (Relational Model) ก็ได้ กล่าวคือการจัดเก็บข้อมูลของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์มีลักษณะเป็นตาราง (Table) ที่ประกอบด้วยแถวข้อมูล (Row) และสดมภ์ (Column) เช่นเดียวกับระบบฐานข้อมูลเชิงสัมพันธ์ แต่ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์มีชนิดข้อมูล 2 ประเภทคือ ชนิดข้อมูลพื้นฐาน (Built-In Datatypes) และชนิดข้อมูลที่ผู้ใช้นิยาม (User-Defined Datatypes) นอกจากนี้ยังมีเมธอดที่ปฏิบัติการกับข้อมูลของตารางได้ ในขณะที่ระบบฐานข้อมูลเชิงสัมพันธ์มีชนิดข้อมูลพื้นฐานเท่านั้น [8]

ชนิดข้อมูล [12]

ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์มีชนิดข้อมูล 2 ประเภท ได้แก่

1) ชนิดข้อมูลพื้นฐาน เป็นชนิดข้อมูลที่ระบบจัดการฐานข้อมูลกำหนดขึ้นเพื่อใช้กำหนดรูปแบบการเก็บข้อมูล ข้อบังคับ (Constraints) และขอบเขตของข้อมูล ตัวอย่างชนิดข้อมูลประเภทนี้ในระบบจัดการฐานข้อมูลออราเคิล ได้แก่ CHAR, VARCHAR2, NUMBER, DATE, ROWID, LOB เป็นต้น

2) ชนิดข้อมูลที่ผู้ใช้นิยาม เป็นชนิดข้อมูลที่ผู้ใช้กำหนดขึ้นเพื่อนิยามข้อมูลซึ่งมีรูปแบบแตกต่างไปจากชนิดข้อมูลพื้นฐาน โดยระบุทั้งโครงสร้างของข้อมูลและปฏิบัติการที่กระทำกับข้อมูล ชนิดข้อมูลประเภทนี้ช่วยให้สามารถเก็บข้อมูลในลักษณะที่เหมือนกับข้อมูลดิบได้มากกว่าชนิดข้อมูลพื้นฐาน ส่งผลให้การออกแบบระบบฐานข้อมูลมีความซับซ้อนน้อยลง ชนิดข้อมูลที่ผู้ใช้นิยามนี้สามารถแบ่งได้เป็น 2 ประเภทคือ

---

[2] อ้างอิงมาจาก Figure 3-16 : Compositions หน้าที่ 59 ของรายการอ้างอิง [6]

2.1) ชนิดข้อมูลออบเจคต์ (Object Types) เป็นการนิยามในลักษณะนามธรรมของเอนทิตีในระบบงานจริง ชนิดข้อมูลออบเจคต์เป็นสกีมาออบเจคต์ (Schema Object) ที่ประกอบด้วย

- ชื่อชนิดข้อมูลออบเจคต์ (Name) เป็นตัวระบุชนิดข้อมูลออบเจคต์ภายในสกีมา กล่าวคือ ชนิดข้อมูลภายในสกีมาหนึ่งๆ ต้องมีชื่อต่างกัน
- แอตทริบิวท์ เป็นการจำลองโครงสร้างและสถานะของเอนทิตีในระบบงานจริง ชนิดข้อมูลของแอตทริบิวท์อาจเป็นชนิดข้อมูลพื้นฐานหรือชนิดข้อมูลที่ผู้ใช้นิยาม ก็ได้
- เมธทอด อาจเป็นฟังก์ชัน (Function) หรือโพรซีเจอร์ (Procedure) เมธทอดเป็นปฏิบัติการที่กระทำกับเอนทิตีในระบบจริง

2.2) ชนิดข้อมูลกลุ่ม (Collection Types) อธิบายหนึ่งหน่วยข้อมูลที่ประกอบไปด้วยส่วนย่อยจำนวนมากโดยส่วนย่อยเหล่านี้มีชนิดข้อมูลเหมือนกัน ชนิดข้อมูลกลุ่มได้แก่ ชนิดข้อมูลอะเรย์ (Array Types) หรือชนิดข้อมูลตาราง (Table Types)

ข้อบังคับ

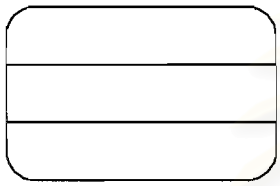
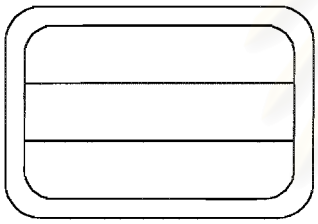
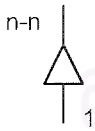
ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ออราเคิล นิยามข้อบังคับหลายประเภท แต่ระบบฐานข้อมูลของงานวิจัยนี้ มีการใช้งานข้อบังคับ ต่อไปนี้ โดยภายในวงเล็บแสดงคำย่อของข้อบังคับแต่ละประเภท

- คีย์หลัก (Primary Key: PK) นิยามบนคีย์ (หนึ่งสดมภ์หรือเซตของสดมภ์) โดยกำหนดว่าแต่ละแถวข้อมูลในตารางจะต้องมีค่าของคีย์แตกต่างกัน
- ข้อบังคับความเฉพาะ (Uniqueness Constraint: Unique) นิยามบนหนึ่งสดมภ์หรือเซตของสดมภ์ ข้อบังคับนี้จะยอมให้มีการแทรกหรือแก้ไขแถวข้อมูลก็ต่อเมื่อในตารางมีค่าของสดมภ์หรือเซตของสดมภ์นั้นไม่ซ้ำกัน
- ข้อบังคับไม่เป็นค่าว่าง (NOT NULL: NN) นิยามบนหนึ่งสดมภ์ของตาราง เมื่อมีการแทรกหรือแก้ไขแถวข้อมูลข้อบังคับนี้จะไม่ยอมให้ค่าของสดมภ์นั้นเป็นค่าว่าง (Null)
- คีย์อ้างอิง(Foreign Key: FK) นิยามบนคีย์(หนึ่งสดมภ์หรือเซตของสดมภ์) ของตารางหนึ่ง โดยกำหนดว่าค่าของคีย์นี้จะสัมพันธ์กับค่าของคีย์ในอีกตารางหนึ่งซึ่งมีความสัมพันธ์กัน

2.1.3 แบบจำลองเชิงวัตถุของ Coad [9]

แบบจำลองเชิงวัตถุของ Coad เป็นแบบจำลองที่แสดงคลาส แอตทริบิวท์และเมธทอดของคลาสและความสัมพันธ์ระหว่างคลาส โดยแบบจำลองนี้แสดงแนวคิดในการออกแบบฐานข้อมูลของระบบ ตารางที่ 2.1 แสดงสัญกรณ์ (Notation) และความหมายของสัญกรณ์ในแบบจำลองเชิงวัตถุของ Coad

ตารางที่ 2.1 สัญกรณ์และคำอธิบายในแบบจำลองเชิงวัตถุของ Coad

สัญกรณ์	คำอธิบาย
	คลาส
	คลาสและออบเจคต์ของคลาส
	Generalization-Specialization Connection
	Whole-Part Object Connection
	Object Connection

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.1.4 เวอร์ชัน

ความหมายของเวอร์ชัน [10]

ความหมายโดยทั่วไปของเวอร์ชัน คือ สถานะหรือคุณลักษณะของออบเจคต์ ณ. เวลาหนึ่ง ในทำนองเดียวกัน ความหมายของเวอร์ชันในระบบฐานข้อมูลคือ การเก็บค่าข้อมูลของเอนทิตีหนึ่งไว้หลายๆ ค่า ซึ่งต่างกับการเก็บข้อมูลทางธุรกิจทั่วไปที่มีการเก็บค่าข้อมูลล่าสุดของเอนทิตีไว้เพียงค่าเดียวเท่านั้น

แบบจำลองเวอร์ชัน (Version Model) [11]

แบบจำลองเวอร์ชันเป็นการนิยามออบเจคต์ ตัวระบุเวอร์ชัน (Version Identifier) การจัดโครงสร้างของเวอร์ชัน (Organization) รวมทั้งปฏิบัติการต่างๆ ที่เกี่ยวกับการเรียกดูเวอร์ชันเก่าและการสร้างเวอร์ชันใหม่

ตัวระบุเวอร์ชันจะถูกใช้ในการแยกความแตกต่างระหว่างแต่ละเวอร์ชันของออบเจคต์ และใช้ในการเข้าถึงเวอร์ชันเมื่อต้องการใช้งานเวอร์ชัน สามารถวิธีการสร้างเวอร์ชันออกเป็น 2 วิธี

1) แบบแจกแจงเวอร์ชั่น (Extensional Versioning) ทุกเวอร์ชันของออบเจคต์จะถูกแจกแจงอย่าง ชัดเจน ซึ่งแสดงได้ดังนิพจน์ต่อไปนี้

$$V = \{v_1, v_2, \ldots, v_n\}$$

โดย V คือ เซตของเวอร์ชันของออบเจคต์

2) แบบกำหนดสภาวะ (Intensional Versioning) เซตของเวอร์ชันจะถูกนิยามโดยเพรดิเคท (Predicate) ต่อไปนี้แทนการแจกแจงเวอร์ชันอย่างชัดเจน

$$V = \{v / c(v)\}$$

โดย c คือ ข้อบังคับ (Constraint) ที่ทุกเวอร์ชันใน V ต้องปฏิบัติตาม

แอนด์/ออร์ กราฟ (AND/OR Graph) [11]

เป็นกราฟที่แสดงแบบจำลองของโครงสร้างความสัมพันธ์ระหว่างออบเจคต์และเวอร์ชันทั้งหมดในระบบงาน แอนด์/ออร์ กราฟประกอบด้วยโหนด (Node) และเส้นเชื่อม (Edge) ที่เรียกว่า แอนด์โหนด ออร์โหนด เส้นเชื่อมแอนด์ และเส้นเชื่อมออร์ โดยเส้นเชื่อมแอนด์เป็นเส้นเชื่อมที่ออกมาจากแอนด์โหนด และเส้นเชื่อมออร์เป็นเส้นเชื่อมที่ออกมาจากออร์โหนด ออร์โหนดใช้แสดงเวอร์ชันในขณะที่แอนด์โหนดใช้แสดงออบเจคต์ ดังรูป 2.3 ซึ่งเป็นตัวอย่างของแอนด์/ออร์ กราฟ

ประเภทของเวอร์ชัน [10]

เนื่องจากระบบงานจริงมีการใช้งานข้อมูลในลักษณะหลายเวอร์ชัน จึงได้มีการจำแนกเวอร์ชันของออบเจคต์ออกเป็นหลายประเภท ตามเกณฑ์ที่แตกต่างกัน ในที่นี้ขอกล่าวถึงเฉพาะการจำแนกเวอร์ชันตามเกณฑ์ 2 ประเภท คือ

- จำแนกประเภทเวอร์ชันตามระดับของระบบงานคอมพิวเตอร์ [1] ได้แก่

1.) การควบคุมเวอร์ชันในระดับของโปรแกรมประยุกต์ (Application-Level Version Control) แบ่งออกได้เป็น 2 ประเภทย่อย

1.1.) เวอร์ชันของออบเจคต์ (Object Versions)

ความจำเป็นในการใช้งานเวอร์ชันของออบเจคต์เกิดขึ้นในโปรแกรมประยุกต์หลายๆ ประเภทได้แก่ โปรแกรมประยุกต์ทางด้านการออกแบบโปรแกรมประยุกต์ทางด้านการจัดการเอกสารและโปรแกรมประยุกต์ที่เกี่ยวข้องกับการเก็บข้อมูลทะเบียนประวัติ เป็นต้น การควบคุมเวอร์ชันในระดับนี้จะเกี่ยวข้องกับข้อมูลสารสนเทศที่ขึ้นอยู่กับเวลาหรือลำดับที่ โดยผู้ใช้หรือโปรแกรมประยุกต์เป็นผู้กำหนดกฎเกณฑ์ที่ใช้ในการควบคุมเวอร์ชัน

โปรแกรมประยุกต์ของระบบงานทางด้านทะเบียนประวัติ คลาสจะใช้แอตทริบิวท์ที่เก็บข้อมูลเกี่ยวกับเวลาหรือช่วงเวลาในการควบคุมเวอร์ชันของข้อมูล [2] โดยเวอร์ชันของข้อมูลประเภทนี้จะเรียงลำดับตามเวลาของการเกิดข้อมูล เช่น โปรแกรมประยุกต์ที่ใช้ในระบบงานทะเบียนประวัติเงินเดือนพนักงาน จะมีคลาสที่เก็บข้อมูลเงินเดือน โดยพนักงานหนึ่งคนจะมีค่าข้อมูลเงินเดือนหลายค่า และแต่ละค่าข้อมูลจะมีช่วงเวลาใช้งาน (Valid Time) ประมาณ 1 ปี ค่าโดยปริยาย (Default Value) ของแอตทริบิวท์เงินเดือนคือค่าที่มีข้อมูลเวลาเป็นปัจจุบัน

โปรแกรมประยุกต์ของระบบงานทางด้านการออกแบบใช้แอตทริบิวท์ที่เก็บหมายเลขเวอร์ชันในการควบคุมการใช้งานและจัดเรียงลำดับเวอร์ชันของออบเจคต์ ค่าโดยปริยายของระบบงานประเภทนี้อาจเป็นค่าที่สัมพันธ์กับหมายเลขเวอร์ชันใดก็ได้ ขึ้นอยู่กับการกำหนดของผู้ใช้

1.2.) เวอร์ชันของชนิดของข้อมูล (Type Versions) คือการจัดการเกี่ยวกับเวอร์ชันของชนิดข้อมูล โดยชนิดข้อมูลนี้จะต้องเป็นชนิดข้อมูลที่ผู้ใช้นิยามซึ่งเป็นชนิดข้อมูลที่มีโครงสร้างซับซ้อนและผู้ใช้สามารถทำการแก้ไขได้ สำหรับชนิดข้อมูลพื้นฐาน ผู้ใช้จะไม่สามารถแก้ไขได้ เนื่องจากเป็นชนิดข้อมูลที่ระบบสร้างขึ้น เมื่อเกิดการเปลี่ยนแปลงชนิดข้อมูล อินสแตนซ์ของข้อมูลก็จะเปลี่ยนแปลงด้วย

2.) การควบคุมเวอร์ชันในระดับระบบ (System-Level Version Control)

เนื่องจากการควบคุมเวอร์ชันประเภทนี้ระบบจะเป็นผู้กำหนดว่าอะไรคือออบเจคต์ ดังนั้นจึงไม่ควรให้ผู้ใช้มีความสามารถในการแก้ไขลำดับเวอร์ชันของระบบ[3]

การควบคุมเวอร์ชันประเภทนี้มักใช้ในงานที่เกี่ยวกับการควบคุมการทำงานพร้อมกันการย้อนกลับ (Rollback) ของรายการและการกู้คืน (Recovery) ของฐานข้อมูลเวอร์ชันของระบบมีความสามารถในการจัดการทางด้านการทำงานพร้อมกันมากกว่าการควบคุมเวอร์ชันประเภทอื่น

- จำแนกประเภทเวอร์ชันตามลำดับการเลือก (Selection Order) ใน แอนด์/ออ กราฟ ได้เป็น 3 ประเภท [11] คือ

1.) Product First (รูปที่ 2.3 (a)) วิธีนี้ความสัมพันธ์ของออบเจคต์จะถูกเลือกก่อน แล้วจึงเลือกเวอร์ชันของส่วนประกอบเป็นลำดับต่อไป ระบบที่ใช้ลำดับการเลือกแบบนี้ได้แก่ Source Code Control System (SCCS) และ Revision Control System (RCS)

2.) Version First (รูปที่ 2.3 (b)) วิธีนี้ตรงกันข้ามกับ Product First โดยเวอร์ชันจะถูกเลือกก่อนแล้วจึงค่อยเลือกออบเจคต์เป็นลำดับต่อไป ระบบที่ใช้ลำดับการเลือกแบบนี้ได้แก่ PCTE

3.) Intertwined (รูปที่ 2.3 (c)) วิธีนี้เป็นวิธีผสมระหว่าง Product First และ Version First โดยจะทำการเลือกสลับกันระหว่างเวอร์ชันและออบเจคต์ ระบบที่ใช้ลำดับการเลือกแบบนี้ได้แก่ ClearCase

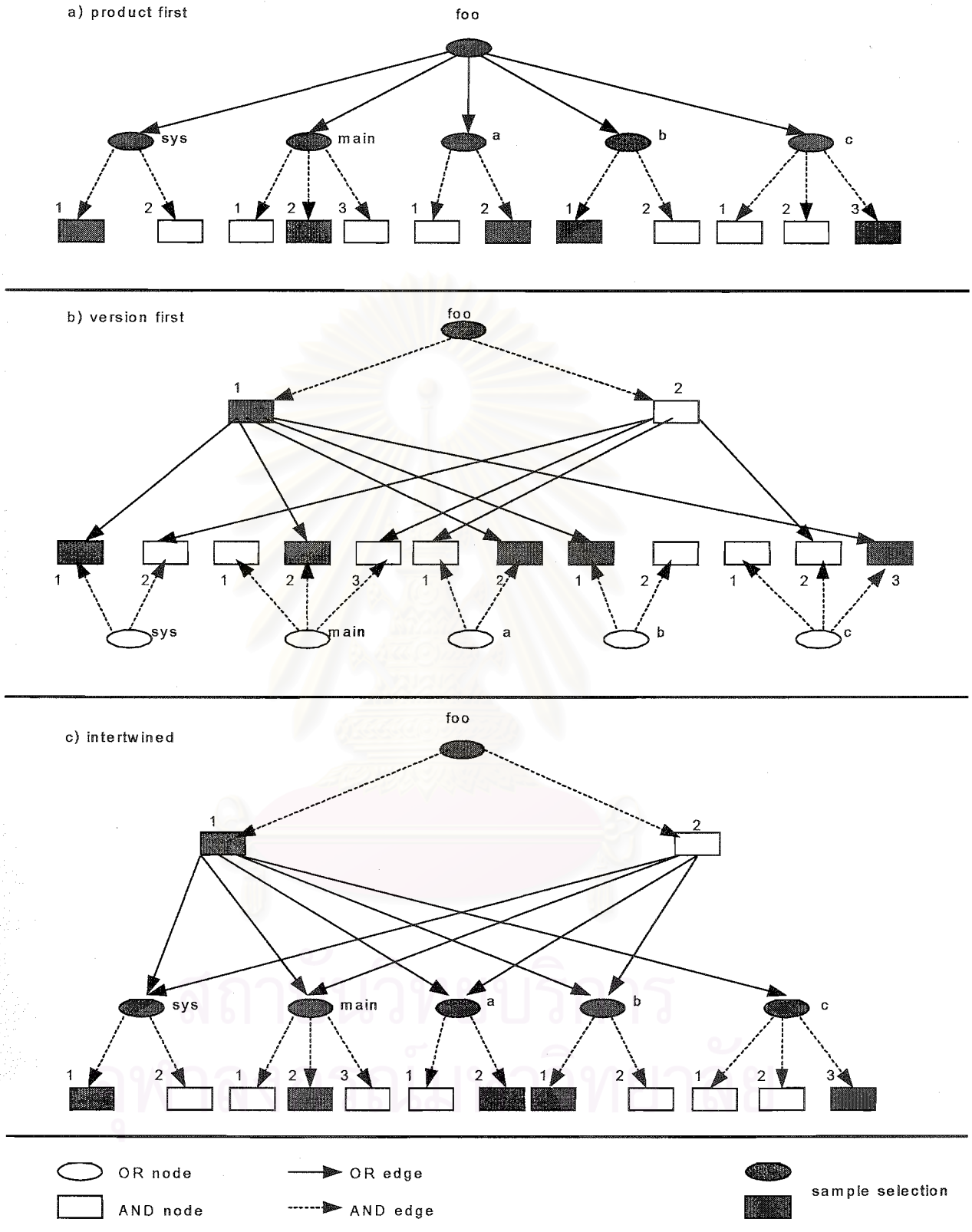
2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Version Support for Engineering Database Systems. [10]

เป็นงานวิจัยที่เกี่ยวกับการจัดการเวอร์ชันของระบบฐานข้อมูลทางวิศวกรรม โดยเสนอสกีมาทั่วไปที่ช่วยให้ผู้ใช้สามารถดำเนินการกับเวอร์ชันของออบเจคต์ได้ ผู้วิจัยเสนอแบบจำลองที่ออบเจคต์การออกแบบประกอบด้วยอินสแตนซ์ของออบเจคต์ในระดับต่ำกว่า ขณะเดียวกัน ออบเจคต์เหล่านี้ก็จะประกอบด้วยออบเจคต์ที่ย่อยลงไปอีก เรียกการประกอบกันของออบเจคต์เหล่านี้

---

[3] การดำเนินการเกี่ยวกับการแก้ไข (Editing Operations) เช่น การลบเวอร์ชันเฉพาะหนึ่งๆ ผู้ใช้สามารถเรียกใช้งานได้ หากการดำเนินการเหล่านั้นไม่ขัดแย้งกับความต้องการของระบบ



รูปที่ 2.3 การใช้ แอนด์/ออร์ กราฟในการแสดงประเภทต่างๆ ของลำดับการเลือก

(a) Product First (b) Version First (c) Intertwined

ว่า การออกแบบแบบลำดับชั้นโครงสร้าง (Hierarchical Design) จากนั้น ผู้วิจัยนิยามออบเจคต์การออกแบบว่าเป็นเซตของเวอร์ชัน ที่แต่ละเวอร์ชันจะมีหมายเลขเวอร์ชันเฉพาะแตกต่างกันภายในหนึ่งออบเจคต์ และนิยามสภาพแวดล้อมว่าเป็นเซตของคู่ลำดับ (ตัวระบุออบเจคต์,หมายเลขเวอร์ชัน) เพื่อใช้ในการอ้างถึงเวอร์ชันของออบเจคต์ย่อยต่างๆ หากการอ้างถึงออบเจคต์เป็นแบบทั่วไป (Generic Reference) ค่าเวอร์ชันของออบเจคต์จะเป็นค่าโดยปริยาย เมื่อสภาพแวดล้อมหนึ่งถูกกระตุ้น ระบบจะสร้างรันไทม์วิว (Runtime View) จากเดฟฟินิชันวิว (Definition View) โดยใช้ขั้นตอนวิธี (Algorithm) ที่ผู้วิจัยกำหนด นอกจากนี้ได้นำแนวคิดของคลัสเตอร์ (Cluster) มาใช้ในการแสดงความสัมพันธ์ระหว่างออบเจคต์กับเวอร์ชันของออบเจคต์ งานวิจัยสร้างคลัสเตอร์ชื่อ Revision สำหรับเก็บเวอร์ชันของออบเจคต์และคลัสเตอร์ Alternative เป็นคลัสเตอร์ของ Revision อีกต่อหนึ่ง ส่วนออบเจคต์การออกแบบจะเสมือนเป็นคลัสเตอร์ที่รวมเอาอินสแตนซ์ของ Alternatives ทั้งหมดเข้าไว้ จะเห็นได้ว่าการนำคลัสเตอร์มาใช้แสดงความสัมพันธ์ระหว่างออบเจคต์กับเวอร์ชันของออบเจคต์เป็นวิธีที่ซับซ้อนและซ้ำซ้อนกับการกำหนดสภาพแวดล้อมซึ่งสามารถแสดงความสัมพันธ์ระหว่างออบเจคต์กับเวอร์ชันของออบเจคต์ได้เช่นกัน ดังนั้น วิทยานิพนธ์นี้จะใช้ตัวระบุเวอร์ชันของออบเจคต์แสดงความสัมพันธ์ระหว่างออบเจคต์ที่อยู่คนละระดับของโครงสร้างลำดับชั้นและความสัมพันธ์ระหว่างเวอร์ชันกับออบเจคต์แทน แต่ได้นำเอาแนวคิดเกี่ยวกับลำดับชั้นโครงสร้างมาใช้แสดงความสัมพันธ์ระหว่างออบเจคต์

2.2.2 Generalized Version Control in an Object-Oriented Database [5]

งานวิจัยนี้อธิบายแฟกซิลิตี (Facilities) ต่างๆ ที่สอดคล้องกับความต้องการของการควบคุมเวอร์ชันในฐานข้อมูลการออกแบบ โดยใช้เคสเป็นระบบต้นแบบในการพัฒนาแฟกซิลิตีเหล่านี้บนระบบฐานข้อมูลเชิงวัตถุชื่อ Iris และเน้นที่การออกแบบซอฟต์แวร์เป็นหลัก

งานวิจัยนี้นิยามสภาพแวดล้อมของการออกแบบว่าประกอบด้วยเครื่องเซิร์ฟเวอร์และเซิร์ฟเวอร์ของฐานข้อมูล (Database Server) ซึ่งเป็นเสมือนที่เก็บข้อมูลการออกแบบสาธารณะ (Public Repository) เช่นเดียวกัน ฐานข้อมูลของการออกแบบจะประกอบด้วยหนึ่งฐานข้อมูลสาธารณะกับกลุ่มของฐานข้อมูลบุคคล (Private Database) ของแต่ละส่วนการทำงาน (Workspace) ผู้วิจัยกำหนดให้การออกแบบถูกแสดงโดยโครงสร้างลำดับชั้นความสัมพันธ์ของออบเจคต์รวมกับออบเจคต์ย่อย ออบเจคต์ซึ่งอยู่ที่ระดับล่างสุดของโครงสร้างลำดับชั้นจะเป็นออบเจคต์ที่ไม่สามารถแยกย่อยออกไปได้อีก และแสดงแบบจำลองเชิงวัตถุที่ออบเจคต์ซึ่งเป็นอินสแตนซ์ของสับคลาสจะเป็นอินสแตนซ์ของซูเปอร์คลาสด้วย ผู้วิจัยแสดงแนวคิดการควบคุมเวอร์ชันโดยแบ่งออบเจคต์ออกเป็นออบเจคต์ธรรมดา (Non-Versioned Object) และออบเจคต์ที่มีเวอร์ชัน (Versioned Object) ซึ่งประกอบด้วยเซตของเวอร์ชันกับอินสแตนซ์ทั่วไปที่เก็บข้อมูลเกี่ยว

กับเวอร์ชันของออบเจคต์และความสัมพันธ์ระหว่างเวอร์ชันของออบเจคต์ การอ้างถึงออบเจคต์ที่มีเวอร์ชันจะกระทำได้ 2 วิธีคือ อ้างถึงแบบเจาะจงเวอร์ชันของออบเจคต์ (Specific Reference) กับการอ้างถึงแบบทั่วไป (Generic Reference) งานวิจัยจะประกอบด้วยคลาส Generic และคลาส Version โดยอินสแตนซ์ทั่วไปของออบเจคต์เป็นอินสแตนซ์ของคลาส Generic ขณะที่เวอร์ชันของออบเจคต์เป็นอินสแตนซ์ของคลาส Version การสร้างเวอร์ชันของออบเจคต์ทำได้ 2 วิธี คือ

1) ผู้ใช้เป็นผู้สร้างเวอร์ชันเองโดยใช้ปฏิบัติการต่างๆ ที่งานวิจัยสร้างไว้

2) ระบบเป็นผู้สร้างเวอร์ชันใหม่ขึ้นเมื่อมีการเปลี่ยนแปลงข้อมูลของเวอร์ชันเก่า โดยระบบจะเปลี่ยนสถานะของเวอร์ชันเก่าและเวอร์ชันของออบเจคต์ทั้งหมดที่สัมพันธ์กับเวอร์ชันนั้นเป็นฟรีซ

วิทยานิพนธ์ได้นำเอาแนวคิดเกี่ยวกับสภาพแวดล้อมของการออกแบบมาประยุกต์ใช้โดยกำหนดให้มีฐานข้อมูลสาธารณะที่เก็บข้อมูลต่างๆ เกี่ยวกับโครงงานของผู้ใช้กับฐานข้อมูลบุคคลที่เก็บคลาสและข้อมูลของการออกแบบ และนำเอาโครงสร้างลำดับชั้นความสัมพันธ์ของออบเจคต์รวมกับออบเจคต์ย่อยมาใช้ในการแสดงความสัมพันธ์ดังกล่าว แต่วิทยานิพนธ์เน้นที่การเก็บออบเจคต์ข้อมูลของการออกแบบใดๆ ซึ่งแต่ละระดับของโครงสร้างลำดับชั้นจะมีออบเจคต์ของคลาสแตกต่างกัน ดังนั้น ออบเจคต์จะเป็นอินสแตนซ์ของคลาสได้เพียงคลาสเดียวเท่านั้น การอ้างถึงออบเจคต์จะทำได้โดยใช้ตัวระบุออบเจคต์โดยตรง ส่วนการอ้างถึงเวอร์ชันจะทำผ่านตัวระบุเวอร์ชัน ซึ่งจะทำให้ทราบว่าเวอร์ชันหนึ่งเป็นของออบเจคต์ใด

2.3 แนวคิดในการวิจัย

2.3.1 แนวคิดเกี่ยวกับออบเจคต์และเวอร์ชันของออบเจคต์

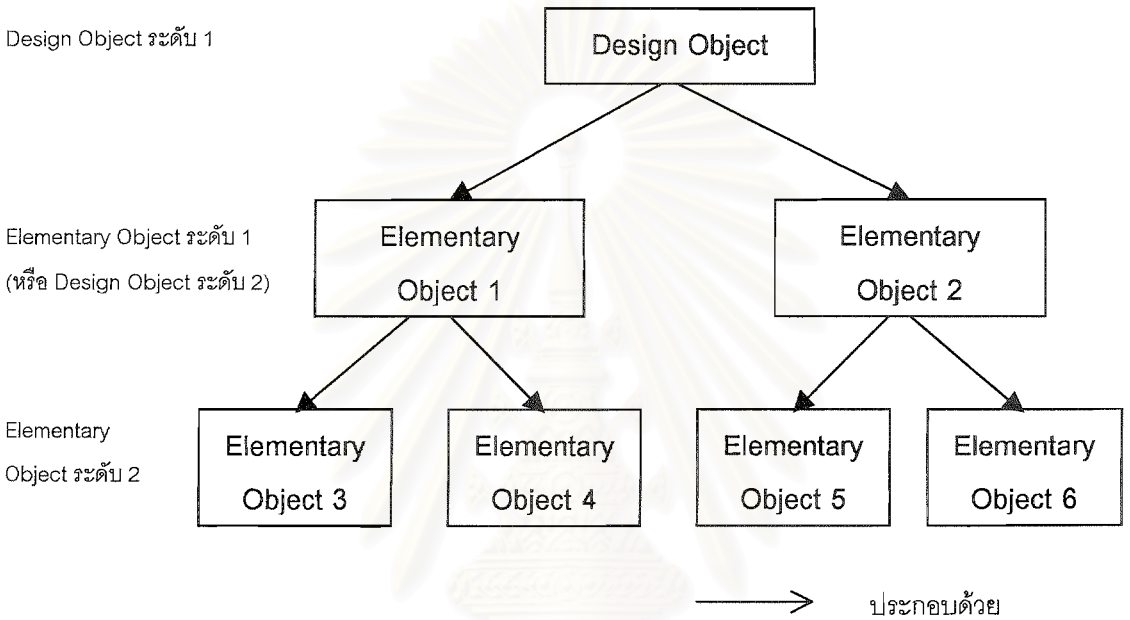
ระบบงานทางด้านการออกแบบส่วนใหญ่มีโครงสร้างในลักษณะลำดับชั้นคล้ายรูปต้นไม้กลับหัว (Tree) จากรูปที่ 2.4 ออบเจคต์ที่อยู่ในระดับบนสุดเป็นออบเจคต์การออกแบบ ออบเจคต์ในระดับที่ 2 (Elementary object level 1) เป็นออบเจคต์ย่อยของออบเจคต์การออกแบบระดับที่ 1 และในขณะเดียวกันออบเจคต์นี้ก็สามารถเป็นออบเจคต์การออกแบบของออบเจคต์ในระดับที่ 3 (Elementary object level 2) ได้ หากผู้ใช้กำหนดลำดับชั้นของการออกแบบเพียง 3 ระดับ (ดังรูป 2.4) ออบเจคต์ในระดับที่ 3 ซึ่งเป็นระดับสุดท้ายของโครงสร้างต้นไม้จะไม่สามารถแยกย่อยออกไปเป็นออบเจคต์ย่อยได้อีก ในโครงสร้างต้นไม้ ออบเจคต์ที่อยู่ต่างระดับกันอาจมีความสัมพันธ์กันได้ 2 ประเภท คือ ความสัมพันธ์แบบ Aggregations และความสัมพันธ์แบบ Compositions

จากรูปที่ 2.5 แต่ละออบเจคต์ภายในโครงสร้างต้นไม้จะประกอบด้วยส่วนที่เป็น ออบเจคต์และเวอร์ชัน โดยเวอร์ชันจะเก็บคุณลักษณะทั่วไปของเวอร์ชันและของออบเจคต์เอาไว้ เวอร์ชันจะมีค่าเฉพาะค่าหนึ่งเรียกว่า ตัวระบุเวอร์ชัน เพื่อให้สามารถระบุความแตกต่างระหว่างแต่ละเวอร์ชัน

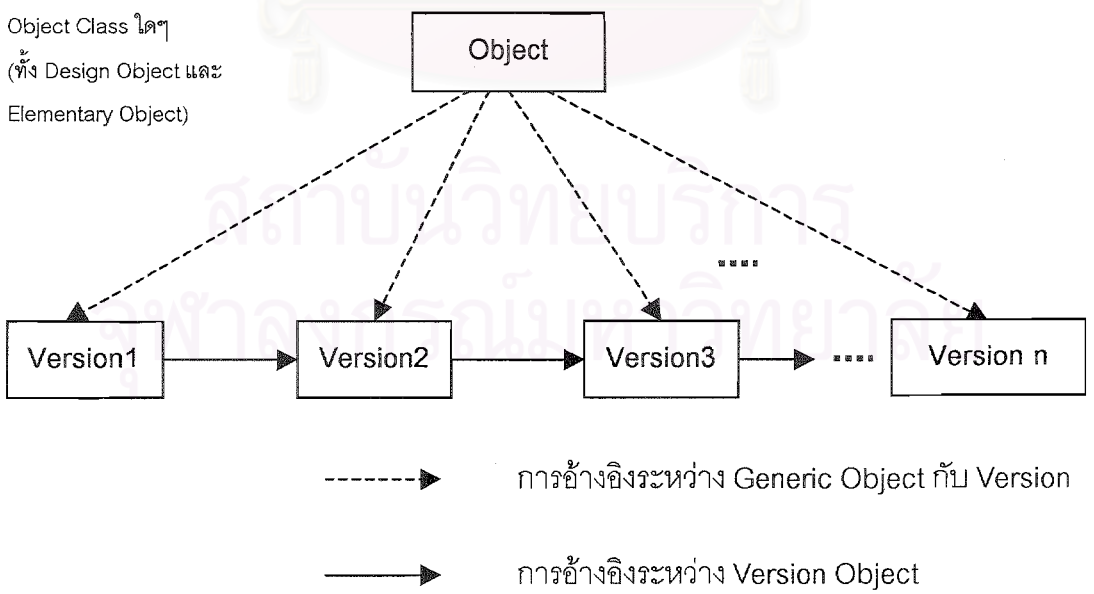
ของออบเจคต์หนึ่ง และใช้ตัวระบุออบเจคต์เป็นสิ่งที่บอกว่าออบเจคต์หนึ่งมีความสัมพันธ์กับเซต ของเวอร์ชันใด

2.3.2 คุณลักษณะทั่วไปของระบบในทางทฤษฎี

เพื่อเป็นแนวคิดในการออกแบบระบบ ผู้วิจัยจึงได้กำหนดคุณลักษณะทางทฤษฎีของระบบควบคุมเวอร์ชันของข้อมูล ดังต่อไปนี้



รูปที่ 2.4 ความสัมพันธ์ระหว่าง Design Object กับ Elementary Object



รูปที่ 2.5 ความสัมพันธ์ระหว่าง Object กับ Version

2.3.2.1 ระบบประกอบด้วย 2 ส่วน คือ

– เนื้อที่ของออบเจคต์ (Object Space) เก็บออบเจคต์และความสัมพันธ์ระหว่างออบเจคต์ของระบบ เช่น หนังสือ บทที่ 2 ส่วนที่ 1 เป็นต้น โดยไม่คำนึงถึงเรื่องเวอร์ชันของออบเจคต์

– เนื้อที่ของเวอร์ชั่น (Version Space) เก็บเวอร์ชันและความสัมพันธ์ระหว่างเวอร์ชันของออบเจคต์

2.3.2.2 เนื้อที่ของออบเจคต์

– ความสัมพันธ์ระหว่างออบเจคต์ที่อยู่คนละระดับของโครงสร้างลำดับชั้นเป็นความสัมพันธ์แบบคอมโพซิชันหรือแอกกรีเกชัน โดยความสัมพันธ์นี้จะแสดงผ่านความสัมพันธ์ระหว่างเวอร์ชันซึ่งอยู่คนละระดับของโครงสร้างลำดับชั้น

– แต่ละออบเจคต์มีแอตทริบิวท์หนึ่งที่เก็บตัวระบุออบเจคต์ที่มีค่าเฉพาะสำหรับออบเจคต์

2.3.2.3 เนื้อที่ของเวอร์ชัน

– โครงสร้างลำดับชั้นมีลักษณะเป็นแบบต้นไม้

– การสร้างเวอร์ชันเป็นแบบแจกแจง

– เวอร์ชันมีลำดับการเลือกเป็นแบบ Version First

– แต่ละเวอร์ชันมีแอตทริบิวท์หนึ่งที่เก็บตัวระบุเวอร์ชันที่มีค่าเฉพาะสำหรับเวอร์ชัน และมีอีกแอตทริบิวท์หนึ่งที่เก็บข้อมูลเกี่ยวกับออบเจคต์

2.3.2.4 Implementation:

– เป็นแบบจำลองเวอร์ชันที่สร้างบนแบบจำลองข้อมูล (Version Model on Top of the Data Model)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การออกแบบระบบ

จากการศึกษา ค้นคว้าทฤษฎีและบทความงานวิจัยต่างๆ ร่วมกับการพิจารณากระบวนการออกแบบของระบบงานจริง ทำให้การออกแบบระบบควบคุมเวอร์ชันของข้อมูลนี้สามารถแบ่งออกเป็นขั้นตอนย่อยๆ ได้ดังนี้

1) การออกแบบกระบวนการใช้งาน (Process Design)
2) การออกแบบฐานข้อมูล (Database Design)
3) การออกแบบส่วนต่อประสานผู้ใช้ (User Interface Design)

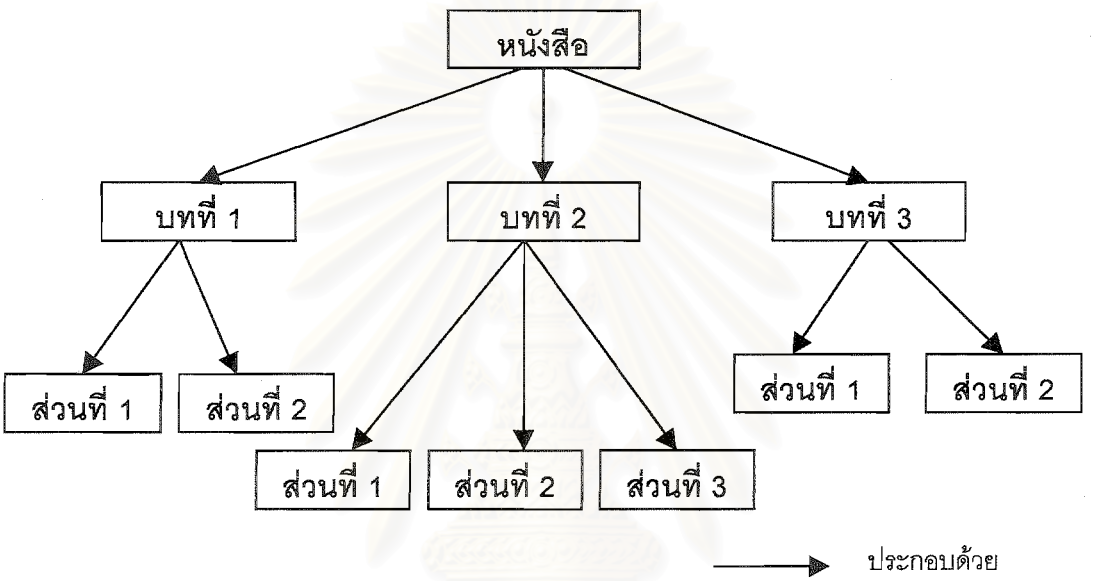
ผู้วิจัยได้นำแนวคิดการวิจัยหัวข้อ 2.3 ร่วมกับแนวคิดเชิงวัตถุมาใช้ในการออกแบบระบบควบคุมเวอร์ชันข้อมูล เนื่องจากช่วยให้การมองเอนทิตีในระบบงานจริงสามารถทำได้ชัดเจนและง่ายขึ้นกว่าการออกแบบโดยใช้แนวคิดเชิงสัมพันธ์ ซึ่งอยู่ในรูปของตารางและการอ้างถึงค่าสดมภ์ระหว่างตาราง แต่เนื่องจากระบบจัดการฐานข้อมูลออราเคิล เวอร์ชัน 8.1.6 เป็นระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ ที่การเก็บข้อมูลในฐานข้อมูลเป็นการเก็บในลักษณะของตาราง ดังนั้น การกล่าวถึงคลาสในระบบฐานข้อมูลจะหมายถึงตาราง แอตทริบิวท์ของคลาสจะหมายถึงสดมภ์ เป็นต้น

