

วิธีการระบุคอล์มน์และตารางของฐานข้อมูลที่ไม่ถูกอ้างอิงในโปรแกรม
ที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์

นายปริรักษ์ หัตถา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

AN APPROACH FOR IDENTIFYING UNREFERENCED DATABASE COLUMNS AND
TABLES IN PROGRAMS USING OBJECT-RELATIONAL MAPPING

Mr. Borirak Hattha

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

วิธีการระบุคอลัมน์และตารางของฐานข้อมูลที่ไม่
ถูกอ้างอิงในโปรแกรม ที่ใช้การแปลงวัตถุเป็น
ตารางความสัมพันธ์

โดย

นายบรรณีย์ หัตถ

สาขาวิชา

วิศวกรรมซอฟต์แวร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

รองศาสตราจารย์ ดร.ชราทิพย์ สุวรรณศาสตร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้แก่นักศึกษานิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

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

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ชราทิพย์ สุวรรณศาสตร์)

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

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

บริษัท หัตถา : วิธีการระบุคอลัมน์และตารางของฐานข้อมูลที่ไม่ถูกอ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์. (AN APPROACH FOR IDENTIFYING UNREFERENCED DATABASE COLUMNS AND TABLES IN PROGRAMS USING OBJECT-RELATIONAL MAPPING) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.ชาราทิพย์ สุวรรณศาสตร์, 143 หน้า.

โปรแกรมต่างๆในองค์กร มีการพัฒนาปรับปรุงเปลี่ยนแปลงอยู่เสมอ เพื่อรองรับความต้องการใหม่ๆทางธุรกิจ ซึ่งการเปลี่ยนแปลงโปรแกรมจะกระทบกับเค้าร่างของฐานข้อมูลด้วย ซึ่งตาราง และคอลัมน์ของฐานข้อมูลบางตัวไม่ได้ถูกอ้างอิงจากโปรแกรมที่พัฒนาแล้ว แต่ไม่มีใครสามารถบอกได้ว่าตาราง และคอลัมน์ใดบ้างที่สามารถลบได้โดยไม่กระทบกับโปรแกรมอื่น ๆ งานวิจัยนี้จึงนำเสนอวิธีการระบุตาราง และคอลัมน์ของฐานข้อมูล โดยวิเคราะห์โปรแกรมทุกโปรแกรมที่ใช้ฐานข้อมูลเดียวกัน เพื่อหาว่าทุกโปรแกรมมีการอ้างอิงตาราง และคอลัมน์ฐานข้อมูลตัวใด เพื่อนำผลลัพธ์ไปเปรียบเทียบกับฐานข้อมูลจริง ซึ่งงานวิจัยนี้แนะนำเฉพาะวิธีที่จะวิเคราะห์โปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ในการติดต่อฐานข้อมูล ซึ่งผลลัพธ์ที่ได้จากวิธีที่นำเสนอในงานวิจัยจะเป็นรายการตาราง และคอลัมน์ที่ไม่มีโปรแกรมใดเลยที่ถูกอ้างอิง

ผู้วิจัยได้พัฒนาเครื่องมือต้นแบบที่ทำงานตามวิธีการที่นำเสนอ โดยพัฒนาเครื่องมือที่สามารถวิเคราะห์โปรแกรมในคลาสของภาษาจาวาที่ใช้เจพีเอ และเทียบกับฐานข้อมูลเพื่อหาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม จากนั้นทดสอบความถูกต้องกับโปรแกรมของธนาคารแห่งหนึ่งเป็นจำนวน 2 โปรแกรม ซึ่งทราบอยู่แล้วว่าแต่ละโปรแกรมอ้างอิงตาราง และคอลัมน์ฐานข้อมูลใด โดยเปรียบเทียบผลลัพธ์ของตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงที่ได้จากเครื่องมือ กับผลลัพธ์ที่คาดหวัง ซึ่งเครื่องมือให้ผลลัพธ์ตรงกับผลลัพธ์ที่คาดหวัง นั้นแสดงให้เห็นว่าเครื่องมือที่ทำงานตามแนวคิดที่นำเสนอให้ผลลัพธ์การทำงานอย่างถูกต้อง

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....
 สาขาวิชา.....วิศวกรรมซอฟต์แวร์.....ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....
 ปีการศึกษา.....2555.....

5470961321 : MAJOR SOFTWARE ENGINEERING

KEYWORDS : DATABASE REFACTORING / OBJECT-RELATIONAL MAPPING / DEAD CODE

BORIRAK HATTHA : AN APPROACH FOR IDENTIFYING UNREFERENCED DATABASE COLUMNS AND TABLES IN PROGRAMS USING OBJECT-RELATIONAL MAPPING. ADVISOR : ASSOC. PROF. TARATIP SUWANNASART, Ph.D., 143 pp.

Programs always have been improved to meet new business requirements, so they may impact database schema. Lots of existing database's tables and columns are not referenced by the programs, but no one can specify correctly which ones can be removed. This thesis proposes a solution to identify unreferenced tables and columns, by analyzing all programs that use same database to find which tables and columns are referenced in these programs, and comparing with database schema information. This thesis proposes only an approach to analyze program using Object-Relational Mapping (ORM) to communicate with database. The result of unreferenced tables and columns are removable ones.

We implemented a prototype tool according to our purposed approach. The tool can analyze Java classes that using JPA and compare with the database to identify which tables and columns have not been referenced in programs. Then we test the implemented prototype tool using two programs from a bank in Thailand. The programs we have already know that which tables and columns have been referenced. The results from the tool are as same as the expected results, showing that our proposed approach has worked correctly.

Department : Computer Engineering Student's Signature.....

Field of Study : Software Engineering Advisor's Signature.....