บทนี้ กล่าวถึงแนวคิดในการออกแบบระบบ เพื่อให้ผู้ศึกษางานวิจัยมีความเข้าใจแนวคิดทั่วไปของระบบก่อน จากนั้นจึงกล่าวถึงรายละเอียดการออกแบบกระบวนการใช้งาน การออกแบบฐานข้อมูล และสุดท้ายเป็นรายละเอียดของออบเจคต์ต่างๆ ที่ถูกสร้างขึ้นในฐานข้อมูลของระบบ ส่วนขั้นตอนการออกแบบส่วนต่อประสานผู้ใช้ สามารถดูตัวอย่างหน้าจอได้ใน ภาคผนวก ก. การอธิบายรายละเอียดของการออกแบบระบบงานวิจัยนี้ ผู้วิจัยขอใช้การเขียนหนังสือเป็นตัวอย่างประกอบคำอธิบาย เนื่องจากการเขียนหนังสือมีวิธีและขั้นตอนคล้ายกับระบบงานการออกแบบจริงทั่วไป ผู้ศึกษางานวิจัยสามารถนำไปประยุกต์ใช้กับระบบงานการออกแบบอื่นๆ ได้ และยังช่วยให้ผู้ศึกษางานวิจัยสามารถทำความเข้าใจแนวคิดการออกแบบระบบงานวิจัยได้ง่ายขึ้น

3.1 แนวคิดในการออกแบบ

ในการเขียนหนังสือ วัตถุประสงค์ของผู้เขียนคือ หนังสือ ในหนังสือหนึ่งเล่มอาจประกอบไปด้วยบทต่างๆ หลายบท และในแต่ละบทก็จะประกอบไปด้วยหัวข้อย่อยๆ สามารถแสดงความสัมพันธ์ระหว่างหนังสือ บท และหัวข้อได้ดังโครงสร้างต้นไม้ในรูปที่ 3.1 สมมุติให้โครงสร้างต้นไม้นี้เป็นของหนังสือเล่มที่หนึ่ง เมื่อผู้เขียนต้องการเขียนหนังสือขึ้นอีกเป็นเล่มที่สอง โครงสร้างต้นไม้ของหนังสือเล่มที่สองจะต่างจากเล่มที่หนึ่ง ด้วยเหตุนี้ หนังสือหนึ่งเล่มจะถูกแสดงได้ด้วยโครงสร้าง

ต้นไม้หนึ่งต้น ส่วนประกอบต่างๆ ภายในหนังสือจะถูกแสดงด้วยออบเจคต์ในแต่ละระดับของโครงสร้างต้นไม้ โดยหัวข้อย่อยในหนังสือจะแทนด้วยออบเจคต์ที่อยู่ในระดับล่างสุดของโครงสร้างต้นไม้ บทจะแทนด้วยออบเจคต์ที่อยู่ในระดับสูงขึ้น และหนังสือก็จะเป็นออบเจคต์ที่อยู่ระดับบนสุดของโครงสร้างต้นไม้ จะเห็นได้ว่า เราสามารถสร้างคลาสเพื่อนิยามแอตทริบิวท์และเมธอดของออบเจคต์ซึ่งอยู่ในระดับเดียวกันของโครงสร้างต้นไม้ได้ เนื่องจากออบเจคต์เหล่านั้นมีคุณลักษณะและปฏิบัติการบางอย่างเหมือนกัน



รูปที่ 3.1 ตัวอย่างหนังสือ และส่วนประกอบของหนังสือ

การออกแบบในระบบงานจริง สิ่งที่ต้องการออกแบบมีอยู่มากมายหลายประเภท แม้กระทั่งการเขียนหรือพิมพ์หนังสือแต่ละเล่ม ก็มีคุณลักษณะเฉพาะตัวตามแต่ผู้เขียนจะกำหนด ดังนั้น ระบบจะต้องมีส่วนที่ใช้ในการเก็บนิยามของคลาสซึ่งอยู่ในระดับต่างๆ ของโครงสร้างต้นไม้ โดยส่วนนี้จะอธิบายคุณลักษณะทางโครงสร้างของคลาส และถูกใช้ในการสร้างคลาสในฐานข้อมูลด้วย เรียกคลาสที่ถูกสร้างในลักษณะนี้ว่า คลาสที่ผู้ใช้นิยาม (User Defined Class)

หลังจากผู้ใช้ทำการสร้างคลาสในฐานข้อมูลแล้ว ระบบจะต้องสร้างส่วนต่อประสานผู้ใช้ของคลาสนั้นขึ้น เพื่อให้ผู้ใช้สามารถทำการสร้าง แก้ไข และทำปฏิบัติการต่างๆ กับออบเจคต์ของคลาสชนิดนี้ได้

ดังได้กล่าวแล้วในบทที่ 2 ว่าออบเจคต์มีความสัมพันธ์กับเซตของเวอร์ชัน แต่ละเวอร์ชันแสดงสถานะของออบเจคต์ที่แตกต่างกันตามกฎเกณฑ์ที่ผู้ใช้กำหนด เช่น ปีที่ตีพิมพ์หนังสือ ภาษาที่ใช้ในการเขียนหนังสือ เป็นต้น ระบบจึงต้องมีส่วนที่ใช้ในการสร้างเวอร์ชันของออบเจคต์

เนื่องจากผู้วิจัยเลือกใช้วิธีของลำดับการเลือกแบบ Version First ประกอบกับระบบงานการออกแบบบางประเภทมีการใช้เวอร์ชันของออบเจคต์ในลักษณะลำดับก่อนหลัง ระบบจึงต้องสามารถแสดงลำดับที่ระหว่างเวอร์ชันของออบเจคต์หนึ่งๆ ได้ และยังต้องมีส่วนที่ใช้ในการสร้างความสัมพันธ์ระหว่างเวอร์ชันของออบเจคต์ซึ่งอยู่ในแต่ละระดับของโครงสร้างต้นไม้ด้วย

เพื่อให้ระบบมีความยืดหยุ่นในการใช้งานกับระบบงานการออกแบบหลายประเภท ระบบจึงไม่มีส่วนในการสร้างข้อมูลดิบ (Raw data หรือ Fact data) ของแต่ละเวอร์ชัน เพราะการสร้างและแก้ไขข้อมูลดิบต้องใช้ซอฟต์แวร์บรรณาธิกรณ์ (Editor Software) หลากหลายชนิดตามประเภทของระบบงานการออกแบบ เช่น ตัวอย่างของการเขียนหนังสือ จะเกี่ยวข้องกับไฟล์ข้อมูลที่มีสกุลเป็น .txt .gif .jpg ซึ่งการสร้างหรือแก้ไขไฟล์ข้อมูลแต่ละสกุลจะเกี่ยวข้องกับซอฟต์แวร์บรรณาธิกรณ์ที่แตกต่างกัน เป็นต้น ดังนั้นจึงต้องมีส่วนในการนำแฟ้มข้อมูลดิบเหล่านี้มาเก็บและใช้งาน

3.2 การออกแบบกระบวนการใช้งาน

กระบวนการใช้งานของระบบควบคุมเวอร์ชันข้อมูลมีขั้นตอน ดังต่อไปนี้

1) การสร้างโครงงาน (Project)

แต่ละโครงงานหมายถึงสิ่งที่ผู้ใช้ต้องการออกแบบ เช่น การเขียนหนังสือหนึ่งเล่มก็คือหนึ่งโครงงาน หากผู้เขียนคนเดียวกันนี้ ต้องการเขียนหนังสือเล่มที่สอง เล่มที่สาม ผู้เขียนคนนี้ก็จะมี โครงงานที่สอง และสาม ตามลำดับ หรืออาจมองในอีกลักษณะหนึ่งได้ว่า แต่ละโครงงานก็คือโครงสร้างต้นไม้แต่ละต้น นั่นเอง

2) กำหนดจำนวนระดับชั้นของโครงสร้างต้นไม้

จำนวนระดับชั้นของโครงสร้างต้นไม้หมายถึงจำนวนครั้งในการทำซ้ำการแยกย่อยออบเจคต์รวมหนึ่งออกเป็นออบเจคต์ย่อย โดยออบเจคต์ที่อยู่ในระดับล่างสุดของโครงสร้างต้นไม้เป็นออบเจคต์ที่ผู้ใช้งานไม่ต้องการจะแยกย่อยออกไปอีก และ ออบเจคต์ที่อยู่ในระดับบนสุดของโครงสร้างต้นไม้เป็นออบเจคต์ที่ผู้ใช้ต้องการออกแบบ

3) กำหนดคุณลักษณะทางโครงสร้างของคลาสที่อยู่ในแต่ละระดับของโครงสร้างต้นไม้ หรือก็คือ การกำหนดแอตทริบิวท์ของคลาสที่ผู้ใช้นิยาม และกำหนดคุณสมบัติของคลาส เช่น ตัวระบุคลาส ระดับชั้นของคลาสในโครงสร้างต้นไม้ เป็นต้น โดยระบบจะใช้คุณสมบัติเหล่านี้ในการทำปฏิบัติการต่างๆ

4) สร้างคลาสที่ผู้ใช้นิยามในระบบฐานข้อมูล เพื่อใช้ในการเก็บออบเจคต์ของคลาสนั้นๆ ในระบบฐานข้อมูล

5) สร้างข้อมูลดิบ

การสร้างข้อมูลดิบในที่นี้หมายถึงการนำเข้าระบบของไฟล์ข้อมูลดิบ ที่ผู้ใช้สร้างขึ้นโดยใช้ซอฟต์แวร์บรรณาธิกรณ์

6) สร้างออบเจคต์ในแต่ละคลาสที่ผู้ใช้นิยาม

ออบเจคต์ของคลาสหนึ่งๆ จะเปรียบได้กับโหนดซึ่งอยู่ในระดับของโครงสร้างต้นไม้ที่กำหนดโดยนิยามของคลาส

7) สร้างเวอร์ชันของออบเจคต์ที่มีอยู่แล้วในระบบฐานข้อมูล และสร้างความสัมพันธ์ระหว่างเวอร์ชันของออบเจคต์ ความสัมพันธ์นี้เป็นความสัมพันธ์ที่แสดงลำดับที่ของเวอร์ชันในออบเจคต์หนึ่งๆ หากเวอร์ชันใดมีการเก็บข้อมูลดิบ เวอร์ชันนั้นก็จะมีความสัมพันธ์กับออบเจคต์ข้อมูลดิบด้วย

8) สร้างความสัมพันธ์ระหว่างเวอร์ชันของออบเจคต์ที่อยู่ในระดับของโครงสร้างต้นไม้ต่างกัน โดยความสัมพันธ์นี้อาจแสดงความสัมพันธ์แบบคอมโพซิชันหรือแอกกรีเกชัน

3.3 การออกแบบฐานข้อมูล (Database Design)

ผู้วิจัยได้นำแนวคิดการออกแบบที่กล่าวข้างต้น มาใช้ในการออกแบบฐานข้อมูลของระบบ และใช้แผนภาพแบบจำลองเชิงวัตถุของ Coad เป็นเครื่องมือช่วยในการออกแบบแบบจำลองข้อมูลเชิงตรรก (Logical Data Model) เพื่อแสดงคุณลักษณะของเอนทิตีและความสัมพันธ์ระหว่างเอนทิตีในระบบงานจริง ดังในรูปที่ 3.2 ประกอบด้วยคลาสหลายๆ คลาส โดยแต่ละคลาสจะแสดงเอนทิตีในระบบงานจริงที่แตกต่างกัน ซึ่งได้อธิบายไว้ในตารางที่ 3.1 แต่แบบจำลองนี้จะไม่แสดงคลาสและความสัมพันธ์ระหว่างคลาสที่ผู้ใช้นิยาม เนื่องจาก ระบบไม่สามารถระบุคุณลักษณะของคลาสและความสัมพันธ์ระหว่างคลาสที่ผู้ใช้นิยามได้เพราะคุณลักษณะของคลาสและความสัมพันธ์ระหว่างคลาสเหล่านี้ผู้ใช้จะเป็นผู้กำหนด ณ. เวลาใช้งานระบบ

รูปที่ 3.2 จะเห็นได้ว่า คลาสประกอบด้วยแอตทริบิวท์ ซึ่งแสดงคุณลักษณะของคลาส และเมธอดที่แสดงปฏิบัติการซึ่งสามารถกระทำกับออบเจคต์อินสแตนซ์ของคลาส หัวข้อย่อยต่อไปจะขอกล่าวเฉพาะรายละเอียดของแอตทริบิวท์ ส่วนรายละเอียดของเมธอดจะกล่าวในบทที่ 4 เนื่องจาก ผู้วิจัยใช้โปรแกรมภาษาจาวา (Java Programming Language) ในการพัฒนาเมธอดของคลาสทั้งหมดบนเครื่องไคล์เอนต์ (Client) แยกออกจากส่วนแอตทริบิวท์ของคลาสซึ่งจะถูกนำไปสร้างเป็นตารางในระบบจัดการฐานข้อมูลออราเคิลบนเครื่องเซิร์ฟเวอร์ (Server)

3.4 ออบเจคต์ต่าง ๆ ในสกีมาของระบบ

หัวข้อนี้แสดงรายละเอียดเกี่ยวกับออบเจคต์หลักทั้งหมดในสกีมาของระบบงานวิจัย ออบเจคต์เหล่านี้เป็นออบเจคต์ที่ถูกสร้างขึ้นก่อนรันไทม์ เพื่อให้ระบบสามารถทำปฏิบัติการต่างๆ กับฐานข้อมูลได้ นอกจากนี้ ระบบยังมีออบเจคต์อีกประเภทหนึ่งที่ถูกสร้างขึ้นตอนรันไทม์ ออบเจคต์ประเภทนี้เป็นออบเจคต์ที่ผู้ใช้นิยามขณะใช้งานระบบ ดังนั้นจึงไม่สามารถแสดงรายละเอียดเกี่ยวกับออบเจคต์ที่ผู้ใช้นิยามได้ในขณะนี้ ภาคผนวก ค แสดงสคริปท์ที่ใช้ในการสร้างออบเจคต์หลักของระบบงานวิจัย

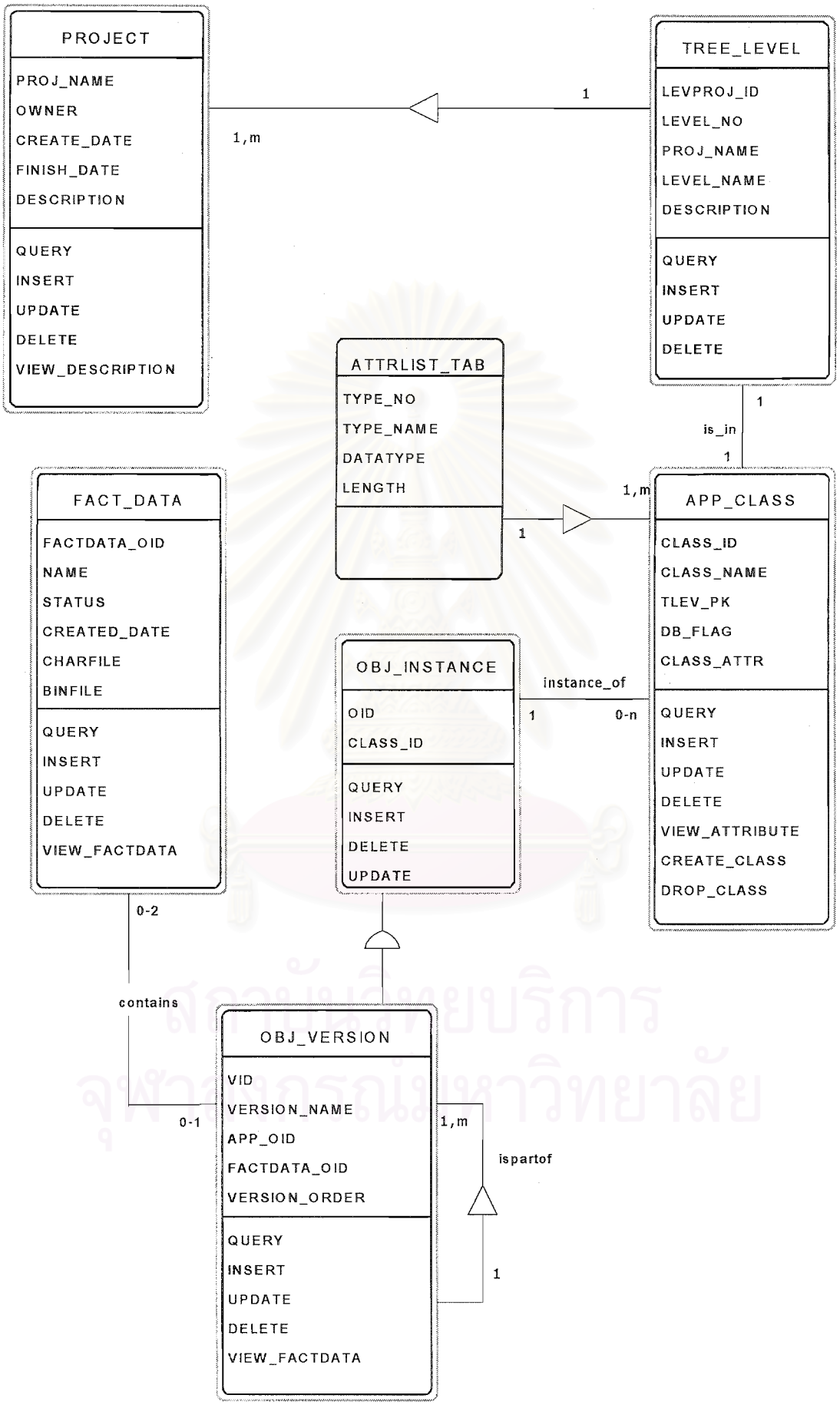
3.4.1 ตารางในระบบฐานข้อมูล

หลังจากขั้นตอนการออกแบบแบบจำลองข้อมูลเชิงตรรกเสร็จสิ้นลง ขั้นต่อไปจะเป็นการแปลงแบบจำลองข้อมูลเชิงตรรกที่ได้ให้เป็นตารางในระบบจัดการฐานข้อมูล

การแปลงแผนภาพแบบจำลองข้อมูลเชิงวัตถุ ให้เป็นตารางในระบบฐานข้อมูลสามารถทำได้โดย แปลงคลาสในแผนภาพให้เป็นตารางในฐานข้อมูล และแปลงแอตทริบิวท์ของแต่ละคลาสให้เป็นแอตทริบิวท์ (หรือสดมภ์) ของตาราง ดังนั้น ออบเจคต์อินสแตนซ์ของคลาสหนึ่งๆ ก็คือ แถวข้อมูลของตารางนั่นเอง เนื่องจากระบบจัดการฐานข้อมูลออราเคิล เวอร์ชัน 8.1.6 เป็นระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ ระบบจัดการฐานข้อมูลออราเคิลจึงมีนิยามของตาราง 2 ประเภท คือ

- ตาราง เป็นตารางที่ประกอบด้วยสดมภ์ซึ่งแต่ละสดมภ์มีชนิดข้อมูลเป็นชนิดข้อมูลพื้นฐาน หรือชนิดข้อมูลที่ผู้ใช้นิยาม การสร้างตารางประเภทนี้จะเหมือนกับการสร้างตารางในระบบจัดการฐานข้อมูลเชิงสัมพันธ์
- ตารางแบบออบเจคต์ (Object Table) เป็นตารางที่ประกอบด้วยสดมภ์ ซึ่งแต่ละสดมภ์มีชนิดข้อมูลเป็นชนิดข้อมูลที่ผู้ใช้นิยามหรือชนิดข้อมูลพื้นฐาน การสร้างตารางประเภทนี้ในระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ จะต้องสร้างชนิดข้อมูลที่ผู้ใช้นิยามขึ้นก่อนแล้วจึงค่อยสร้างตารางแบบออบเจคต์ของชนิดข้อมูลนั้น

ดังนั้นคลาสในแผนภาพแบบจำลองเชิงวัตถุของ Coad รูปที่ 3.2 จะถูกแปลงให้เป็นตารางได้ 2 ประเภทตามวัตถุประสงค์การใช้งานและความซับซ้อนของข้อมูลที่ต้องการเก็บในระบบฐานข้อมูล ตารางที่ 3.1 แสดงผลของการแปลงแผนภาพแบบจำลองนี้ และรายละเอียดการใช้งานของคลาส โดยฐานข้อมูลของระบบควบคุมเวอร์ชันข้อมูลจะมี PROJECT0, TREE_LEVEL0, ATTRLIST_TAB, FACTDATA, ISPART0F, OBJ_INSTANCE, OBJ_VERSION เป็นตารางแบบออบเจคต์ และ APP_CLASS เป็นตารางที่มีลักษณะเป็นตารางซ้อน (Nested Table) แต่ละ



รูปที่ 3.2 แบบจำลองเชิงออบเจคต์ของการออกแบบฐานข้อมูล

แถวข้อมูลของ APP_CLASS จะมีความสัมพันธ์กับหนึ่งกลุ่มของแถวข้อมูลในตาราง ATTRLIST_TAB ซึ่งเป็นตารางที่มีหนึ่งสดมภ์ โดยสดมภ์นี้มีชนิดข้อมูลที่ผู้ใช้นิยามชื่อ ATTRIBUTE_TYPE ในการนิยามโครงสร้างข้อมูลแต่ละแอตทริบิวท์ในคลาสที่ผู้ใช้นิยาม ระบบควบคุมเวอร์ชันข้อมูลจะใช้แถวข้อมูลในตาราง APP_CLASS เป็นข้อมูลในการสร้างตารางที่ผู้ใช้นิยาม เพื่อเก็บออบเจคต์ของคลาสที่ผู้ใช้นิยาม

ตารางที่ 3.2 ถึง ตารางที่ 3.9 แสดงแอตทริบิวท์ของตารางทั้งหมดในฐานข้อมูล รวมทั้งชนิดข้อมูล ความยาวข้อมูล และชื่อประเภทข้อบังคับของแอตทริบิวท์ โดยความยาวข้อมูลจะถูกระบุในวงเล็บหลังชื่อชนิดข้อมูล หากชนิดข้อมูลใดไม่มีการระบุความยาวข้อมูล ระบบจะใช้ค่าปริยายของชนิดข้อมูลนั้นในการสร้างตารางในฐานข้อมูล สดมภ์ประเภทข้อบังคับเก็บข้อบังคับของแอตทริบิวท์ ถ้าแอตทริบิวท์ใดไม่มีการกำหนดข้อบังคับ สดมภ์นี้ของแอตทริบิวท์ก็จะไม่มีค่า ข้อบังคับที่นิยามบนแอตทริบิวท์ตั้งแต่สองแอตทริบิวท์ขึ้นไป เป็นข้อบังคับแบบรวม (Composite Constraint) เพื่อแยกความแตกต่างระหว่างข้อบังคับที่นิยามบนแอตทริบิวท์เดียวกับข้อบังคับแบบรวม ผู้วิจัยขอใช้ 'C_' นำหน้าชื่อประเภทข้อบังคับแบบรวม

ตารางที่ 3.1 การใช้งานของคลาส ในแผนภาพแบบจำลองข้อมูลเชิงวัตถุ

คลาสในแผนภาพแบบจำลองข้อมูล	คำอธิบาย	ตารางและแอตทริบิวท์ในระบบฐานข้อมูล
PROJECT	เก็บข้อมูลเกี่ยวกับโครงงาน โดยหนึ่งโครงงาน คือ ระบบงานการออกแบบหนึ่งระบบ ซึ่งจะได้ผลลัพธ์เป็นชิ้นงานหนึ่งชิ้น	PROJECT0 (ตารางที่ 3.2)
TREE_LEVEL	เก็บข้อมูลเกี่ยวกับลำดับชั้นของโครงสร้างต้นไม้ โดยแต่ละอินสแตนซ์ในคลาสจะหมายถึงแต่ละระดับในโครงสร้างต้นไม้ของโครงงานหนึ่ง	TREE_LEVEL0 (ตารางที่ 3.3)
APP_CLASS	เก็บข้อมูลเกี่ยวกับคลาสที่ผู้ใช้นิยาม ซึ่งเป็นข้อมูลของคลาสและถูกใช้งานโดยระบบ คลาสนี้มีความสัมพันธ์กับคลาส ATTRLIST_TAB ในลักษณะที่หนึ่งออบเจคต์ของ APP_CLASS สัมพันธ์กับหนึ่งกลุ่มออบเจคต์ของคลาส ATTRLIST_TAB	APP_CLASS (ตารางที่ 3.5)

ตารางที่ 3.1 (ต่อ) การใช้งานของคลาสในแผนภาพแบบจำลองข้อมูลเชิงวัตถุ

คลาสในแผนภาพแบบจำลองข้อมูล	คำอธิบาย	ตารางและแอตทริบิวท์ในระบบฐานข้อมูล
ATTRLIST_TAB	คลาสนี้เป็นคลาสนามธรรมที่ไม่มีออบเจคต์ และมีความสัมพันธ์กับคลาส APP_CLASS ในลักษณะที่เป็นค่าแอตทริบิวท์ CLASS_ATTR ของคลาส APP_CLASS ซึ่งระบบจะนำข้อมูลนี้ไปใช้ในการสร้างคลาสที่ผู้ใช้นิยาม	ATTRLIST_TAB (ตารางที่ 3.4)
OBJ_INSTANCE	เก็บข้อมูลเกี่ยวกับออบเจคต์ทั้งหมดของคลาสที่ผู้ใช้นิยาม	OBJ_INSTANCE (ตารางที่ 3.7)
OBJ_VERSION	เก็บข้อมูลเกี่ยวกับเวอร์ชันทั้งหมดของออบเจคต์ที่ผู้ใช้สร้างในระบบ	OBJ_VERSION (ตารางที่ 3.8)
FACTDATA	เก็บข้อมูลเกี่ยวกับข้อมูลดิบทั้งหมดในระบบ	FACTDATA (ตารางที่ 3.6)
ISPARTOF	เก็บข้อมูลที่แสดงความสัมพันธ์แบบทวิภาค (Binary Relationship) ระหว่างเวอร์ชันของออบเจคต์ โดยเวอร์ชันทั้งสองจะอยู่ในระดับที่ต่างกันของโครงสร้างต้นไม้	ISPARTOF (ตารางที่ 3.9)

ตารางที่ 3.2 แอตทริบิวท์และข้อบังคับของตาราง PROJECT0

แอตทริบิวท์	ชนิดข้อมูล และความยาว	ประเภทข้อบังคับ	คำอธิบาย
PROJ_NAME	VARCHAR2(60)	PK	ชื่อโครงงาน
OWNER	VARCHAR2(60)	NN	เจ้าของโครงงานหรือผู้ออกแบบ
CREATE_DATE	DATE		วันที่สร้างโครงงาน
FINISH_DATE	DATE		วันที่สิ้นสุดโครงงาน
DESCRIPTION	VARCHAR2(150)		คำอธิบายรายละเอียดโครงงาน

ตารางที่ 3.3 แอตทริบิวท์และข้อบังคับของตาราง TREE_LEVEL0

แอตทริบิวท์	ชนิดข้อมูล และความยาว	ประเภทข้อบังคับ	คำอธิบาย
LEVPROJ_ID	NUMBER(5)	PK	คีย์หลักของคลาส ระบบจะมีส่วนที่ทำหน้าที่ในการกำหนดค่าให้กับแอตทริบิวท์นี้โดยอัตโนมัติ
LEVEL_NO	NUMBER(2)	C_UNIQUE	เลขแสดงระดับของโครงสร้างต้นไม้
PROJ_NAME	VARCHAR2(60)	C_UNIQUE,NN FK(PROJECT)	ชื่อโครงงาน
LEVEL_NAME	VARCHAR2(70)		ชื่อระดับของโครงสร้างต้นไม้
DESCRIPTION	VARCHAR2(150)		คำอธิบายรายละเอียดระดับโครงสร้างต้นไม้

ตารางที่ 3.4 แอตทริบิวท์และข้อบังคับของตาราง ATTRLIST_TAB

แอตทริบิวท์	ชนิดข้อมูล และความยาว	ประเภทข้อบังคับ	คำอธิบาย
TYPE_NO	NUMBER(2)	PK	ลำดับที่แอตทริบิวท์ของคลาสที่ผู้ใช้นิยาม
TYPE_NAME	VARCHAR2(45)	UNIQUE	ชื่อแอตทริบิวท์ของคลาสที่ผู้ใช้นิยาม
DATATYPE	VARCHAR2(30)		ชนิดข้อมูล
LENGTH	NUMBER(4)		ขนาดความยาวของข้อมูล

ตารางที่ 3.5 แอตทริบิวท์และข้อบังคับของตาราง APP_CLASS

แอตทริบิวท์	ชนิดข้อมูล และความยาว	ประเภทข้อบังคับ	คำอธิบาย
CLASS_ID	NUMBER(4)	PK	รหัสของคลาสที่ผู้ใช้นิยาม ระบบจะมีส่วนที่ทำหน้าที่ในการกำหนดค่าให้กับแอตทริบิวท์นี้โดยอัตโนมัติ
CLASS_NAME	VARCHAR2(70)		ชื่อคลาสที่ผู้ใช้นิยาม
TLEV_PK	TLEV_VIEW_TYPE	NN	ระดับของโครงสร้างต้นไม้
DB_FLAG	NUMBER(1)		แฟลกการสร้างคลาสในระบบฐานข้อมูล
CLASS_ATTR	ATTRLIST_TYPE		แอตทริบิวท์ของคลาสที่ผู้ใช้นิยาม

ตารางที่ 3.6 แอตทริบิวท์และข้อบังคับของตาราง FACTDATA

แอตทริบิวท์	ชนิดข้อมูล และความยาว	ประเภทข้อบังคับ	คำอธิบาย
FACTDATA_OID	NUMBER(10)	PK	รหัสของออบเจคต์ข้อมูลดิบ ระบบจะมีส่วนที่ทำหน้าที่ในการกำหนดค่าให้กับแอตทริบิวท์นี้โดยอัตโนมัติ
NAME	VARCHAR2(70)		ชื่อออบเจคต์ข้อมูลดิบ
STATUS	VARCHAR2(40)		สถานะของออบเจคต์ข้อมูลดิบ
CREATED_DATE	DATE		วันที่สร้างออบเจคต์ข้อมูลดิบ
CHARFILE	BFILE		ข้อมูลดิบชนิดแฟ้มข้อความ
BINFILE	BFILE		ข้อมูลดิบชนิดแฟ้มรูปภาพ

ตารางที่ 3.7 แอตทริบิวท์และข้อบังคับของตาราง OBJ_INSTANCE

แอตทริบิวท์	ชนิดข้อมูล และความยาว	ประเภทข้อบังคับ	คำอธิบาย
OID	NUMBER(10)	PK	รหัสออบเจคต์ ระบบจะมีส่วนที่ทำหน้าที่ในการกำหนดค่าให้กับแอตทริบิวท์นี้โดยอัตโนมัติ
CLASS_ID	NUMBER(4)	NN, FK(APP_CLASS)	คลาสที่ผู้ใช้นิยาม

ตารางที่ 3.8 แอตทริบิวท์และข้อบังคับของตาราง OBJ_VERSION

แอตทริบิวท์	ชนิดข้อมูล และ ความยาว	ประเภทข้อบังคับ	คำอธิบาย
VID	NUMBER(14)	PK	รหัสเวอร์ชันของออบเจคต์ ระบบจะมีส่วนที่ทำหน้าที่ในการกำหนดค่าให้กับแอตทริบิวท์นี้โดยอัตโนมัติ
VERSION_NAME	VARCHAR2(50)		ชื่อเวอร์ชัน
APP_OID	OBJ_INSTANCE_TYPE	NN	ข้อมูลออบเจคต์
FACTDATA_OID	FACTDATA_VIEW_TYPE		ข้อมูลออบเจคต์ข้อมูลดิบ
VERSION_ORDER	NUMBER(4)	NN	ลำดับที่ของเวอร์ชันภายในออบเจคต์หนึ่ง ระบบจะกำหนดและเมนเทน (Maintain) ค่าของแอตทริบิวท์นี้โดยอัตโนมัติ

ตารางที่ 3.9 แอตทริบิวท์และข้อบังคับของตาราง ISPARTOF

แอตทริบิวท์	ชนิดข้อมูล และ ความยาว	ประเภทข้อบังคับ	คำอธิบาย
CHILD_VID	NUMBER(14)	C_PK, FK(OBJ_VERSION)	เวอร์ชันของออบเจคต์ส่วนย่อย
PARENT_VID	NUMBER(14)	C_PK, FK(OBJ_VERSION)	เวอร์ชันของออบเจคต์รวม
CHILD_ORDER	NUMBER(4)		ลำดับที่เวอร์ชันของออบเจคต์ส่วนย่อยภายในเวอร์ชันของออบเจคต์รวม ระบบจะกำหนดและเมนเทนค่าของแอตทริบิวท์นี้โดยอัตโนมัติ

3.4.2 วิว

นอกจากตารางและข้อบังคับต่างๆ ของตารางแล้ว ฐานข้อมูลของระบบประกอบด้วยวิวต่างๆระบบใช้วิวในการเรียกดูข้อมูลบางส่วนของตารางที่สัมพันธ์กับวิวนั้นเช่น FACTDATA_VIEW ประกอบด้วยแอตทริบิวท์ FACTDATA_OID, NAME, STATUS, และCREATE_DATE ซึ่งก็คือแอตทริบิวท์ของตาราง FACTDATA นั่นเอง เรียก FACTDATA ว่า ตารางฐาน (Base Table) ของวิว FACTDATA_VIEW ระบบจัดการฐานข้อมูลจะไม่สร้างวิวขึ้นอย่างถาวรเพื่อใช้เก็บข้อมูลเหมือนตาราง แต่จะเก็บนิยามของวิวไว้ใช้ในกรณีที่มีการทำปฏิบัติการผ่านวิว

ตารางที่ 3.10 แสดงวิวและคุณลักษณะของวิวที่ถูกสร้างขึ้นในฐานข้อมูลของระบบ พร้อมทั้งแสดงตารางฐานที่สัมพันธ์กับวิวนั้นๆ ด้วย

ตารางที่ 3.10 วิว คุณลักษณะของวิว และตารางฐาน

วิว	คุณลักษณะของวิวและชนิดข้อมูล	ตารางฐาน
TLEV_VIEW	Levproj_id, NUMBER Level_no, NUMBER Proj_name, VARCHAR2 Level_name, VARCHAR2	TREE_LEVEL0
FACTDATA_VIEW	Factdata_oid, NUMBER Name, VARCHAR2 Status, VARCHAR2 Created_date, DATE	FACTDATA
APPCLASS_VIEW	Class_id, NUMBER Class_name, VARCHAR2 Tlev_pk, tlev_view_type Db_flag, NUMBER	APP_CLASS
VERSION_VIEW	Vid, NUMBER Version_name, VARCHAR2 App_oid, obj_instance_type Version_order, NUMBER	OBJ_VERSION

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
การพัฒนาโปรแกรม

งานวิจัยนี้ใช้ซอฟต์แวร์ช่วยพัฒนาชื่อออราเคิล เจเดฟเวอลอปเปอร์ เวอร์ชัน 3.0 (Oracle JDeveloper) ซึ่งซอฟต์แวร์นี้สร้างไฟล์ต้นฉบับ (Source File) เป็นโปรแกรมภาษาจาวาที่รันบนเครื่องไคล์เอนต์ และติดต่อกับระบบจัดการฐานข้อมูลออราเคิลบนเครื่องเซิร์ฟเวอร์ผ่านเจดีบีซี (JDBC-Java Database Connectivity)

บทนี้อธิบายรายละเอียดของการพัฒนาโปรแกรม โดยกล่าวถึงผังโครงสร้างของโปรแกรมเป็นประการแรกเพื่อสะดวกแก่การทำความเข้าใจในส่วนประกอบต่างๆ ของโปรแกรม จากนั้นจะกล่าวถึงผังโครงสร้างหน้าจอ ซึ่งแสดงความเกี่ยวเนื่องกันของหน้าจอ และสุดท้ายเป็นคำอธิบายเกี่ยวกับคลาสและหน้าที่การทำงานของคลาสต่างๆ ในโปรแกรม

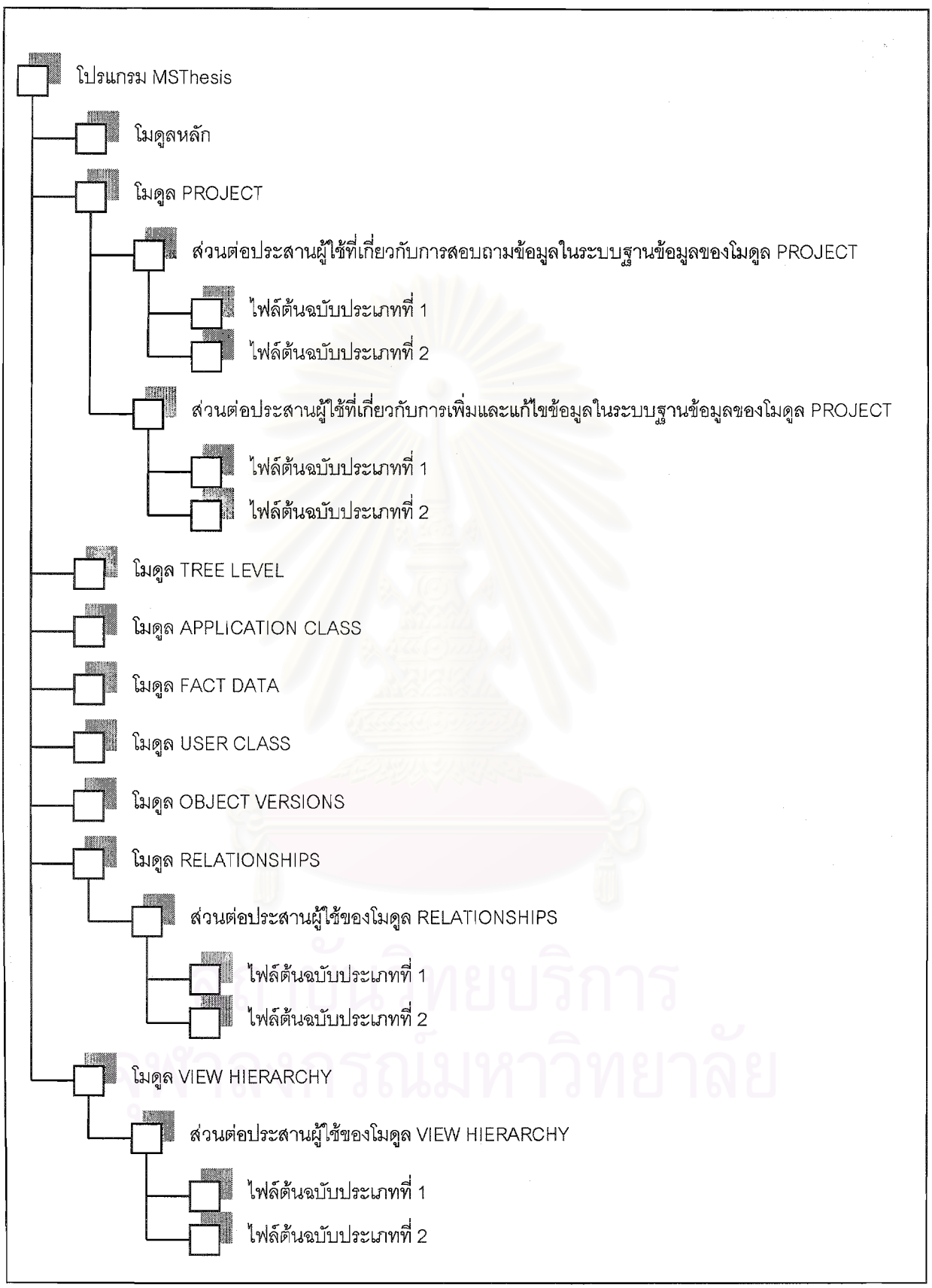
4.1 ผังโครงสร้างของโปรแกรม

ผังโครงสร้างโปรแกรมเป็นการแสดงโครงสร้างโปรแกรมทางแนวคิด (Conceptual Program Structure) ซึ่งแตกต่างจากโปรแกรมต้นฉบับ กล่าวคือโปรแกรมทางแนวคิดประกอบด้วยโมดูล ส่วนประสานผู้ใช้ และไฟล์ต้นฉบับ ตามลำดับ แต่โปรแกรมต้นฉบับมีเพียงไฟล์ต้นฉบับเท่านั้น เนื่องจากโปรแกรมต้นฉบับประกอบด้วยไฟล์ต้นฉบับหลายๆ ไฟล์ ทำให้ยากต่อการศึกษาโปรแกรม ผู้วิจัยจึงใช้วัตถุประสงค์การทำงานเป็นเกณฑ์ในการจัดกลุ่มไฟล์ต้นฉบับทั้งหมดเป็นผังโครงสร้างโปรแกรมทางแนวคิด ดังรูปที่ 4.1

โปรแกรมทางแนวคิดของงานวิจัยประกอบด้วยหลายๆ โมดูล (Module) แต่ละโมดูลจะทำปฏิบัติการที่เกี่ยวกับแต่ละตารางในระบบฐานข้อมูล และโมดูลจะประกอบด้วยส่วนต่อประสานผู้ใช้ตั้งแต่หนึ่งส่วนขึ้นไป โดยแต่ละส่วนต่อประสานผู้ใช้ประกอบด้วยไฟล์ต้นฉบับ 2 ไฟล์ ดังรูปที่ 4.1 สามารถแบ่งประเภทไฟล์ต้นฉบับออกเป็น 2 ประเภทตามลักษณะการทำงานของเมธอดของคลาสซึ่งอยู่ภายในไฟล์ ได้แก่