Academic Year : 2012.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลืออย่างดียิ่งจาก รองศาสตราจารย์ ดร.ชราทิพย์ สุวรรณศาสตร์ ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้สละเวลา ให้คำแนะนำ ให้คำปรึกษา และแนวทางการวิจัย รวมไปถึงความรู้ทางวิชาการต่างๆที่มีประโยชน์ อย่างยิ่งกับวิทยานิพนธ์นี้ อีกทั้งให้ความเอาใจใส่ ให้คำปรึกษาในเรื่องอื่นๆ อาทิ การติดต่อเพื่อไป ต่างประเทศ เพื่อ ไปงานประชุมทางวิชาการ เป็นต้น ทำให้ผู้วิจัยสามารถดำเนินงานวิจัยจนกระทั่ง ประสบความสำเร็จ และมีคุณภาพ

ขอขอบพระคุณ รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ ประธานกรรมการสอบ ผู้ช่วย ศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์ และ ผู้ช่วยศาสตราจารย์ ดร.ภัทรชัย ลลิตโรจน์วงศ์ กรรมการ สอบวิทยานิพนธ์ที่ได้กรุณาสละเวลา ให้คำแนะนำต่างๆเกี่ยวกับการทำวิจัย เพื่อให้วิทยานิพนธ์ ฉบับนี้มีคุณภาพ

ขอขอบพระคุณอาจารย์ทุกๆท่าน ที่ได้ให้ความรู้ คำแนะนำในเรื่องเนื้อหาวิชาการต่างๆ และการทำวิจัย

ขอขอบคุณ เพื่อนๆ พี่ๆ และน้องๆ สำหรับคำปรึกษา คำแนะนำ และเป็นกำลังใจในการทำงานวิจัยตลอดมา

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา และสมาชิกในครอบครัวทุกๆท่าน ที่ให้ สติปัญญา ส่งเสริมสนับสนุนด้านการเรียน และให้กำลังใจแก่ผู้วิจัยเสมอมา

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	3
1.3 ขอบเขตของงานวิจัย.....	4
1.4 ขั้นตอนและวิธีการดำเนินการวิจัย.....	5
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	6
1.6 ลำดับขั้นตอนในการนำเสนองานวิจัย.....	6
1.7 บทความวิชาการที่ได้รับการตีพิมพ์.....	6
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	8
2.1 ทฤษฎีที่เกี่ยวข้อง.....	8
2.1.1 การรีแฟกเตอร์ฐานข้อมูล.....	8
2.1.2 การแปลงวัตถุเป็นตารางความสัมพันธ์.....	12
2.2 งานวิจัย และเครื่องมือที่เกี่ยวข้อง.....	13
2.2.1 เจพีเอ.....	14
2.2.2 โค้ดที่เข้าไม่ถึง.....	17
2.2.3 เครื่องมือแฮมเชดที่เข้าไม่ถึง.....	18
2.2.4 การวิเคราะห์ผลกระทบเมื่อเค้าร่างฐานข้อมูลเปลี่ยน.....	18

	หน้า
2.2.5 ภาษาสืบค้นข้อมูล.....	19
2.2.5.1 ภาษาเจพีคิวแอล.....	19
2.2.5.1 ภาษาเอสคิวแอล.....	20
2.2.5.3 วิธีการใช้งานคำสั่งสืบค้น.....	20
2.2.6 เครื่องมือบีซีอีแอล.....	21
บทที่ 3 วิธีการดำเนินงานวิจัย.....	25
3.1 ภาพรวมของงานวิจัย.....	25
3.1.1 ขั้นตอนการหาตาราง และคอลัมน์ที่โปรแกรมอ้างอิง.....	29
3.2 แนวทางการพัฒนาเครื่องมือต้นแบบ.....	34
3.2.1 ภาพรวมขั้นตอนการทำงานของเครื่องมือ.....	34
3.2.2 ฟังก์ชันของเครื่องมือ.....	35
3.2.3 โครงสร้างของเครื่องมือ.....	50
3.2.4 พฤติกรรมของเครื่องมือ.....	53
บทที่ 4 การพัฒนาเครื่องมือ.....	62
4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ.....	62
4.2 ส่วนต่อประสานกับผู้ใช้ของเครื่องมือ.....	63
4.3 โครงสร้างไฟล์โครงการสำหรับเก็บข้อมูล.....	65
บทที่ 5 การทดสอบเครื่องมือ.....	68
5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ.....	68
5.2 การทดสอบเครื่องมือ.....	69
5.2.1 ทดสอบเครื่องมือกับกรณีทดสอบ.....	69
5.2.2 ทดสอบเครื่องมือกับองค์กรจริง.....	70
5.3 ผลการทดสอบ.....	71
5.3.1 ผลการทดสอบเครื่องมือกับกรณีทดสอบ.....	71
5.3.2 ผลการทดสอบเครื่องมือกับองค์กรจริง.....	72

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	76
6.1 สรุปผลการวิจัย.....	76
6.2 ข้อจำกัดของวิธีที่ทำเสนอ.....	77
6.3 ข้อจำกัดของเครื่องมือต้นแบบ.....	77
6.4 แนวทางการพัฒนาต่อ.....	78
รายการอ้างอิง.....	80
ภาคผนวก.....	82
ภาคผนวก ก รายละเอียดประเภทการรีแฟคเตอร์ และคำสั่งดำเนินการ.....	83
ภาคผนวก ข วิธีการใช้บรรณทัศน์สัจเจพีเอระบุความสัมพันธ์	
การแปลงวัตถุเป็นตารางความสัมพันธ์.....	87
ภาคผนวก ค รายการรหัสคำสั่งเครื่องของภาษาจาวา.....	96
ภาคผนวก ง คำบรรยายยูสเคส.....	106
ภาคผนวก จ ผลการทดสอบเครื่องมือ.....	115
ภาคผนวก ฉ คู่มือการใช้งานเครื่องมือระบุตาราง และคอลัมน์	
ที่ไม่ถูกอ้างอิงในโปรแกรม.....	129
ประวัติผู้เขียนวิทยานิพนธ์.....	143

สารบัญตาราง

	หน้า
ตารางที่ 3.1 หน้าที่ของคลาสในเครื่องมือ.....	52
ตารางที่ 3.2 คำสั่งภาษาเอสคิวแอลสำหรับลบตาราง และคอลัมน์.....	61
ตารางที่ 4.1 โครงสร้างไฟล์โครงการ.....	65
ตารางที่ 5.1 ผลการทดสอบเครื่องมือกับกรณีทดสอบ.....	71
ตารางที่ 5.2 ข้อมูลนำเข้าของเครื่องมือ.....	72
ตารางที่ 5.3 ผลการทดสอบเครื่องมือกับองค์กรจริง.....	73
ตารางที่ ก.1 ประเภทของการรีเฟลคเตอร์ฐานข้อมูล และคำสั่งดำเนินการที่เกี่ยวข้อง กับการ รีเฟลคเตอร์ฐานข้อมูลแต่ละประเภท.....	83
ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา.....	96
ตารางที่ ง.1 คำบรรยายยูสเคสการสร้างโครงการ.....	106
ตารางที่ ง.2 คำบรรยายยูสเคสการเปิดโครงการ.....	107
ตารางที่ ง.3 คำบรรยายยูสเคสการบันทึก.....	107
ตารางที่ ง.4 คำบรรยายยูสเคสการเพิ่มโปรแกรมเพื่อวิเคราะห์.....	108
ตารางที่ ง.5 คำบรรยายยูสเคสการเพิ่มฐานข้อมูลเพื่อวิเคราะห์.....	109
ตารางที่ ง.6 คำบรรยายยูสเคสการลบโปรแกรม.....	110
ตารางที่ ง.7 คำบรรยายยูสเคสการลบฐานข้อมูล.....	110
ตารางที่ ง.8 คำบรรยายยูสเคสการนำเข้าโปรแกรมที่วิเคราะห์แล้ว.....	111
ตารางที่ ง.9 คำบรรยายยูสเคสการนำออกโปรแกรมที่วิเคราะห์แล้ว.....	111
ตารางที่ ง.10 คำบรรยายยูสเคสการวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง.....	112
ตารางที่ ง.11 คำบรรยายยูสเคสการหาคำสั่งสืบค้นที่ได้รับผลกระทบ.....	113
ตารางที่ ง.12 คำบรรยายยูสเคสการสร้างคำสั่งลบตาราง และคอลัมน์.....	114
ตารางที่ จ.1 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์แบบง่าย.....	115
ตารางที่ จ.2 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์ แบบใช้วัตถุแบบฝังตัว.....	116
ตารางที่ จ.3 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์ แบบที่มีความสัมพันธ์ระหว่างคลาส.....	117

ตารางที่ จ.4 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์ แบบที่สัมพันธ์มากกว่า 1 ตาราง.....	118
ตารางที่ จ.5 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์ แบบความสัมพันธ์สี่ทิศทางคุณสมบัติ.....	119
ตารางที่ จ.6 รายการกรณีทดสอบการหาคลาสเอนติตี้ และแอตทริบิวต์ของเอนติตี้ที่เข้าไม่ถึง	119
ตารางที่ จ.7 รายการกรณีทดสอบการหาคำสั่งสืบค้นในโปรแกรม.....	120
ตารางที่ จ.8 รายการกรณีทดสอบการหาตาราง และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบค้น ภาษาเจพีควีแอล.....	123
ตารางที่ จ.9 รายการกรณีทดสอบการหาตาราง และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบค้น ภาษาเอสควีแอล.....	124
ตารางที่ จ.10 รายการกรณีทดสอบการรวมผลลัพธ์รายการตาราง 2 รายการ.....	125
ตารางที่ จ.11 รายการกรณีทดสอบการหารายการตารางที่ไม่ถูกอ้างอิง.....	126

สารบัญภาพ

	หน้า
ภาพที่ 2.1 เอนติตีของฐานข้อมูลบุคคลก่อนรีแฟคเตอร์.....	9
ภาพที่ 2.2 เอนติตีของฐานข้อมูลบุคคลหลังจากรีแฟคเตอร์แล้ว.....	9
ภาพที่ 2.3 กรณียของการรีแฟคเตอร์ฐานข้อมูลที่ยากที่สุด.....	11
ภาพที่ 2.4 คลาสที่ใช้จัดเก็บข้อมูลบุคคล และหมายเลขโทรศัพท์.....	12
ภาพที่ 2.5 ตารางฐานข้อมูลที่ใช้จัดเก็บข้อมูลบุคคล และหมายเลขโทรศัพท์.....	13
ภาพที่ 2.6 การแปลงเอนติตีคลาส เป็นตารางฐานข้อมูล ด้วยเจพีเอ.....	14
ภาพที่ 2.7 การแปลงเอนติตีคลาสที่มีความสัมพันธ์แบบหนึ่งต่อหลายเป็นตารางฐานข้อมูล ด้วยเจพีเอ.....	15
ภาพที่ 2.8 การใช้คำสั่ง แทรก เปลี่ยนแปลง และลบข้อมูล ตามลำดับ.....	19
ภาพที่ 2.9 ตัวอย่างการสืบค้นข้อมูลด้วยภาษาเจพีควีแอล.....	19
ภาพที่ 2.10 ตัวอย่างการสืบค้นข้อมูลด้วยภาษาเอสควีแอล.....	20
ภาพที่ 2.11 การใช้คำสั่งสืบค้นผ่าน เอนติตีเมเนเจอร์.....	20
ภาพที่ 2.12 การตั้งชื่อให้แก่คำสั่งสืบค้นในคลาสเอนติตี.....	21
ภาพที่ 2.13 ซอร์สโค้ดก่อนคอมไพล์.....	21
ภาพที่ 2.14 รหัสคำสั่งที่เครื่องมือบีซีอีแอลอ่านได้ จากไฟล์คลาสที่คอมไพล์แล้ว.....	22
ภาพที่ 2.15 ตัวอย่างรหัสคำสั่งที่อ่านได้ จากการใช้คำสั่งสืบค้นภาษาเจพีควีแอล.....	22
ภาพที่ 2.16 ตัวอย่างรหัสคำสั่งที่อ่านได้ จากการใช้คำสั่งสืบค้นภาษาเอสควีแอล.....	23
ภาพที่ 2.17 ตัวอย่างรหัสคำสั่งที่อ่านได้ จากการใช้คำสั่งสืบค้นแบบระบุชื่อคำสั่งสืบค้น.....	23
ภาพที่ 3.1 ภาพรวมของการค้นหาโครงสร้างของฐานข้อมูลที่โปรแกรมไม่ได้ใช้งาน.....	25
ภาพที่ 3.2 ขั้นตอนการหาตาราง และคอลัมน์ที่โปรแกรมใช้งาน โดยใช้เทคนิค การแปลงวัตถุเป็นตารางความสัมพันธ์ และโค้ดที่เข้าไม่ถึง.....	29
ภาพที่ 3.3 บรรณนิพจน์ @Entity ที่ระบุให้กับคลาสเอนติตี.....	29
ภาพที่ 3.4 ภาพรวมการทำงานของเครื่องมือต้นแบบ.....	34
ภาพที่ 3.5 แผนภาพยูสเคสฟังก์ชันงานของเครื่องมือ.....	36
ภาพที่ 3.6 แผนภาพยูสเคสระบบย่อยจัดการ โครงการ.....	37
ภาพที่ 3.7 แผนภาพยูสเคสระบบย่อยจัดการ โปรแกรม และฐานข้อมูล.....	37
ภาพที่ 3.8 แผนภาพยูสเคสระบบย่อยวิเคราะห์ตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง.....	38