ประเภทที่ 1 เป็นไฟล์ที่ประกอบด้วยเมธอด ซึ่งทำปฏิบัติการบางอย่างกับตารางในระบบฐานข้อมูล เช่น การสอบถามข้อมูล (Query) การแทรกแถวข้อมูล (Insert) การแก้ไขข้อมูล (Update) และการลบข้อมูล (Delete) เป็นต้น

ประเภทที่ 2 เป็นไฟล์ที่ประกอบด้วยเมธอด ซึ่งทำปฏิบัติการเกี่ยวกับหน้าจอ เช่น จัดการกับส่วนประกอบภายในหน้าจอ แสดงข้อความบอกสถานะการทำงานของโปรแกรม แสดงหน้าจอย่อย เป็นต้น



รูปที่ 4.1 ผังโครงสร้างของโปรแกรมของงานวิจัย

I20066582

จากรูปที่ 4.1 จะเห็นว่า โมดูล RELATIONSHIPS และ โมดูล VIEW HIERARCHY เป็นโมดูลที่มีส่วนต่อประสานผู้ใช้เพียงหนึ่งส่วน ในขณะที่โมดูลอื่นๆ จะมีส่วนประสานผู้ใช้ 2 ส่วน โดยส่วนต่อประสานผู้ใช้ของโมดูล RELATIONSHIPS จะมีทั้งปฏิบัติการสอบถามข้อมูลและปฏิบัติการสร้างออบเจคต์ในหน้าจอเดียวกัน ขณะที่โมดูล VIEW HIERARCHY เป็นโมดูลที่ทำปฏิบัติการสอบถามข้อมูลเพียงอย่างเดียว

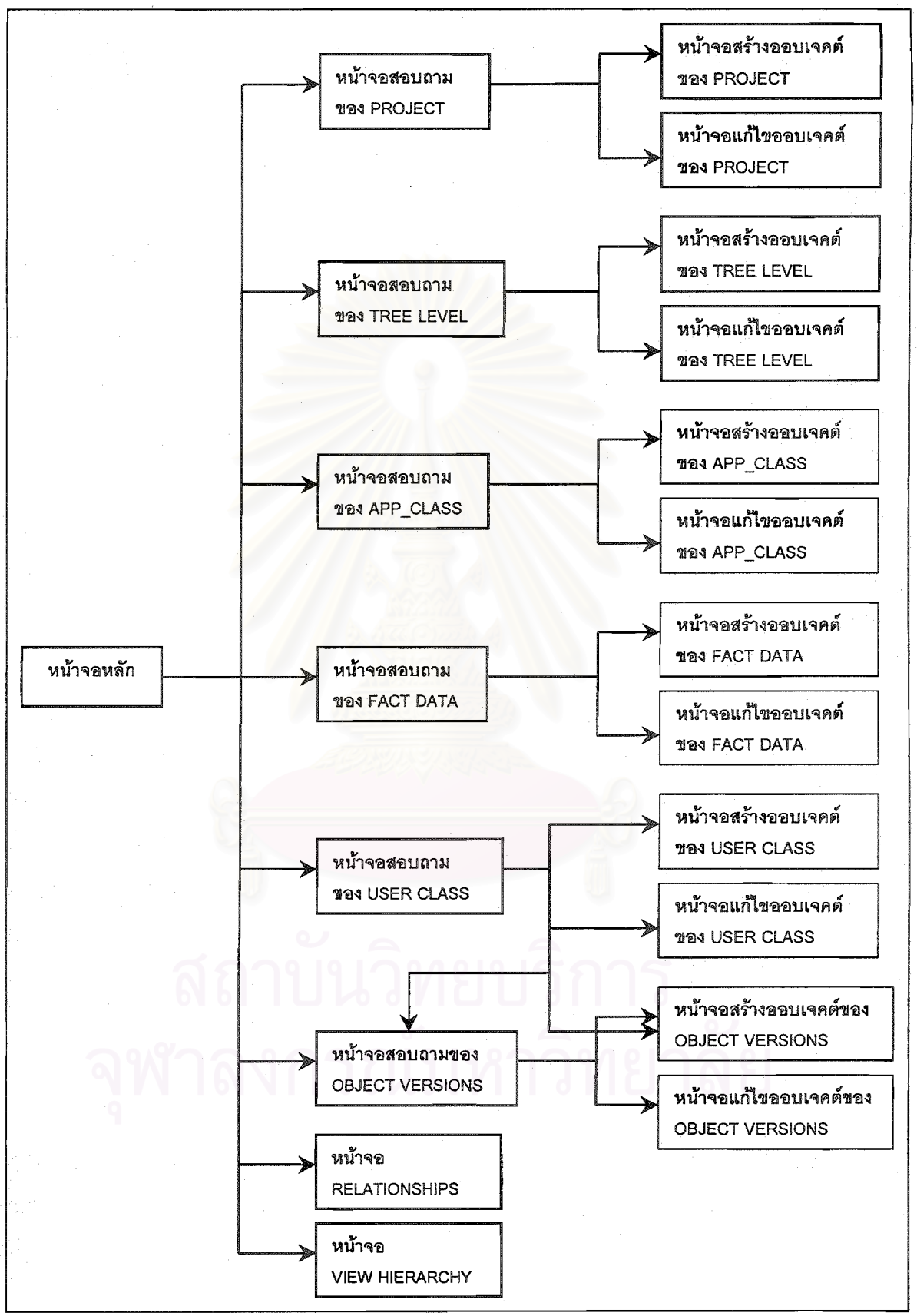
4.2 ผังโครงสร้างหน้าจอของระบบ

เพื่อง่ายในการทำความเข้าใจเกี่ยวกับขั้นตอนของกระบวนการใช้งานระบบควบคุมเวอร์ชันของข้อมูล ดังกล่าวในหัวข้อ 3.2 ผังโครงสร้างหน้าจอของระบบในรูปที่ 4.2 แสดงลำดับการเรียกใช้งานและความเกี่ยวเนื่องกันของหน้าจอทั้งหมดในระบบ

เมื่อผู้ใช้รันโปรแกรม ระบบควบคุมเวอร์ชันข้อมูลจะแสดงหน้าจอหลัก เป็นหน้าจอเริ่มต้นของการใช้งานระบบ หน้าจอนี้ประกอบด้วยปุ่มที่ทำหน้าที่เสมือนเมนูในการเรียกใช้โมดูลต่างๆ โดยแต่ละปุ่มจะเรียกใช้โมดูลเฉพาะที่แตกต่างกัน หลังจากผู้ใช้กดปุ่มใดปุ่มหนึ่ง ระบบจะแสดงหน้าจอสอบถามของโมดูลที่เกี่ยวข้องกับปุ่มนั้น เพื่อให้ผู้ใช้ทำปฏิบัติการต่างๆ หากผู้ใช้ต้องการแทรกแถวข้อมูลของตารางในระบบฐานข้อมูล สามารถทำได้โดยเลือกเมนูที่เรียกใช้เมธทอดที่ทำหน้าที่แสดงหน้าจอสร้างออบเจคต์ แต่ถ้าผู้ใช้ต้องการแก้ไขแถวข้อมูลของตารางในระบบฐานข้อมูล ผู้ใช้จะต้องระบุแถวข้อมูลที่ต้องการแก้ไขก่อน แล้วจึงเลือกเมนูที่เรียกใช้เมธทอดซึ่งทำหน้าที่แสดงหน้าจอแก้ไขออบเจคต์ จากการใช้งานระบบ ผู้ใช้จะเห็นเสมือนมีหน้าจอหนึ่งสำหรับใช้ในการแทรกแถวข้อมูลและมีอีกหน้าจอหนึ่งสำหรับใช้ในการแก้ไขค่าแอตทริบิวท์ของแถวข้อมูล แต่ในการพัฒนาระบบ โปรแกรมจะสร้างเพียงหนึ่งหน้าจอเท่านั้นสำหรับใช้ในการแทรกแถวข้อมูลใหม่และแก้ไขแอตทริบิวท์ของแถวข้อมูล

4.3 โปรแกรมของระบบควบคุมเวอร์ชันข้อมูล

หัวข้อ 4.1 กล่าวถึงโครงสร้างอย่างกว้างๆ ของโปรแกรมระบบควบคุมเวอร์ชันข้อมูลในแง่การออกแบบโปรแกรม เพื่อให้ผู้ศึกษางานวิจัยเข้าใจถึงส่วนประกอบของโปรแกรมและสามารถศึกษารายละเอียดการพัฒนาโปรแกรมได้อย่างเป็นระบบ หัวข้อนี้จะอธิบายรายละเอียดของโปรแกรมในแง่ของการพัฒนา โดยแสดงความสัมพันธ์ระหว่างไฟล์ต้นฉบับกับโมดูล คลาสในไฟล์ต้นฉบับ และตารางในระบบฐานข้อมูล จากนั้นจะเป็นคำอธิบายหน้าที่การทำงานของคลาสในไฟล์ต้นฉบับ



รูปที่ 4.2 ผังแสดงโครงสร้างหน้าจอของระบบ

4.3.1 ไฟล์ต้นฉบับ

โปรแกรมของระบบประกอบด้วยไฟล์ต้นฉบับจำนวน 37 ไฟล์ ไฟล์ต้นฉบับเหล่านี้จะสัมพันธ์กับตารางในระบบฐานข้อมูลและโมดูลในโครงสร้างโปรแกรม ดังตารางที่ 4.1 ค่าในสดมภ์คลาสในไฟล์ต้นฉบับแสดงให้เห็นว่า ไฟล์ต้นฉบับแต่ละไฟล์ประกอบด้วยคลาสตั้งแต่หนึ่งคลาสขึ้นไป และหากแถวของไฟล์ต้นฉบับใดไม่มีค่าในสดมภ์ตารางในระบบฐานข้อมูล หมายความว่าไฟล์ต้นฉบับนั้นไม่สัมพันธ์กับตารางใดในระบบฐานข้อมูลเลย

เมื่อนำไฟล์ต้นฉบับในตารางที่ 4.1 มาพิจารณาร่วมกับผังโครงสร้างโปรแกรม รูปที่ 4.1 ชื่อของไฟล์ต้นฉบับที่ลงท้ายด้วยคำว่า Frame จะเป็นไฟล์ต้นฉบับประเภทที่ 2 กล่าวคือ เป็นไฟล์ที่ประกอบด้วยคลาสซึ่งจัดการเกี่ยวกับหน้าจอต่างๆ ของระบบ ตัวอย่างของไฟล์ประเภทนี้ ได้แก่ AppCQryFrame.java, AppCUpdFrame.java, ProjQFrame.java, ProjUpdFrame.java เป็นต้น ไฟล์ต้นฉบับที่มีชื่อแตกต่างไปจากนี้จะเป็นไฟล์ต้นฉบับประเภทที่ 1

เนื่องจากแต่ละส่วนต่อประสานผู้ใช้ประกอบด้วยไฟล์ต้นฉบับ 2 ไฟล์ คือไฟล์ประเภทที่ 1 จำนวนหนึ่งไฟล์และไฟล์ประเภทที่ 2 อีกหนึ่งไฟล์ ไฟล์ต้นฉบับทั้งสองไฟล์นี้จะทำงานร่วมกัน เพื่อให้ผู้ใช้สามารถทำปฏิบัติการต่างๆ บนหน้าจอของโมดูลได้ตามต้องการ ตัวอย่างเช่น โมดูล PROJECT มีส่วนประสานผู้ใช้ 2 ส่วน คือ ส่วนต่อประสานผู้ใช้ที่เกี่ยวกับการสอบถามข้อมูลในระบบฐานข้อมูล ซึ่งประกอบด้วยไฟล์ต้นฉบับ ProjectClass.java และ ProjQFrame.java ไฟล์ต้นฉบับทั้งสองไฟล์นี้จะทำงานร่วมกันเพื่อให้ผู้ใช้ทำปฏิบัติการต่างๆ บนหน้าจอสอบถามของโมดูล และส่วนต่อประสานผู้ใช้ที่เกี่ยวกับการเพิ่มและแก้ไขข้อมูลในระบบฐานข้อมูล ประกอบด้วยไฟล์ต้นฉบับชื่อ ProjUpd.java กับ ProjUpdFrame.java ไฟล์ต้นฉบับทั้งสองไฟล์นี้จะทำงานร่วมกันเพื่อให้ผู้ใช้ทำปฏิบัติการต่างๆ บนหน้าจอสร้างออบเจคต์และหน้าจอแก้ไขออบเจคต์ของโมดูล เป็นต้น

4.3.2 คลาสในไฟล์ต้นฉบับ

การพัฒนาโปรแกรมประยุกต์จาวาเพื่อทำปฏิบัติการต่างๆ กับชนิดข้อมูลที่ผู้ใช้นิยามในระบบฐานข้อมูลออราเคิล 8 ผ่านเจดีบีซีนั้น จะต้องมีคลาสซึ่งถูกใช้โดยเจดีบีซีในการแปลงชนิดข้อมูลระหว่างชนิดข้อมูลของระบบฐานข้อมูลออราเคิลกับชนิดข้อมูลของโปรแกรมจาวา และคลาสนี้จะช่วยให้เจดีบีซีสามารถส่งผ่านข้อมูลระหว่างระบบฐานข้อมูลกับโปรแกรมจาวาได้อย่างถูกต้อง

ตารางที่ 4.1 ความสัมพันธ์ระหว่างไฟล์ต้นฉบับกับโมดูลและตารางในระบบฐานข้อมูล

ไฟล์ต้นฉบับ	โมดูล	คลาสในไฟล์ต้นฉบับ	ตารางในระบบฐานข้อมูล
AppClass.java	โมดูล APPLICATION CLASS	AppClass	APP_CLASS
AppCQryFrame.java	โมดูล APPLICATION CLASS	AppCQryFrame, AttrsQryPane	APP_CLASS
AppCUpd.java	โมดูล APPLICATION CLASS	AppCUpd	APP_CLASS
AppCUpdFrame.java	โมดูล APPLICATION CLASS	AppCUpdFrame, TLevOptPane	APP_CLASS
AttributeType.java	โมดูล APPLICATION CLASS	AttributeType	ATTRLIST_TAB
AttributeTypeRef.java	โมดูล APPLICATION CLASS	AttributeTypeRef	ATTRLIST_TAB
AttrlistType.java	โมดูล APPLICATION CLASS	AttrlistType	APP_CLASS
connectParams.java	โมดูลหลัก	connectParams	
FactData.java	โมดูล FACT DATA	FactData	FACTDATA
FactDataUpd.java	โมดูล FACT DATA	FactDataUpd	FACTDATA
FactQFrame.java	โมดูล FACT DATA	FactQFrame	FACTDATA
FactUpdFrame.java	โมดูล FACT DATA	FactUpdFrame	FACTDATA
genTable.java	โมดูลหลัก	genTable	
LinkFrame.java	โมดูล RELATIONSHIPS	LinkFrame	ISPARTOF
LinkVerClass.java	โมดูล RELATIONSHIPS	LinkVerClass	ISPARTOF
MainFrame.java	โมดูลหลัก	MainFrame	
MyFileFilter.java	โมดูล FACT DATA	MyFileFilter	
ProjectClass.java	โมดูล PROJECT	ProjectClass	PROJECT0
ProjQFrame.java	โมดูล PROJECT	ProjQFrame	PROJECT0

ตารางที่ 4.1 (ต่อ) ความสัมพันธ์ระหว่างไฟล์ต้นฉบับกับโมดูลและตารางในระบบฐานข้อมูล

ไฟล์ต้นฉบับ	โมดูล	คลาสในไฟล์ต้นฉบับ	ตารางในระบบฐานข้อมูล
ProjUpd.java	โมดูล PROJECT	ProjUpd	PROJECT0
ProjUpdFrame.java	โมดูล PROJECT	ProjUpdFrame	PROJECT0
QryFrame.java	โมดูล USER CLASS	QryFrame, DynamicTableModel	User Defined Class
TLevClass.java	โมดูล TREE LEVEL	TLevClass	TREE_LEVEL0
TLevelQryFrame.java	โมดูล TREE LEVEL	TLevelQryFrame	TREE_LEVEL0
TLevelUpdFrame.java	โมดูล TREE LEVEL	TLevelUpdFrame	TREE_LEVEL0
TLevUpd.java	โมดูล TREE LEVEL	TLevUpd	TREE_LEVEL0
TlevViewType.java	โมดูล APPLICATION CLASS	TlevViewType	APP_CLASS
TlevViewTypeRef.java	โมดูล APPLICATION CLASS	TlevViewTypeRef	APP_CLASS
UpdFrame.java	โมดูล USER CLASS	UpdFrame	User Defined Class
UsrClass.java	โมดูล USER CLASS	UsrClass	User Defined Class
UsrUpdate.java	โมดูล USER CLASS	UsrUpdate	User Defined Class
VersionQFrame.java	โมดูล OBJECT VERSIONS	VersionQFrame, FactDataPnl	OBJ_VERSION
VersionQry.java	โมดูล OBJECT VERSIONS	VersionQry	OBJ_VERSION
VersionUpd.java	โมดูล OBJECT VERSIONS	VersionUpd	OBJ_VERSION
VersionView.java	โมดูล VIEW HIERARCHY	VersionView	
VerUpdFrame.java	โมดูล OBJECT VERSIONS	VerUpdFrame, FactDataOptPane	OBJ_VERSION

ตารางที่ 4.1 (ต่อ) ความสัมพันธ์ระหว่างไฟล์ต้นฉบับกับโมดูลและตารางในระบบฐานข้อมูล

ไฟล์ต้นฉบับ	โมดูล	คลาสในไฟล์ต้นฉบับ	ตารางในระบบฐานข้อมูล
VerViewFrame.java	โมดูล VIEW HIERARCHY	VerViewFrame, ChildVerPnl	

การใช้เจดีบีซีเพื่อเข้าถึงออบเจคต์ของชนิดข้อมูลที่ผู้ใช้นิยามในระบบฐานข้อมูลออราเคิล มี 2 ขั้นตอน [14] ได้แก่

1) ขั้นตอนการเลือกใช้คลาสสำหรับการแปลงชนิดข้อมูลของเจดีบีซี โดยออราเคิลกำหนดวิธีในการเลือกใช้คลาส 2 วิธีคือ
   - ออบเจคต์ของชนิดข้อมูลถาวร (Strongly Typed Object) วิธีนี้เป็นการสร้างจาวาคลาส สำหรับแต่ละชนิดข้อมูลออบเจคต์ที่ผู้ใช้นิยาม ซึ่งโปรแกรมเมอร์ (Programmer) อาจสร้างจาวาคลาสขึ้นเอง หรืออาจใช้โปรแกรมเจพับลิชเชอร์ (Jpublisher) ช่วยสร้างจาวาคลาสให้ ก็ได้
   - ออบเจคต์ของชนิดข้อมูลชั่วคราว (Weakly Typed Object) เป็นการใช้คลาส oracle.sql.STRUCT, oracle.sql.REF,และ oracle.sql.ARRAY ซึ่งเป็นคลาสที่ออราเคิลนิยามขึ้นสำหรับโปรแกรมภาษาจาวา
2) ขั้นตอนการสร้างและใช้ออบเจคต์อินสแตนซ์ของคลาสในขั้นตอนที่ 1

การพัฒนาโปรแกรมของระบบควบคุมเวอร์ชันข้อมูล ผู้วิจัยเลือกใช้ทั้ง 2 วิธีในการกำหนดจาวาคลาสสำหรับการแปลงชนิดข้อมูล โดยมีโมดูล APPLICATION CLASS เป็นโมดูลที่ใช้วิธีออบเจคต์ของชนิดข้อมูลถาวรเพียงโมดูลเดียว ส่วนโมดูลอื่นๆ จะใช้วิธีออบเจคต์ของชนิดข้อมูลชั่วคราว ดังในตารางที่ 4.2 จะเห็นได้ว่า โมดูล APPLICATION CLASS ประกอบด้วยคลาสซึ่งถูกสร้างโดยโปรแกรม JPublisher จำนวน 6 คลาส ได้แก่ AttributeType, AttributeTypeRef, AttrlistType, TlevViewType, และ TlevViewTypeRef คลาสเหล่านี้จะถูกใช้โดยเจดีบีซีเพื่อแปลงชนิดข้อมูลที่ ผู้ใช้นิยาม ATTRIBUTE_TYPE และ TLEV_VIEW_TYPE ในระบบฐานข้อมูลให้เป็นชนิดข้อมูลของโปรแกรมจาวา คลาสที่โปรแกรมเจพับลิชเชอร์สร้างขึ้นนี้ ประกอบด้วยเมธทอดซึ่งทำหน้าที่เกี่ยวกับการกำหนดค่า (Set) และการอ่านค่า (Get) แต่ละแอตทริบิวท์ของชนิดข้อมูลออบเจคต์ที่ผู้ใช้นิยาม รวมทั้งเมธทอดคอนสตรัคเตอร์ (Constructor) ที่ใช้ในการสร้างออบเจคต์ของชนิดข้อมูลนั้น

ตารางที่ 4.2 คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ

คลาส	หน้าที่การทำงาน
AppClass	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานของโมดูล APPLICATION CLASS คลาสนี้มีเมธอดที่เกี่ยวข้องกับการสอบถามข้อมูล การลบแถวข้อมูล การแสดงหน้าจอสอบถาม หน้าจอสร้าง ออบเจคต์และหน้าจอแก้ไขออบเจคต์ของตาราง APP_CLASS ในระบบฐานข้อมูล เพื่อแสดงหน้าจอและทำปฏิบัติการต่างๆ บนหน้าจอ คลาสนี้จะทำงานร่วมกับคลาส AppCQryFrame โดยมีการเรียกใช้งานและส่งผ่านค่าพารามิเตอร์ (Parameters) ไปมาระหว่างเมธอดของคลาสทั้งสอง
AppCQryFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ (Components) ภายในหน้าจอสอบถามของโมดูล APPLICATION CLASS เช่น การวางส่วนประกอบบนหน้าจอ การจัดการเกี่ยวกับเหตุการณ์ที่เกิดกับส่วนประกอบ (Event Handler) เป็นต้น
AttrsQryPane	เป็นคลาสย่อยภายในคลาส AppCQryFrame โดยคลาสนี้จัดการเกี่ยวกับหน้าจอย่อยที่แสดงรายละเอียดของแอตทริบิวท์ของคลาสที่ผู้ใช้เลือกบนหน้าจอสอบถาม เช่น ชื่อแอตทริบิวท์ ชนิดข้อมูล ความยาวข้อมูล เป็นต้น
AppCUpd	คลาสนี้มีเมธอดที่เกี่ยวกับการแทรกแถวข้อมูลและแก้ไขค่าแอตทริบิวท์ของแถวข้อมูลในตาราง APP_CLASS ในระบบฐานข้อมูล คลาสนี้จะทำงานร่วมกับคลาส AppCUpdFrame โดยมีการเรียกใช้งานและส่งค่าพารามิเตอร์ไปมาระหว่างเมธอดของคลาสทั้งสอง
AppCUpdFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสร้าง ออบเจคต์ (หรือก็คือ หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์) ของโมดูล APPLICATION CLASS เมื่อผู้ใช้เลือกเมนูที่แสดงหน้าจอการสร้างออบเจคต์ หรือเมนูที่แสดงหน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ ระบบจะแสดงหน้าจอของคลาสนี้ เพื่อให้ผู้ใช้สามารถทำปฏิบัติการได้ตามต้องการ
TlevOptPane	เป็นคลาสย่อยภายในคลาส AppCUpdFrame โดยคลาสนี้จัดการเกี่ยวกับหน้าจอย่อยที่แสดงรายละเอียดของระดับทั้งหมดในโครงสร้างต้นไม้ที่ผู้ใช้ต้องการเพื่อให้ผู้ใช้เลือกว่าจะสร้างคลาสอยู่ในระดับใดของโครงสร้างต้นไม้

ตารางที่ 4.2 (ต่อ) คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ

คลาส	หน้าที่การทำงาน
AttributeType	คลาสนี้ถูกสร้างโดยโปรแกรมชื่อเจพับลิชเชอร์ ซึ่งเป็นโปรแกรมหนึ่งในซอฟต์แวร์ช่วยพัฒนาเจเดฟเวอลอปเปอร์ เพื่อประโยชน์ในการปฏิบัติการกับแต่ละแถวข้อมูลภายในแอตทริบิวท์ CLASS_ATTR ของคลาส APP_CLASS โดยแถวข้อมูลภายในแอตทริบิวท์นี้มีชนิดข้อมูลเป็น ATTRIBUTE_TYPE ซึ่งเป็นชนิดข้อมูลที่ผู้ใช้นิยามในระบบฐานข้อมูล
AttributeTypeRef	คลาสนี้ถูกสร้างโดยโปรแกรมชื่อเจพับลิชเชอร์ เพื่อใช้ในการทำปฏิบัติการกับแต่ละแถวข้อมูลภายในแอตทริบิวท์ CLASS_ATTR ของตาราง APP_CLASS โดยแถวภายในแอตทริบิวท์นี้มีชนิดข้อมูลเป็น REF ATTRIBUTE_TYPE ในระบบฐานข้อมูล
AttrlistType	คลาสนี้ถูกสร้างโดยโปรแกรมชื่อเจพับลิชเชอร์ เพื่อใช้ในการทำปฏิบัติการกับแอตทริบิวท์ CLASS_ATTR ของคลาส APP_CLASS ซึ่งแอตทริบิวท์นี้มีลักษณะเป็นอะเรย์ของชนิดข้อมูล ATTRIBUTE_TYPE จึงทำให้คลาส APP_CLASS เป็นตารางซ้อน
connectParams	คลาสนี้เก็บค่าพารามิเตอร์ต่างๆ ที่ใช้ในการสร้างเซสชั่น (Session) ในระบบจัดการฐานข้อมูลออราเคิล ได้แก่ ชื่อเครื่องที่ติดตั้งระบบจัดการฐานข้อมูล หมายเลขพอร์ต (Port), ชื่อรหัสของฐานข้อมูล (SID) ชื่อและรหัสผ่านผู้ใช้สำหรับเข้าใช้งาน (UserName and Password) ระบบจัดการฐานข้อมูล
FactData	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานโมดูล FACT DATA คลาสนี้มีเมธทอดที่เกี่ยวข้องกับการสอบถามข้อมูล การลบแถวข้อมูล, การแสดงหน้าจอสอบถาม หน้าจอสร้างออบเจคต์และหน้าจอแก้ไข ออบเจคต์ของตาราง FACTDATA ในระบบฐานข้อมูล เพื่อแสดงหน้าจอและทำปฏิบัติการต่างๆ บนหน้าจอ คลาสนี้จะทำงานร่วมกับคลาส FactQFrame โดยมีการเรียกใช้งานและส่งผ่านค่าพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง
FactDataUpd	คลาสนี้มีเมธทอดที่เกี่ยวกับการแทรกแถวข้อมูลและแก้ไขค่าแอตทริบิวท์ของแถวข้อมูลในตาราง FACTDATA ในระบบฐานข้อมูล คลาสนี้จะทำงานร่วมกับคลาส FactUpdFrame โดยมีการเรียกใช้งานและส่งค่าพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง

ตารางที่ 4.2 (ต่อ) คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ

คลาส	หน้าที่การทำงาน
FactQFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสอบถามของโมดูล FACT DATA เช่น การวางส่วนประกอบบนหน้าจอ การจัดการเกี่ยวกับเหตุการณ์ที่เกิดกับส่วนประกอบ เป็นต้น
FactUpdFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสร้างออบเจคต์ (หรือก็คือ หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์) ของโมดูล FACT DATA เมื่อผู้ใช้เลือกเมนูที่แสดงหน้าจอการสร้างออบเจคต์ หรือเมนูแสดงหน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ ระบบจะแสดงหน้าจอนี้เพื่อให้ผู้ใช้สามารถทำปฏิบัติการได้ตามต้องการ
genTable	คลาสนี้เป็นแบบจำลองตารางของระบบ โดยประกอบด้วยเมธทอดที่ทำหน้าที่เกี่ยวข้องกับตาราง เช่น การแทรกแถวข้อมูล การลบและแก้ไขแถวข้อมูล การอ่านค่าเซล (Cell) ในตาราง เป็นต้น หน้าจอสอบถามทุกหน้าจอเรียกใช้คลาสนี้ เพื่อแสดงข้อมูลในลักษณะตาราง
LinkFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสอบถามของโมดูล RELATIONSHIPS เช่น การวางส่วนประกอบบนหน้าจอ การจัดการเกี่ยวกับเหตุการณ์ที่เกิดกับส่วนประกอบ เป็นต้น
LinkVerClass	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานโมดูล RELATIONSHIPS คลาสนี้มีเมธทอดที่เกี่ยวกับการสอบถามข้อมูล การลบแถวข้อมูล การสร้างออบเจคต์ของตาราง ISPARTOF ในระบบฐานข้อมูล คลาสนี้จะทำงานร่วมกับคลาส LinkFrame โดยมีการเรียกใช้งานและส่งผ่านพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง
MainFrame	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานระบบ นอกจากนี้ยังจัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอหลัก และมีเมธทอดที่ใช้ในการเชื่อมต่อ (Connect) กับระบบจัดการฐานข้อมูล
MyFileFilter	เป็นคลาสที่ทำหน้าที่ในการกรองไฟล์ โดยเลือกเอาเฉพาะไฟล์ที่มีสกุลตามที่กำหนด[13]

ตารางที่ 4.2 (ต่อ) คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ

คลาส	หน้าที่การทำงาน
ProjectClass	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานโมดูล PROJECT คลาสนี้มีเมธอดที่เกี่ยวข้องกับการสอบถามข้อมูล การลบแถวข้อมูล การแสดงหน้าจอสอบถาม หน้าจอสร้างออบเจคต์และหน้าจอแก้ไข ออบเจคต์ของตาราง PROJECT0 ในระบบฐานข้อมูล เพื่อแสดงหน้าจอและทำปฏิบัติการต่างๆ บนหน้าจอ คลาสนี้จะทำงานร่วมกับคลาส ProjQFrame โดยมีการเรียกใช้งานและส่งผ่านค่าพารามิเตอร์ไปมาระหว่างเมธอดของคลาสทั้งสอง
ProjQFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสอบถามของโมดูล PROJECT เช่น การวางส่วนประกอบบนหน้าจอ การจัดการเกี่ยวกับเหตุการณ์ที่เกิดกับส่วนประกอบ เป็นต้น
ProjUpd	คลาสนี้มีเมธอดที่เกี่ยวกับการแทรกแถวข้อมูลและแก้ไขค่าแอตทริบิวท์ของแถวข้อมูลในตาราง PROJECT0 ในระบบฐานข้อมูล คลาสนี้จะทำงานร่วมกับคลาส ProjUpdFrame โดยมีการเรียกใช้งานและส่งค่าพารามิเตอร์ไปมาระหว่างเมธอดของคลาสทั้งสอง
ProjUpdFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสร้างออบเจคต์ (หรือก็คือ หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์) ของโมดูล PROJECT เมื่อผู้ใช้เลือกเมนูที่แสดงหน้าจอการสร้างออบเจคต์ หรือเมนูที่แสดงหน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ ระบบจะแสดงหน้าจอนี้เพื่อให้ผู้ใช้สามารถทำปฏิบัติการได้ตามต้องการ
QryFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสอบถามของโมดูล USER CLASS เช่น การวางส่วนประกอบบนหน้าจอ การจัดการเกี่ยวกับเหตุการณ์ที่เกิดกับส่วนประกอบ เป็นต้น
DynamicTableModel	คลาสนี้ขยายมาจากคลาส genTable คลาสนี้เป็นแบบจำลองตารางของระบบ เช่นเดียวกับคลาสแม่ ซึ่งประกอบด้วยเมธอดที่ทำหน้าที่เกี่ยวข้องกับตาราง เช่น การแทรกแถวข้อมูล การลบและแก้ไขแถว ข้อมูล การอ่านค่าเซล(Cell)ในตาราง เป็นต้น หน้าจอสอบถามของโมดูล USER CLASS เรียกใช้คลาสนี้ เพื่อแสดงข้อมูลในลักษณะตารางแบบไดนามิกซึ่งเป็นตารางที่มีจำนวนสดมภ์ของตารางไม่แน่นอนขึ้นอยู่กับการกำหนดของผู้ใช้

ตารางที่ 4.2 (ต่อ) คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ

คลาส	หน้าที่การทำงาน
TlevClass	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานโมดูล TREE LEVEL คลาสนี้มีเมธอดที่เกี่ยวข้องกับการแทรกแถวข้อมูล การลบแถวข้อมูล การแสดงหน้าจอสอบถาม หน้าจอสร้างออบเจคต์และหน้าจอแก้ไข ออบเจคต์ของตาราง TREE_LEVEL0 ในระบบฐานข้อมูล เพื่อแสดง หน้าจอและทำปฏิบัติการต่างๆ บนหน้าจอ คลาสนี้จะทำงานร่วมกับคลาส TlevelQryFrame โดยมีการเรียกใช้งานและส่งผ่านค่าพารามิเตอร์ไปมาระหว่างเมธอดของคลาสทั้งสอง
TlevelQryFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสอบถามของโมดูล TREE LEVEL เช่น การวางส่วนประกอบบนหน้าจอ การจัดการเกี่ยวกับเหตุการณ์ที่เกิดกับส่วนประกอบ เป็นต้น
TlevelUpdFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสร้างออบเจคต์ (หรือก็คือ หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์) ของโมดูล TREE LEVEL เมื่อผู้ใช้เลือกเมนูที่แสดงหน้าจอสร้างออบเจคต์ หรือเมนูที่แสดงหน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ ระบบจะแสดงหน้าจอนี้เพื่อให้ผู้ใช้สามารถทำปฏิบัติการได้ตามต้องการ
TlevUpd	คลาสนี้มีเมธอดที่เกี่ยวกับการแทรกแถวข้อมูลและแก้ไขค่าแอตทริบิวท์ของแถวข้อมูลในตาราง TREE_LEVEL0 ในระบบฐานข้อมูล คลาสนี้จะทำงานร่วมกับคลาส TlevelUpdFrame โดยมีการเรียกใช้งานและส่งค่าพารามิเตอร์ไปมาระหว่างเมธอดของคลาสทั้งสอง
TlevViewType	คลาสนี้ถูกสร้างโดยโปรแกรมเจพับลิชเชอร์ โดยมีเมธอดที่ปฏิบัติการเกี่ยวกับการอ่านค่าแอตทริบิวท์ของวิว TLEV_VIEW ซึ่งเป็นตารางที่ประกอบด้วยหนึ่งสดมภ์และมีชนิดข้อมูล TLEV_VIEW_TYPE ที่ผู้ใช้นิยามในระบบฐานข้อมูล
TlevViewTypeRef	คลาสนี้ถูกสร้างโดยโปรแกรมเจพับลิชเชอร์ เพื่อใช้ในการทำปฏิบัติการกับแต่ละแถวข้อมูลของวิว TLEV_VIEW โดยแถวข้อมูลภายในวิวนี้มีชนิดข้อมูลเป็น REF ATTRIBUTE_TYPE ในระบบฐานข้อมูล

ตารางที่ 4.2 (ต่อ) คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ

คลาส	หน้าที่การทำงาน
UpdFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสร้างออบเจคต์ (หรือก็คือ หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์) ของโมดูล USER CLASS เมื่อผู้ใช้เลือกเมนูที่แสดงหน้าจอสร้างออบเจคต์ หรือเมนูที่แสดงหน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ ระบบจะแสดงหน้าจอนี้เพื่อให้ผู้ใช้สามารถทำปฏิบัติการได้ตามต้องการ
UsrClass	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานโมดูล USER CLASS คลาสนี้มีเมธทอดที่เกี่ยวข้องกับการสอบถามข้อมูล การลบออบเจคต์ การแสดงหน้าจอสอบถาม หน้าจอสร้างออบเจคต์และหน้าจอแก้ไข ออบเจคต์ของตารางที่ผู้ใช้สร้างในระบบฐานข้อมูล เพื่อแสดงหน้าจอและทำปฏิบัติการต่างๆ บนหน้าจอ คลาสนี้จะทำงานร่วมกับคลาส QryFrame โดยมีการเรียกใช้งานและส่งผ่านค่าพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง
UsrUpdate	คลาสนี้มีเมธทอดที่เกี่ยวกับการแทรกแถวข้อมูลและแก้ไขค่าแอตทริบิวท์ของแถวข้อมูลในตารางที่ผู้ใช้นิยาม คลาสนี้จะทำงานร่วมกับคลาส UpdFrame โดยมีการเรียกใช้งานและส่งค่าพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง
VersionQFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสอบถามของโมดูล OBJECT VERSIONS เช่น การวางส่วนประกอบบนหน้าจอ การจัดการเกี่ยวกับเหตุการณ์ที่เกิดกับส่วนประกอบ เป็นต้น
FactDataPnl	เป็นคลาสภาย (Inner class) ของคลาส VersionQFrame ทำหน้าที่ในการแสดงข้อความภายใน (Content) ของไฟล์ข้อมูลดิบซึ่งเกี่ยวข้องกับแถวข้อมูลในตาราง FactData ที่ผู้ใช้เลือก
VersionQry	เป็นคลาสหลักและเป็นคลาสเริ่มต้นในการใช้งานโมดูล OBJECT VERSIONS คลาสนี้มีเมธทอดที่เกี่ยวข้องกับการสอบถามข้อมูล การลบออบเจคต์ การแสดงหน้าจอสอบถาม หน้าจอสร้างออบเจคต์และหน้าจอแก้ไขออบเจคต์ของตารางที่ผู้ใช้สร้างในระบบฐานข้อมูล เพื่อแสดงหน้าจอและทำปฏิบัติการต่างๆ บนหน้าจอ คลาสนี้จะทำงานร่วมกับคลาส VersionQFrame โดยมีการเรียกใช้งานและส่งผ่านค่าพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง

ตารางที่ 4.2 (ต่อ) คลาสและหน้าที่การทำงานของคลาสทั้งหมดในระบบ

คลาส	หน้าที่การทำงาน
VersionUpd	คลาสนี้มีเมธทอดที่เกี่ยวกับการแทรกแถวข้อมูลและแก้ไขค่าแอตทริบิวท์ของแถวข้อมูลในตาราง OBJ_VERSION ในระบบฐานข้อมูล คลาสนี้จะทำงานร่วมกับคลาส VerUpdFrame โดยมีการเรียกใช้งานและส่งค่าพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง
VersionView	เป็นคลาสหลักและเป็นคลาสเริ่มต้นของโมดูล VIEW HIERARCHY คลาสนี้มีเมธทอดที่เกี่ยวข้องกับการสอบถามข้อมูลจากตาราง ISPARTOF แล้วนำข้อมูลที่ได้มาแสดงในลักษณะเสมือนลำดับชั้นของโครงสร้างต้นไม้ แต่ละระดับของโครงสร้างจะถูกแสดงในรูปของตาราง คลาสนี้จะทำงานร่วมกับคลาส VerViewFrame โดยมีการเรียกใช้งานและส่งผ่านค่าพารามิเตอร์ไปมาระหว่างเมธทอดของคลาสทั้งสอง
VerUpdFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอสร้างออบเจคต์ (หรือก็คือ หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์) ของโมดูล OBJECT VERSIONS เมื่อผู้ใช้เลือกเมนูที่แสดงหน้าจอสร้างออบเจคต์ หรือเมนูที่แสดงหน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ ระบบจะแสดงหน้าจอนี้ เพื่อให้ผู้ใช้สามารถทำปฏิบัติการได้ตามต้องการ
FactDataOptPane	เป็นคลาสภายในของคลาส VerUpdFrame ทำหน้าที่เสมือน LOV ของ Factdata โดยแสดงข้อมูลที่จำเป็นเกี่ยวกับ Factdata ทั้งหมดที่มีอยู่ในระบบ
VerViewFrame	เป็นคลาสที่จัดการเกี่ยวกับส่วนประกอบต่างๆ ภายในหน้าจอของโมดูล VIEW HIERARCHY
ChildVerPnl	เป็นคลาสภายในของคลาส VerViewFrame คลาสนี้ทำหน้าที่ในการแสดงข้อมูลของเวอร์ชันลูกทั้งหมดที่มีความสัมพันธ์กับ เวอร์ชันแม่ซึ่งผู้ใช้เลือก ระบบจะเรียกใช้คลาสนี้แบบไดนามิก กล่าวคือ ตราบใดที่ผู้ใช้เลือกเวอร์ชันที่มีความสัมพันธ์กับเวอร์ชันลูก คลาสนี้ก็จะถูกเรียกใช้ และหากผู้ใช้เลือกเวอร์ชันที่ไม่มีความสัมพันธ์กับเวอร์ชันลูกเลยคลาสนี้ก็จะไม่ถูกเรียกใช้

บทที่ 5
การทดสอบเพื่อประเมินผลการวิจัย

ขั้นตอนการทดสอบการทำงานของระบบงานวิจัยเป็นขั้นตอนที่ต่อจากขั้นตอนการออกแบบและพัฒนาโปรแกรม เพื่อตรวจสอบว่าระบบงานวิจัยที่ถูกพัฒนาขึ้นนั้นสามารถทำงานได้ตรงตามวัตถุประสงค์หรือไม่

บทนี้จะกล่าวถึงสภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการทดสอบระบบงานวิจัย และผลการทดสอบระบบงานวิจัย โดยใช้ตัวอย่างข้อมูลการแต่งหนังสือ

5.1 สภาพแวดล้อมของระบบงานที่ใช้ในการทดสอบ

แม้ว่าระบบงานวิจัยจะมีสถาปัตยกรรมเป็นไคล์เอนต์-เซิร์ฟเวอร์ แต่เนื่องจาก ระบบงานวิจัยมีลักษณะการทำงานแบบเครื่องเดี่ยว (Stand-Alone Machine) ดังนั้น การพัฒนาและทดสอบการทำงานของระบบงานวิจัยจึงกระทำบนเครื่องคอมพิวเตอร์เพียงเครื่องเดียว ซึ่งต้องมีสภาพแวดล้อมทั้งทางด้านฮารดแวร์และซอฟต์แวร์ดังนี้