ภาพที่ 3.9 แผนภาพกิจกรรมขั้นตอนการสร้างโครงการ.....	39
ภาพที่ 3.10 แผนภาพกิจกรรมขั้นตอนการเปิดโครงการ.....	40
ภาพที่ 3.11 แผนภาพกิจกรรมขั้นตอนการบันทึกโครงการ.....	41
ภาพที่ 3.12 แผนภาพกิจกรรมขั้นตอนการเพิ่มโปรแกรมเข้าในโครงการ.....	42
ภาพที่ 3.13 แผนภาพกิจกรรมขั้นตอนการเพิ่มฐานข้อมูลเข้าในโครงการ.....	43
ภาพที่ 3.14 แผนภาพกิจกรรมขั้นตอนการลบโปรแกรมออกจากโครงการ.....	44
ภาพที่ 3.15 แผนภาพกิจกรรมขั้นตอนการลบฐานข้อมูลออกจากโครงการ.....	45
ภาพที่ 3.16 แผนภาพกิจกรรมขั้นตอนการนำเข้าโปรแกรมที่วิเคราะห์แล้ว.....	46
ภาพที่ 3.17 แผนภาพกิจกรรมขั้นตอนการนำออกโปรแกรมที่วิเคราะห์แล้ว.....	47
ภาพที่ 3.18 แผนภาพกิจกรรมขั้นตอนการวิเคราะห์เพื่อระบุตารางและคอลัมน์ ที่ไม่ถูกอ้างอิงในโปรแกรม.....	48
ภาพที่ 3.19 แผนภาพกิจกรรมขั้นตอนการวิเคราะห์หาคำสั่งสืบค้นที่ได้รับผลกระทบ.....	49
ภาพที่ 3.20 แผนภาพกิจกรรมขั้นตอนการสร้างคำสั่งสำหรับลบตารางและคอลัมน์.....	50
ภาพที่ 3.21 แผนภาพคลาสแสดงโครงสร้างของเครื่องมือ.....	51
ภาพที่ 3.22 แผนภาพลำดับพฤติกรรมเครื่องมือขณะสร้างโครงการ.....	54
ภาพที่ 3.23 แผนภาพลำดับพฤติกรรมเครื่องมือขณะเปิดโครงการ.....	54
ภาพที่ 3.24 แผนภาพลำดับพฤติกรรมเครื่องมือขณะบันทึกโครงการ.....	55
ภาพที่ 3.25 แผนภาพลำดับพฤติกรรมเครื่องมือขณะเพิ่มโปรแกรมเพื่อวิเคราะห์.....	55
ภาพที่ 3.26 แผนภาพลำดับพฤติกรรมเครื่องมือขณะเพิ่มฐานข้อมูลเพื่อวิเคราะห์.....	56
ภาพที่ 3.27 แผนภาพลำดับแสดงพฤติกรรมเครื่องมือขณะลบโปรแกรม.....	56
ภาพที่ 3.28 แผนภาพลำดับพฤติกรรมเครื่องมือขณะลบฐานข้อมูล.....	57
ภาพที่ 3.29 แผนภาพลำดับพฤติกรรมเครื่องมือขณะนำเข้าโปรแกรมที่วิเคราะห์แล้ว.....	57
ภาพที่ 3.30 แผนภาพลำดับพฤติกรรมเครื่องมือขณะนำออกโปรแกรมที่วิเคราะห์แล้ว.....	58
ภาพที่ 3.31 แผนภาพลำดับพฤติกรรมเครื่องมือขณะวิเคราะห์หาตาราง และคอลัมน์ ที่ไม่ถูกอ้างอิง.....	59
ภาพที่ 3.32 แผนภาพลำดับพฤติกรรมเครื่องมือขณะหาคำสั่งสืบค้นที่ได้รับผลกระทบ.....	60
ภาพที่ 3.33 แผนภาพลำดับพฤติกรรมเครื่องมือขณะสร้างคำสั่งลบตาราง และคอลัมน์.....	61
ภาพที่ 4.1 ส่วนต่อประสานกับผู้ใช้ของเครื่องมือ.....	63
ภาพที่ 4.2 เมนูต่าง ๆ ในเครื่องมือ.....	63

ภาพที่ 5.1 เวลาที่เครื่องมือใช้งานในแต่ละขั้นตอน.....	73
ภาพที่ 5.2 เปรียบเทียบเวลาที่ใช้ในการวิเคราะห์โปรแกรมระหว่างโปรแกรมขนาดเล็ก และโปรแกรมขนาดใหญ่.....	74
ภาพที่ ข.1 การใช้บรรทัดที่ระบุคลาสให้เป็นเอนติตี้คลาส.....	87
ภาพที่ ข.2 การใช้บรรทัดที่ระบุกำหนดแอตทริบิวต์ที่ไม่ต้องการจัดเก็บลงฐานข้อมูล.....	87
ภาพที่ ข.3 การใช้บรรทัดที่ระบุกำหนดชื่อตาราง และคอลัมน์ฐานข้อมูล.....	88
ภาพที่ ข.4 การใช้บรรทัดที่ระบุการใช้แอตทริบิวต์แบบวัตถุแบบฝังตัว.....	88
ภาพที่ ข.5 การใช้บรรทัดที่ระบุ@AttributeOverridesกำหนดชื่อคอลัมน์ใหม่ ให้วัตถุแบบฝังตัว.....	89
ภาพที่ ข.6 การใช้บรรทัดที่ระบุ@AttributeOverrideกำหนดชื่อคอลัมน์ใหม่ ให้วัตถุแบบฝังตัว.....	90
ภาพที่ ข.7 การใช้บรรทัดที่ระบุระบุคีย์หลัก.....	90
ภาพที่ ข.8 การใช้บรรทัดที่ระบุระบุคีย์ผสมโดยใช้ @EmbeddedId.....	91
ภาพที่ ข.9 การใช้บรรทัดที่ระบุระบุคีย์ผสมโดยใช้ @IdClass.....	91
ภาพที่ ข.10 การกำหนดเอนติตี้คลาสให้สัมพันธ์กับตารางมากกว่า 1 ตาราง.....	92
ภาพที่ ข.11 การสร้างความสัมพันธ์แบบหนึ่งต่อหลาย.....	93
ภาพที่ ข.12 การสร้างความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....	93
ภาพที่ ข.13 แผนภาพอีอาร์ตารางสำหรับเก็บข้อมูลที่มีความสัมพันธ์แบบหลายต่อหลาย.....	94
ภาพที่ ข.14 การสร้างความสัมพันธ์แบบหลายต่อหลาย.....	95
ภาพที่ ฉ.1 ส่วนประกอบต่างๆของเครื่องมือ.....	129
ภาพที่ ฉ.2 วิธีการเรียกใช้ฟังก์ชันสร้างโครงการ.....	130
ภาพที่ ฉ.3 หน้าจอรายละเอียดข้อมูลโครงการที่จะสร้าง.....	130
ภาพที่ ฉ.4 วิธีการเรียกใช้ฟังก์ชันบันทึกโครงการ.....	131
ภาพที่ ฉ.5 วิธีการเรียกใช้ฟังก์ชันเปิดโครงการ.....	131
ภาพที่ ฉ.6 วิธีการเรียกใช้ฟังก์ชันเพิ่มโปรแกรมเพื่อวิเคราะห์.....	132
ภาพที่ ฉ.7 หน้าจอรายละเอียดข้อมูลโปรแกรมที่จะเพิ่มเข้าสู่โครงการ.....	133
ภาพที่ ฉ.8 วิธีการเรียกใช้ฟังก์ชันเพิ่มฐานข้อมูลเพื่อวิเคราะห์.....	134
ภาพที่ ฉ.9 หน้าจอรายละเอียดข้อมูลฐานข้อมูลที่จะเพิ่มเข้าสู่โครงการ.....	134
ภาพที่ ฉ.10 วิธีการเรียกใช้ฟังก์ชันลบโปรแกรม หรือฐานข้อมูล.....	135

ภาพที่ ฉ.11 วิธีการเรียกใช้ฟังก์ชันนำออกโปรแกรมที่วิเคราะห์แล้ว.....	135
ภาพที่ ฉ.12 วิธีการเรียกใช้ฟังก์ชันนำเข้าโปรแกรมที่วิเคราะห์แล้ว.....	136
ภาพที่ ฉ.13 วิธีการเรียกใช้ฟังก์ชันวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง.....	136
ภาพที่ ฉ.14 หน้าจอแสดงรายการ โปรแกรมให้ผู้เลือกใช้ก่อนทำการวิเคราะห์.....	137
ภาพที่ ฉ.15 หน้าจอแสดงความคืบหน้าในระหว่างการวิเคราะห์.....	137
ภาพที่ ฉ.16 หน้าจอแสดงรายละเอียดผลลัพธ์ที่ได้จากการวิเคราะห์โปรแกรม.....	138
ภาพที่ ฉ.17 หน้าจอแสดงรายละเอียดรายการตาราง และคอลัมน์ของฐานข้อมูล.....	139
ภาพที่ ฉ.18 หน้าจอแสดงรายละเอียดรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม.....	140
ภาพที่ ฉ.19 หน้าจอคำสั่งสืบค้นที่ได้รับผลกระทบ.....	141
ภาพที่ ฉ.20 หน้าจอแสดงคำสั่งเอสคิวแอลมาตรฐานสำหรับลบตาราง และคอลัมน์.....	142

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ข้อมูลในระบบสารสนเทศของแต่ละองค์กร เป็นสิ่งที่มีความจำเป็น มีมูลค่า มีความสำคัญต่อองค์กรเป็นอย่างยิ่ง ซึ่งข้อมูลที่มีความสัมพันธ์กันสามารถนำไปสู่ความรู้ต่างๆ ขององค์กรได้ ซึ่งสามารถนำไปใช้ประโยชน์ทางการตลาด และด้านอื่นๆ ได้ [1] ตั้งแต่อดีตจนถึงปัจจุบันข้อมูลในองค์กรมีแนวโน้มที่จะเพิ่มขึ้นเรื่อยๆ นับวันยิ่งจะมีปริมาณข้อมูลที่มากขึ้นเรื่อยๆ ซึ่งส่งผลให้ใช้พื้นที่สำหรับเก็บข้อมูลมากขึ้น [2,3] การบริหารจัดการข้อมูลที่ไม่ดีจะทำให้ใช้ประโยชน์จากทรัพยากรที่มีได้อย่างไม่เต็มที่

ในองค์กรต่างๆ จะมีการปรับปรุงพัฒนาโปรแกรมอยู่เสมอ ซึ่งการปรับปรุงพัฒนาแต่ละครั้งย่อมต้องมีการใช้งานฐานข้อมูลที่แตกต่างกัน เช่น โปรแกรมใหม่ อาจต้องเพิ่มตาราง หรือคอลัมน์ใหม่ๆ แต่ในทางกลับกัน อาจมีตาราง หรือคอลัมน์บางตัว ที่โปรแกรมใหม่ไม่ได้อ้างอิงถึงแล้ว แต่ผู้พัฒนาก็เลือกที่จะไม่ลบตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึงแล้ว ทั้งนี้เนื่องจากผู้พัฒนาเองก็ไม่ทราบถึงผลกระทบที่จะเกิดขึ้นกับโปรแกรม อื่นๆ ถ้าลบตาราง หรือคอลัมน์ออกจากฐานข้อมูล [4]