5.1.1 สภาพแวดล้อมทางซอฟต์แวร์

เครื่องคอมพิวเตอร์ที่ใช้ทำงานจะต้องติดตั้งซอฟต์แวร์ ต่อไปนี้

- ระบบปฏิบัติการ Windows NT 4.0 SP 5.0
- ระบบจัดการฐานข้อมูลออรเคิล เวอร์ชัน 8.1.6 ขึ้นไป
- ซอฟต์แวร์ช่วยพัฒนา Oracle JDeveloper 3.0
- เจดีบีซี

5.1.2 สภาพแวดล้อมทางฮาร์ดแวร์

เครื่องคอมพิวเตอร์ที่ใช้จะต้องมีทรัพยากร ดังนี้

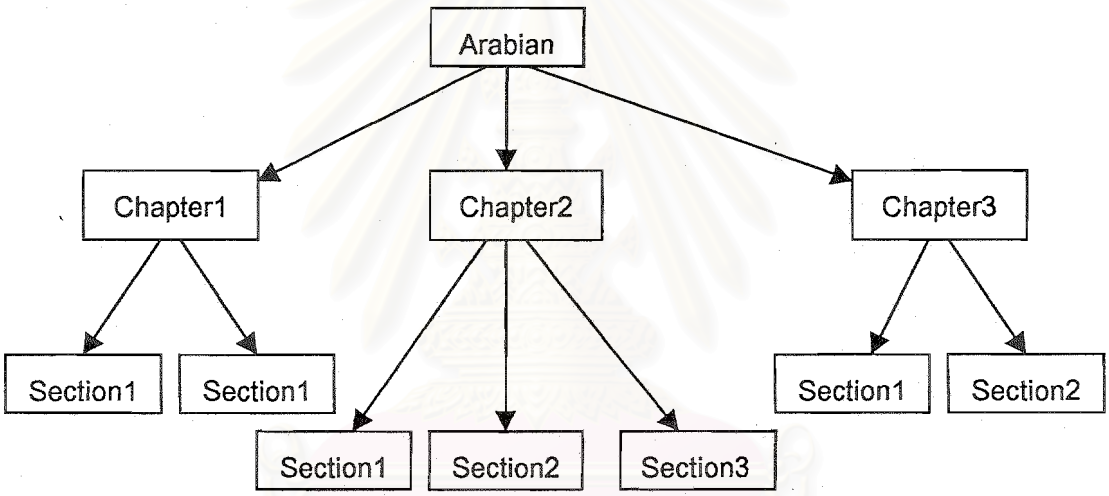
- ความเร็วของซีพียู ต้องไม่น้อยกว่า 450 MHz.
- หน่วยความจำอย่างน้อย 256 Mbyte
- ฮาร์ดดิสก์ประมาณ 10 Gbyte

5.2 ตัวอย่างข้อมูลที่ใช้ในการทดสอบ

การออกแบบระบบงานวิจัยนี้มีวัตถุประสงค์หลักที่จะใช้จัดการกับข้อมูลซึ่งมีความสัมพันธ์แบบคอมโพซิชัน ผู้วิจัยจึงได้สมมติข้อมูลของการเขียนหนังสือขึ้นเพื่อใช้เป็นตัวอย่างข้อมูลในการ

ทดสอบการทำงานของระบบควบคุมเวอร์ชันข้อมูล เนื่องจาก ตัวอย่างการเขียนหนังสือมีลักษณะคล้ายคลึงกับกระบวนการออกแบบที่ความสัมพันธ์ของเอนทิตีรวมและเอนทิตีย่อยมีลักษณะเป็นคอมโพซิชัน

ตัวอย่างข้อมูลที่ใช้ในการทดสอบระบบงานวิจัยเป็นข้อมูลของการแต่งหนังสือหนึ่งเล่มชื่อ Tales ที่มีโครงสร้างต้นไม้ดังแสดงในรูปที่ 5.1 กำหนดให้หนังสือเล่มนี้ประกอบด้วย 3 บทได้แก่ Chapter1 ประกอบด้วย 2 ส่วนย่อย, Chapter2 ประกอบด้วย 3 ส่วนย่อย, และ Chapter3 ประกอบด้วย 2 ส่วนย่อย ในขณะเดียวกันแต่ละเอนทิตีก็จะมีหลายๆ เวอร์ชัน เช่น Arabian มี 3 เวอร์ชัน, Chapter1 มี 3 เวอร์ชัน, Chapter2 มี 4 เวอร์ชัน, Chapter3 มี 4 เวอร์ชัน เป็นต้น โครงสร้างต้นไม้ของเวอร์ชันและส่วนประกอบต่างๆ ของตัวอย่างข้อมูลการแต่งหนังสือ Arabian แสดงในรูป ข.1 ภาคผนวก ข แสดงข้อมูลตัวอย่างสำหรับแอตทริบิวท์ของตารางทั้งหมดในฐานข้อมูล



รูปที่ 5.1 ตัวอย่างหนังสือ Arabian และส่วนประกอบของหนังสือ

5.3 การทดสอบการทำงานของระบบงานวิจัย

เนื่องจาก โปรแกรมของระบบงานวิจัยประกอบด้วยหลายๆ โมดูล และแต่ละโมดูลจะมีหน้าที่การทำงานต่างๆ ที่สัมพันธ์กับตารางในฐานข้อมูล ยกเว้นโมดูลหลักที่ไม่สัมพันธ์กับตารางใดในฐานข้อมูลเลย ประกอบกับตารางในฐานข้อมูลมีความสัมพันธ์กันในลักษณะของการอ้างอิงถึงข้อมูลของอีกตารางหนึ่ง โดยตารางที่ถูกอ้างถึงจะต้องมีข้อมูลนั้นอยู่ ดังนั้น การใช้งานระบบงานวิจัยจึงต้องทำตามลำดับขั้นตอนของกระบวนการใช้งานระบบในหัวข้อ 3.2 สำหรับวิธีการใช้งานและหน้าจอต่างๆ ของระบบควบคุมเวอร์ชันข้อมูล สามารถศึกษาได้จาก ภาคผนวก ก

ด้วยเหตุนี้ การทดสอบการทำงานของระบบงานวิจัยจึงต้องกระทำทีละโมดูลตามลำดับขั้นตอนในหัวข้อ 3.2 เช่นเดียวกัน โดยเริ่มจากโมดูล PROJECT, TREE LEVEL, APPLICATION

CLASS, FACT DATA, USER CLASS, OBJECT VERSIONS, RELATIONSHIPS และ VIEW HIERARCHY ตามลำดับ จากการทดสอบปฏิบัติการต่างๆ ของแต่ละโมดูลทำให้ทราบว่าระบบควบคุมเวอร์ชันข้อมูลมีคุณลักษณะอย่างไร และมีข้อจำกัดในการใช้งานอะไรบ้าง หัวข้อต่อไปเป็นการกล่าวถึงคุณลักษณะของระบบ ส่วนข้อจำกัดในการใช้งานระบบจะกล่าวไว้ในบทที่ 6

5.4 คุณลักษณะของระบบจากการทดสอบระบบ

จากการทดสอบระบบงานวิจัย ระบบควบคุมเวอร์ชันข้อมูลมีคุณลักษณะเพิ่มเติมจากคุณลักษณะที่ผู้วิจัยได้กำหนดไว้ในหัวข้อ 2.3.2 ซึ่งกล่าวถึงคุณลักษณะทั่วไปของระบบในทางทฤษฎี ดังต่อไปนี้

5.4.1 สามารถนำระบบควบคุมเวอร์ชันข้อมูลนี้ไปใช้กับระบบงานจริงที่มีลักษณะความสัมพันธ์แบบแอกกรีเกชันได้

5.4.2 ในแต่ละระดับชั้นของโครงสร้าง ผู้ใช้สามารถสร้างคลาสได้หลายๆ คลาส ยกเว้นระดับชั้นที่ 1 ซึ่งเป็นที่เก็บคลาสของออบเจคต์การออกแบบ เช่น ระดับที่ 2 ของโครงงาน Project2 มีคลาสอยู่ 2 คลาส คือ Component และ Subcomponent เป็นต้น

5.4.3 ในแต่ละระดับชั้นของโครงสร้างต้นไม้ ออบเจคต์รวมหนึ่งออบเจคต์อาจประกอบด้วยออบเจคต์ย่อยจำนวนเท่าไรก็ได้ โดยไม่จำเป็นว่าทุกออบเจคต์รวมในระดับชั้นโครงสร้างต้นไม้นั้นต้องประกอบด้วยออบเจคต์จำนวนที่เท่ากัน

5.4.4 ออบเจคต์รวมหนึ่งๆ อาจประกอบด้วยออบเจคต์ย่อยที่อยู่ต่างระดับชั้นของโครงสร้างต้นไม้ เช่น ออบเจคต์รวมระดับที่2 ประกอบด้วยออบเจคต์ย่อย 2 ออบเจคต์ โดยออบเจคต์ย่อยหนึ่งอยู่ในระดับที่ 3 ของโครงสร้างต้นไม้และออบเจคต์ย่อยอีกออบเจคต์หนึ่งอยู่ในระดับที่ 4 ของโครงสร้างต้นไม้

5.4.5 ผู้ใช้สามารถนำออบเจคต์ข้อมูลดิบที่มีอยู่แล้วในระบบมาใช้ได้หลายครั้งตามต้องการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

ระบบงานทางการค้าส่วนใหญ่ใช้ระบบสารสนเทศในการเก็บและจัดการกับเอนทิตีที่มีค่าข้อมูลเพียงค่าเดียวเท่านั้น แต่ในความเป็นจริงยังมีระบบงานบางประเภทที่เกี่ยวข้องกับเอนทิตีซึ่งมีหลายค่าข้อมูล ระบบงานเหล่านี้ ได้แก่ ระบบงานทางด้านการออกแบบ ระบบงานทะเบียนประวัติ เป็นต้น ด้วยเหตุนี้ผู้วิจัยจึงได้เสนอระบบควบคุมเวอร์ชันข้อมูลเชิงวัตถุขึ้น เพื่อใช้จัดการกับข้อมูลข้อมูลที่มีหลายเวอร์ชัน

หลังจากศึกษางานวิจัยและบทความทางวิชาการต่างๆ แล้ว ผู้วิจัยได้ทำการออกแบบระบบฐานข้อมูล โดยใช้แบบจำลองเชิงวัตถุของ Coad และสร้างสกีมาออบเจคต์ในระบบจัดการฐานข้อมูลออราเคิล เวอร์ชัน 8.1.6 โปรแกรมของระบบงานวิจัยถูกแบ่งออกเป็นโมดูลเพื่อประโยชน์ในการพัฒนาและศึกษาโปรแกรม ผู้วิจัยใช้ซอฟต์แวร์ช่วยพัฒนาชื่อออราเคิลเจเดฟเวอลอปเปอร์ เวอร์ชัน 3.0 บนระบบปฏิบัติการ Windows NT 4.0 SP 5.0 และใช้ข้อมูลการเขียนหนังสือเป็นตัวอย่างสำหรับแสดงการใช้งานและทดสอบระบบงานวิจัย

ระบบงานวิจัยนี้เป็นระบบงานที่ใช้ในการจัดการและจัดเก็บเวอร์ชันข้อมูลของระบบงานทางด้านการออกแบบ โดยผู้ใช้สามารถกำหนดคุณลักษณะของคลาสในระบบงานการออกแบบได้ตามต้องการ ระบบงานวิจัยจะสร้างตารางในฐานข้อมูลเพื่อใช้เก็บและทำปฏิบัติการกับออบเจคต์ของคลาสเหล่านั้น นอกจากนี้ ระบบยังมีส่วนที่ควบคุมดูแลเกี่ยวกับลำดับที่ของแต่ละเวอร์ชันในออบเจคต์หนึ่งๆ และส่วนที่ควบคุมดูแลเกี่ยวกับลำดับที่ของความสัมพันธ์ระหว่างแต่ละเวอร์ชันของออบเจคต์ส่วนประกอบย่อยกับหนึ่งเวอร์ชันของออบเจคต์รวม

งานวิจัยนี้แสดงให้เห็นว่าระบบจัดการฐานข้อมูลสามารถนำมาใช้ในการจัดเก็บและจัดการกับเวอร์ชันข้อมูลของระบบงานทางด้านการออกแบบได้ และยังสามารถนำไปประยุกต์ใช้กับระบบงานประเภทอื่นๆ ที่มีลักษณะคล้ายคลึงกันได้อีกด้วย

6.2 ข้อจำกัด

ระบบควบคุมเวอร์ชันข้อมูลมีข้อจำกัดในการใช้งาน ดังต่อไปนี้

6.2.1 ระบบงานนี้ต้องใช้ทรัพยากรทางด้านฮาร์ดแวร์มาก ในกรณีที่ผู้ใช้ติดตั้งระบบจัดการฐานข้อมูลออราเคิล เวอร์ชัน 8.1.6 และซอฟต์แวร์ช่วยพัฒนาเจเดฟเวอลอปเปอร์ 3.0 ในเครื่องเดียวกัน เครื่องคอมพิวเตอร์ที่สามารถใช้งานระบบควบคุมเวอร์ชันข้อมูลได้ จะต้องประกอบด้วย

หน่วยความจำอย่างน้อย 256 Mbyte ความเร็วของซีพียูต้องไม่น้อยกว่า 450 MHz และมีฮาร์ดดิสก์ประมาณ 10 Gbyte ด้วยเหตุนี้ การใช้งานระบบบางครั้งอาจเกิดข้อผิดพลาดจากการที่ทรัพยากรของระบบไม่เพียงพอ เนื่องจาก การดำเนินงานของบางโมดูลกระทำ ณ. เวลารันไทม์ และบางโมดูลทำปฏิบัติการกับตารางในฐานข้อมูลในลักษณะของการวนลูป

6.2.2 ในการแทรกแถวข้อมูลหรือแก้ไขค่าสดมภ์ของแถวข้อมูลในตาราง ผู้ใช้จะต้องใส่ค่าข้อมูลในฟิลด์ข้อความ (Text Field) ให้ครบทุกฟิลด์ ยกเว้นสดมภ์ CHARFILE และ BINFILE ของตาราง FACTDATA และสดมภ์ FACTDATA_OID ของตาราง OBJ_VERSION ที่สามารถเป็น Null ได้ สดมภ์อื่นของตารางในระบบฐานข้อมูลที่นอกเหนือไปจากนี้ จะเป็น Null ไม่ได้

6.2.3 ไฟล์ข้อมูลดิบทั้งหมดที่ใช้ในระบบงานวิจัย จะต้องถูกเก็บภายใต้ไดเรกทอรี (Directory) E:/Thesis/implement/database/files เท่านั้น เนื่องจาก เป็นข้อจำกัดของระบบจัดการฐานข้อมูลออราเคิลที่กำหนดให้การอ่านค่าในไฟล์ข้อมูลดิบเข้าสู่สดมภ์ซึ่งมีชนิดข้อมูลเป็น BFILE ในระบบฐานข้อมูล ไฟล์ข้อมูลนั้นจะต้องอยู่ในไดเรกทอรีที่กำหนดตอนสร้างสกีมาออบเจคต์ในฐานข้อมูลเท่านั้น สคริปท์สำหรับการสร้างสกีมาออบเจคต์นี้สามารถดูได้จากตาราง ค3 ในภาคผนวก ค

6.2.4 บางโมดูลของโปรแกรมระบบงานวิจัย เช่น โมดูล VIEW HIERARCHY ต้องการใช้เคอร์เซอร์ (Cursor) เป็นจำนวนมาก ในการอ่านข้อมูลจากระบบฐานข้อมูล ดังนั้น การใช้งานโมดูลนี้ ผู้ใช้อาจต้องทำการปิดหน้าจอเป็นระยะๆ เพื่อคืนทรัพยากรระบบให้แก่ระบบจัดการฐานข้อมูลสำหรับการเข้าถึงข้อมูลในครั้งต่อไป

6.2.5 แต่ละออบเจคต์ข้อมูลดิบจะประกอบด้วย หนึ่งไฟล์ข้อมูลดิบชนิดข้อความกับหนึ่งไฟล์ข้อมูลดิบชนิดรูปภาพ หรืออาจประกอบด้วยหนึ่งไฟล์ข้อมูลดิบชนิดข้อความเพียงอย่างเดียวหรือหนึ่งไฟล์ข้อมูลดิบชนิดรูปภาพเพียงอย่างเดียว

6.3 ข้อเสนอแนะ

ข้อเสนอแนะเพิ่มเติมสำหรับผู้ที่ต้องการศึกษาและพัฒนาระบบควบคุมเวอร์ชันข้อมูลเชิงวัตถุในระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ เพื่อใช้กับระบบงานจริงที่มีลักษณะเช่นเดียวกับกระบวนการออกแบบ ได้แก่

6.3.1 ควรสร้างฟังก์ชั่นที่ทำงานเกี่ยวกับการตรวจสอบความแตกต่างระหว่างแต่ละเวอร์ชัน ของออบเจคต์ และแสดงค่าความแตกต่างนั้นออกทางหน้าจอเพื่อเป็นข้อมูลให้กับผู้ใช้ในการตัดสินใจสร้างเวอร์ชันใหม่ของออบเจคต์

6.3.2 ควรเพิ่มประเภทของไฟล์ข้อมูลดิบให้หลากหลายมากขึ้น เพราะระบบงานวิจัยนี้สามารถใช้งานกับไฟล์ข้อมูลดิบได้เพียง 3 สกุลเท่านั้น คือ .txt, .jpg, และ .gif

6.3.3 ควรเพิ่มฟังก์ชั่นสำหรับเรียกใช้งานซอฟต์แวร์บรรณาธิกรณ์ต่างๆ เพื่อให้ผู้ใช้สามารถสร้างไฟล์ข้อมูลดิบได้ในทันทีที่ต้องการ

6.3.4 ควรเพิ่มส่วนที่จัดการเกี่ยวกับการควบคุมการทำงานพร้อมกันเพื่อให้ผู้ใช้หลายคนสามารถใช้งานระบบงานวิจัยในเวลาเดียวกันได้

6.3.5 ควรปรับปรุงหน้าจอซึ่งเป็นส่วนต่อประสานผู้ใช้ให้สวยงามน่าใช้มากกว่าเดิม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. W. Kim, and F. Lochovsky. Object-Oriented Concepts: Databases, and Applications. New York : ACM Press, 1989.

2. J. Won, and R. Elmasri. Representing Retroactive and Proactive Versions in Bi-Temporal DataBases(2TDB). Proceedings of the IEEE Data Engineering Conference 1996 (Texas : Feb. 1996) : pp.85-94.

3. W. Kafer, and H. Schoning. Mapping a Version Model to a Complex-Object Data Model. Proceedings of the IEEE Data Engineering Conference 1992 (Germany : Jul. 1992) : pp.348-357.

4. E. Sciore. Using Annotations to Support Multiple Kinds of Versioning in an Object-Oriented Database System. ACM Transactions on Database Systems Vol.16,3 (Sept. 1991) : pp.417-438.

5. D. Beech, and B. Mahbod. Generalized Version Control in an Object-Oriented Database. Proceedings of the IEEE Data Engineering Conference 1988 (California : Feb. 1988) : pp.14-22.

6. S. Alhir. UML in a Nutshell: A Desktop Quick Reference. US : O'Reilly & Associates, Inc., 1998.

7. R.G.G Cattell. Object Data Management: Object-Oriented and Extended Relational Database Systems. : Addison-Wesley Publishing Company, 1994

8. A. Silberschatz, and H.F. Korth, and S. Sudarshan. Database system Concepts. 3rd Edition. Singapore : McGraw-Hill, 1997.

9. R.J. Norman. Object-Oriented Systems Analysis and Design. US : Prentice Hall International, Inc., 1996.

10. K. Dittrich, and R. Lorie. Version Support for Engineering Database Systems. IEEE Transactions on Software Engineering Vol.14,4 (April 1988) : pp.429-437.

11. R. Conradi, and B. Westfechtel. Version Models for Software Configuration Management. ACM Computing Surveys Vol.30,2 (June 1998) : pp.232-282.

12. Oracle Systems (Thailand) Co.,Ltd. Oracle8I On-Line Generic Documentation CD-ROM Release2 (8.1.6) US : Oracle Corporation 2000, 2000.

13. Sun Microsystems. http://www.java.sun.com/docs/books/tutorial/information/download.html. US : Sun Microsystem,

14. Oracle Corporation. http://technet.oracle.com/products/jdev/index2.htm. US : Oracle Corporation, 1999

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก
คู่มือการใช้งานระบบควบคุมเวอร์ชันข้อมูลเชิงวัตถุ

ภาคผนวกนี้แสดงวิธีการใช้งานระบบควบคุมเวอร์ชันข้อมูลตามกระบวนการใช้งานระบบที่ได้กล่าวแล้วในหัวข้อ 3.2 โดยใช้ตัวอย่างข้อมูลของการเขียนหนังสือในภาคผนวก ข ประกอบการอธิบายและแสดงหน้าจอซึ่งเป็นผลที่เกิดจากการทำปฏิบัติการต่างๆ กับระบบ

ขั้นตอนการใช้งานระบบควบคุมเวอร์ชันข้อมูล

รูป ก1 แสดงหน้าจอหลักซึ่งเป็นหน้าจอเริ่มต้นในการใช้งานระบบ ด้านขวาของหน้าจอมีปุ่มอยู่ 8 ปุ่ม โดยแต่ละปุ่มทำหน้าที่ในการเรียกใช้งานโมดูลเฉพาะที่แตกต่างกัน การอธิบายขั้นตอนการใช้งานระบบจะกระทำตามกระบวนการใช้งานระบบในหัวข้อ3.2 โดยจะเริ่มจากปุ่มบนสุดลงมาจนถึงปุ่มล่างสุด และในแต่ละโมดูลก็จะแสดงวิธีการสอบถามข้อมูล การแทรกแถวข้อมูล การแก้ไขค่าแอตทริบิวท์ การลบแถวข้อมูลออกจากตาราง และปฏิบัติการอื่นๆ

รูป ก1 หน้าจอหลักของระบบควบคุมเวอร์ชันข้อมูลเชิงวัตถุ

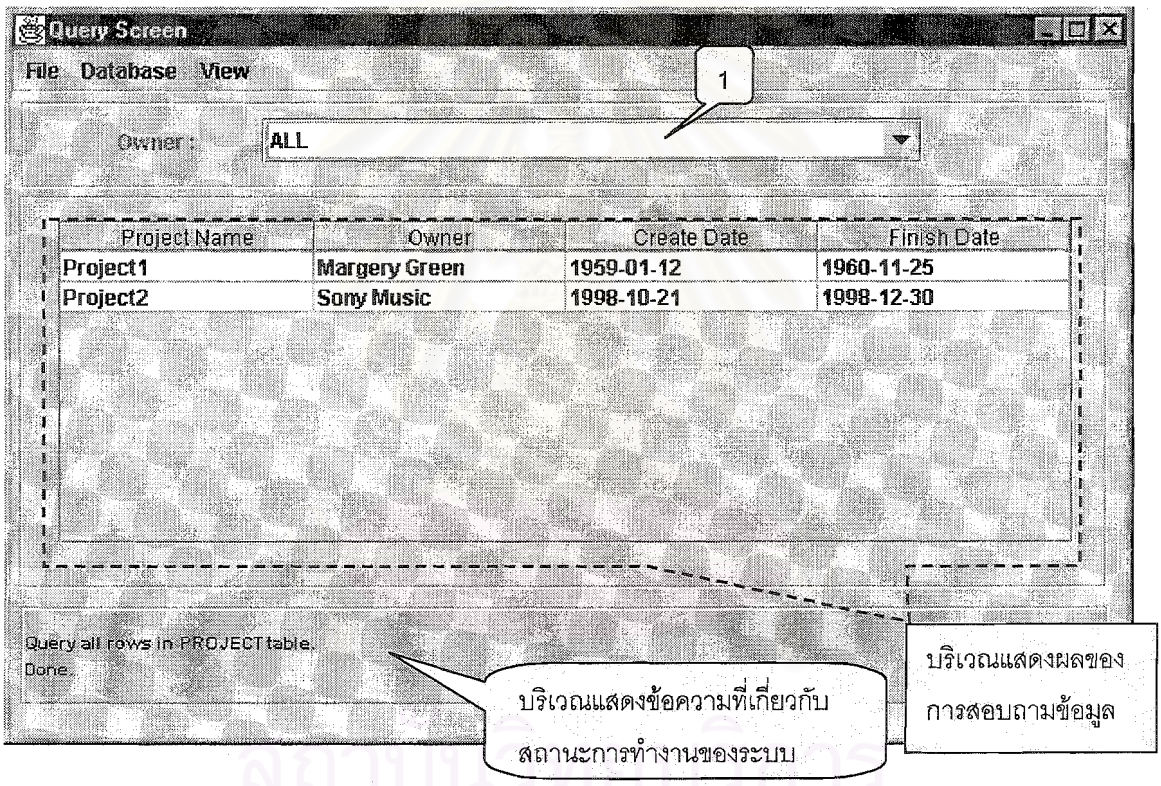
หากผู้ใช้ต้องการออกจากระบบ จะทำได้โดยการกดปุ่ม

ก.1 ปฏิบัติการต่าง ๆ ของโมดูล PROJECT

ก.1.1 การเริ่มต้นใช้งานโมดูลและสอบถามข้อมูล

เมื่อผู้ใช้กดปุ่ม PROJECT ระบบจะแสดงหน้าจอสอบถามของโมดูล PROJECT รูป ก2 เป็นหน้าจอเริ่มต้นในการใช้งานโมดูลนี้

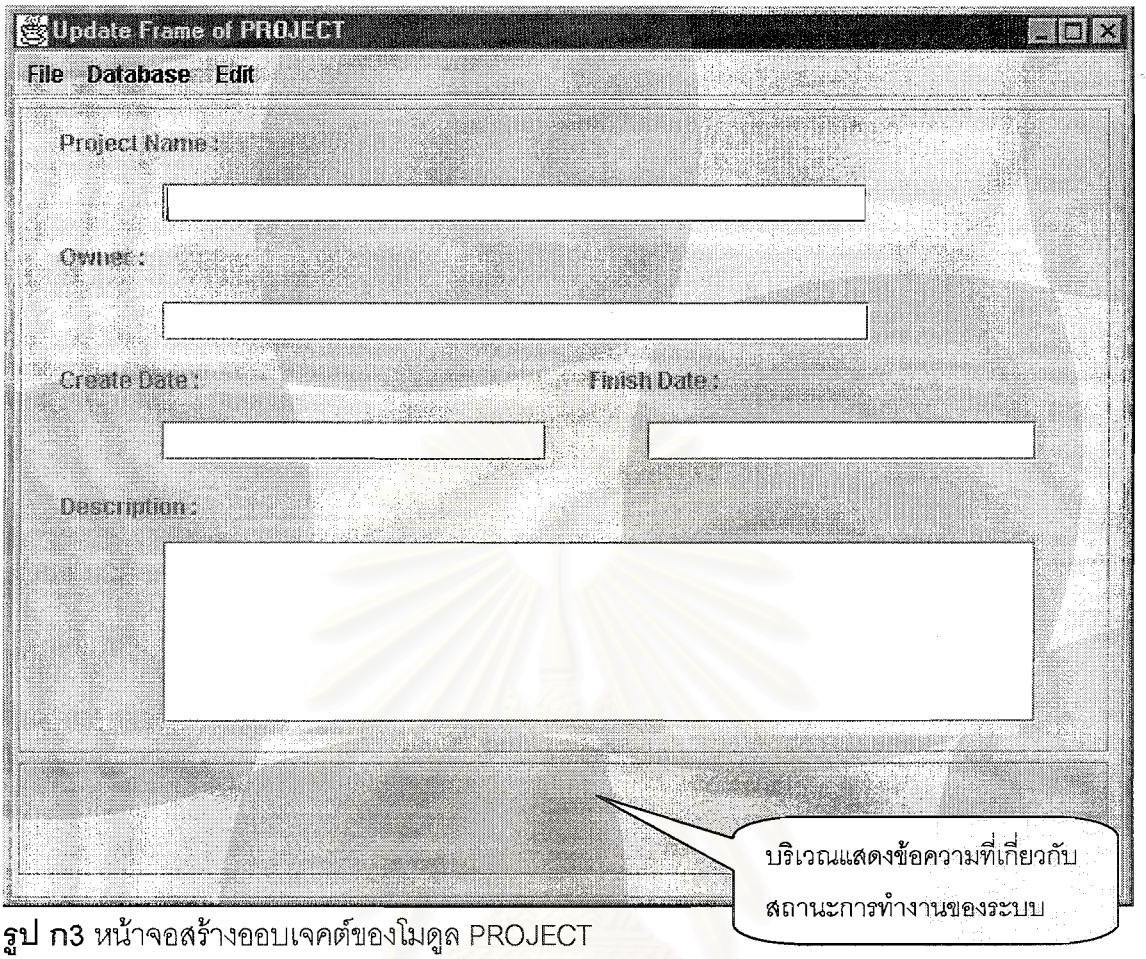
ผู้ใช้สามารถกำหนดเงื่อนไขการสอบถามข้อมูลของตาราง PROJECT0 ได้โดยเลือกเจ้าของโครงงานจากส่วนประกอบหมายเลข 1 บริเวณแสดงผลการสอบถามข้อมูลจะแสดงข้อมูลของโครงงานที่มีคุณสมบัติตรงตามเงื่อนไข แต่หากผู้ใช้ไม่กำหนดเงื่อนไขใดเลย ระบบจะแสดงข้อมูลของโครงงานทั้งหมด



รูป ก2 หน้าจอสอบถามของโมดูล PROJECT

ก.1.2 การแทรกแถวข้อมูลหรือการสร้างออบเจคต์

การแทรกแถวข้อมูลในตาราง PROJECT0 ผู้ใช้ต้องเลือกเมนู Database และเมนูย่อย Insert Screen ระบบจะแสดงหน้าจอสร้างออบเจคต์ของโมดูล PROJECT



รูป ก3 หน้าจอสร้างออบเจคต์ของโมดูล PROJECT

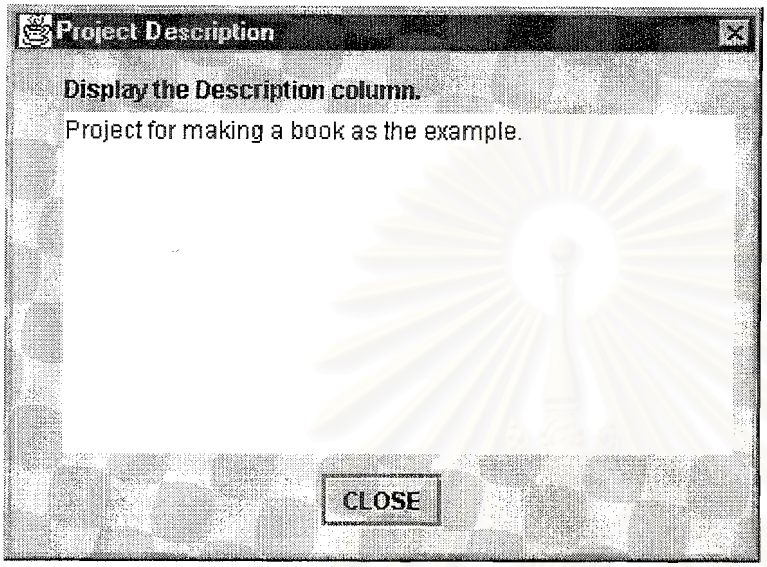
หลังจากผู้ใช้ป้อนค่าข้อมูลในฟิลด์ต่างๆ ของหน้าจอสร้างออบเจคต์ ดังรูป ก3 แล้ว ให้ผู้ใช้เลือกเมนูย่อย Insert ของเมนู Database เพื่อแทรกแถวข้อมูลเข้าตาราง PROJECT0 ในฐานข้อมูล เมื่อปฏิบัติการแทรกแถวข้อมูลสำเร็จแล้วระบบจะทำการเคลียร์ค่าในฟิลด์ของหน้าจอ ถ้าผู้ใช้ต้องการทำปฏิบัติการแทรกแถวข้อมูลต่อไป ก็สามารถทำได้โดยป้อนค่าข้อมูลในทุกฟิลด์ของหน้าจอ แต่หากผู้ใช้ไม่ต้องการทำปฏิบัติการแทรกแถวข้อมูลแล้ว ให้เลือกเมนูย่อย Close ของเมนู File จะเป็นการปิดหน้าจอนี้

ก.1.3 การปรับปรุงหน้าจอสอบถามหลังทำปฏิบัติการต่าง ๆ

ในหน้าจอสอบถาม หากผู้ใช้ต้องการดูผลการทำปฏิบัติการต่างๆ ให้เลือกเมนูย่อย Query ของเมนู Database หน้าจอสอบถามจะแสดงผลของปฏิบัติการนั้น

ก.1.4 การเรียกดูค่าข้อมูลแอตทริบิวท์ DESCRIPTION

ผู้ใช้ต้องเลือกโครงงานที่ต้องการดูค่าสดมภ์ DESCRIPTION ก่อน จากนั้น เลือกเมนูย่อย Description ในเมนู View ระบบจะแสดงค่าข้อมูลของแอตทริบิวท์ DESCRIPTION ดังรูป ก4



รูป ก4 หน้าจอย่อยแสดงค่าข้อมูลแอตทริบิวท์ของโครงงาน Project1

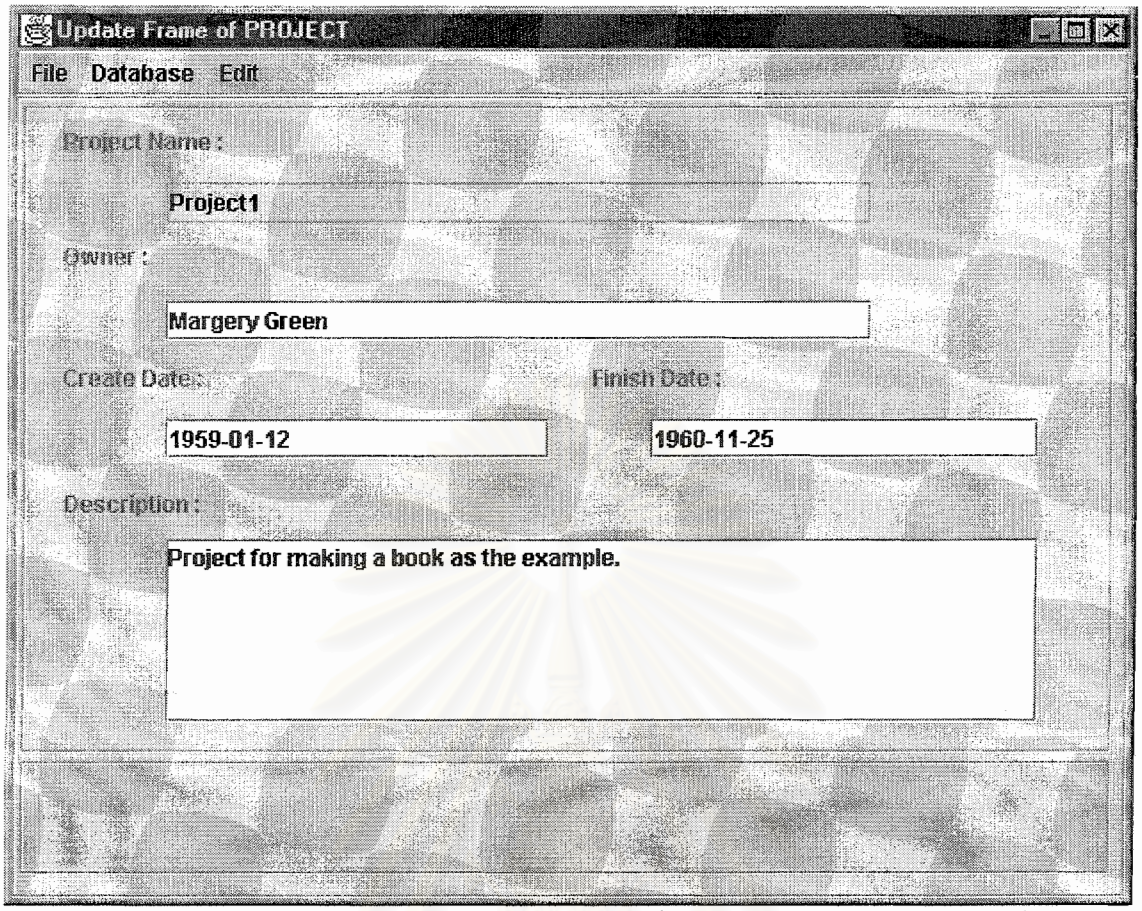
ก.1.5 การแก้ไขค่าแอตทริบิวท์ของแถวข้อมูล

ผู้ใช้ต้องเลือกแถวข้อมูลของโครงงานที่ต้องการแก้ไขค่าแอตทริบิวท์จากหน้าจอสอบถาม และเลือกเมนูย่อย Update Screen ของเมนู Database ระบบจะแสดงหน้าจอแก้ไขออบเจคต์ โดยทุกฟิลด์ในหน้าจอจะมีค่าแอตทริบิวท์ของแถวข้อมูลก่อนการแก้ไข รูป ก5 ผู้ใช้ไม่สามารถทำการแก้ไขค่าของฟิลด์ Project Name ได้ เนื่องจากเป็นค่าคีย์หลักของตาราง PROJECT0

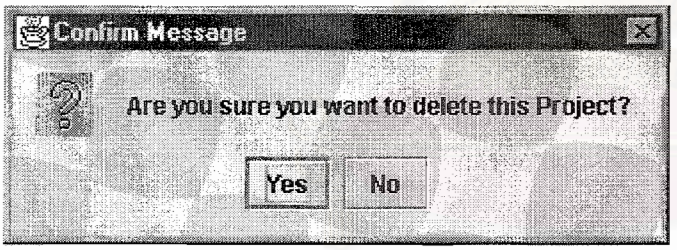
หลังจากผู้ใช้ได้ทำการแก้ไขค่าข้อมูลในฟิลด์แล้วและต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตาราง PROJECT0 ในฐานข้อมูล ผู้ใช้ต้องเลือกเมนูย่อย Update จากเมนู Database แต่หากผู้ใช้ไม่ต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตาราง PROJECT0 ก็สามารถทำได้โดยการปิดหน้าจอแก้ไขนี้

ก.1.6 การลบแถวข้อมูล

ในหน้าจอสอบถาม ผู้ใช้ต้องเลือกแถวข้อมูลของโครงงานที่ต้องการลบก่อน จากนั้นเลือกเมนูย่อย Delete ของเมนู Database ระบบจะแสดงหน้าจอย่อย รูป ก6 ซึ่งเป็นหน้าจอที่แสดงข้อความขอคำยืนยันการลบแถวข้อมูล หากผู้ใช้กดปุ่ม Yes ระบบจะทำปฏิบัติการลบแถวข้อมูลนั้นออก



รูป ก5 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล PROJECT: ก่อนการแก้ไข



รูป ก6 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบแถวข้อมูลจากตาราง PROJECT0

จากตาราง PROJECT0 ในระบบฐานข้อมูล แต่หากผู้ให้กดปุ่ม No ระบบจะไม่ทำปฏิบัติการลบแถวข้อมูล

ก.1.7 การปิดหน้าจอ

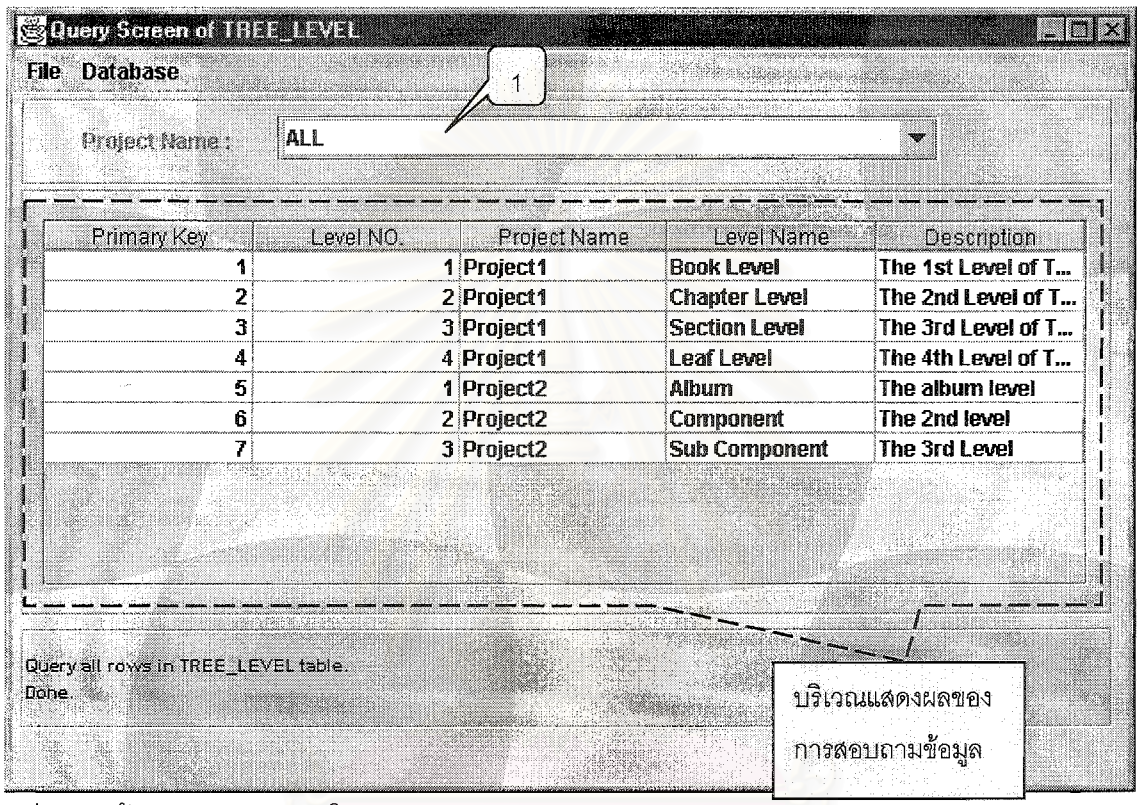
ทุกหน้าจอของระบบจะมีเมนูย่อย Close ของเมนู File เมนูนี้จะทำการปิดหน้าจอที่แสดงอยู่