เนื่องจากสาเหตุดังกล่าวจึงเป็นที่มาของปัญหาอย่างหนึ่งที่พบมากในองค์กร คือ ฐานข้อมูลมีตาราง และคอลัมน์เป็นจำนวนมากที่โปรแกรมไม่ได้อ้างอิง แต่ยังคงเก็บไว้อยู่ในฐานข้อมูล ซึ่งจะทำให้ฐานข้อมูลมีขนาดใหญ่กว่าความจำเป็น ซึ่งพื้นที่เหล่านี้สามารถบริหารโดยนำไปใช้จัดเก็บข้อมูลที่มีความจำเป็นอื่นๆ เพื่อใช้ประโยชน์ของพื้นที่สำหรับจัดเก็บข้อมูลให้คุ้มค่าที่สุด

ฐานข้อมูลเมื่อมีขนาดข้อมูลที่ใหญ่ขึ้น ยังเป็นปัจจัยที่มีผลกระทบต่อการทำงานต่างๆ ได้แก่

- ประสิทธิภาพการเข้าถึงข้อมูลในฐานข้อมูล คือ ฐานข้อมูลที่มีขนาดใหญ่กว่า จะใช้เวลาเข้าถึงข้อมูลได้ช้ากว่าขนาดข้อมูลที่ขนาดเล็กกว่าที่อยู่บนสภาพแวดล้อมอื่นๆ ที่เหมือนกัน [5,6] ซึ่งประสิทธิภาพเป็นสิ่งที่องค์กรต่างๆ ให้ความสำคัญเป็นอย่างมาก เนื่องจากผู้ใช้งานต้องการทำงานได้อย่างรวดเร็ว
- เวลาที่ใช้ในการสำรองข้อมูล และกู้คืนข้อมูล จะใช้เวลามากกว่าการสำรองข้อมูลเมื่อฐานข้อมูลมีขนาดน้อยกว่า [7] ซึ่งในองค์กร การสำรองข้อมูลมีความสำคัญมาก เนื่องจากต้องสำรองข้อมูลป้องกันกรณีเกิดความผิดพลาดของระบบ หรือข้อมูลเสียหาย เพื่อที่จะทำให้กู้คืนข้อมูลได้
- พื้นที่ใช้สำหรับการสำรองข้อมูลมีมากขึ้น เมื่อข้อมูลในฐานข้อมูลมีปริมาณที่ใหญ่ขึ้น การสำรองข้อมูลจึงจำเป็นต้องใช้พื้นที่ในการสำรองข้อมูลมากขึ้นด้วย [8] ซึ่งองค์กรจะทำการสำรองข้อมูลอยู่เป็นประจำ ข้อมูลจึงมีขนาดใหญ่ขึ้น

นอกจากนี้การลบตาราง และคอลัมน์ของฐานข้อมูลที่ไม่ได้ใช้งานออก ยังส่งผลดีต่อการบำรุงรักษาระบบด้วย คือปรับปรุงฐานข้อมูล ให้มีเฉพาะตาราง และคอลัมน์ที่จำเป็น และใช้งานจริง ทำให้ผู้พัฒนาระบบ และผู้จัดการฐานข้อมูลทำความเข้าใจเค้าร่าง (Schema) ฐานข้อมูลได้ดีขึ้น ไม่เกิดความสับสน และเกิดคำถามในภายหลังว่าคอลัมน์ หรือตารางในฐานข้อมูลนี้มีไว้เพื่อประโยชน์อะไร มีความสำคัญอย่างไร ทำไมถึงต้องมีคอลัมน์หรือตารางนี้ อันเป็นการนำข้อดีของการทำรีแฟคเตอร์ (Refactoring) ในส่วนของการช่วยให้ฐานข้อมูลง่ายต่อการอ่าน และการทำความเข้าใจความหมายมาใช้ประโยชน์ [9]

ในปัจจุบันมีวิธีการระบุตาราง และคอลัมน์ของฐานข้อมูลที่ไม่ได้ใช้งาน ที่ผู้เชี่ยวชาญได้นำเสนออยู่ด้วยกันหลายวิธี ได้แก่

1. การใช้คำสั่งเฉพาะที่มีในระบบจัดการฐานข้อมูลของออร์اكل (Oracle DBMS) ค้นคืนข้อมูลคอลัมน์ครรชนี (Index) ที่ไม่ได้ถูกใช้งานได้ โดยจะสามารถค้นคืนได้เฉพาะคอลัมน์ครรชนีเท่านั้น [10]

2. การใช้คำสั่งเฉพาะที่มีในระบบจัดการฐานข้อมูลของอราเคิล ค้นคืนรายชื่อคอลัมน์ที่ถูกระบุไว้ก่อนหน้านี้ว่าเป็นคอลัมน์ที่ไม่ได้ใช้งาน [10]
3. การวิเคราะห์จากคำสั่งที่ใช้งานกับฐานข้อมูล โดยหาคำสั่งต่างๆ จากแฟ้มลงบันทึกเข้าออก (Log File) [11] และนำคำสั่งเหล่านี้เฉพาะช่วงเวลาที่น่าสนใจมาวิเคราะห์หาตารางและคอลัมน์ของฐานข้อมูลที่มีการใช้งาน ซึ่งตาราง และคอลัมน์นอกเหนือจากนี้จะถือว่าไม่ได้ถูกใช้งาน

วิธีการดังกล่าว ในวิธีที่ 1 และวิธีที่ 2 เป็นวิธีที่ต้องทราบโดยคร่าวๆ แล้วว่าคอลัมน์ใดบ้างที่มีแนวโน้มที่จะไม่ใช้งาน จึงเน้นให้ความสนใจเฉพาะกับคอลัมน์นั้นๆ ส่วนในวิธีที่ 3 จำเป็นต้องเลือกช่วงเวลาที่เหมาะสม เนื่องจากฟังก์ชันบางฟังก์ชันผู้ใช้ใช้งานน้อย ซึ่งถ้าเลือกช่วงเวลาผิดพลาดเป็นช่วงที่ฟังก์ชันดังกล่าวไม่ได้เรียกใช้งาน จะทำให้ผลลัพธ์ที่วิเคราะห์ออกมาผิดได้ นอกจากนี้ทั้งวิธีที่ 1 และวิธีที่ 3 ยังจำเป็นต้องให้ฐานข้อมูลใช้งานแล้ว เพื่อให้ฐานข้อมูลมีสถิติเก็บไว้วิเคราะห์