ก.2 ปฏิบัติการต่าง ๆ ของโมดูล TREE LEVEL

ก.2.1 การเริ่มต้นใช้งานโมดูลและสอบถามข้อมูล

เมื่อผู้ใช้กดปุ่ม TREE LEVEL ระบบจะแสดงหน้าจอสอบถามของโมดูล TREE LEVEL รูป ก7 เป็นหน้าจอเริ่มต้นในการใช้งานโมดูลนี้

ผู้ใช้สามารถกำหนดเงื่อนไขการสอบถามข้อมูลของตาราง TREE_LEVEL0 ได้โดยเลือกชื่อโครงงานจากส่วนประกอบหมายเลข 1 บริเวณแสดงผลการสอบถามข้อมูลจะแสดงข้อมูลของระดับชั้นโครงสร้างต้นไม้ของโครงงานที่ผู้ใช้กำหนด แต่หากผู้ใช้ไม่กำหนดเงื่อนไขใดเลย ระบบจะแสดงข้อมูลของทุกระดับชั้นโครงสร้างต้นไม้ในทุกโครงงาน



รูป ก7 หน้าจอสอบถามของโมดูล TREE LEVEL

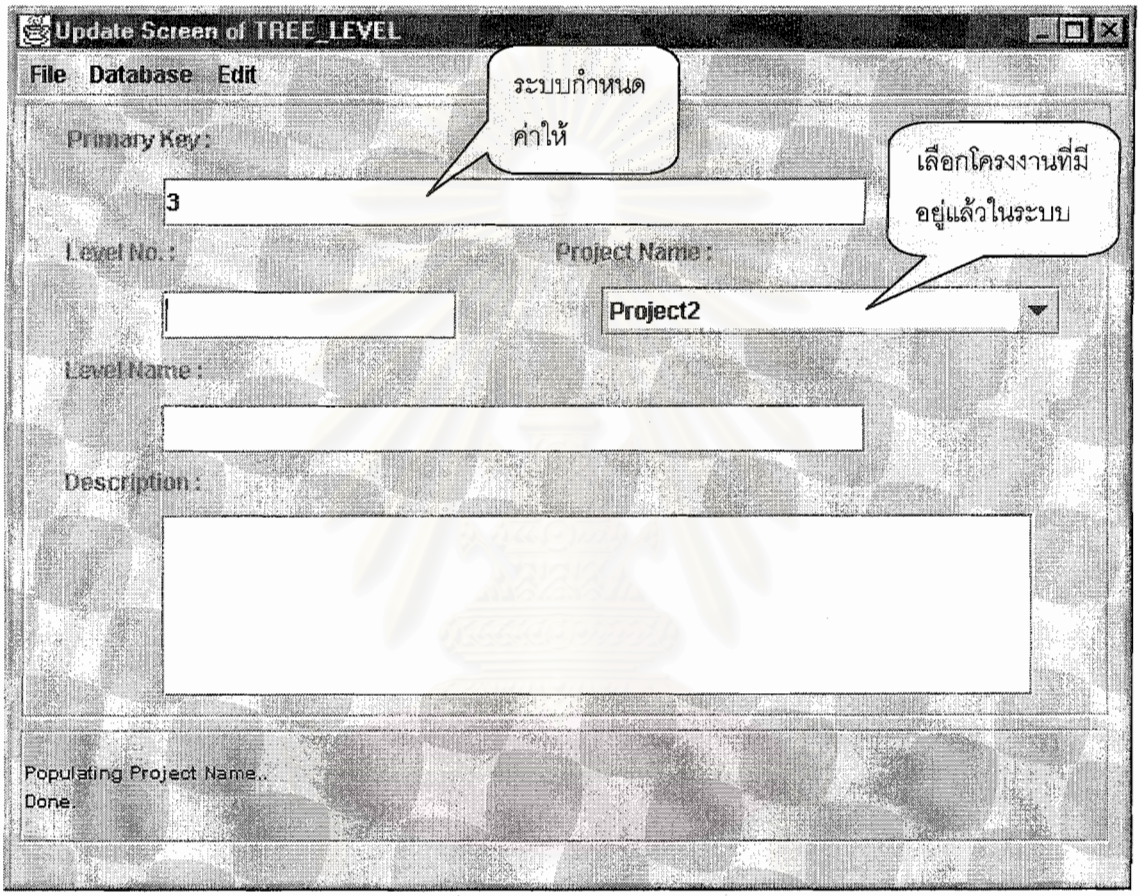
ก.2.2 การแทรกแถวข้อมูล

การแทรกแถวข้อมูลในตาราง TREE_LEVEL0 ผู้ใช้ต้องเลือกเมนูย่อย Insert Screen ของเมนู Database ในหน้าจอสอบถาม ระบบจะแสดงหน้าจอสร้างออบเจคต์ของโมดูล TREE LEVEL ดังรูป ก8 เพื่อให้ผู้ใช้ใส่ค่าข้อมูลให้กับแต่ละแอตทริบิวท์ของระดับชั้นโครงสร้างต้นไม้

จากนั้นถ้าผู้ใช้ต้องการทำปฏิบัติการแทรกแถวข้อมูลในตาราง TREE_LEVEL0 ที่หน้าจอสร้างออบเจคต์ ให้เลือกเมนูย่อย Insert ของเมนู Database เมื่อปฏิบัติการแทรกแถวข้อมูลสำเร็จแล้วระบบจะเคลียร์ค่าในฟิลด์ของหน้าจอ หากผู้ใช้ต้องการทำปฏิบัติการแทรกแถวข้อมูลแถวต่อไปก็สามารถทำได้โดยป้อนค่าข้อมูลในทุกฟิลด์ของหน้าจอ รวมทั้งฟิลด์ Primary Key ด้วย ผู้ใช้ควรระวังในการกำหนดค่าให้แก่ฟิลด์ Primary Key โดยพยายามกำหนดค่าที่ไม่ซ้ำกับค่าในสดมภ์

Levproj_id ของตาราง TREE_LEVEL0 แต่ถ้าผู้ใช้ต้องการให้ระบบกำหนดค่าในฟิลด์ Primary Key ให้ ผู้ใช้ต้องปิดหน้าจอสร้างออบเจคต์นี้ก่อน แล้วจึงค่อยเปิดหน้าจอสร้างออบเจคต์ขึ้นใหม่

หากผู้ใช้เปลี่ยนใจไม่ทำปฏิบัติการแทรกแถวข้อมูล ก็สามารถทำได้โดยการเลือกเมนูย่อย Close จากเมนู File เพื่อปิดหน้าจอสร้างออบเจคต์ หรือเลือกเมนูย่อย Clear ของเมนู Edit เพื่อเคลียร์ค่าในฟิลด์ของหน้าจอ



รูป ก8 หน้าจอสร้างออบเจคต์ของโมดูล TREE LEVEL

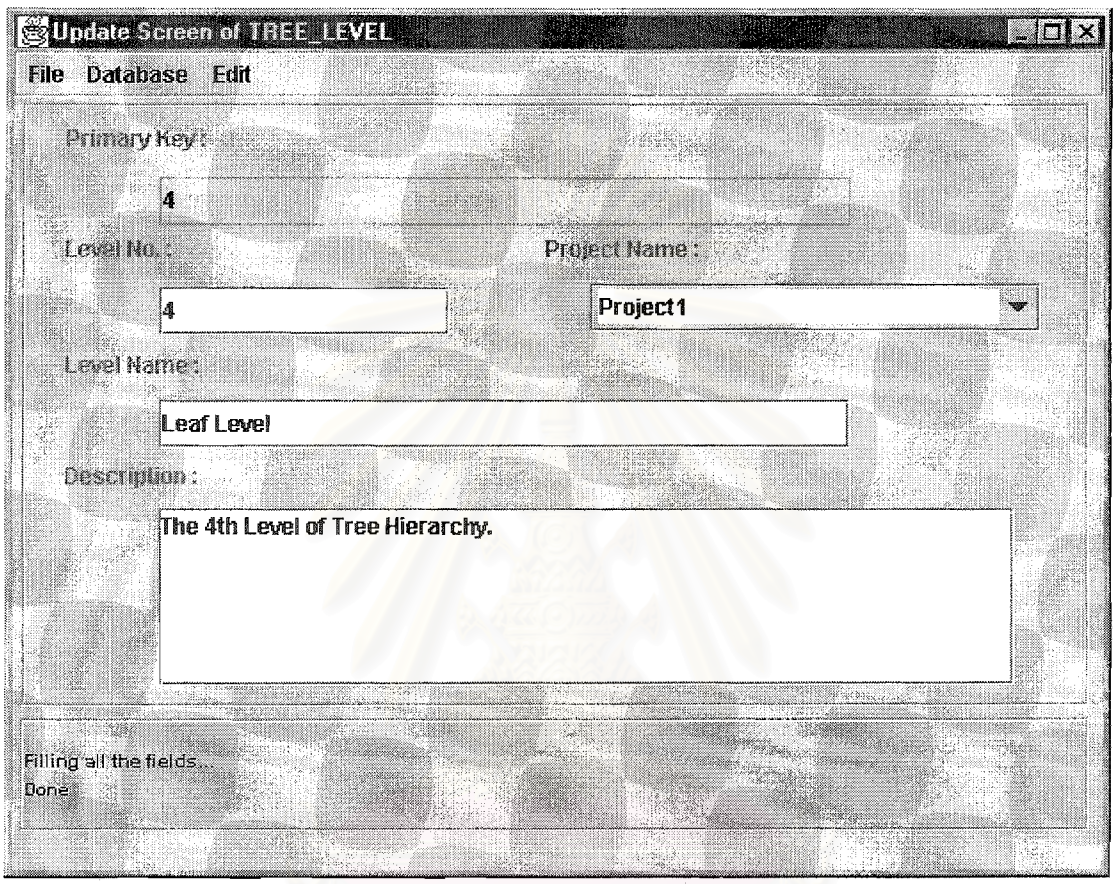
ก.2.3 การปรับปรุงหน้าจอสอบถามหลังทำปฏิบัติการต่าง ๆ

หากผู้ใช้ต้องการดูผลการทำปฏิบัติการต่างๆ ให้เลือกเมนูย่อย Query จากเมนู Database หน้าจอสอบถามของโมดูลจะแสดงผลของปฏิบัติการนั้น

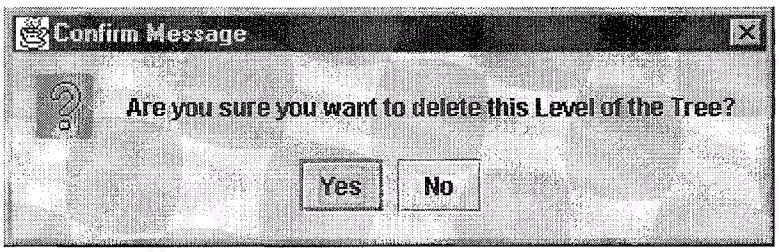
ก.2.4 การแก้ไขค่าแอตทริบิวท์ของแถวข้อมูล

ผู้ใช้ต้องเลือกแถวข้อมูลของระดับชั้นโครงสร้างต้นไม้ที่ต้องการแก้ไขค่าแอตทริบิวท์จากหน้าจอสอบถาม และเลือกเมนูย่อย Update Screen ของเมนู Database ระบบจะแสดงหน้าจอแก้ไขออบเจคต์ โดยทุกฟิลด์ในหน้าจอนี้จะมีค่าแอตทริบิวท์ของแถวข้อมูลก่อนการแก้ไข ดังรูป ก9

หลังจากผู้ใช้ได้ทำการแก้ไขค่าข้อมูลในฟิลด์แล้วและต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตาราง TREE_LEVEL0 ในฐานข้อมูล ผู้ใช้ต้องเลือกเมนูย่อย Update จากเมนู Database แต่หากผู้ใช้ไม่ต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตาราง TREE_LEVEL0 ก็สามารถทำได้โดยปิดหน้าจอแก้ไขนี้



รูป ก9 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล TREE LEVEL: ก่อนการแก้ไข



รูป ก10 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบแถวข้อมูลจากตาราง TREE_LEVEL0

ก.2.5 การลบแถวข้อมูล

ในหน้าจอสอบถาม ผู้ใช้ต้องเลือกแถวข้อมูลของระดับชั้นโครงสร้างต้นไม้ที่ต้องการลบก่อน จากนั้นเลือกเมนูย่อย Delete ของเมนู Database ระบบจะแสดงหน้าจอย่อย ดังรูป ก10 เพื่อให้ผู้ใช้

ยืนยันการทำปฏิบัติการลบแถวข้อมูล หากผู้ใช้กดปุ่ม Yes ระบบจะทำการลบแถวข้อมูลนั้นออกจากตาราง TREE_LEVEL0 ในฐานข้อมูล แต่ถ้าผู้ใช้กดปุ่ม No ระบบจะไม่ทำปฏิบัติการนี้

ก.3 ปฏิบัติการต่าง ๆ ของโมดูล APPLICATION CLASS

ก.3.1 การเริ่มต้นใช้งานโมดูลและสอบถามข้อมูล

เมื่อผู้ใช้กดปุ่ม APPLICATION CLASS ระบบจะแสดงหน้าจอสอบถามของโมดูล APPLICATION CLASS รูป ก11 เป็นหน้าจอเริ่มต้นในการใช้งานโมดูลนี้

ผู้ใช้สามารถกำหนดเงื่อนไขการสอบถามข้อมูลของตาราง APP_CLASS ได้โดยเลือกชื่อโครงงานจากส่วนประกอบหมายเลข 1 บริเวณแสดงผลการสอบถามข้อมูลจะแสดงข้อมูลของคลาสในโครงงานที่ผู้ใช้กำหนด แต่หากผู้ใช้ไม่กำหนดเงื่อนไขใดเลย ระบบจะแสดงข้อมูลของทุกคลาสในระบบ

ก.3.2 การแทรกแถวข้อมูลในตาราง APP_CLASS

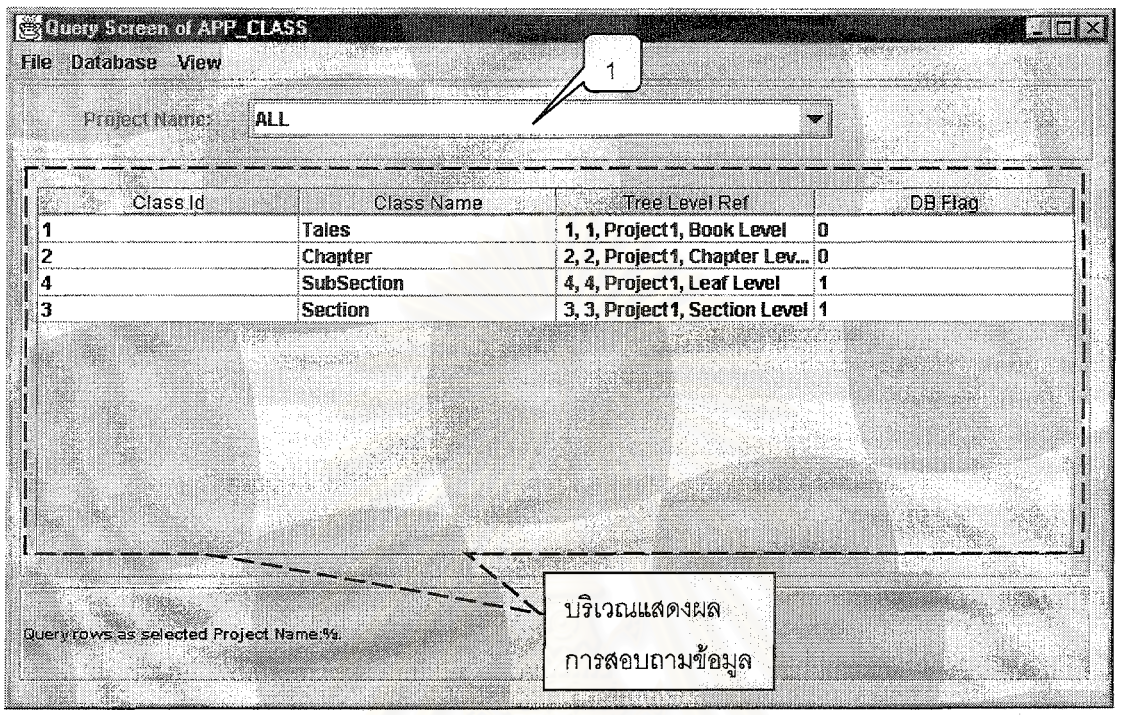
แต่ละแถวข้อมูลในตาราง APP_CLASS เป็นนิยามของคลาสที่ผู้ใช้กำหนดขึ้น โดยระบบงานวิจัยจะนำนิยามของคลาสนี้ไปใช้ในการสร้างตารางในฐานข้อมูลเพื่อเป็นที่สำหรับเก็บออบเจคต์ของคลาส หากผู้ใช้กำหนดคุณลักษณะของคลาสใหม่ขึ้นและต้องการเก็บคลาสนั้นเพื่อใช้ในการสร้างตาราง ผู้ใช้ต้องทำปฏิบัติการแทรกแถวข้อมูลเข้าในตาราง APP_CLASS โดยเลือกเมนูย่อย Insert Screen ของเมนู Database ระบบจะแสดงหน้าจอสร้างออบเจคต์ของโมดูล APPLICATION CLASS ดังรูป ก13 จะเห็นว่าหน้าจอนี้ประกอบด้วย 2 ส่วน คือ ส่วนที่กำหนดข้อมูลของคลาสและส่วนที่กำหนดคุณลักษณะของแอตทริบิวท์ในคลาสนั้น

การกดปุ่ม LIST ซึ่งอยู่ในส่วนข้อมูลของคลาส ระบบจะแสดงหน้าจอย่อย ดังรูป ก12 หน้าจอนี้แสดงรายละเอียดเกี่ยวกับระดับชั้นของโครงสร้างต้นไม้ทั้งหมดที่มีในระบบ เพื่อให้ผู้ใช้เลือกว่าคลาสที่กำลังนิยามนี้ควรอยู่ในระดับใดของโครงสร้างต้นไม้

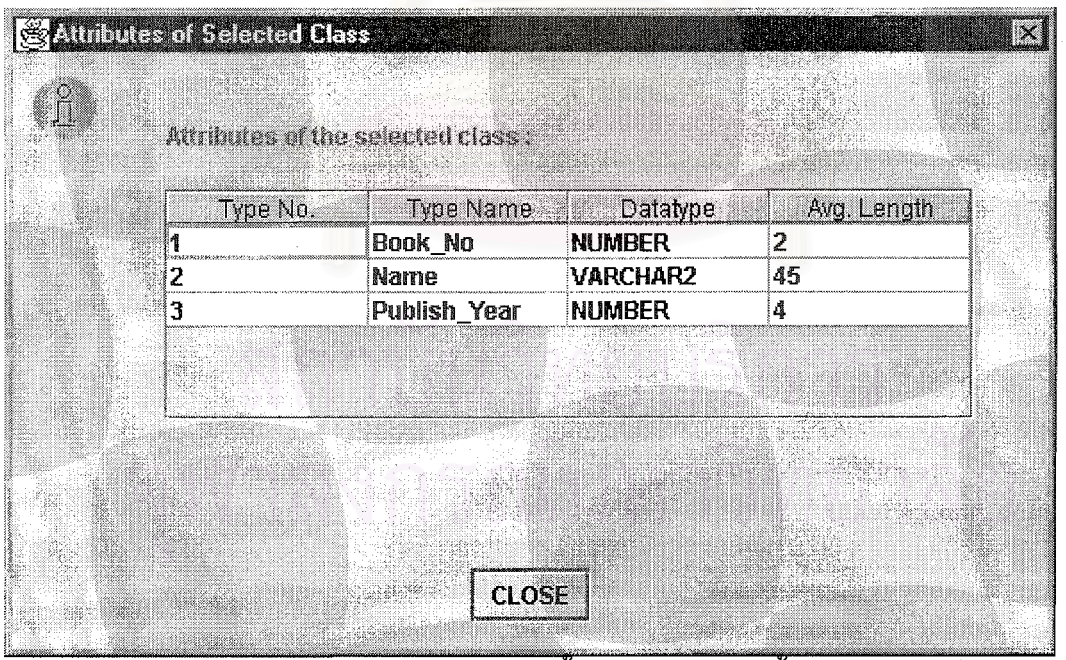
ส่วนกำหนดแอตทริบิวท์ของคลาสเป็นส่วนที่ใช้สำหรับกำหนดแอตทริบิวท์ใหม่ แก้ไขคุณลักษณะของแอตทริบิวท์เดิม หรือลบแอตทริบิวท์ที่ไม่ต้องการ

ในการนิยามแอตทริบิวท์ของคลาส ผู้ใช้จะนิยามได้ครั้งละแอตทริบิวท์ โดยกำหนดค่าให้กับฟิลด์ Attribute Order, Attribute Name, Datatype, และ Average Length แล้วกดปุ่ม ADD ระบบจะนำค่าของฟิลด์เหล่านั้นแสดงในตารางแอตทริบิวท์ พร้อมกับเคลียร์ค่าในฟิลด์ เพื่อให้ผู้ใช้กำหนดคุณลักษณะของแอตทริบิวท์ต่อไป ถ้าต้องการแก้ไขคุณลักษณะบางอย่างของแอตทริบิวท์ ผู้ใช้จะต้องเลือกแอตทริบิวท์ที่ต้องการแก้ไขจากตารางก่อน ค่าคุณลักษณะก่อนแก้ไขของแอตทริบิวท์จะถูกอ่านจากตารางและนำมาใส่ในฟิลด์ เมื่อผู้ใช้ทำการแก้ไขค่าคุณลักษณะในฟิลด์แล้ว ให้กดปุ่ม

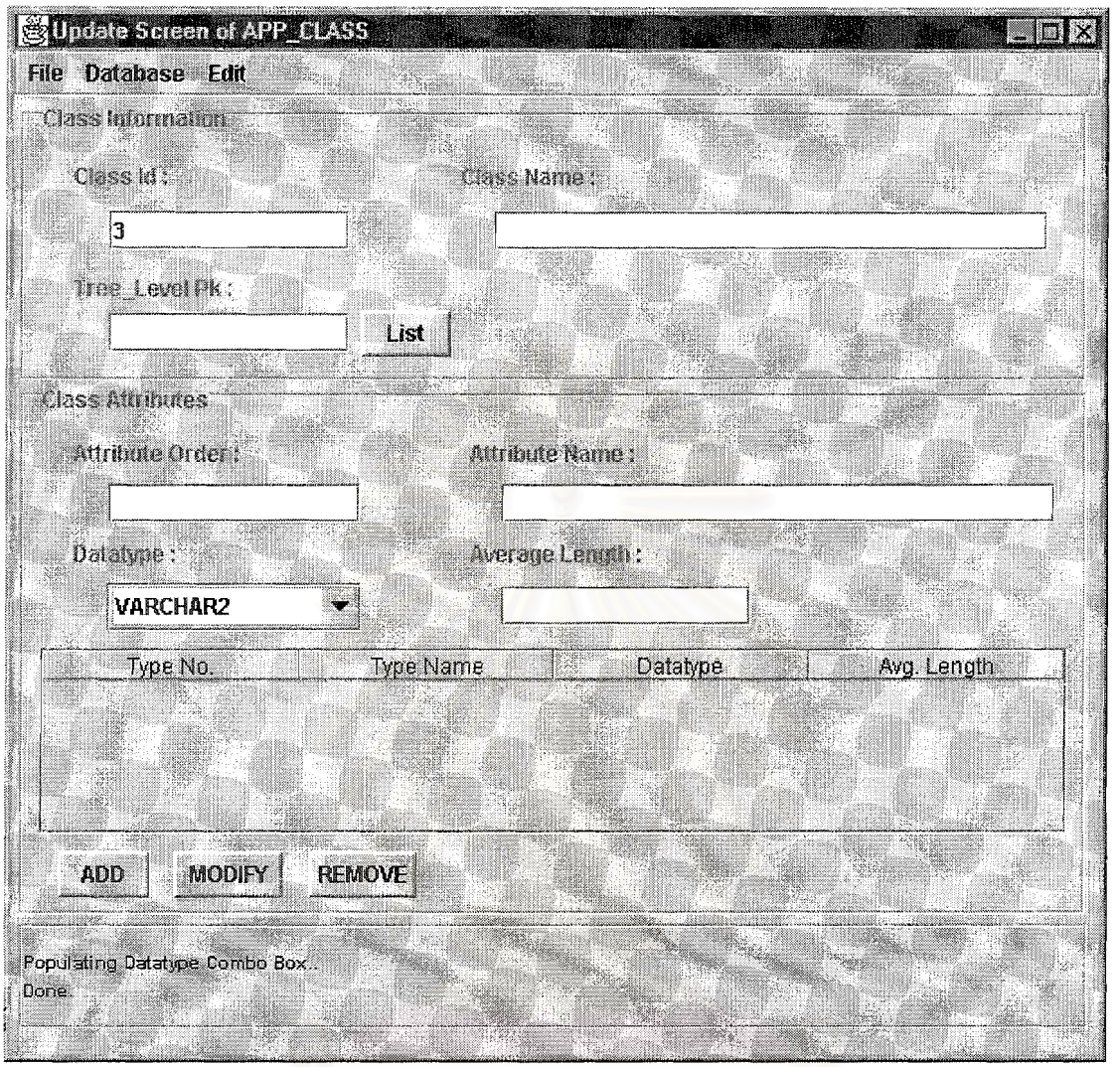
MODIFY เพื่อปรับปรุงคุณลักษณะแอตทริบิวท์ให้ถูกต้อง สำหรับการลบแอตทริบิวท์ ผู้ใช้จะต้องเลือกแอตทริบิวท์ที่ต้องการลบในตารางก่อน แล้วจึงกดปุ่ม REMOVE



รูป ก11 หน้าจอสอบถามของโมดูล APPLICATION CLASS



รูป ก12 หน้าจอย่อยแสดงรายละเอียดของระดับชั้นโครงสร้างต้นไม้ทั้งหมดในระบบ



รูป ก13 หน้าจอสร้างออบเจคต์ของโมดูล APPLICATION CLASS

หลังจากกำหนดค่าให้กับฟิลด์ต่างๆ ของหน้าจอเรียบร้อยแล้ว ให้ผู้ใช้เลือกเมนูย่อย Insert จากเมนู Database เพื่อแทรกแถวข้อมูลเข้าตาราง APP_CLASS ในฐานข้อมูล ถ้าปฏิบัติการแทรกแถวข้อมูลกระทำสำเร็จ ระบบจะเคลียร์ค่าในฟิลด์ของหน้าจอ หากผู้ใช้ต้องการทำปฏิบัติการแทรกแถวข้อมูลแถวต่อไป ก็สามารถทำได้โดยป้อนค่าข้อมูลในทุกฟิลด์ของหน้าจอ รวมทั้งฟิลด์ Class Id ด้วย ผู้ใช้ควรระวังในการกำหนดค่าให้แก่ฟิลด์ Class Id โดยพยายามกำหนดค่าที่ไม่ซ้ำกับค่าในสดมภ์ class_id ของตาราง APP_CLASS แต่ถ้าผู้ใช้ต้องการให้ระบบกำหนดค่าในฟิลด์ Class Id ให้ ผู้ใช้ต้องปิดหน้าจอสร้างออบเจคต์นี้ก่อน แล้วจึงค่อยเปิดหน้าจอสร้างออบเจคต์ขึ้นใหม่

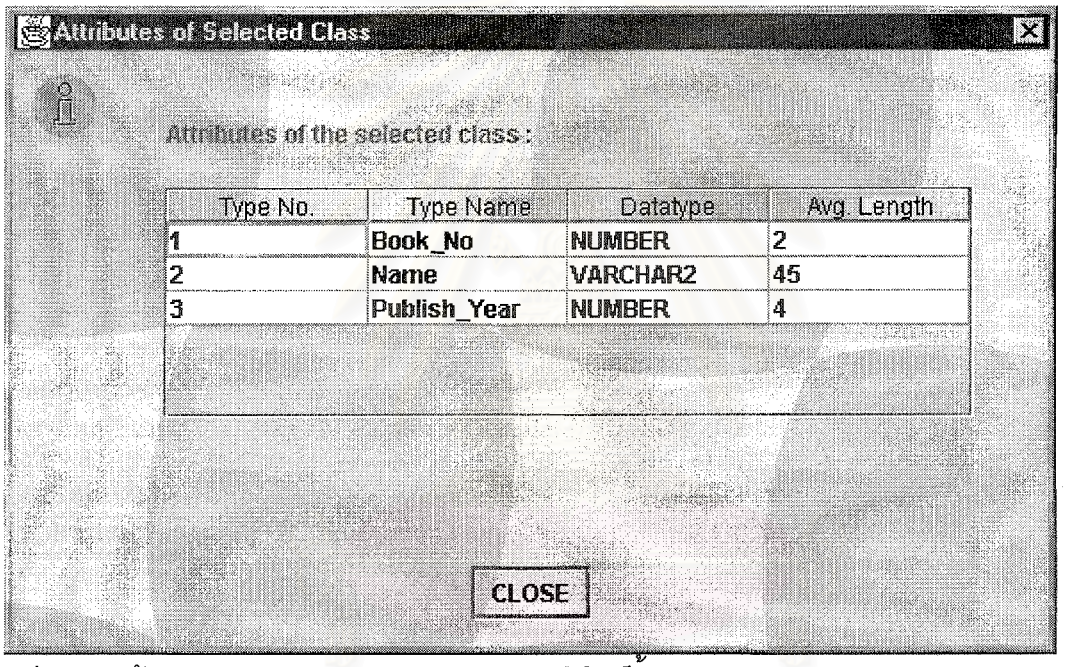
หากผู้ใช้ไม่ต้องการทำปฏิบัติการแทรกแถวข้อมูลแล้ว ให้เลือกเมนูย่อย Close ของเมนู File เพื่อปิดหน้าจอสร้างออบเจคต์

ก.3.3 การปรับปรุงหน้าจอสอบถามหลังทำปฏิบัติการต่าง ๆ

หากผู้ใช้ต้องการดูผลการทำปฏิบัติการต่างๆ ให้เลือกเมนูย่อย Query จากเมนู Database หน้าจอสอบถามของโมดูลจะแสดงผลของปฏิบัติการนั้น

ก.3.4 การเรียกดูแอตทริบิวท์และคุณลักษณะของแอตทริบิวท์ของคลาสที่ผู้ใช้นิยาม

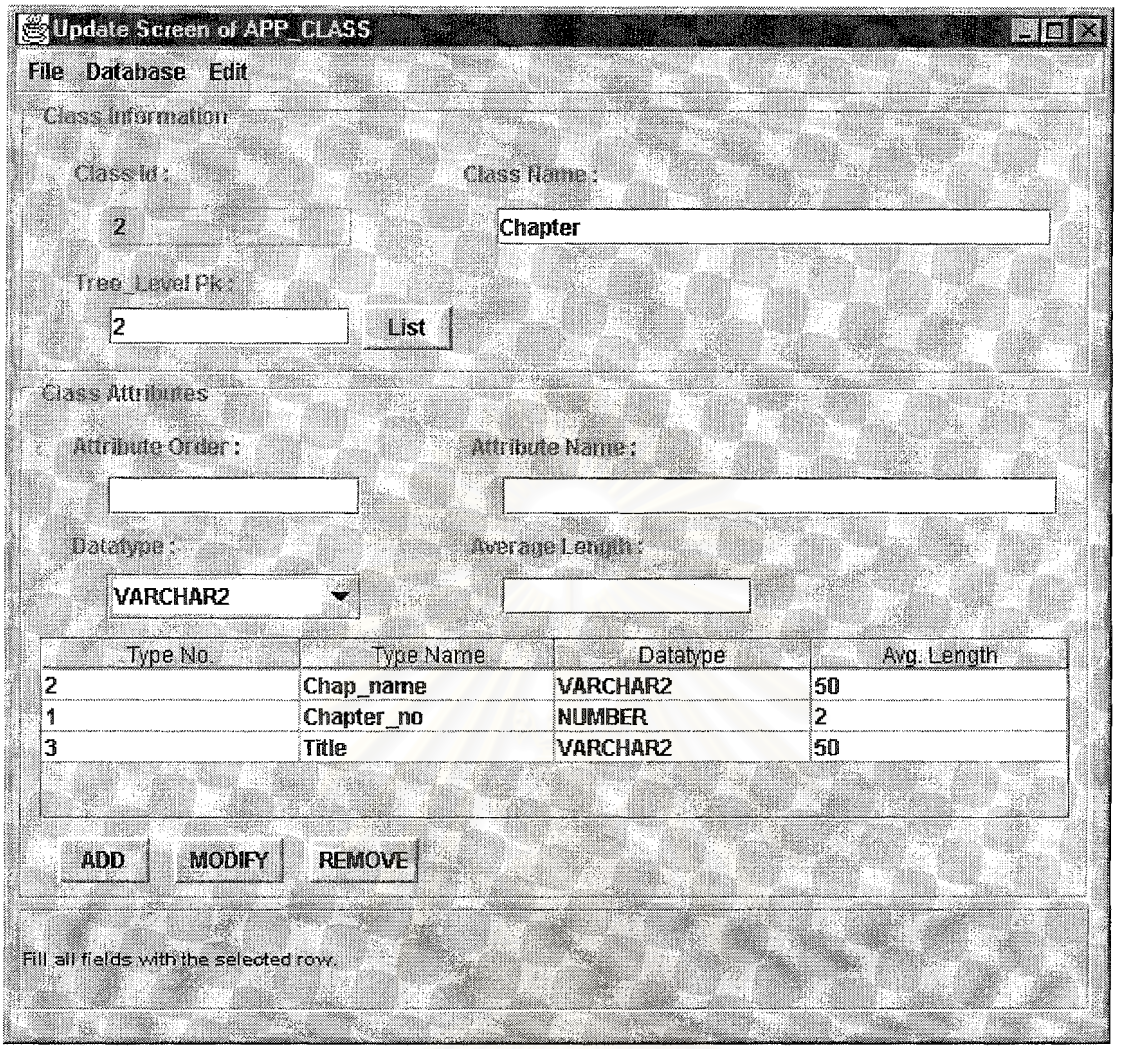
ในหน้าจอสอบถามของโมดูล ผู้ใช้ต้องเลือกคลาสที่ต้องการดูแอตทริบิวท์ก่อน จากนั้นเลือกเมนูย่อย Attributes ในเมนู View ระบบจะแสดงหน้าจอย่อย ดังรูป ก14 ซึ่งเป็นหน้าจอที่แสดงคุณลักษณะของแอตทริบิวท์ทั้งหมดของคลาสที่ผู้ใช้ระบุ



รูป ก14 หน้าจอย่อยแสดงคุณลักษณะของแอตทริบิวท์ทั้งหมดของคลาส Tales

ก.3.5 การแก้ไขค่าข้อมูลของคลาสที่ผู้ใช้นิยาม

เนื่องจาก การทำปฏิบัติการแก้ไขค่าข้อมูลของคลาสที่ผู้ใช้นิยามจะกระทบกับคุณลักษณะของตารางในฐานข้อมูล ดังนั้น ระบบจึงกำหนดให้การทำปฏิบัติการนี้ขึ้นอยู่กับค่าในสดมภ์ DB_FLAG ของตาราง APP_CLASS ถ้าสดมภ์นี้มีค่าเป็น 1 จะหมายความว่าคลาสนั้นถูกนำไปใช้สร้างตารางในฐานข้อมูลแล้ว แต่หากสดมภ์นี้มีค่าเป็น 0 คลาสนั้นยังไม่ถูกนำไปใช้สร้างตารางในฐานข้อมูล ด้วยเหตุนี้ การทำปฏิบัติการแก้ไขค่าข้อมูลของคลาสจึงแบ่งออกได้เป็น 2 กรณี คือ



รูป ก15 หน้าจอแก้ไขออบเจคต์ของโมดูล APPLICATION CLASS: ก่อนการแก้ไข

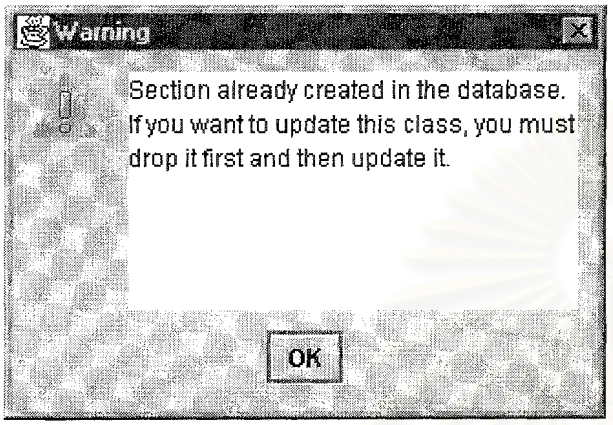
ก.3.5.1 คลาสที่ยังไม่ถูกสร้างเป็นตารางในฐานข้อมูล

ผู้ใช้ต้องเลือกคลาสที่ต้องการแก้ไขค่าข้อมูลจากหน้าจอสอบถามก่อน แล้วจึงเลือกเมนูย่อย Update Screen ของเมนู Database ระบบจะแสดงหน้าจอแก้ไขออบเจคต์ของโมดูล APPLICATION CLASS ดังรูป ก15 เป็นหน้าจอที่แสดงค่าข้อมูลต่างๆ ของคลาสก่อนการแก้ไข

ก.3.5.2 คลาสที่ถูกสร้างเป็นตารางในฐานข้อมูลแล้ว

ถ้าผู้ใช้ต้องการแก้ไขค่าแอตทริบิวท์ของคลาสในตาราง APP_CLASS โดยที่คลาสนั้นได้ถูกสร้างเป็นตารางในฐานข้อมูลแล้ว (คือคลาสที่ค่าสดมภ์ DB_FLAG มีค่าเป็น 1) ระบบจะแสดงหน้าจอย่อย ดังรูป ก16 เพื่อแจ้งให้ทราบว่า ผู้ใช้ไม่สามารถทำปฏิบัติการแก้ไขแอตทริบิวท์ได้ในขณะนี้ แต่ผู้ใช้จะต้องทำปฏิบัติการดรอป (Drop) ตารางในฐานข้อมูลก่อน จึงจะสามารถทำการแก้ไขค่าแอตทริบิวท์ของคลาสได้ เพราะการแก้ไขค่าแอตทริบิวท์ของคลาสในตาราง APP_CLASS เป็นการแก้ไขนิยามต่างๆ ของคลาสที่จะกระทบกับคุณลักษณะของตารางในฐานข้อมูล

หลังจากผู้ใช้ทำการแก้ไขค่าข้อมูลต่างๆ ของคลาสแล้ว หากต้องการทำปฏิบัติการปรับปรุงข้อมูลของคลาสที่ผู้ใช้นิยาม ผู้ใช้ต้องเลือกเมนูย่อย Update ของเมนู Database แต่ถ้าผู้ใช้ไม่ต้องการทำปฏิบัติการปรับปรุงข้อมูลของตาราง APP_CLASS ในฐานข้อมูล ก็สามารถทำได้โดยการปิดหน้าจอแก้ไขนี้



รูป ก16 หน้าจอย่อยแสดงข้อความเตือน ในกรณีที่ผู้ใช้ต้องการแก้ไขแอตทริบิวท์ของคลาสซึ่งถูกสร้างเป็นตารางในฐานข้อมูลแล้ว

ก.3.6 การลบคลาสในตาราง APP_CLASS

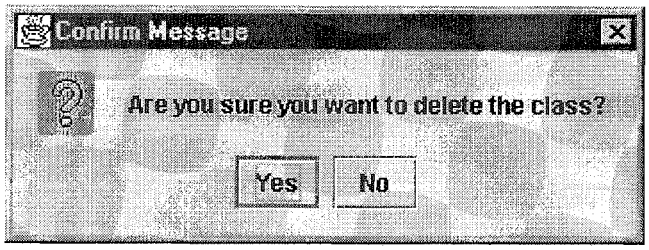
เช่นเดียวกับปฏิบัติการแก้ไขค่าข้อมูลของคลาสที่ผู้ใช้นิยาม การลบคลาสในตาราง APP_CLASS จะกระทบกับตารางในฐานข้อมูลที่ถูกสร้างขึ้นจากคุณลักษณะของคลาสนั้น ดังนั้นการลบคลาสออกจากตาราง APP_CLASS จึงแบ่งออกได้เป็น 2 กรณี ต่อไปนี้

ก.3.6.1 คลาสที่ยังไม่ถูกสร้างเป็นตารางในฐานข้อมูล

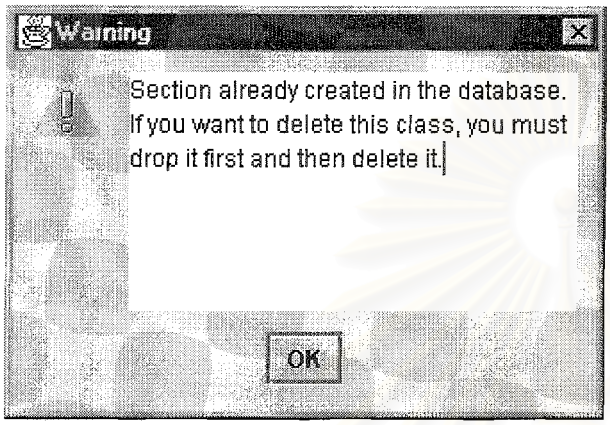
ในหน้าจอสอบถาม ผู้ใช้ต้องเลือกคลาสที่ต้องการลบก่อน จากนั้นเลือกเมนูย่อย Delete ของเมนู Database ระบบจะแสดงหน้าจอย่อย ดังรูป ก17 เพื่อให้ผู้ใช้ยืนยันการทำปฏิบัติการลบคลาส หากผู้ใช้กดปุ่ม Yes ระบบจะทำการลบคลาสนั้นออกจากตาราง APP_CLASS ในฐานข้อมูล แต่ถ้าผู้ใช้กดปุ่ม No ระบบจะไม่ทำปฏิบัติการนี้

ก.3.6.2 คลาสที่ถูกสร้างเป็นตารางในฐานข้อมูลแล้ว

การลบคลาสในตาราง APP_CLASS โดยที่คลาสนั้นได้ถูกสร้างเป็นตารางในฐานข้อมูล (คือคลาสที่ค่าสดมภ์ DB_FLAG มีค่าเป็น 1) ระบบจะแสดงหน้าจอย่อย ดังรูป ก18 เพื่อแจ้งให้ทราบว่า ผู้ใช้ไม่สามารถทำปฏิบัติการลบคลาสได้ในขณะนี้ แต่ผู้ใช้จะต้องทำปฏิบัติการดรอปตารางในฐานข้อมูลก่อน จึงจะสามารถทำการลบคลาสได้



รูป ก17 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบคลาสออกจากตาราง APP_CLASS



รูป ก18 หน้าจอย่อยแสดงข้อความเตือน ในกรณีที่ผู้ใช้ต้องการลบคลาสซึ่งถูกสร้างเป็นตารางในฐานข้อมูลแล้ว

ก.3.7 การทำปฏิบัติการสร้างตารางในฐานข้อมูล

ดังกล่าวแล้วว่า แต่ละแถวข้อมูลของตาราง APP_CLASS ในฐานข้อมูล เก็บนิยามของคลาสที่ผู้ใช้กำหนดเพื่อให้ระบบงานวิจัยนำไปใช้สร้างตารางในฐานข้อมูล ตารางที่ถูกสร้างขึ้นในลักษณะนี้เป็นตารางที่ใช้เก็บออบเจคต์ในแต่ละระดับชั้นของโครงสร้างต้นไม้ตามนิยามของคลาส

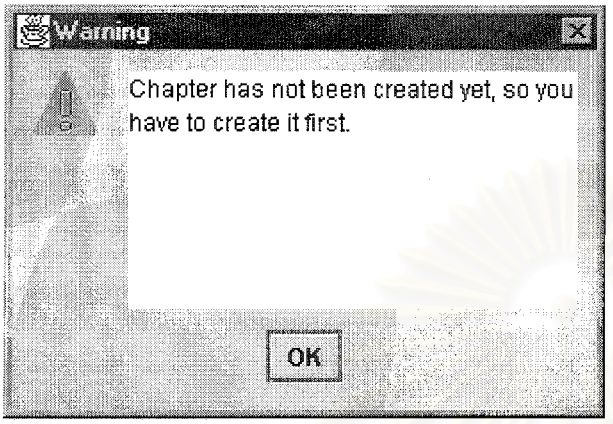
การทำปฏิบัติการแทรกแถวข้อมูลในตาราง APP_CLASS เป็นเพียงการเก็บนิยามของคลาสเท่านั้น ระบบจะยังไม่ทำการสร้างตารางของคลาสนั้นในฐานข้อมูล จนกระทั่งผู้ใช้เลือกเมนูย่อย Create Class ของเมนู Database ในหน้าจอสอบถาม ระบบจึงจะทำการสร้างตารางขึ้นจากนิยามของคลาส โดยสดมภ์ของตารางจะถูกสร้างจากคุณลักษณะต่างๆ ของแอตทริบิวท์ในคลาสที่ผู้ใช้นิยาม นั่นเอง

ระบบจะแสดงข้อความแจ้งให้ผู้ใช้ทราบว่าการทำปฏิบัติการสร้างตารางในฐานข้อมูลสำเร็จหรือไม่ หากสำเร็จระบบงานวิจัยจะทำการปรับปรุงค่าในสดมภ์ DB_FLAG ให้เป็น 1 เพื่อดูผลของการปรับปรุงค่านี้ ผู้ใช้ต้องเลือกเมนูย่อย Query จากเมนู Database ดังในหัวข้อ ก.3.3 การปรับปรุงหน้าจอหลังทำปฏิบัติการต่างๆ

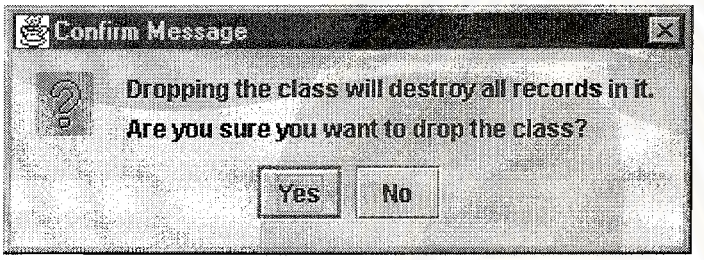
ก.3.8 การดรอปตารางในระบบฐานข้อมูล

ผู้ใช้ต้องเลือกคลาสในตาราง APP_CLASS ที่ต้องการดรอปก่อน จากนั้นเลือกเมนูย่อย Drop Class ของเมนู Database ในหน้าจอสอบถาม ระบบจะทำการตรวจสอบว่า คลาสนั้นได้ถูก

สร้างเป็นตารางในฐานข้อมูลหรือยังโดยดูจากค่าข้อมูลในสดมภ์ DB_FLAG หากมีค่าเป็น 0 ระบบจะแสดงหน้าจอย่อย ดังรูป ก19 เพื่อแจ้งให้ทราบว่า ผู้ใช้ไม่สามารถทำปฏิบัติการดรอปตารางในฐานข้อมูลได้ เนื่องจาก ตารางนั้นยังไม่ได้ถูกสร้างขึ้นในฐานข้อมูล



รูป ก19 หน้าจอย่อยแสดงข้อความเตือน ในกรณีที่ผู้ใช้ต้องการดรอปตารางซึ่งยังไม่ถูกสร้างขึ้นในฐานข้อมูล



รูป ก20 หน้าจอย่อยแสดงข้อความขอคำยืนยันการดรอปตาราง

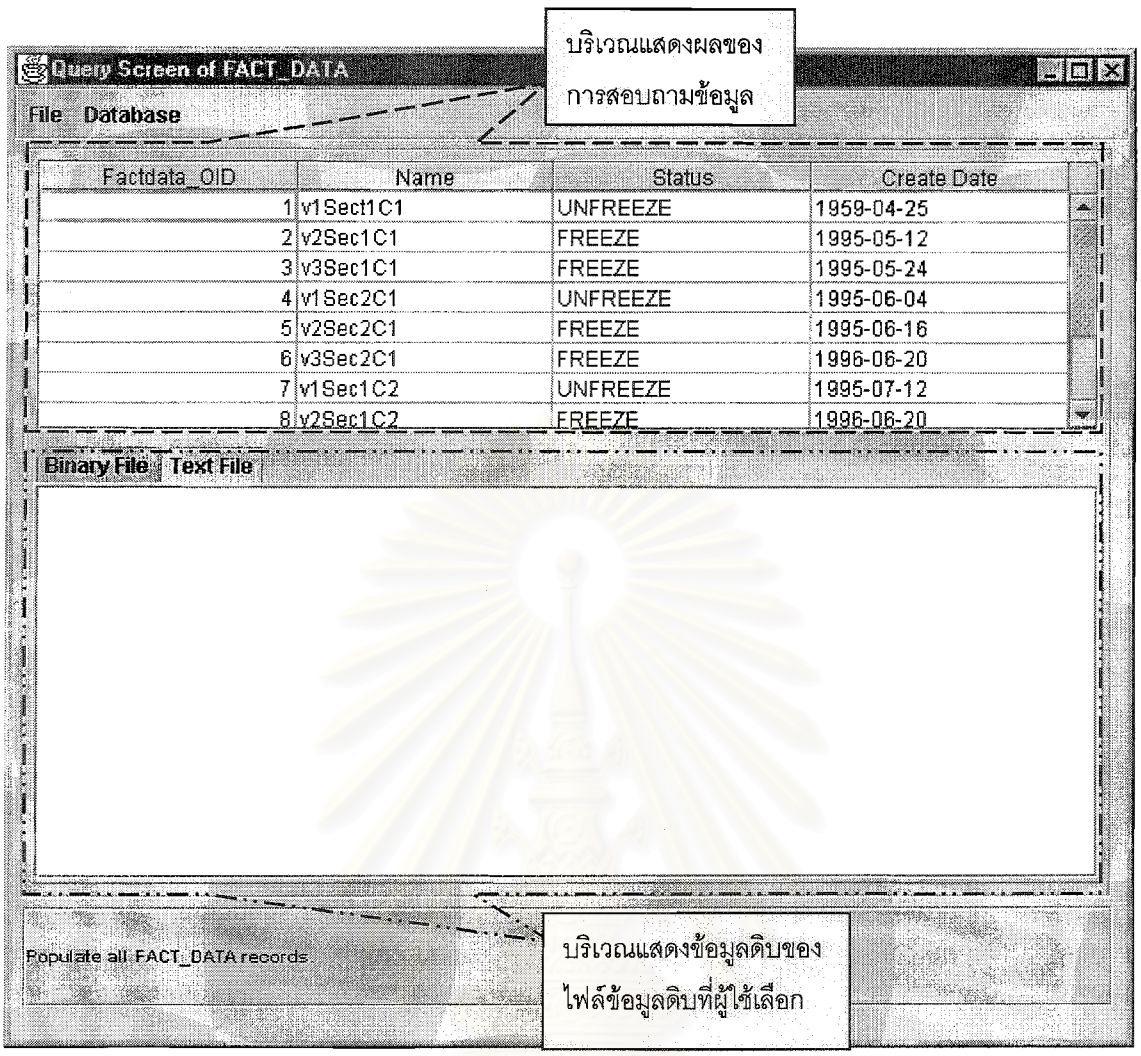
แต่ถ้าค่าในสดมภ์ DB_FLAG มีค่าเป็น 1 ระบบจะหน้าจอย่อย รูป ก20 ซึ่งเป็นหน้าจอที่แสดงข้อความขอคำยืนยันการดรอปตารางออกจากฐานข้อมูล หากผู้ใช้กดปุ่ม Yes ระบบจะทำปฏิบัติการดรอปตารางนั้น พร้อมกับแก้ไขค่าในสดมภ์ DB_FLAG ให้มีค่าเป็น 0 การกดปุ่ม No ระบบจะไม่ทำปฏิบัติการใดเลย

ก.4 ปฏิบัติการต่าง ๆ ของโมดูล FACT DATA

ก.4.1 การเริ่มต้นใช้งานโมดูลและสอบถามข้อมูล

เมื่อผู้ใช้กดปุ่ม FACT DATA ระบบจะแสดงหน้าจอสอบถามของโมดูล FACT DATA รูป ก21 เป็นหน้าจอเริ่มต้นในการใช้งานโมดูลนี้

ผู้ใช้สามารถกำหนดเงื่อนไขการสอบถามข้อมูลของตาราง FACTDATA ได้โดยเลือกชื่อโครงงานจากส่วนประกอบหมายเลข 1 บริเวณแสดงผลการสอบถามข้อมูลจะแสดงข้อมูลของคลาสในโครงงานที่ผู้ใช้กำหนด แต่หากผู้ใช้ไม่กำหนดเงื่อนไขใดเลย ระบบจะแสดงข้อมูลของทุกคลาสในระบบ



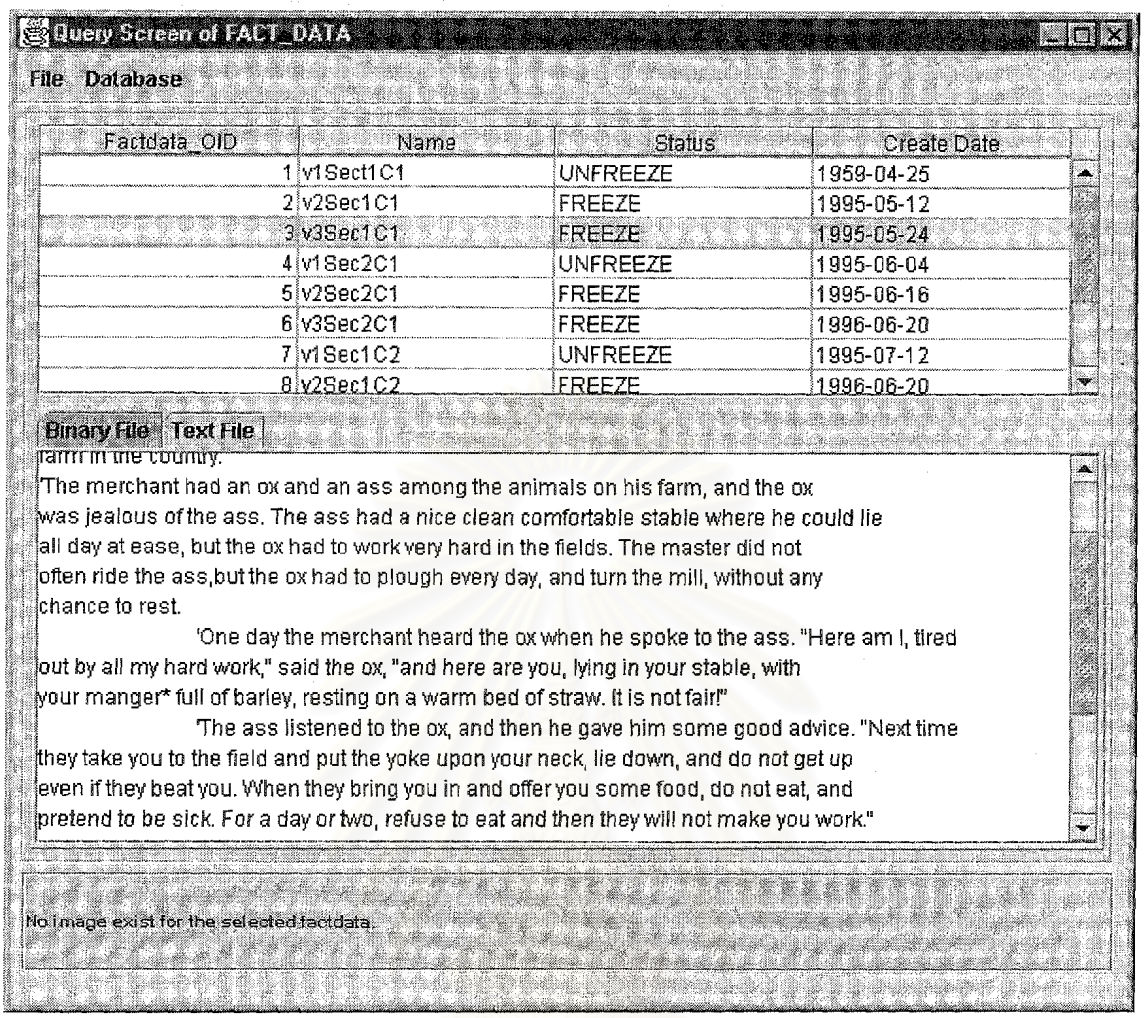
รูป ก21 หน้าจอสอบถามของโมดูล FACT DATA

ก.4.2 การดูข้อมูลดิบในไฟล์ข้อมูลดิบของออบเจคต์ Fact Data

ผู้ใช้ต้องเลือกออบเจคต์ข้อมูลดิบที่ต้องการจากตารางในหน้าจอสอบถาม ระบบจะแสดงข้อมูลดิบที่เกี่ยวข้องกับออบเจคต์ข้อมูลดิบนั้นในบริเวณที่ 2 ดังรูป ก22

ก.4.3 การแทรกแถวข้อมูลหรือการสร้างออบเจคต์ข้อมูลดิบ

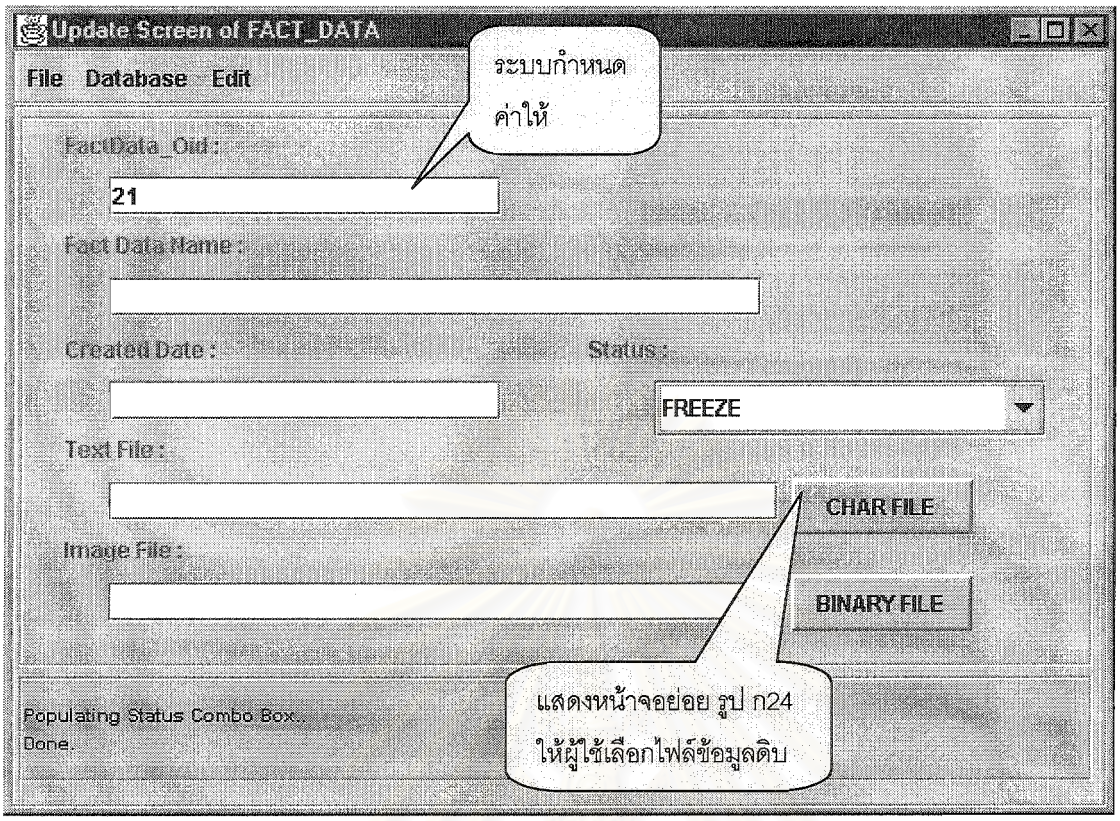
การทำปฏิบัติการแทรกแถวข้อมูลในตาราง FACTDATA ผู้ใช้ต้องเลือกเมนูย่อย Insert Screen จากเมนู Database ในหน้าจอสอบถาม ระบบจะแสดงหน้าจอสร้างออบเจคต์ของโมดูล FACT DATA ดังรูป ก23 เพื่อให้ผู้ใช้ใส่ค่าข้อมูลให้กับแต่ละแอตทริบิวท์ของออบเจคต์ข้อมูลดิบ ผู้ใช้สามารถเลือกไฟล์ข้อมูลดิบให้กับออบเจคต์ข้อมูลดิบได้โดยกดปุ่ม CHAR FILE สำหรับไฟล์ข้อมูลดิบชนิดตัวอักษร และปุ่ม BINARY FILE สำหรับไฟล์ข้อมูลดิบชนิดรูปภาพ ซึ่งระบบจะแสดงหน้าจอย่อย รูป ก24 ข้อจำกัดประการหนึ่งเกี่ยวกับการเลือกไฟล์ข้อมูลดิบก็คือ ไฟล์ข้อมูลดิบทั้งหมดที่ผู้ใช้ต้องการใช้ในระบบงานวิจัยจะต้องเก็บอยู่ภายใต้ไดเรกทอรี E:/Thesis/implement/database/files/



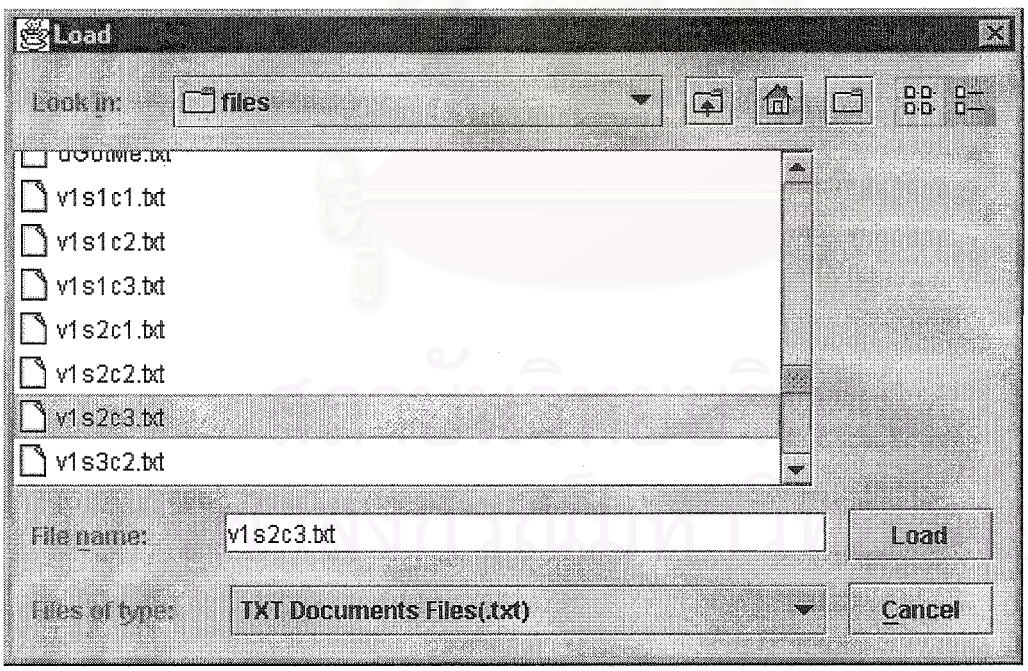
รูป ก22 แสดงข้อมูลดิบของออบเจคต์ข้อมูลดิบ

จากนั้นหากผู้ใช้ต้องการทำปฏิบัติการแทรกแถวข้อมูลในตาราง FACTDATA ให้เลือกเมนูย่อย Insert ของเมนู Database เมื่อปฏิบัติการแทรกแถวข้อมูลสำเร็จแล้ว ระบบจะเคลียร์ค่าในฟิลด์ของหน้าจอ หากผู้ใช้ต้องการทำปฏิบัติการแทรกแถวข้อมูลแถวต่อไปก็สามารถทำได้โดยป้อนค่าข้อมูลในทุกฟิลด์ของหน้าจอ รวมทั้งฟิลด์ FactData_Oid ด้วย ผู้ใช้ควรระวังในการกำหนดค่าให้แก่ฟิลด์ FactData_Oid โดยพยายามกำหนดค่าที่ไม่ซ้ำกับค่าในสดมภ์ Factdata_oid ของตาราง FACTDATA แต่ถ้าผู้ใช้ต้องการให้ระบบกำหนดค่าในฟิลด์ FactData_Oid ให้ ผู้ใช้ต้องปิดหน้าจอสร้างออบเจคต์นี้ก่อน แล้วจึงค่อยเปิดหน้าจอสร้างออบเจคต์ขึ้นใหม่

หากผู้ใช้เปลี่ยนใจไม่ทำปฏิบัติการแทรกแถวข้อมูล ก็สามารถทำได้โดยเลือกเมนูย่อย Close จากเมนู File เพื่อปิดหน้าจอสร้างออบเจคต์ หรือเลือกเมนูย่อย Edit ของเมนู Clear เพื่อเคลียร์ค่าในฟิลด์ทั้งหมดของหน้าจอ



รูป ก23 หน้าจอสร้างออบเจคต์ของโมดูล FACT DATA



รูป ก24 หน้าจอย่อยสำหรับเลือกไฟล์ข้อมูลดิบให้กับออบเจคต์ข้อมูลดิบ

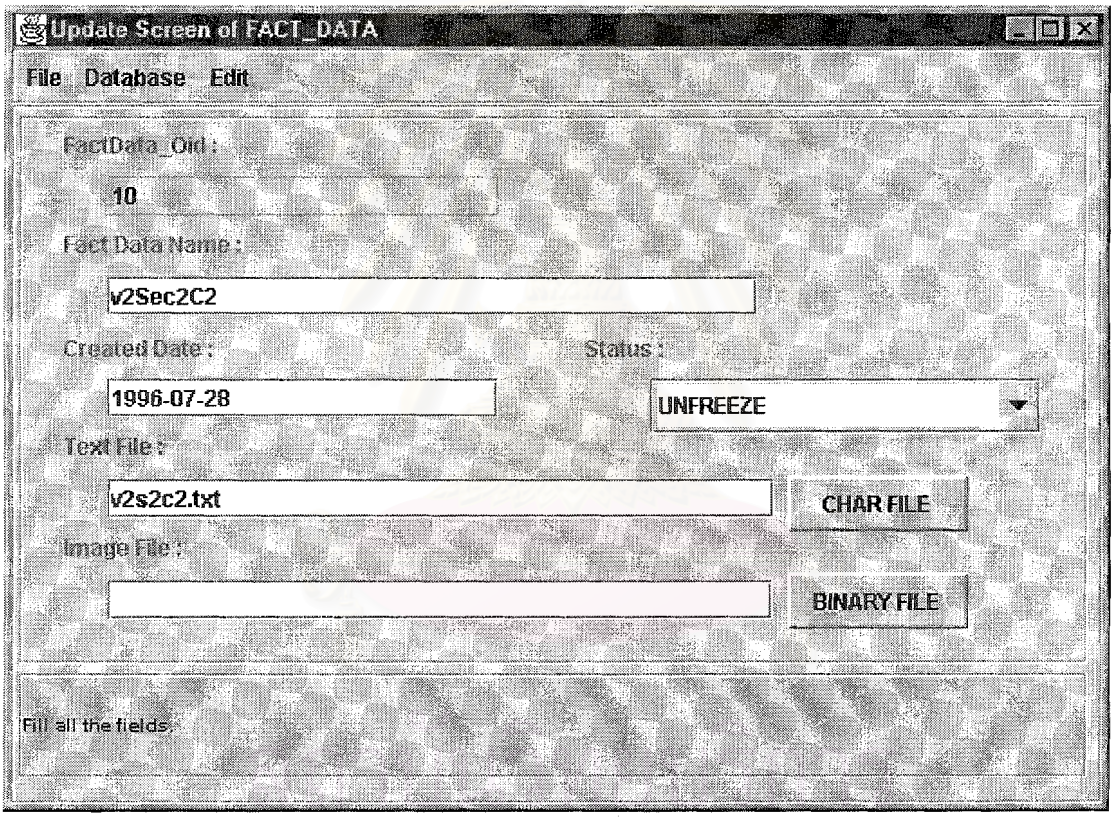
ก.4.4 การปรับปรุงหน้าจอสอบถามหลังทำปฏิบัติการต่าง ๆ

หากผู้ใช้ต้องการดูผลการทำปฏิบัติการต่างๆ ให้เลือกเมนูย่อย Query จากเมนู Database หน้าจอสอบถามของโมดูลจะแสดงผลของปฏิบัติการนั้น

ก.4.5 การแก้ไขค่าแอตทริบิวท์ของออบเจคต์ข้อมูลดิบ

ผู้ใช้ต้องเลือกออบเจคต์ข้อมูลดิบที่ต้องการแก้ไขค่าแอตทริบิวท์จากหน้าจอสอบถาม และเลือกเมนูย่อย Update Screen ของเมนู Database ระบบจะแสดงหน้าจอแก้ไขออบเจคต์ โดยทุกฟิลด์ในหน้าจอนี้จะมีค่าแอตทริบิวท์ของออบเจคต์ข้อมูลดิบก่อนการแก้ไข ดังรูป ก25

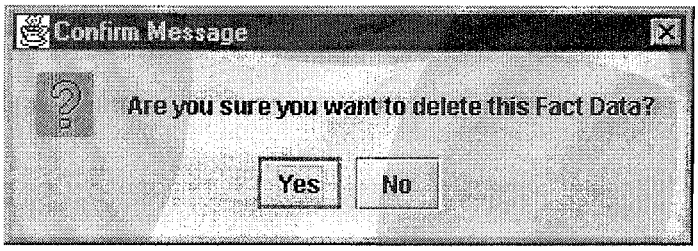
หลังจากผู้ใช้ได้ทำการแก้ไขค่าข้อมูลในฟิลด์แล้วและต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตาราง FACTDATA ในฐานข้อมูล ผู้ใช้ต้องเลือกเมนูย่อย Update จากเมนู Database แต่หากผู้ใช้ไม่ต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตาราง FACTDATA ก็สามารถทำได้โดยปิดหน้าจอแก้ไขนี้



รูป ก25 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล FACT DATA: ก่อนการแก้ไข

ก.4.6 การลบออบเจคต์ข้อมูลดิบ

ในหน้าจอสอบถาม ผู้ใช้ต้องเลือกออบเจคต์ข้อมูลดิบที่ต้องการลบก่อน จากนั้นเลือกเมนูย่อย Delete ของเมนู Database ระบบจะแสดงหน้าจอย่อย ดังรูป ก26 เพื่อให้ผู้ใช้ยืนยันการทำปฏิบัติการลบออบเจคต์ข้อมูลดิบ หากผู้ใช้กดปุ่ม Yes ระบบจะทำการลบออบเจคต์ข้อมูลดิบนั้นออกจากตาราง FACTDATA ในฐานข้อมูล แต่ถ้าผู้ใช้กดปุ่ม No ระบบจะไม่ทำปฏิบัติการนี้



รูป ก26 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบออบเจคต์ข้อมูลดิบจากตาราง FACTDATA

ก.5 ปฏิบัติการต่าง ๆ ของโมดูล USER CLASS

ก.5.1 การเริ่มต้นใช้งานโมดูลและสอบถามข้อมูล

เมื่อผู้ใช้กดปุ่ม USER CLASSES ระบบจะแสดงหน้าจอสอบถามของโมดูล USER CLASS รูป ก27 เป็นหน้าจอเริ่มต้นในการใช้งานโมดูลนี้

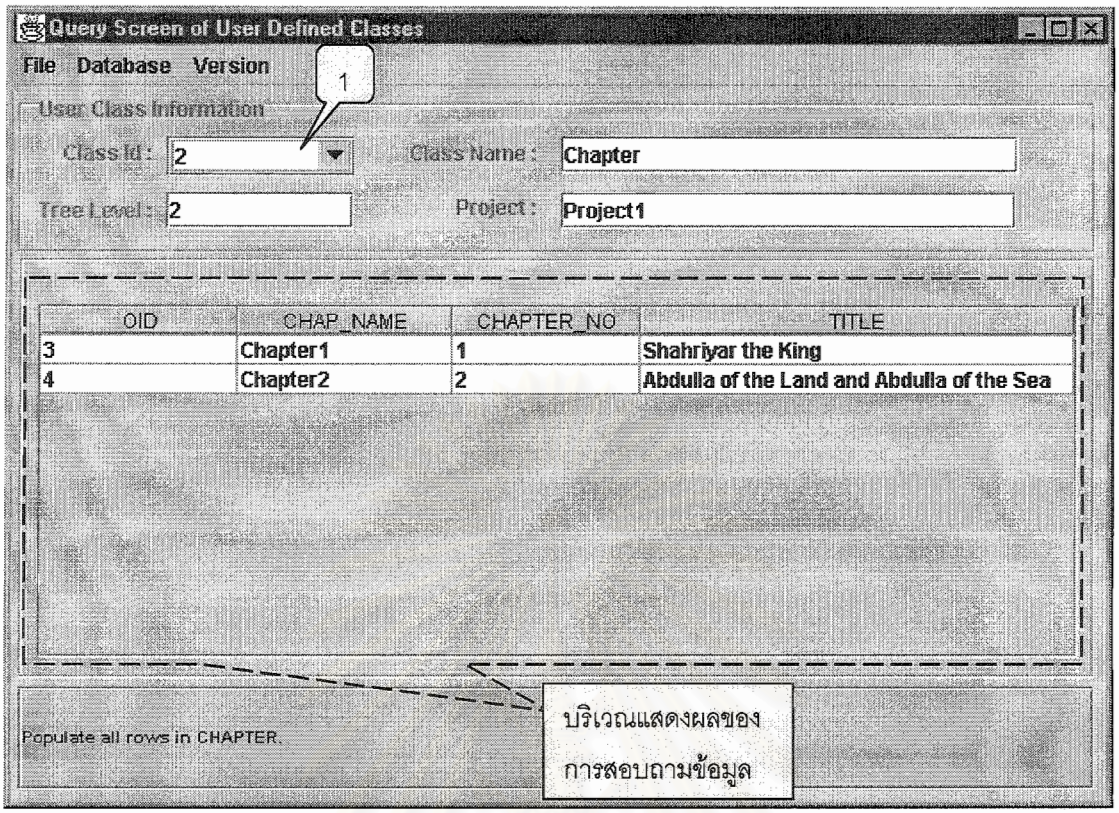
ผู้ใช้สามารถกำหนดเงื่อนไขการสอบถามข้อมูลของตารางที่ผู้ใช้นิยามได้โดยเลือกค่าของตัวระบุคลาสจากส่วนประกอบหมายเลข 1 บริเวณแสดงผลการสอบถามข้อมูลจะแสดงข้อมูลของออบเจคต์ในคลาสที่ผู้ใช้เลือก

ก.5.2 การแทรกแถวข้อมูลหรือการสร้างออบเจคต์ของคลาสที่ผู้ใช้นิยาม

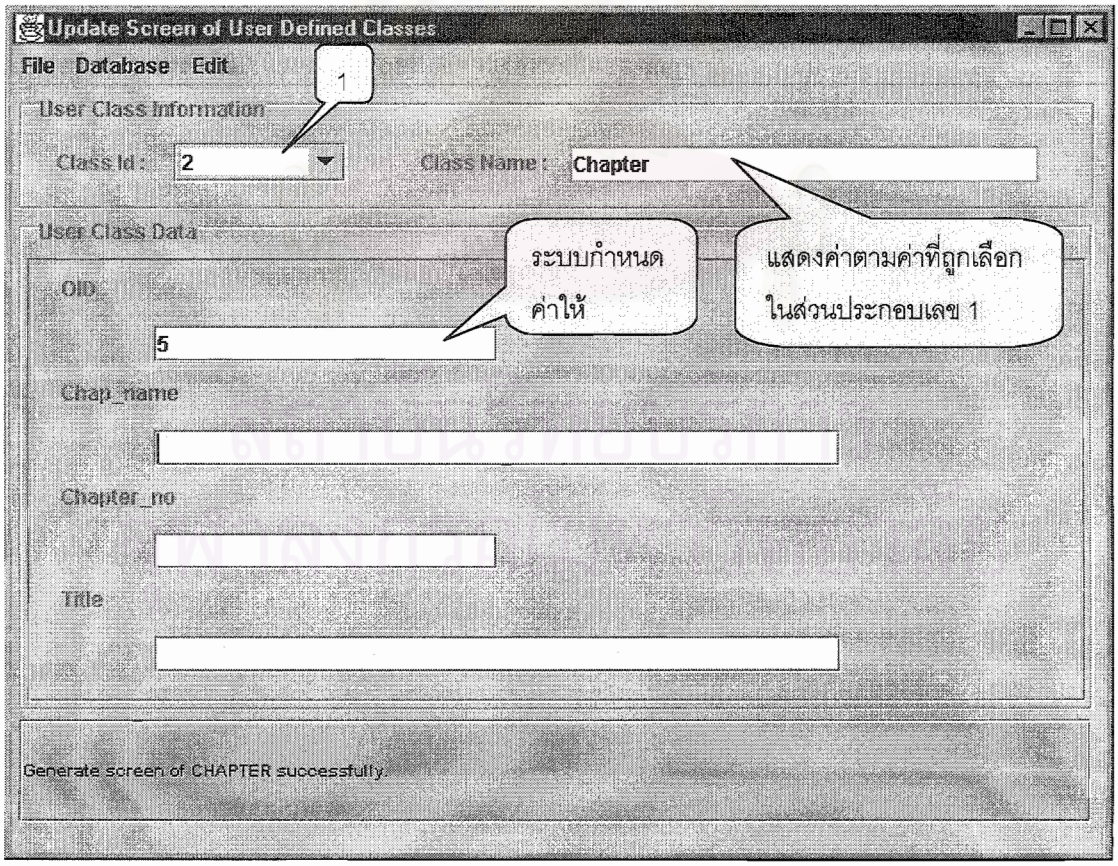
การสร้างออบเจคต์ของคลาสที่ผู้ใช้นิยาม ผู้ใช้ต้องเลือกค่าของตัวระบุคลาสจากส่วนประกอบหมายเลข 1 ก่อน จากนั้นเลือกเมนู Database และเมนูย่อย Insert Screen ระบบจะแสดงหน้าจอสร้างออบเจคต์ของโมดูล USER CLASS ดังรูป ก28

หลังจากผู้ใช้ป้อนค่าข้อมูลในฟิลด์ต่างๆ ของคลาสเรียบร้อยแล้ว ให้ผู้ใช้เลือกเมนูย่อย Insert ของเมนู Database เพื่อแทรกแถวข้อมูลเข้าตารางในฐานข้อมูลที่ถูกสร้างจากคุณลักษณะของคลาสนั้น เมื่อปฏิบัติการแทรกแถวข้อมูลสำเร็จแล้วระบบจะทำการเคลียร์ค่าในฟิลด์ของหน้าจอ ถ้าผู้ใช้ต้องการทำปฏิบัติการสร้างออบเจคต์ต่อไป ก็สามารถทำได้โดยป้อนค่าข้อมูลในทุกฟิลด์ของคลาส รวมทั้งฟิลด์ Oid ด้วย ผู้ใช้ควรระวังในการกำหนดค่าให้แก่ฟิลด์ Oid โดยพยายามกำหนดค่าที่ไม่ซ้ำกับค่าในสดมภ์ Oid ของตาราง OBJ_INSTANCE แต่ถ้าผู้ใช้ต้องการให้ระบบกำหนดค่าในฟิลด์ Oid ให้ ผู้ใช้ต้องปิดหน้าจอสร้างออบเจคต์นี้ก่อน แล้วจึงค่อยเปิดหน้าจอสร้างออบเจคต์ขึ้นใหม่

หากผู้ใช้เปลี่ยนใจไม่ทำปฏิบัติการสร้างออบเจคต์ของคลาสที่ผู้ใช้นิยาม ก็สามารถทำได้โดยเลือกเมนูย่อย Close จากเมนู File เพื่อปิดหน้าจอสร้างออบเจคต์ หรือเลือกเมนูย่อย Clear ของเมนู Edit เพื่อเคลียร์ค่าในฟิลด์ของหน้าจอ



รูป ก27 หน้าจอสอบถามของโมดูล USER CLASS



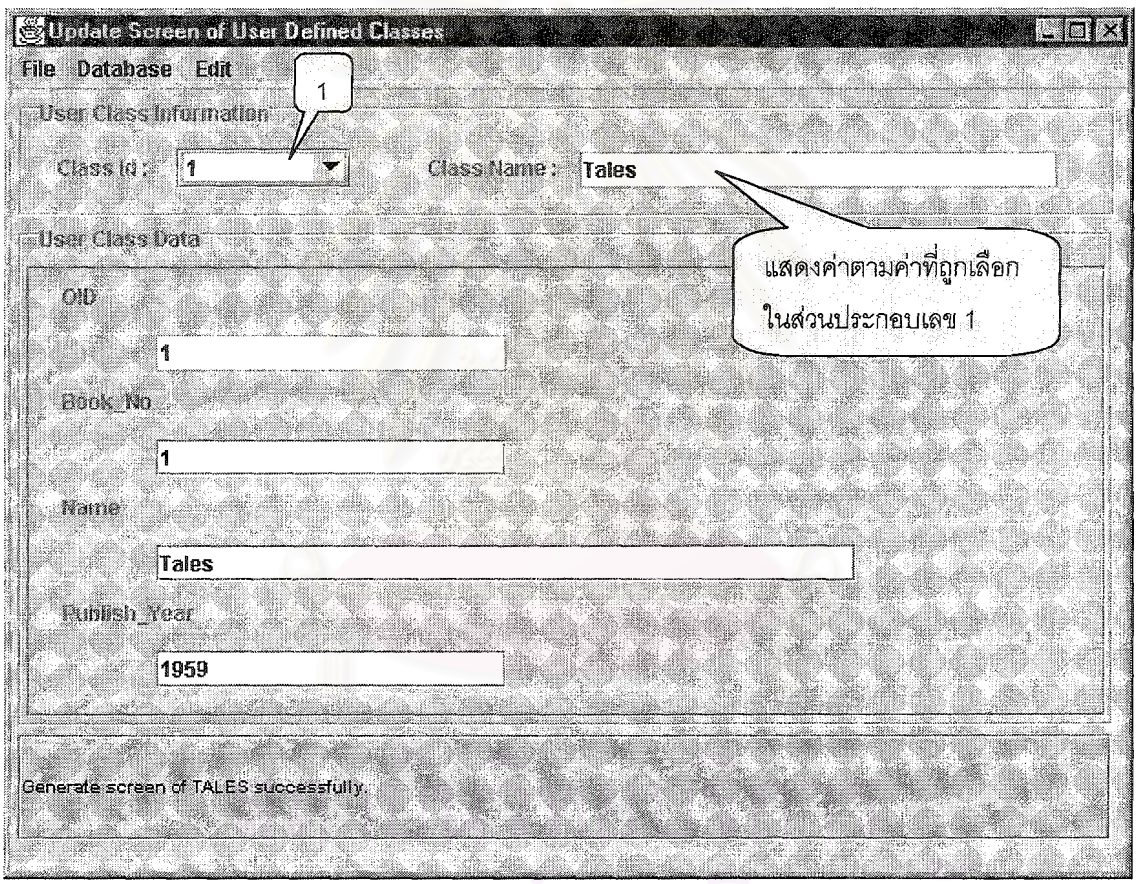
รูป ก28 หน้าจอสร้างออบเจคต์ของโมดูล USER CLASS