งานวิจัยนี้จึงนำเสนอวิธีการระบุตาราง และคอลัมน์ของฐานข้อมูลที่โปรแกรมไม่ได้อ้างอิงถึง โดยจะวิเคราะห์โปรแกรมทุกโปรแกรมที่ใช้งานฐานข้อมูลเดียวกัน เพื่อให้ทราบถึงตารางและคอลัมน์ทุกตัวที่โปรแกรมทุกโปรแกรมอ้างอิง เนื่องจากในโปรแกรมหนึ่งๆ อาจใช้งานเพียงแค่บางคอลัมน์ในตาราง และอีกโปรแกรมหนึ่งอาจใช้งานคอลัมน์ที่เหลือในตารางเดียวกัน และเปรียบเทียบผลลัพธ์ที่วิเคราะห์ได้กับฐานข้อมูลจริงว่ามีตาราง และคอลัมน์ใดบ้างที่โปรแกรมไม่ได้อ้างอิง เนื่องจากปัญหาที่พบคือเค้าร่างของฐานข้อมูลมีตาราง และคอลัมน์เกินกว่าที่ใช้งานจริง จึงพิจารณาจากโปรแกรมเป็นหลัก โดยในวิธีที่นำเสนอในงานวิจัยนี้จะใช้เทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์เป็นหลัก (Object-Relational Mapping) ซึ่งเป็นเทคนิคที่ใช้กันอย่างแพร่หลายในปัจจุบัน เนื่องจากทำให้พัฒนาโปรแกรมในส่วนที่เชื่อมต่อกับฐานข้อมูลได้ง่าย อีกทั้งเทคนิคนี้ยังช่วยระบุขอบเขตตาราง และคอลัมน์ที่โปรแกรมใช้งานได้อีกด้วย

1.2 วัตถุประสงค์ของการวิจัย

เพื่อเสนอวิธีการและพัฒนาเครื่องมือ สำหรับหาตารางและคอลัมน์ของฐานข้อมูลที่ไม่ได้ถูกอ้างอิงถึงในโปรแกรมที่ใช้ เจพีเอ ในการติดต่อฐานข้อมูล

1.3 ขอบเขตของงานวิจัย

1. วิธีการที่นำเสนอจะวิเคราะห์ตาราง และคอลัมน์ฐานข้อมูล ที่ไม่ได้ถูกใช้งานจากโปรแกรมภาษาจาวาที่ใช้เจพีเอเวอร์ชัน 2.0 ขึ้นไป โดยใช้บรรณนิทัศน์ (Annotation) ในการระบุการแปลงระหว่างวัตถุ กับตารางความสัมพันธ์เท่านั้น
2. เครื่องมือที่พัฒนาขึ้น จะใช้เทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์ จึงสามารถวิเคราะห์โปรแกรมที่เชื่อมต่อกับฐานข้อมูล โดยการใช้อินติตีเมเนเจอร์ และเจพีเอ เท่านั้น เครื่องมือนี้ไม่สามารถวิเคราะห์โปรแกรมที่ใช้วิธีอื่นๆ ในการเชื่อมต่อกับฐานข้อมูล
3. งานวิจัยนี้ โปรแกรมที่นำมาวิเคราะห์จะต้องใช้งานฐานข้อมูล เป็นฐานข้อมูลเชิงสัมพันธ์เท่านั้น
4. เครื่องมือรองรับการหาคำสั่งสืบค้นที่เป็นคำสั่งสืบค้นแบบคงที่ (Static Query) ที่ใช้งานกับเมธอดของอินติตีเมเนเจอร์เท่านั้น คือคำสั่ง จะต้องเป็นข้อมูลประเภทสายอักขระ (String) ที่มีข้อความคงที่เพียงหนึ่งข้อความ ไม่มีการเชื่อมข้อความมากกว่าหนึ่งข้อความเข้าด้วยกัน (Concatenation String)
5. เครื่องมือจะวิเคราะห์โปรแกรมที่มาจากไฟล์คลาส (*.class) เท่านั้น นั่นคือจะไม่พิจารณาโค้ดที่อยู่ในหน้าเว็บ อาทิ ไฟล์เจเอสพี (*.jsp) เป็นต้น
6. เครื่องมือที่พัฒนาขึ้นไม่สามารถวิเคราะห์โปรแกรมที่ใช้งานสตอร์โปรซีเจอร์ (Stored procedure)
7. เครื่องมือที่พัฒนาขึ้นไม่สามารถวิเคราะห์โปรแกรมที่ใช้งานตารางแบบทรศนะ (View)
8. เครื่องมือที่พัฒนาขึ้นไม่มีตัวไคร์ฟเวอร์ สำหรับเชื่อมต่อฐานข้อมูลให้ ผู้ใช้งาน เครื่องมือ จำเป็นต้องมีไคร์ฟเวอร์ไว้ใช้สำหรับเชื่อมต่อฐานข้อมูลเอง
9. นำเครื่องมือที่พัฒนาขึ้นนี้ ไปทดลองหาความถูกต้องกับโปรแกรมในองค์กร 2 โปรแกรม

1.4 ขั้นตอนและวิธีการดำเนินการวิจัย

1. ศึกษางานที่เกี่ยวข้อง
 - 1.1 ศึกษาวิธีการแปลงของเจพีเอ
 - 1.2 ศึกษาวิธีการใช้งาน com.mebigfatguy.deadmethods
 - 1.3 ศึกษาการใช้งานบีซีอีแอล (BCEL)
 - 1.4 ศึกษาการเขียนคำสั่งสืบค้นภาษาเจพีคิวแอล (JPQL)
 - 1.5 ศึกษาการเขียนคำสั่งสืบค้นภาษาเอสคิวแอล (SQL)
2. ออกแบบเครื่องมือ
 - 2.1 ออกแบบส่วนต่อประสานกับผู้ใช้
 - 2.2 ออกแบบโครงสร้างการเก็บข้อมูล
 - 2.3 ออกแบบโครงสร้างเครื่องมือ
3. พัฒนาเครื่องมือ
 - 3.1 พัฒนาส่วนต่อประสานผู้ใช้
 - 3.2 พัฒนาฟังก์ชันวิเคราะห์โค้ดโปรแกรม
 - 3.3 พัฒนาฟังก์ชันการหาเค้าร่างฐานข้อมูล
 - 3.4 พัฒนาฟังก์ชันการรวม และการหาผลต่างของไฟล์ผลลัพธ์
 - 3.5 พัฒนาฟังก์ชันสร้างเอสคิวแอลผลลัพธ์
4. ทดสอบเครื่องมือ
 - 4.1 ทดสอบฟังก์ชันวิเคราะห์โค้ดโปรแกรม
 - 4.2 ทดสอบฟังก์ชันการหาเค้าร่างฐานข้อมูล
 - 4.3 ทดสอบฟังก์ชันการรวม และการหาผลต่างของไฟล์ผลลัพธ์
 - 4.4 ทดสอบฟังก์ชันสร้างเอสคิวแอลผลลัพธ์
 - 4.5 ทดสอบเครื่องมือโดยรวม
5. ปรับปรุงแก้ไขเครื่องมือ
6. สรุปและประเมินผลการทดสอบ
7. จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เครื่องมือที่พัฒนาขึ้นสามารถลดเวลาในส่วนของการบำรุงรักษาระบบ เนื่องจากใช้เวลาในการค้นหาตาราง และคอลัมน์ที่โปรแกรมไม่ได้ใช้งานน้อยลง
2. ค้นหาตาราง และคอลัมน์ที่ไม่ได้ใช้งาน ภายหลัง หรือ ระหว่างขั้นตอนการพัฒนาโปรแกรม เพื่อรองรับการพัฒนาแบบอไจล์ (Agile) ที่มีการเปลี่ยนแปลงโปรแกรม และเค้าร่างฐานข้อมูลอยู่ตลอดเวลา
3. ช่วยลดปริมาณข้อมูลจากฐานข้อมูลที่มีเกินความจำเป็นต่อการใช้งาน
4. ช่วยให้ฐานข้อมูลเป็นระเบียบมากยิ่งขึ้น เนื่องจากโครงสร้างการเก็บข้อมูลในฐานข้อมูล มีข้อมูลเฉพาะที่ใช้งานอยู่จริง จึงง่ายต่อการอ่านเพื่อทำความเข้าใจ

1.6 ลำดับขั้นตอนในการนำเสนองานวิจัย

วิทยานิพนธ์เล่มนี้แบ่งเนื้อหาออกเป็น 6 บทดังต่อไปนี้ บทที่ 1 บทนำ กล่าวถึงที่มา และความสำคัญของปัญหา รวมไปถึง วัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย ขั้นตอนและวิธีการวิจัย ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย และบทความทางวิชาการที่ได้รับการตีพิมพ์ บทที่ 2 จะกล่าวถึงทฤษฎี งานวิจัยต่างๆที่เกี่ยวข้อง รวมไปถึงเครื่องมือต่างๆที่ใช้ ในบทที่ 3 จะกล่าวถึงวิธีการดำเนินงานวิจัย ได้แก่ แนวความคิดของงานวิจัย และการออกแบบเครื่องมือต้นแบบ บทที่ 4 จะกล่าวถึงการพัฒนาเครื่องมือ บทที่ 5 จะกล่าวถึงการทดลอง ทั้งการทดลองระดับหน่วย และการทดลองจริงในองค์กร และในบทที่ 6 จะกล่าวถึงข้อจำกัดของงานวิจัย และแนวทางการพัฒนาต่อไปในอนาคต

1.7 บทความวิชาการที่ได้รับการตีพิมพ์

ในการวิจัยนี้ ผู้วิจัยมีผลงานที่ได้รับการตีพิมพ์ทางวิชาการร่วมกับคณะผู้วิจัย ทั้งในระดับชาติ และระดับนานาชาติ รวม 2 บทความ ได้แก่

- 1) บทความทางวิชาการเรื่อง “เครื่องมือระบุตารางและคอลัมน์ของฐานข้อมูลที่ ไม่ได้ถูกอ้างอิงในโปรแกรมที่ใช้เจพีเอ” ซึ่งได้รับคัดเลือกเพื่อนำเสนอ และตีพิมพ์ในงาน “การประชุมวิชาการระดับประเทศด้านเทคโนโลยีสารสนเทศ ครั้งที่ 5 : NCIT 2013” ระหว่างวันที่ 26-27 กุมภาพันธ์ 2556 ณ โรงแรมยูนิโค แกรนด์ เซนดารา ชะอำ อำเภอหัวหิน จังหวัดประจวบคีรีขันธ์ ประเทศไทย

- 2) บทความทางวิชาการเรื่อง “AN APPROACH FOR IDENTIFYING UNREFERENCED DATABASE COLUMNS AND TABLES IN PROGRAMS USING OBJECT-RELATIONAL MAPPING” ซึ่งได้รับคัดเลือกเพื่อนำเสนอ และตีพิมพ์ในงาน “The 2nd International Conference on Electrical Engineering and Computer Sciences : EECS 2013” ระหว่างวันที่ 15-17 มีนาคม 2556 ณ โรงแรมโทชิเซนเตอร์โฮเทล เมืองโตเกียว ประเทศญี่ปุ่น

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

สำหรับการนำเสนอวิธีการระบุตาราง และคอลัมน์ของฐานข้อมูลที่โปรแกรมไม่ได้อ้างอิงถึง เพื่อที่จะลบตาราง และคอลัมน์เหล่านั้นออกจากฐานข้อมูล เป็นหนึ่งในประเภทของการรีแฟคเตอร์ฐานข้อมูล เพื่อเพิ่มประสิทธิภาพของฐานข้อมูล อีกทั้งยังช่วยให้ฐานข้อมูลง่