ก.5.3 การปรับปรุงหน้าจอสอบถามหลังทำปฏิบัติการต่าง ๆ

หากผู้ใช้ต้องการดูผลการทำปฏิบัติการต่างๆ ให้เลือกเมนูย่อย Query จากเมนู Database หน้าจอสอบถามของโมดูลจะแสดงผลของปฏิบัติการนั้น

ก.5.4 การแก้ไขค่าแอตทริบิวท์ของออบเจคต์ในคลาสที่ผู้ใช้นิยาม

ผู้ใช้ต้องเลือกคลาสและออบเจคต์ของคลาสที่ต้องการแก้ไขค่าแอตทริบิวท์จากหน้าจอสอบถาม จากนั้น เลือกเมนูย่อย Update Screen ของเมนู Database ระบบจะแสดงหน้าจอแก้ไขออบเจคต์ โดยทุกฟิลด์ในหน้าจอนี้จะมีค่าแอตทริบิวท์ของออบเจคต์ก่อนการแก้ไข ดังรูป ก29

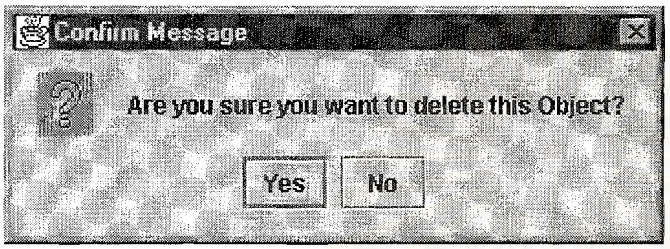


รูป ก29 หน้าจอแก้ไขแอตทริบิวท์ของออบเจคต์ในคลาส Tales: ก่อนการแก้ไข

หลังจากผู้ใช้ได้ทำการแก้ไขค่าข้อมูลในฟิลด์แล้วและต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตารางในฐานข้อมูลที่ถูกสร้างจากคุณลักษณะของคลาสนั้น ผู้ใช้ต้องเลือกเมนูย่อย Update จากเมนู Database แต่หากผู้ใช้ไม่ต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตารางในฐานข้อมูล ก็สามารถทำได้โดยปิดหน้าจอแก้ไขนี้

ก.5.5 การลบออบเจคต์ของคลาสที่ผู้ใช้นิยาม

ในหน้าจอสอบถาม ผู้ใช้ต้องเลือกคลาสและออบเจคต์ของคลาสที่ต้องการลบก่อน จากนั้นเลือกเมนูย่อย Delete ของเมนู Database ระบบจะแสดงหน้าจอย่อย ดังรูป ก30 เพื่อให้ผู้ใช้ยืนยันการทำปฏิบัติการลบออบเจคต์ หากผู้ใช้กดปุ่ม Yes ระบบจะทำการลบแถวข้อมูลนั้นออกจากตารางที่ผู้ใช้นิยามในฐานข้อมูล แต่ถ้าผู้ใช้กดปุ่ม No ระบบจะไม่ทำปฏิบัติการนี้



รูป ก30 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบออบเจคต์ของคลาสที่ผู้ใช้นิยาม

ก.6 ปฏิบัติการต่าง ๆ ของโมดูล OBJECT VERSIONS

ก.6.1 การเริ่มต้นใช้งานโมดูลและสอบถามข้อมูล

เมื่อผู้ใช้กดปุ่ม OBJECT VERSIONS ระบบจะแสดงหน้าจอสอบถามของโมดูล OBJECT VERSIONS รูป ก31 เป็นหน้าจอเริ่มต้นในการใช้งานโมดูลนี้

ผู้ใช้สามารถกำหนดเงื่อนไขการสอบถามข้อมูลของตาราง OBJ_VERSION ได้โดยเลือกค่าตัวระบุออบเจคต์จากส่วนประกอบหมายเลข 1 บริเวณแสดงผลการสอบถามข้อมูลจะแสดงข้อมูลเกี่ยวกับเวอร์ชันทั้งหมดของออบเจคต์ที่ผู้ใช้เลือก

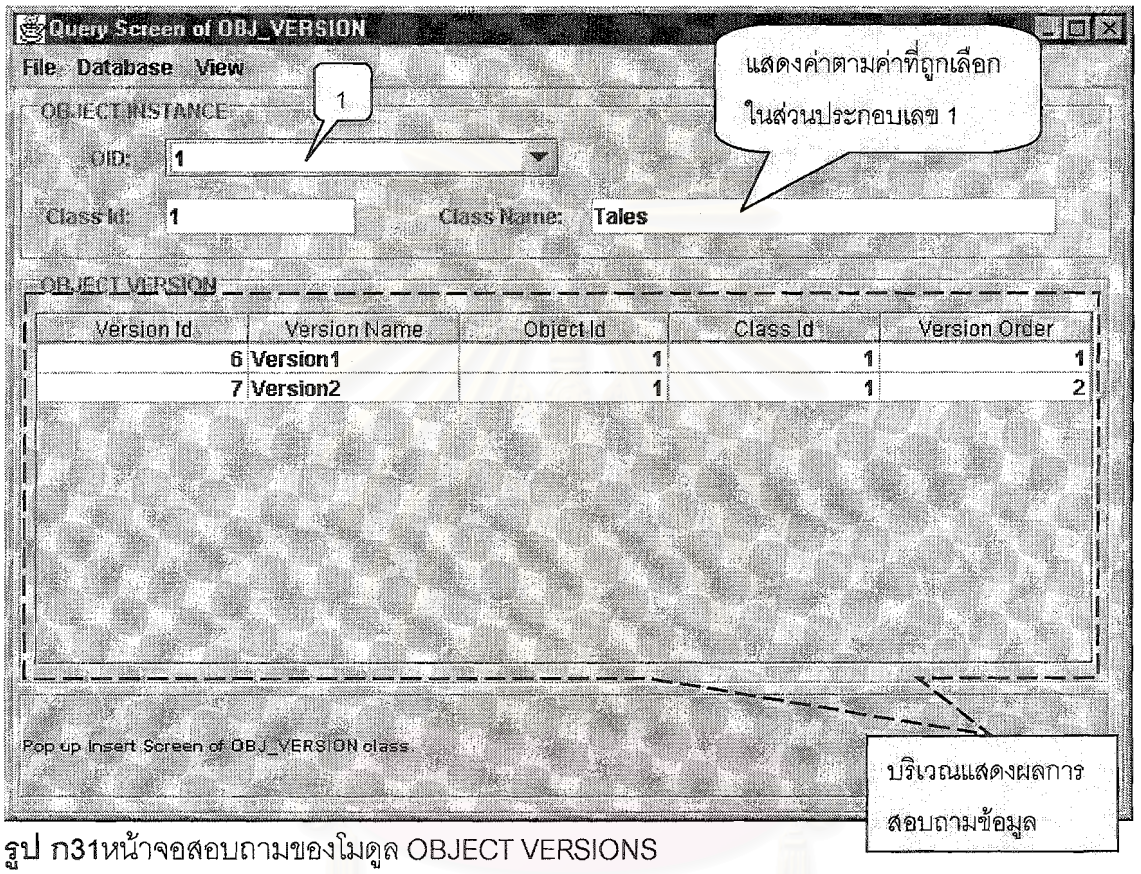
ก.6.2 การแทรกแถวข้อมูลหรือการสร้างเวอร์ชันของออบเจคต์

การสร้างเวอร์ชันของออบเจคต์ หรือก็คือการแทรกแถวข้อมูลเข้าตาราง OBJ_VERSION ในฐานข้อมูลนั่นเอง ผู้ใช้ต้องเลือกค่าของตัวระบุคลาสจากส่วนประกอบหมายเลข 1 ก่อน จากนั้นเลือกเมนู Database และเมนูย่อย Insert Screen ระบบจะแสดงหน้าจอสร้างออบเจคต์ของโมดูล OBJECT VERSIONS ดังรูป ก32

หลังจากผู้ใช้ป้อนค่าข้อมูลในฟิลด์ต่างๆ ของเวอร์ชันแล้ว ให้ผู้ใช้เลือกเมนูย่อย Insert ของเมนู Database เพื่อแทรกแถวข้อมูลเข้าตาราง OBJ_VERSION ในฐานข้อมูล เมื่อปฏิบัติการแทรกแถวข้อมูลสำเร็จแล้ว ระบบจะทำการเคลียร์ค่าในฟิลด์ของหน้าจอ ถ้าผู้ใช้ต้องการทำปฏิบัติการสร้างเวอร์ชันของออบเจคต์ต่อไป ก็สามารถทำได้โดยป้อนค่าข้อมูลในทุกฟิลด์ของเวอร์ชัน รวมทั้งฟิลด์ Vid ด้วย ผู้ใช้ควรระวังในการกำหนดค่าให้แก่ฟิลด์ Vid โดยพยายามกำหนดค่าที่ไม่ซ้ำกับค่าใน

สดมภ์ Vid ของตาราง OBJ_VERSION แต่ถ้าผู้ใช้ต้องการให้ระบบกำหนดค่าในฟิลด์ Vid ให้ ผู้ใช้ต้องปิดหน้าจอนี้ก่อน แล้วจึงค่อยเปิดหน้าจอสร้างออบเจคต์ขึ้นใหม่

หากผู้ใช้เปลี่ยนใจไม่ทำปฏิบัติการสร้างเวอร์ชันของออบเจคต์ ก็สามารถทำได้โดยเลือกเมนูย่อย Close จากเมนู File เพื่อปิดหน้าจอสร้างออบเจคต์ หรือเลือกเมนูย่อย Clear ของเมนู Edit เพื่อเคลียร์ค่าในฟิลด์ของหน้าจอ



รูป ก31หน้าจอสอบถามของโมดูล OBJECT VERSIONS

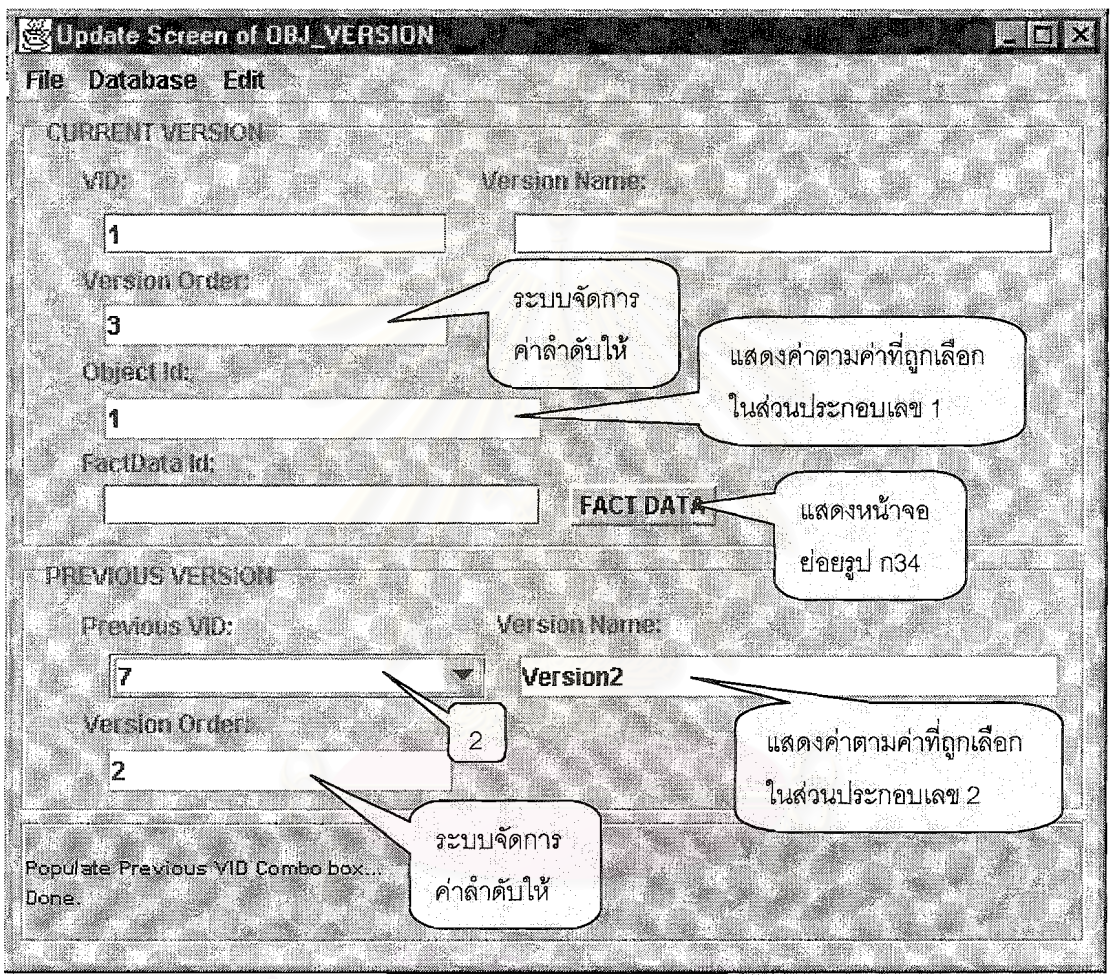
ก.6.3 การปรับปรุงหน้าจอสอบถามหลังทำปฏิบัติการต่าง ๆ

หากผู้ใช้ต้องการดูผลการทำปฏิบัติการต่างๆ ให้เลือกเมนูย่อย Query จากเมนู Database หน้าจอสอบถามของโมดูลจะแสดงผลของปฏิบัติการนั้น

ก.6.4 การแก้ไขค่าแอตทริบิวท์ของแถวข้อมูล

ผู้ใช้ต้องเลือกออบเจคต์และเวอร์ชันของออบเจคต์ที่ต้องการแก้ไขค่าแอตทริบิวท์จากหน้าจอสอบถาม จากนั้น เลือกเมนูย่อย Update Screen ของเมนู Database ระบบจะแสดงหน้าจอแก้ไขออบเจคต์ โดยทุกฟิลด์ในหน้าจอนี้จะมีค่าแอตทริบิวท์ของเวอร์ชันก่อนการแก้ไข ดังรูป ก33

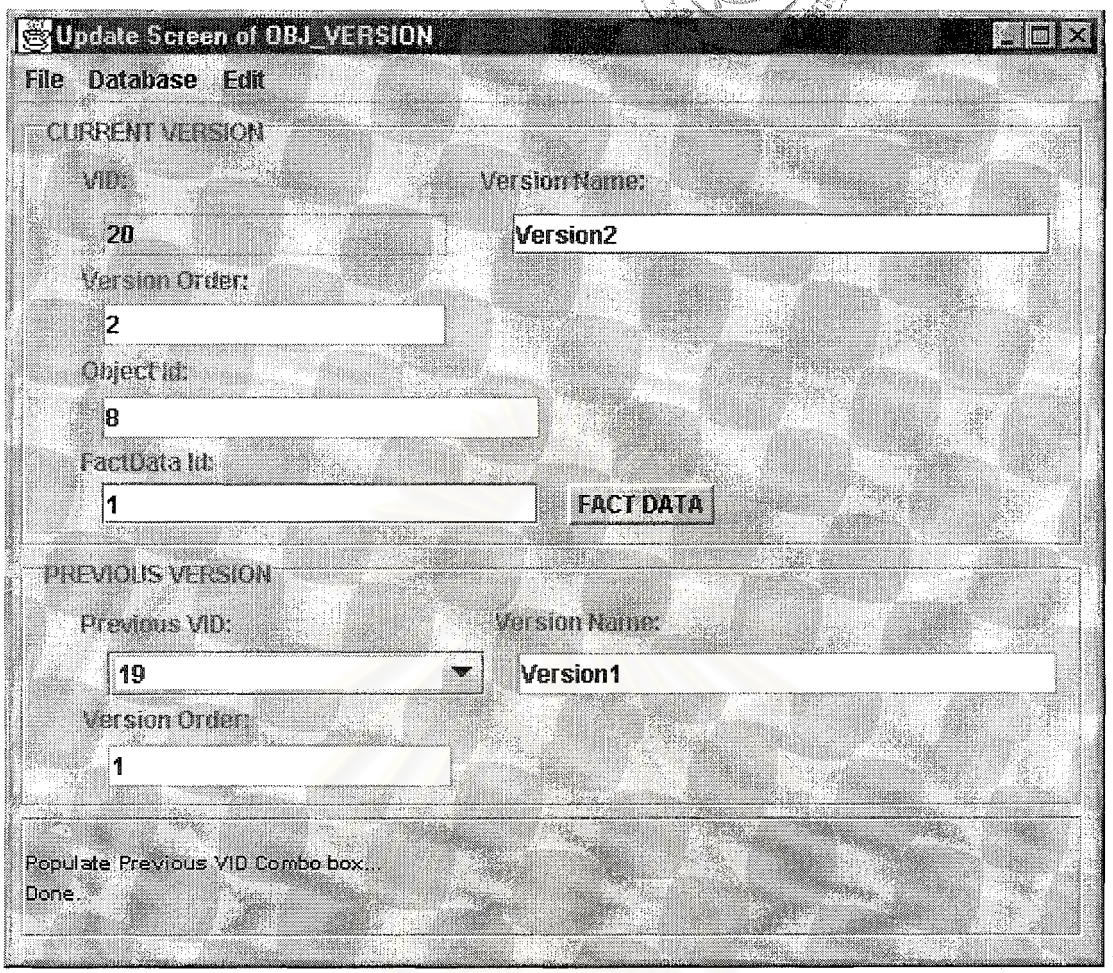
หลังจากผู้ใช้ได้ทำการแก้ไขค่าข้อมูลในฟิลด์แล้วและต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตารางในฐานข้อมูลที่ถูกสร้างจากคุณลักษณะของคลาสนั้น ผู้ใช้ต้องเลือกเมนูย่อย Update จากเมนู Database แต่หากผู้ใช้ไม่ต้องการทำปฏิบัติการปรับปรุงค่าแอตทริบิวท์ของตารางในฐานข้อมูล ก็สามารถทำได้โดยปิดหน้าจอแก้ไขนี้



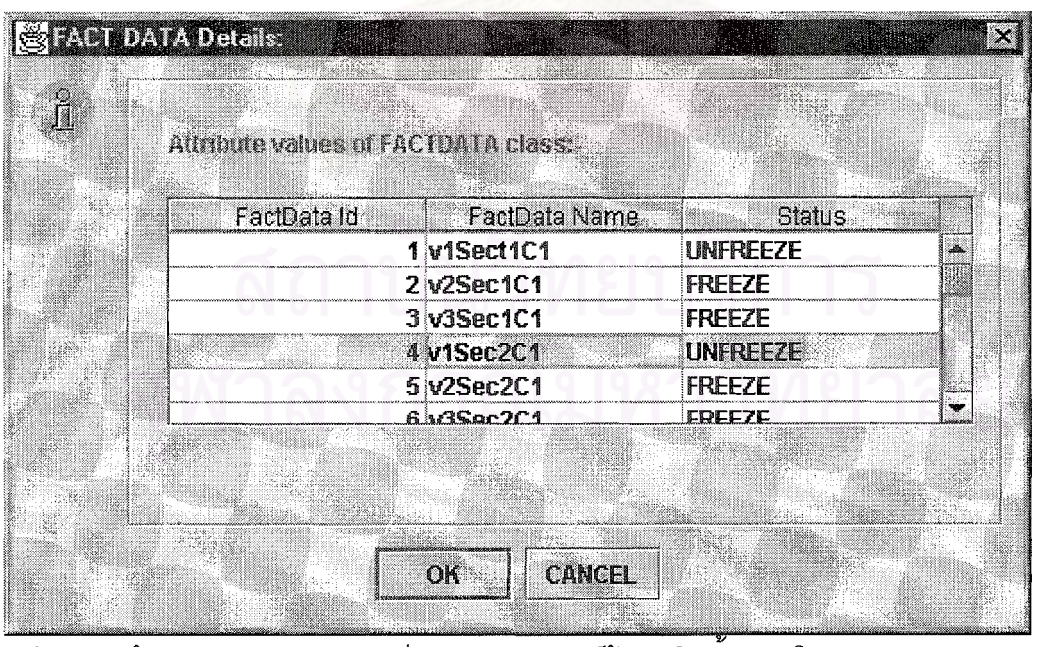
รูป ก32 หน้าจอสร้างออบเจคต์ของโมดูล OBJECT VERSIONS

ก.6.5 การเรียกดูข้อมูลดิบที่สัมพันธ์กับเวอร์ชันหนึ่ง ๆ

ในหน้าจอสอบถามของโมดูล ผู้ใช้ต้องเลือกเวอร์ชันที่ต้องการดูข้อมูลดิบก่อน จากนั้นเลือกเมนูย่อย Fact Data ในเมนู View ระบบจะแสดงหน้าจอย่อย ดังรูป ก35 ซึ่งเป็นหน้าจอที่แสดงข้อมูลดิบของเวอร์ชันที่ผู้ใช้ระบุ



รูป ก33 หน้าจอแก้ไขค่าแอตทริบิวท์ของโมดูล OBJECT VERSIONS: ก่อนการแก้ไข



รูป ก34 หน้าจอย่อยแสดงรายละเอียดของออบเจคต์ข้อมูลดิบทั้งหมดในระบบ

ก.6.6 การลบแถวข้อมูลออกจากตาราง หรือการลบเวอร์ชันออกจากออบเจคต์

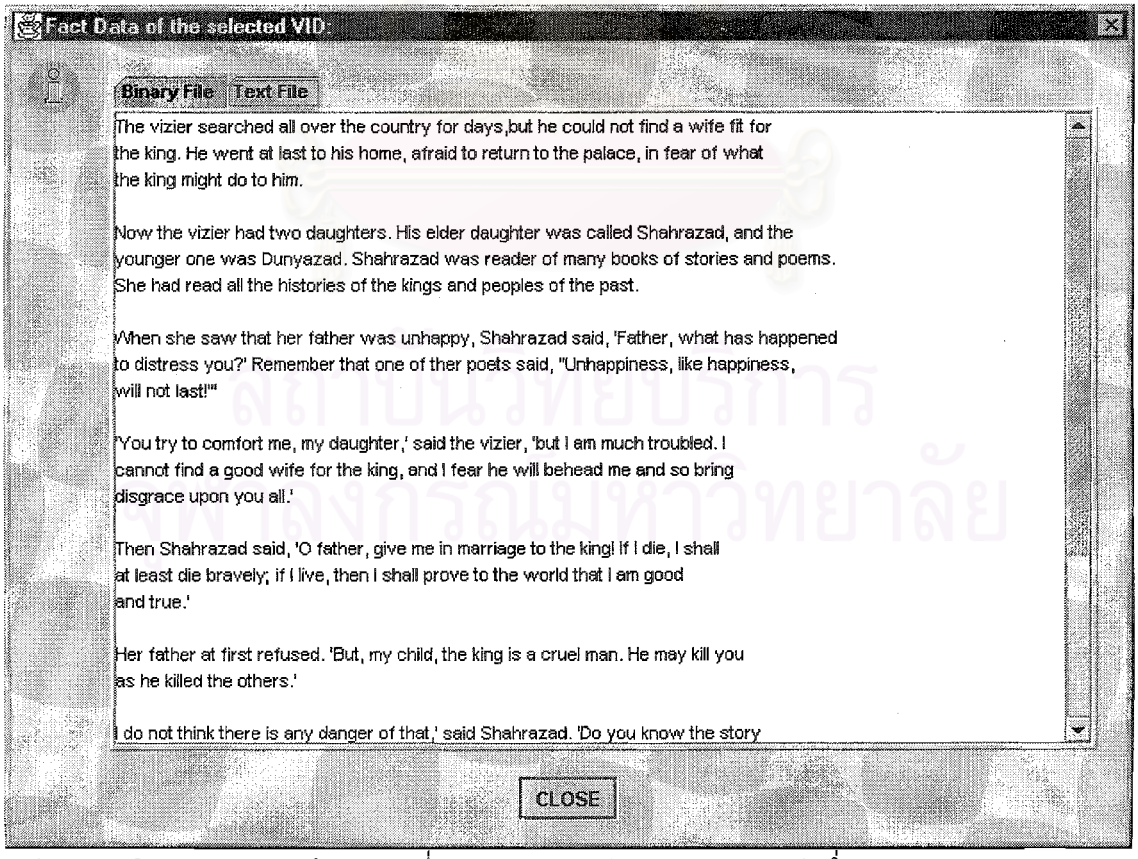
ในหน้าจอสอบถาม ผู้ใช้ต้องเลือกออบเจคต์และเวอร์ชันของออบเจคต์ที่ต้องการลบก่อน จากนั้นเลือกเมนูย่อย Delete ของเมนู Database ระบบจะแสดงหน้าจอย่อย ดังรูป ก36 เพื่อให้ผู้ใช้ยืนยันการทำปฏิบัติการลบเวอร์ชัน หากผู้ใช้กดปุ่ม Yes ระบบจะทำการลบแถวข้อมูลนั้นออกจากตาราง OBJ_VERSION ในฐานข้อมูล แต่ถ้าผู้ใช้กดปุ่ม No ระบบจะไม่ทำปฏิบัติการนี้

ก.7 ปฏิบัติการต่าง ๆ ของโมดูล RELATIONSHIPS

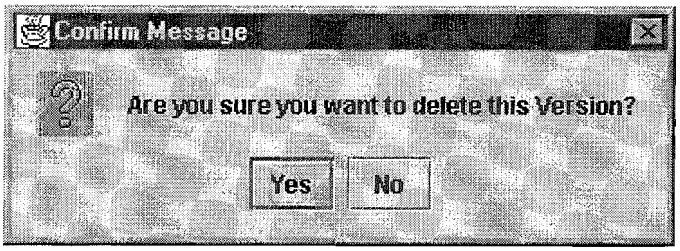
ก.7.1 การเริ่มต้นใช้งานโมดูลและสอบถามข้อมูล

เมื่อผู้ใช้กด RELATIONSHIPS ระบบจะแสดงหน้าจอของโมดูล RELATIONSHIPS รูป ก37 หน้าจอนี้แบ่งออกเป็น 2 ส่วนได้แก่ ส่วนที่แสดงรายละเอียดเกี่ยวกับเวอร์ชันแม่ (Parent Version) และส่วนที่แสดงรายละเอียดเกี่ยวกับเวอร์ชันลูก (Child Version)

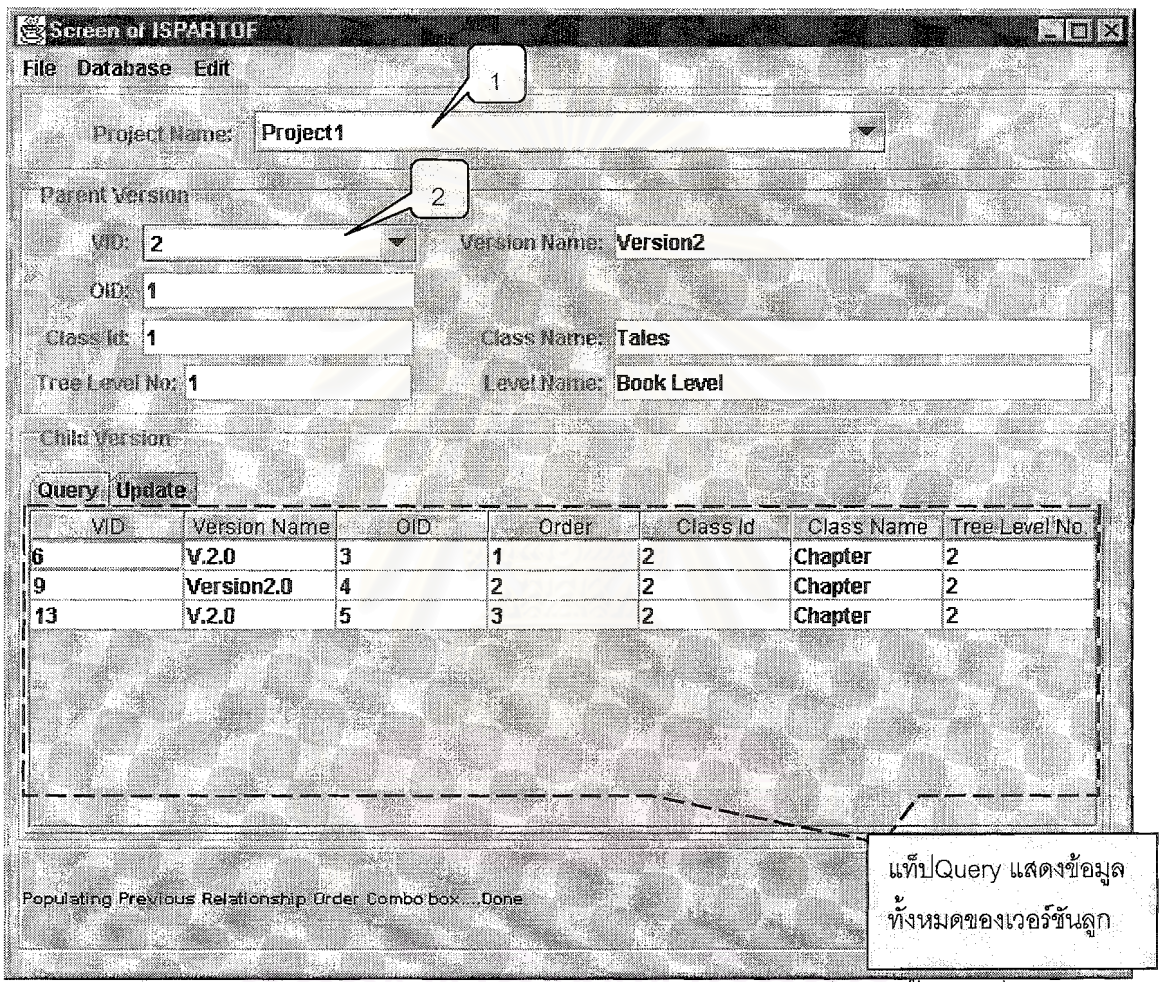
ผู้ใช้สามารถกำหนดเงื่อนไขการสอบถามข้อมูลของตาราง ISPARTOF ได้โดยเลือกชื่อโครงงานจากส่วนประกอบหมายเลข 1 และเลือกค่าของตัวระบุเวอร์ชันแม่จากส่วนประกอบหมายเลข 2 แท็ปชื่อ Query แสดงผลการสอบถามข้อมูลของเวอร์ชันลูกทั้งหมดที่สัมพันธ์กับเวอร์ชันแม่



รูป ก35 หน้าจอย่อยแสดงข้อมูลดิบที่สัมพันธ์กับเวอร์ชันของออบเจคต์หนึ่ง



รูป ก36 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบแถวข้อมูลจากตาราง OBJ_VERSION



รูป ก37 หน้าจอของโมดูล RELATIONSHIPS: แสดงข้อมูลของเวอร์ชันลูกทั้งหมดที่สัมพันธ์กับเวอร์ชันแม่

ก.7.2 การแทรกแถวข้อมูลหรือการสร้างออบเจคต์ความสัมพันธ์

การสร้างออบเจคต์ความสัมพันธ์ระหว่างเวอร์ชันแม่กับเวอร์ชันลูกเป็นการทำปฏิบัติการแทรกแถวข้อมูลเข้าตาราง ISPARTOF ในฐานข้อมูล ซึ่งมีขั้นตอนการทำงาน ต่อไปนี้

1. ในส่วนของเวอร์ชันลูก ให้เลือกแท็ป Update เพื่อแสดงส่วนของหน้าจอสำหรับการเลือกเวอร์ชันลูก

2. เลือกโครงงานที่ต้องการ จากส่วนประกอบหมายเลข 1 ระบบจะดึงค่าตัวระบุเวอร์ชันที่มีคุณสมบัติเป็นเวอร์ชันแม่ได้ มาใส่ไว้ในส่วนประกอบหมายเลข 2
3. เลือกค่าตัวระบุเวอร์ชันแม่จากส่วนประกอบหมายเลข 2 ผลก็คือ
   - ระบบดึงข้อมูลทั้งหมดของเวอร์ชันแม่นั้นมา และใส่ไว้ในฟิลด์ของเวอร์ชันแม่
   - ระบบจะดึงเวอร์ชันลูกทั้งหมดที่มีความสัมพันธ์กับเวอร์ชันแม่ที่เลือกนั้นมาใส่ในส่วนประกอบหมายเลข 3
   - ระบบดึงค่าตัวระบุเวอร์ชันลูกที่มีคุณลักษณะเป็นเวอร์ชันลูกได้ มาใส่ไว้ในส่วนประกอบหมายเลข 4
4. ถ้าผู้ใช้ต้องการให้ออบเจคต์ความสัมพันธ์ที่จะสร้างเป็นออบเจคต์ลำดับแรก ที่ส่วนประกอบหมายเลข 3 ให้เลือกค่า NONE ดังรูป ก38 ในทางตรงกันข้าม ถ้าออบเจคต์ความสัมพันธ์นั้นเป็นออบเจคต์ลำดับที่2 ขึ้นไป ผู้ใช้ต้องเลือกตัวระบุเวอร์ชันจากส่วนประกอบหมายเลข 3 ก่อน ดังรูป ก39 เพื่อกำหนดลำดับที่ให้กับออบเจคต์ความสัมพันธ์ที่ต้องการสร้าง

5. เลือกค่าตัวระบุเวอร์ชันของเวอร์ชันลูกจากส่วนประกอบหมายเลข 4 เพื่อใช้ในการสร้างออบเจคต์ความสัมพันธ์

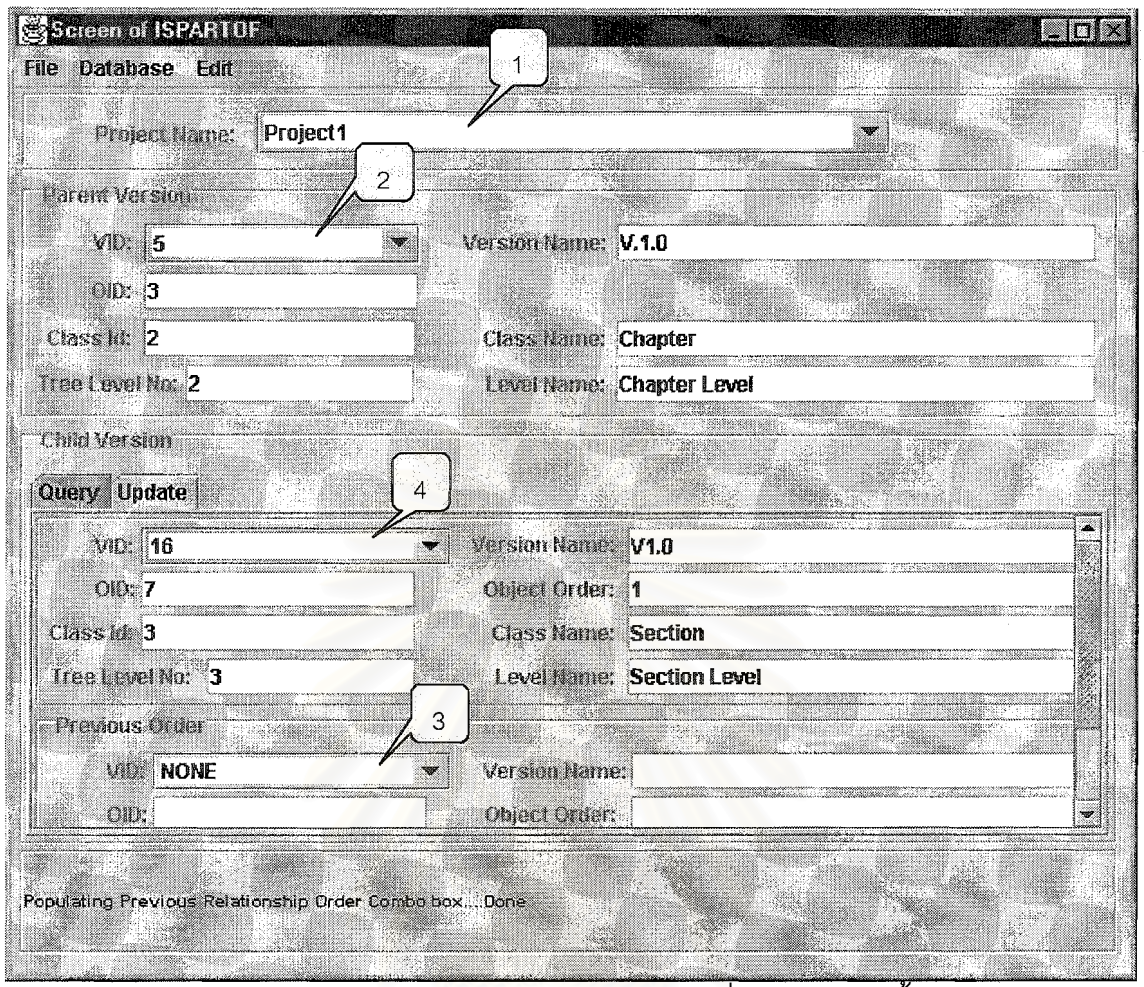
จากนั้นให้ผู้ใช้เลือกเมนูย่อย Insert ของเมนู Database เพื่อแทรกแถวข้อมูลเข้าตาราง ISPARTOF ในฐานข้อมูล เมื่อปฏิบัติการแทรกแถวข้อมูลสำเร็จแล้ว ระบบจะทำการเคลียร์ค่าในฟิลด์ทั้งหมดของเวอร์ชันลูก

ก.7.3 การปรับปรุงแท็ป Query หลังทำปฏิบัติการต่าง ๆ

โมดูล RELATIONSHIPS จะมีเพียงหน้าจอเดียว หากผู้ใช้ต้องการดูผลของการทำปฏิบัติการต่างๆ ให้ผู้ใช้เลือกแท็ป Query จากนั้นเลือกเมนูย่อย Query ของเมนู Database ที่แท็ป Query ก็จะแสดงผลของปฏิบัติการนั้น

ก.7.4 การแก้ไขค่าแอตทริบิวท์ของออบเจคต์ความสัมพันธ์

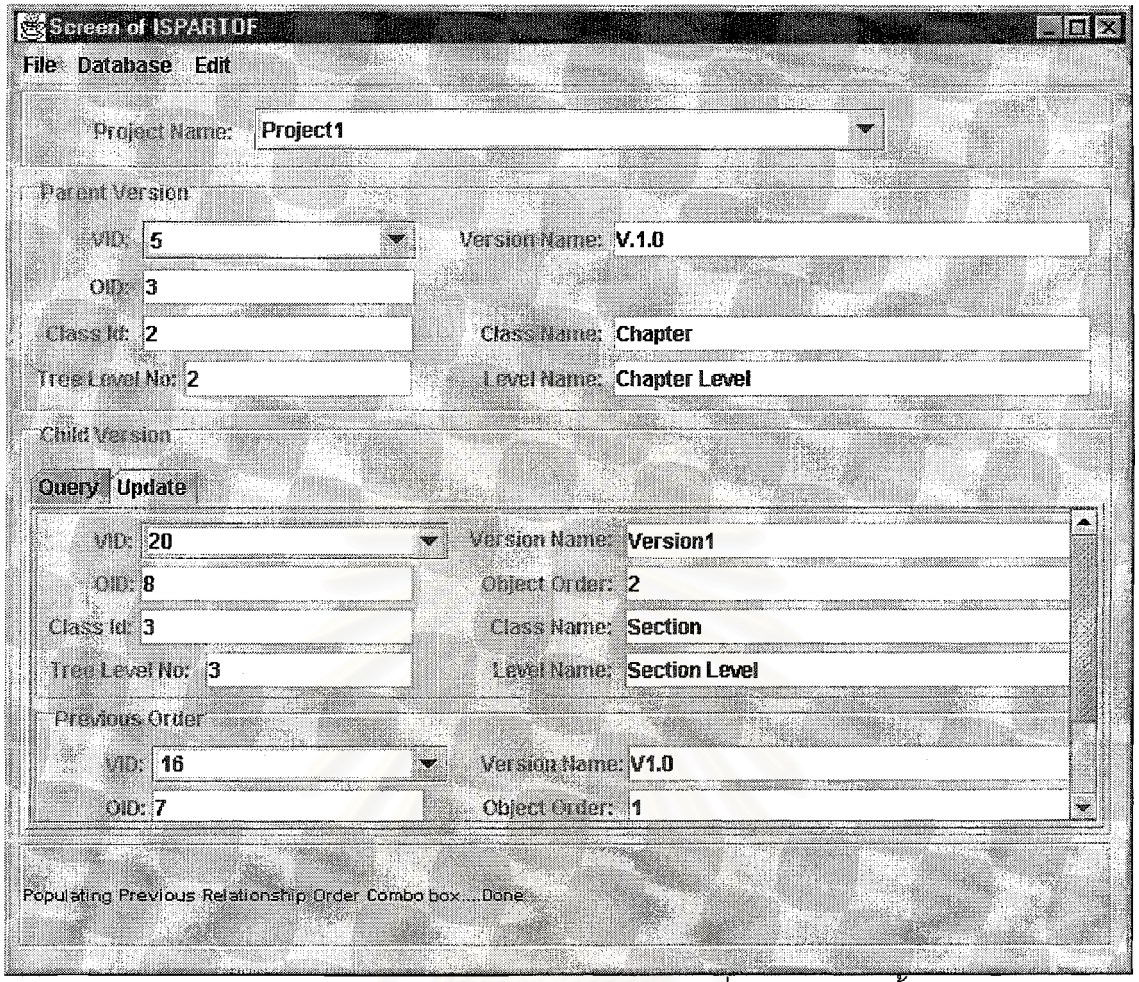
จากการสร้างออบเจคต์ความสัมพันธ์ จะเห็นได้ว่ามีลักษณะการทำงานที่คล้ายคลึงกับการโยงเส้นความสัมพันธ์ระหว่างเวอร์ชันแม่กับเวอร์ชันลูก ส่วนค่าแอตทริบิวท์ของเวอร์ชันแม่และเวอร์ชันลูกเป็นค่าข้อมูลที่ระบบงานวิจัยดึงมาจากตาราง OBJ_VERSION ในฐานข้อมูล ดังนั้นโมดูล RELATIONSHIPS จึงไม่มีปฏิบัติการแก้ไขค่าแอตทริบิวท์ของออบเจคต์ความสัมพันธ์ เช่นเดียวกับการโยงเส้นความสัมพันธ์ที่ไม่มีการแก้ไขคุณลักษณะของเส้นที่ได้โยงไปแล้ว



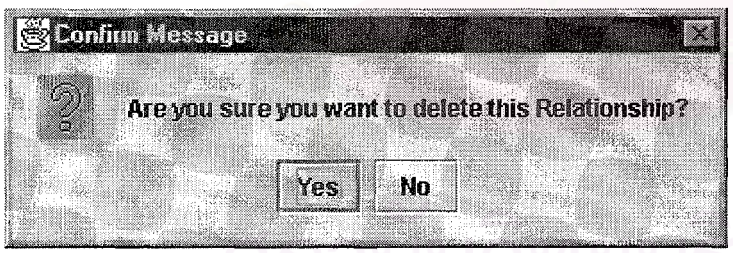
รูป ก38 หน้าจอแสดงการสร้างออบเจคต์ความสัมพันธ์: ลำดับที่ของออบเจคต์นี้มีค่าเป็น 1

ก.7.5 การลบออบเจคต์ความสัมพันธ์ออกจากตาราง ISPARTOF

ในแท็ป Query ผู้ใช้ต้องเลือกเวอร์ชันแม่จากส่วนประกอบหมายเลข 2 และเลือกแถวของเวอร์ชันลูกที่ต้องการลบ จากนั้นเลือกเมนูย่อย Delete ในเมนู Database ระบบจะแสดงหน้าจอย่อยดังรูป ก40 เพื่อให้ผู้ใช้ยืนยันการทำปฏิบัติการลบออบเจคต์ความสัมพันธ์ หากผู้ใช้กดปุ่ม Yes ระบบจะทำการลบออบเจคต์ความสัมพันธ์นั้นออกจากตาราง ISPARTOF ในฐานข้อมูล แต่ถ้าผู้ใช้กดปุ่ม No ระบบจะไม่ทำปฏิบัติการนี้



รูป ก39 หน้าจอแสดงการสร้างออบเจคต์ความสัมพันธ์: ลำดับที่ของออบเจคต์นี้มีค่าเป็น 2



รูป ก40 หน้าจอย่อยแสดงข้อความขอคำยืนยันการลบออบเจคต์ความสัมพันธ์

ก.8 การใช้งานโมดูล VIEW HIERARCHY

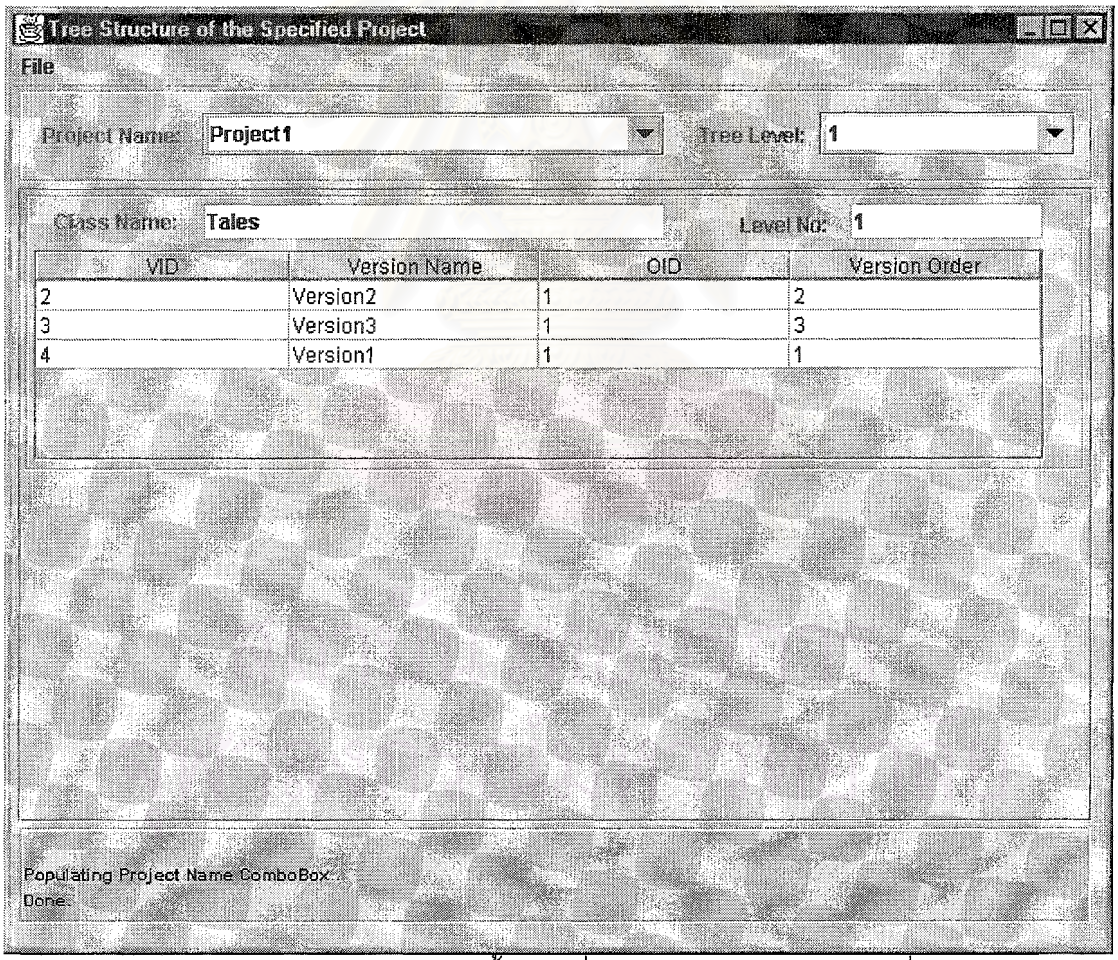
โมดูล VIEW HIERARCHY เป็นโมดูลที่ใช้สำหรับดูค่าความสัมพันธ์แบบคอมโพซิชัน หรือแอกกรีเกชัน ระหว่างเวอร์ชันแม่กับเวอร์ชันลูก โดยแสดงในลักษณะของโครงสร้างต้นไม้ ซึ่งมีขั้นตอนการใช้งาน ต่อไปนี้

1. เลือกโครงงานที่ต้องการดูโครงสร้างต้นไม้ จากส่วนประกอบหมายเลข 1

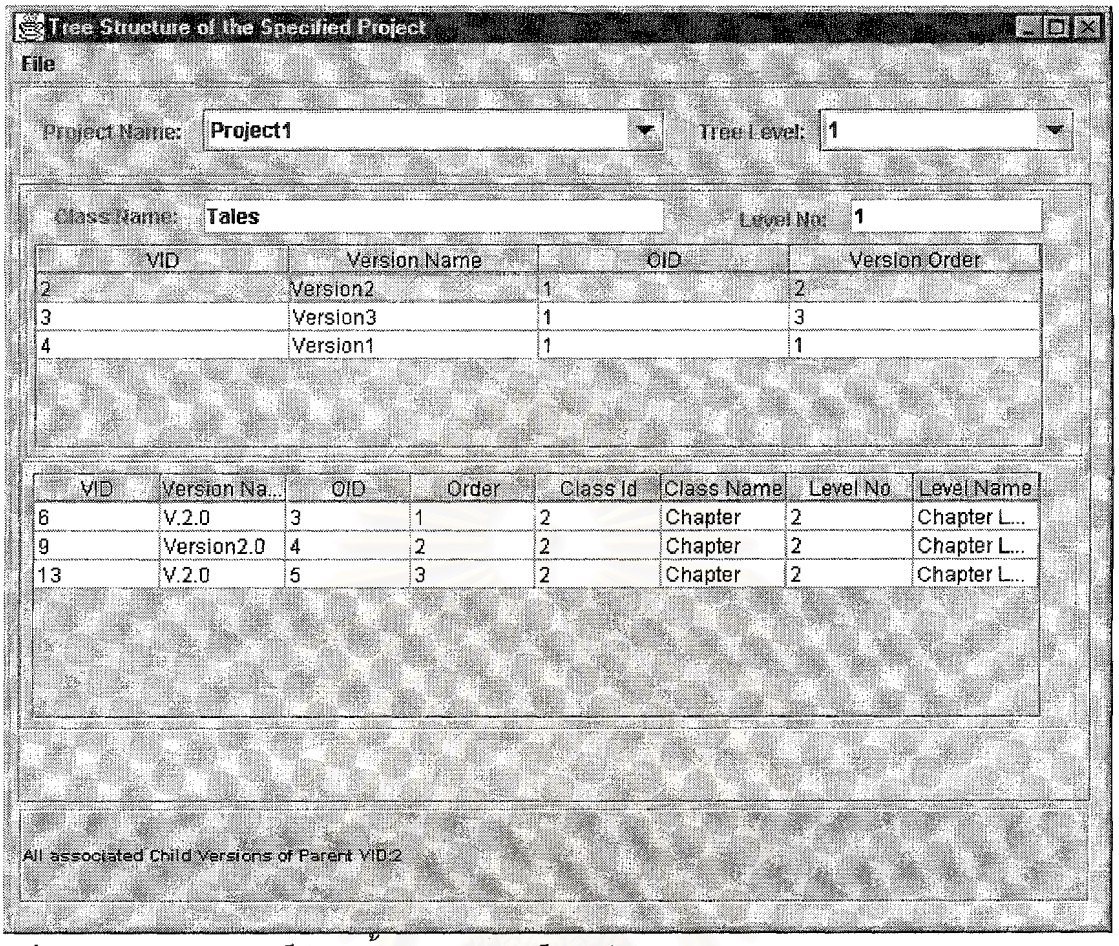
2. เลือกระดับชั้นของโครงสร้างต้นไม้ เพื่อเป็นระดับชั้นแรกสุดของการดู ระบบจะแสดงข้อมูลของเวอร์ชันทั้งหมดที่เป็นของออบเจคต์ซึ่งอยู่ในระดับชั้นที่ผู้ใช้เลือกในตารางแรกของหน้าจอ ดังรูป ก41

3. เลือกเวอร์ชันออบเจคต์จากตารางแรก ระบบจะกำหนดให้เวอร์ชันนี้เป็นเวอร์ชันแม่ และแสดงแอตทริบิวท์ของเวอร์ชันลูกทั้งหมดที่สัมพันธ์กับเวอร์ชันแม่ในตารางที่ 2 ซึ่งอยู่ถัดลงมาจากตารางที่ 1 ดังรูป ก42 ทำเช่นนี้เรื่อยไปจนถึงเวอร์ชันออบเจคต์ในระดับล่างสุดของโครงสร้างต้นไม้ รูป ก43

เนื่องจากโมดูล VIEW HIERARCHY เป็นโมดูลที่ใช้ในการดูความสัมพันธ์ระหว่างเวอร์ชั่นของออบเจคต์ซึ่งอยู่คนละระดับชั้นของโครงสร้างต้นไม้ โมดูลนี้จึงไม่มีปฏิบัติการอื่นๆ ในการใช้งาน

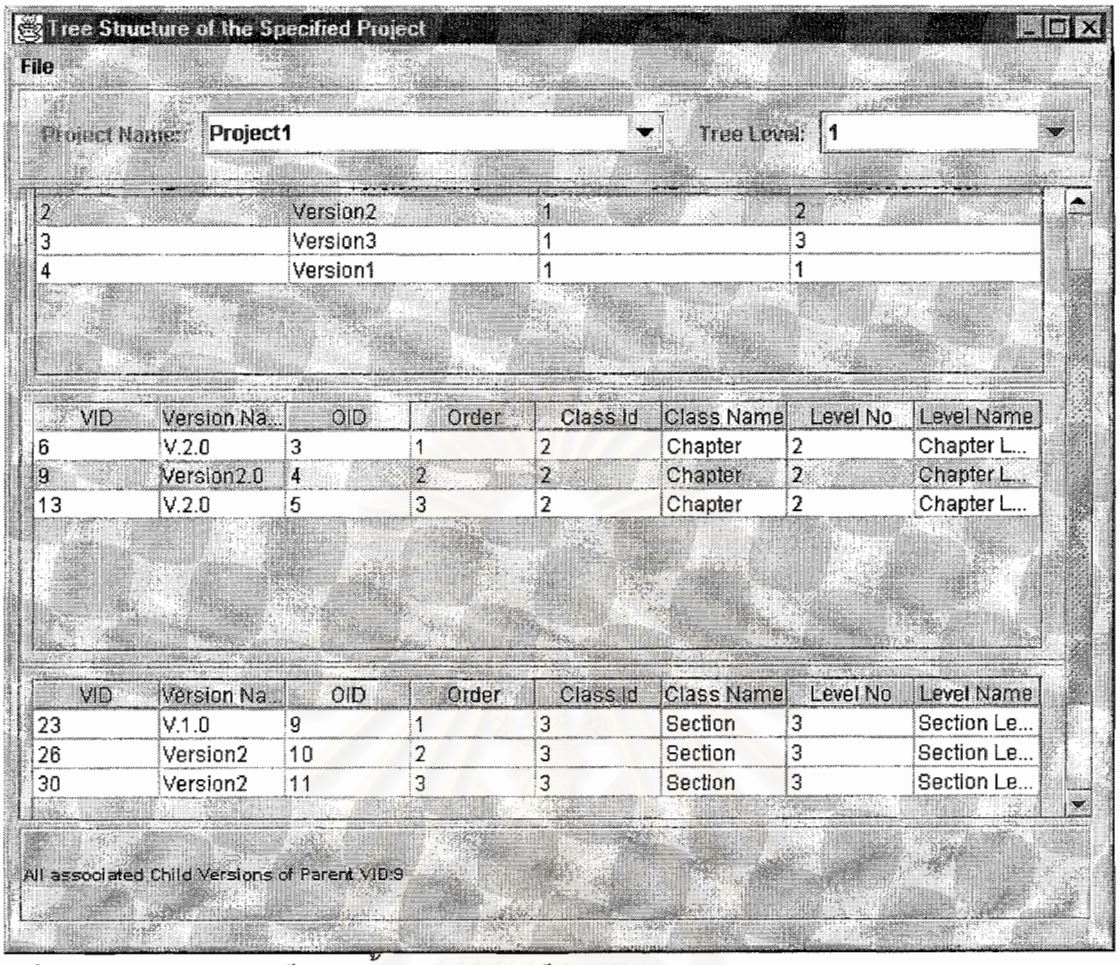


รูป ก41 ระบบแสดงเวอร์ชันของออบเจคต์ทั้งหมดที่อยู่ในระดับโครงสร้างต้นไม้ซึ่งผู้ใช้เลือก



รูป ก42 ระบบแสดงเวอร์ชันลูกทั้งหมดของเวอร์ชันแม่ VID:2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูป ก43 ระบบแสดงเวอร์ชันลูกทั้งหมดของเวอร์ชันแม่ VID:9

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข
ตัวอย่างข้อมูลที่ใช้ในการทดสอบ

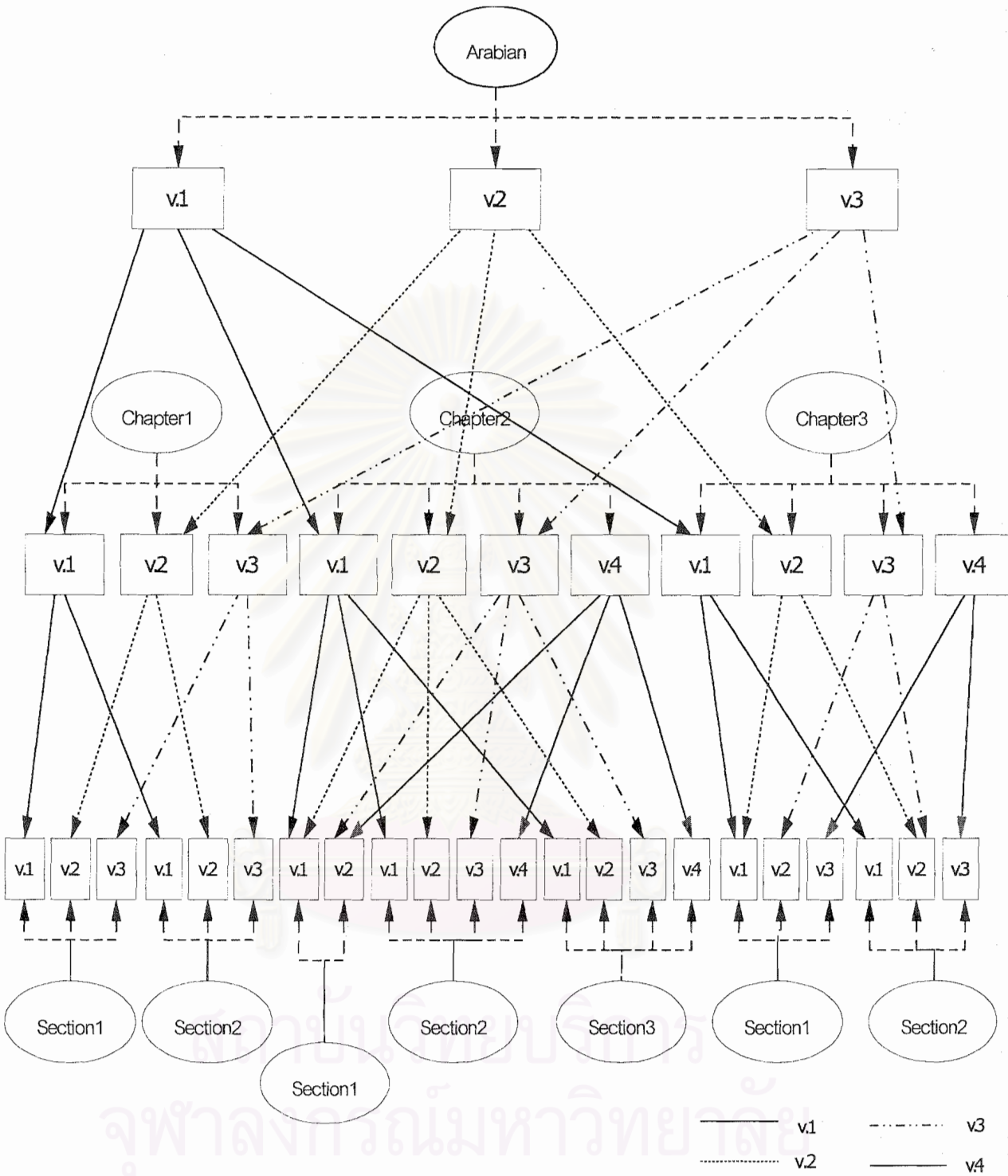
ระบบงานจริงที่ต้องการคุณสมบัติของระบบซึ่งสามารถจัดการกับข้อมูลที่มีหลายๆเวอร์ชัน ได้แก่ ระบบงานที่เกี่ยวข้องกับกระบวนการออกแบบชิ้นงาน ระบบที่เกี่ยวข้องกับข้อมูลซึ่งมีค่าแปรผันตามเวลา เป็นต้น ประกอบกับการออกแบบระบบงานวิจัยนี้มีวัตถุประสงค์หลักที่จะใช้จัดการกับข้อมูลซึ่งมีความสัมพันธ์แบบคอมโพซิชัน และแอกกรีเกชัน ดังนั้น การทดสอบการทำงานของระบบควบคุมเวอร์ชันข้อมูล ผู้วิจัยได้สมมติข้อมูลของการเขียนหนังสือขึ้นเพื่อใช้เป็นตัวแทนข้อมูลที่แสดงความสัมพันธ์แบบคอมโพซิชัน

เนื่องจาก กระบวนการออกแบบชิ้นงานหนึ่ง ประกอบด้วยหลายๆ ออบเจคต์ ในแต่ละออบเจคต์จะมีหลายเวอร์ชัน ทำให้ยากแก่การสังเกตุผลการทดสอบระบบ ดังนั้นจึงนำโครงสร้างต้นไม้มาใช้ในการแสดงความสัมพันธ์ระหว่างออบเจคต์และเวอร์ชันต่างๆ ของออบเจคต์ที่อยู่ภายในโครงงานการออกแบบหนึ่งๆ โดยนำแนวคิดของ แอนด์/ออร์ กราฟ มาใช้ในการวาดโครงสร้างต้นไม้ และใช้ตารางในการแสดงตัวอย่างค่าข้อมูลต่างๆ ทั้งที่ผู้ใช้ป้อนเข้าระบบและค่าข้อมูลที่ระบบสร้างขึ้น เช่น ค่าของตัวระบุออบเจคต์ ค่าของตัวระบุเวอร์ชัน เป็นต้น เพื่อแสดงความแตกต่างระหว่างค่าข้อมูลทั้งสองประเภทจึงใช้เครื่องหมาย * ต่อท้ายชื่อสดมภ์ที่เก็บค่าข้อมูลซึ่งถูกสร้างโดยระบบ ชื่อสดมภ์ใดที่ไม่มีเครื่องหมาย * ต่อท้ายชื่อสดมภ์จะเป็นสดมภ์ที่เก็บค่าข้อมูลที่ผู้ใช้ป้อน

ข.1 ตัวอย่างการแต่งหนังสือ

ข.1.1 โครงสร้างต้นไม้ของการแต่งหนังสือ

โครงสร้างต้นไม้ของการแต่งหนังสือในรูป ข1 แสดงความสัมพันธ์ระหว่างออบเจคต์กับเวอร์ชันของออบเจคต์ และความสัมพันธ์ระหว่างออบเจคต์ที่อยู่ต่างระดับกันของโครงสร้างต้นไม้ โดยออร์โหนดจะแทนออบเจคต์อินสแตนซ์ของคลาสในระดับต่างๆ ของโครงสร้างต้นไม้ และแอนด์โหนดจะแทนเวอร์ชันของออบเจคต์ เช่น ออร์โหนด Arabian เป็นออบเจคต์อินสแตนซ์ของคลาส Tales ขณะที่ Chapter1, Chapter2, และ Chapter3 เป็นออบเจคต์อินสแตนซ์ของคลาส Chapter เป็นต้น



รูป ข1 โครงสร้างต้นไม้ของตัวอย่างข้อมูลการเขียนหนังสือ

ข.1.2 ตารางแสดงข้อมูลของการแต่งหนังสือ

ตาราง ข1 ตัวอย่างค่าข้อมูลของตาราง PROJECT0 ในระบบฐานข้อมูล

PROJ_NAME	OWNER	CREATE_DATE	FINISH_DATE	DESCRIPTION
Project1	Margery Green	12-JAN-59	25-NOV-60	Project for making a book as the example.

ตาราง ข2 ตัวอย่างค่าข้อมูลของตาราง TREE_LEVEL0 ในระบบฐานข้อมูล

LEVPROJ_ID*	LEVEL_NO	PROJ_NAME	LEVEL_NAME	DESCRIPTION
1	1	Project1	Book Level	The 1st Level of Tree Hierarchy.
2	2	Project1	Chapter Level	The 2nd Level of Tree Hierarchy.
3	3	Project1	Section Level	The 3rd Level of Tree Hierarchy

ตาราง ข3 ตัวอย่างค่าข้อมูลของตาราง APP_CLASS ในระบบฐานข้อมูล

CLASS_ID*	CLASS_NAME	TLEV_PK	DB_FLAG	CLASS_ATTR
1	Tales	1	0	1,Book_No,NUMBER,2 2,Name,VARCHAR2,45 3,Publish_Year,NUMBER,4
2	Chapter	2	0	1,Chapter_No,NUMBER,2 2,Chapter_name,VARCHAR2,50 3,Title,VARCHAR2,50
3	Section	3	0	1,Sect_No,NUMBER,3 2,Sect_name,VARCHAR2,45 3,Sect_title,VARCHAR2,50

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง ข4 ตัวอย่างค่าข้อมูลของตาราง FACTDATA ในระบบฐานข้อมูล

FACTDATA_OID*	NAME	STATUS	CREATED_DATE	CHARFILE	BINFILE
1	V1Sect1C1	Unfreeze	1959-04-25	v1s1c1.txt	ALASKA1.jpg
2	V2Sec1C1	Freeze	1995-05-12	v2s2c1.txt	QLND1.jpg
3	V3Sec1C1	Freeze	1995-05-24	v3s1c1.txt	Null
4	V1Sec2C1	Unfreeze	1995-06-04	v1s2c1.txt	Mk22.jpg
5	V2Sec2C1	Freeze	1995-06-16	v2s2c1.txt	Null
6	V3Sec2C1	Freeze	1995-06-20	v3s2c1.txt	Null
7	V1Sec1C2	Unfreeze	1995-07-12	v1s1c2.txt	Mk30.jpg
8	V2Sec1C2	Freeze	1996-07-28	v2s1c2.txt	Null
9	V1Sec2C2	Unfreeze	1996-07-15	v1s2c2.txt	Null
10	V2Sec2C2	Unfreeze	1996-07-28	v2s2c2.txt	Null
11	V3Sec2C2	Unfreeze	1996-07-29	v3s2c2.txt	Null
12	V4Sec2C2	Freeze	1996-08-05	v3s2c2.txt	Null
13	V1Sec1C3	Unfreeze	1959-08-11	v1s1c3.txt	Null
14	V2Sec1C3	Unfreeze	1959-08-16	v2s1c3.txt	Null
15	V3Sec1C3	Freeze	1959-08-17	v3s1c3.txt	Null
21	V1Sec3C2	Unfreeze	1995-07-24	v1s3c2.txt	Mk46.jpg
22	V2Sec3C2	Unfreeze	1959-07-28	v2s3c2.txt	Null
23	V3Sec3C2	Unfreeze	1959-08-02	v3s3c2.txt	Msw1.jpg
24	V4Sec3C2	Freeze	1959-08-08	v4s3c2.txt	Null
25	V1Sec2C3	Unfreeze	1959-08-20	v1s2c3.txt	Null
26	V2Sec2C3	Unfreeze	1959-08-24	v2s3c3.txt	Null
27	V3Sec2C3	Freeze	1959-08-28	v3s2c3.txt	Cow.gif

ตาราง ข5 ตัวอย่างค่าข้อมูลของตาราง SECTION ในระบบฐานข้อมูล

OID*	SECT_NO	SECT_NAME	SECT_TITLE
7	1	Section1	Section1 of Chapter1
8	2	Section2	Section2 of Chapter1
9	1	Section1	Section1 of Chapter2
10	2	Section2	Section2 of Chapter2
11	3	Section3	Section3 of Chapter2
12	1	Section1	Section1 of Chapter3
13	2	Section2	Section2 of Chapter3

ตาราง ข6 ตัวอย่างค่าข้อมูลของตาราง CHAPTER ในระบบฐานข้อมูล

OID*	CHAPTER_NO	CHAPTER_NAME	TITLE
3	1	Chapter1	Shahriyar the King
4	2	Chapter2	Abdulla of the Land and Abdulla of the Sea
5	3	Chapter3	The fable of the peacock and the peahen,Etc.

ตาราง ข7 ตัวอย่างค่าข้อมูลของตาราง TALES ในระบบฐานข้อมูล

OID*	BOOK_NO	NAME	PUBLISH_YEAR
1	1	Arabian	1959

ตาราง ข8 ตัวอย่างค่าข้อมูลของตาราง OBJ_VERSION ในระบบฐานข้อมูล

VID*	VERSION NAME	OID	VERSION ORDER*	FACTDATA OID
2	Version2	1	2	Null
3	Version3	1	3	Null
4	Version1	1	1	Null
5	V.1.0	3	1	Null
6	V.2.0	3	2	Null
7	V.3.0	3	3	Null

ตาราง ข8(ต่อ) ตัวอย่างค่าข้อมูลของตาราง OBJ_VERSION ในระบบฐานข้อมูล

VID*	VERSION NAME	OID	VERSION ORDER*	FACTDATA OID
8	V.1.0	4	1	Null
9	Version2.0	4	2	Null
10	V.3.0	4	3	Null
11	Version4.0	4	4	Null
12	V.1.0	5	1	Null
13	V.2.0	5	2	Null
14	Version3.0	5	3	Null
15	Version4.0	5	4	Null
16	V1.0	7	1	1
17	V.2.0	7	2	2
18	V.3.0	7	3	3
20	Version1	8	1	4
21	Version2.0	8	2	5
22	Version3.0	8	3	6
23	V.1.0	9	1	7
24	V2.0	9	2	8
25	Version1	10	1	9
26	Version2	10	2	10
27	Version3	10	3	11
28	Version4	10	4	12
29	V1.0	11	1	21
30	Version2	11	2	22
31	Version3	11	3	23
32	Version4	11	4	24
33	V1.0	12	1	13
34	V2	12	2	14
35	V3	12	3	15

ตาราง ข8(ต่อ) ตัวอย่างค่าข้อมูลของตาราง OBJ_VERSION ในระบบฐานข้อมูล

VID*	VERSION NAME	OID	VERSION ORDER*	FACTDATA OID
36	V1	13	1	25
37	V2	13	2	26
38	V3	13	3	27

ตาราง ข9 ตัวอย่างค่าข้อมูลของตาราง ISPARTOFในระบบฐานข้อมูล

CHILD VID	PARENT VID	CHILD ORDER*
6	2	1
9	2	2
13	2	3
7	3	1
10	3	2
14	3	3
5	4	1
8	4	2
12	4	3
16	5	1
20	5	2
17	6	1
21	6	2
18	7	1
22	7	2
23	8	1
25	8	2
29	8	3
23	9	1
26	9	2
30	9	3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง ข9(ต่อ) ตัวอย่างค่าข้อมูลของตาราง ISPARTOFในระบบฐานข้อมูล

CHILD VID	PARENT VID	CHILD ORDER*
24	10	1
27	10	2
31	10	3
33	12	1
36	12	2
33	13	1
37	13	2
34	14	1
37	14	2
35	15	1
38	15	2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค
สคริปท์สำหรับสร้างออบเจคต์หลักในฐานข้อมูล

ภาคผนวกนี้แสดงสคริปท์สำหรับสร้างออบเจคต์หลักต่างๆ ในฐานข้อมูล ได้แก่ ตาราง และวิว โดยออบเจคต์เหล่านี้เป็นออบเจคต์ที่ถูกสร้างขึ้นก่อนเพื่อให้ระบบงานวิจัยสามารถทำปฏิบัติการต่างๆ กับระบบฐานข้อมูล เนื่องจาก ระบบงานวิจัยใช้ระบบจัดการฐานข้อมูลออราเคิล ดังนั้น สคริปท์ที่แสดงในภาคผนวกนี้ จึงเป็นสคริปท์ที่ใช้กับระบบจัดการฐานข้อมูลออราเคิล

ค.1 สคริปท์สำหรับสร้างตารางและข้อบังคับต่าง ๆ ของตาราง

ตาราง ค1 สคริปท์สำหรับสร้างตาราง PROJECT0

ตาราง: PROJECT0
ประเภทตาราง:ตารางแบบออบเจคต์
สคริปท์:

```
CREATE OR REPLACE TYPE project_type AS OBJECT
( proj_name       VARCHAR2(60),
  owner  VARCHAR2(60),
  create_date   DATE,
  finish_date     DATE,
  description       VARCHAR2(150));
CREATE TABLE project0 OF project_type
(CONSTRAINT pk_project0 PRIMARY KEY (proj_name),
 CONSTRAINT nn_owner CHECK (owner IS NOT NULL),
 create_date DEFAULT SYSDATE) OBJECT IDENTIFIER IS PRIMARY KEY;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง ค2 สคริปท์สำหรับสร้างตาราง TREE_LEVEL0

ตาราง: TREE_LEVEL0
ประเภทตาราง: ตารางแบบออบเจคต์
สคริปท์:

```
CREATE OR REPLACE TYPE tree_level_type AS OBJECT
( levproj_id        NUMBER(5),
  level_no          NUMBER(2),
  proj_name         VARCHAR2(60),
  level_name        VARCHAR2(70),
  description       VARCHAR2(150));
CREATE TABLE tree_level0 OF tree_level_type
( CONSTRAINT pk_treelevel0 PRIMARY KEY (levproj_id),
  CONSTRAINT fk_projname0 FOREIGN KEY (proj_name)
              REFERENCES project0(proj_name) ON DELETE CASCADE,
  CONSTRAINT uniq_levno_projn0 UNIQUE (level_no,proj_name),
  CONSTRAINT nn_projName CHECK (proj_name IS NOT NULL))
  OBJECT IDENTIFIER IS PRIMARY KEY;
```

ตาราง ค3 สคริปท์สำหรับสร้างตาราง FACTDATA

ตาราง: FACTDATA
ประเภทตาราง: ตารางแบบออบเจคต์
สคริปท์:

```
CREATE OR REPLACE TYPE factdata_type AS OBJECT
(factdata_oid NUMBER(10),
 name VARCHAR2(70),
 status VARCHAR2(40),
 created_date DATE,
 charfile BFILE,
 binfile BFILE);
CREATE TABLE factdata OF factdata_type
(CONSTRAINT pk_factdata PRIMARY KEY (factdata_oid),
  charfile DEFAULT NULL,
  binfile DEFAULT NULL) OBJECT IDENTIFIER IS PRIMARY KEY;
CREATE OR REPLACE DIRECTORY bfile_dir AS 'e:/Thesis/implement/database/files/';
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง ค4 สคริปท์สำหรับสร้างตาราง APP_CLASS

ตาราง: APP_CLASS
ประเภทตาราง: ตารางซ้อน (Nested Table)
สคริปท์:

```
CREATE OR REPLACE TYPE attribute_type AS OBJECT
(type_noNUMBER(2),
 type_name      VARCHAR2(45),
 datatype       VARCHAR2(30),
 length         NUMBER(4));
CREATE TYPE attrlist_type AS TABLE OF attribute_type;
CREATE TABLE app_class
(class_id       NUMBER(4) CONSTRAINT pk_appclass PRIMARY KEY,
 class_name     VARCHAR2(70),
 tlev_pk tlev_view_type  CONSTRAINT nn_tlevpk NOT NULL,
 db_flag NUMBER(1),
 class_attr     attrlist_type) NESTED TABLE class_attr STORE AS attrlist_tab;
ALTER TABLE attrlist_tab
    ADD (CONSTRAINT pk_attrlisttab PRIMARY KEY (nested_table_id,type_no));
ALTER TABLE attrlist_tab
    ADD (CONSTRAINT uniq_typename UNIQUE (nested_table_id,type_name));
```

ตาราง ค5 สคริปท์สำหรับสร้างตาราง OBJ_INSTANCE

ตาราง: OBJ_INSTANCE
ประเภทตาราง: ตารางแบบออบเจคต์
สคริปท์:

```
CREATE OR REPLACE TYPE obj_instance_type AS OBJECT
(oid            NUMBER(10),
 class_id       NUMBER(4));
CREATE TABLE obj_instance OF obj_instance_type
(CONSTRAINT pk_objinst PRIMARY KEY (oid),
 CONSTRAINT fk_classid FOREIGN KEY (class_id)
        REFERENCES app_class(class_id) ON DELETE CASCADE,
 CONSTRAINT nn_classid CHECK (class_id IS NOT NULL))
 OBJECT IDENTIFIER IS PRIMARY KEY;
```

ตาราง ค6 สคริปท์สำหรับสร้างตาราง OBJ_VERSION

ตาราง: OBJ_VERSION
ประเภทตาราง: ตารางแบบออบเจคต์
สคริปท์:

```
CREATE TYPE obj_version_type AS OBJECT
(vid              NUMBER(14),
 version_name     VARCHAR2(50),
 app_oid          obj_instance_type,
 factdata_oid     factdata_view_type,
 version_order    NUMBER(4));
CREATE TABLE obj_version OF obj_version_type
(CONSTRAINT pk_objversion PRIMARY KEY (vid),
 CONSTRAINT nn_appoid CHECK (app_oid IS NOT NULL),
 CONSTRAINT nn_versionorder CHECK (version_order IS NOT NULL))
 OBJECT IDENTIFIER IS PRIMARY KEY;
```

ตาราง ค7 สคริปท์สำหรับสร้างตาราง ISPARTOF

ตาราง: ISPARTOF
ประเภทตาราง: ตารางแบบออบเจคต์
สคริปท์:

```
CREATE OR REPLACE TYPE ispartof_type AS OBJECT
(child_vid    NUMBER(14),
 parent_vid   NUMBER(14),
 child_order  NUMBER(4));
CREATE TABLE ispartof OF ispartof_type
(CONSTRAINT pk_ispartof PRIMARY KEY (child_vid,parent_vid),
 CONSTRAINT fk_childvid FOREIGN KEY (child_vid)
        REFERENCES obj_version(vid) ON DELETE CASCADE,
 CONSTRAINT fk_parentvid FOREIGN KEY (parent_vid)
        REFERENCES obj_version(vid) ON DELETE CASCADE)
 OBJECT IDENTIFIER IS PRIMARY KEY;
```

ค.2 สคริปท์สำหรับสร้างวิว

ตาราง ค8 สคริปท์สำหรับสร้างวิว TLEV_VIEW

วิว: TLEV_VIEW **ประเภทวิว:** วิวแบบออบเจคต์ **สคริปท์:**
CREATE TYPE tlev_view_type AS OBJECT (levproj_id NUMBER(5), level_no NUMBER(2), proj_name VARCHAR2(60), level_name VARCHAR2(70)); CREATE VIEW tlev_view OF tlev_view_type WITH OBJECT IDENTIFIER(levproj_id) AS SELECT levproj_id,level_no,proj_name,level_name FROM tree_level0;

ตาราง ค9 สคริปท์สำหรับสร้างวิว FACTDATA_VIEW

วิว: FACTDATA_VIEW **ประเภทวิว:** วิวแบบออบเจคต์ **สคริปท์:**
CREATE TYPE factdata_view_type AS OBJECT (factdata_oid NUMBER(10), name VARCHAR2(70), status VARCHAR2(40), created_date DATE); CREATE VIEW factdata_view OF factdata_view_type WITH OBJECT IDENTIFIER(factdata_oid) AS SELECT factdata_oid,name,status,created_date FROM factdata;

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง ค10 สคริปท์สำหรับสร้างวิว APPCLASS_VIEW

วิว: APPCLASS_VIEW
ประเภทวิว: วิวแบบออบเจคต์
สคริปท์:

```
CREATE TYPE appclass_view_type AS OBJECT
(class_id        NUMBER(4),
 class_name      VARCHAR2(70),
 tlev_pk  tlev_view_type,
 db_flag NUMBER(1));
CREATE VIEW appclass_view OF appclass_view_type
     WITH OBJECT IDENTIFIER(class_id) AS
     SELECT class_id,class_name,tlev_pk,db_flag FROM app_class;
```

ตาราง ค11 สคริปท์สำหรับสร้างวิว VERSION_VIEW

วิว: VERSION_VIEW
ประเภทวิว: วิวแบบออบเจคต์
สคริปท์:

```
CREATE OR REPLACE TYPE version_view_type AS OBJECT
(vid             NUMBER(14),
 version_name    VARCHAR2(50),
 app_oid         obj_instance_type,
 version_order   NUMBER(4));
CREATE VIEW version_view OF version_view_type
     WITH OBJECT IDENTIFIER(vid) AS
     SELECT vid,version_name,app_oid,version_order FROM obj_version;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนงานวิจัย

นางสาว จารุวรรณ สุขุมลักษณ์ เกิดวันที่ 9 ธันวาคม พ.ศ.2513 ที่จังหวัดเพชรบุรี สำเร็จการศึกษาระดับปริญญาตรีสาขาคอมพิวเตอร์ จากคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าคุณทหาร ลาดกระบัง ปี พ.ศ. 2536 และเข้าทำงานในตำแหน่งผู้จัดการระบบฐานข้อมูล บริษัท Information Facility Services Co.,Ltd. จากนั้นเปลี่ยนมาทำงานเป็นผู้ควบคุมระบบยูนิกซ์ของธนาคารกรุงเทพฯพาณิชย์การ จำกัด(มหาชน) ขณะที่ทำงานเป็นผู้ช่วยผู้วิเคราะห์ระบบ บริษัทออราเคิล (ประเทศไทย) จำกัด ได้เข้าศึกษาต่อในหลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย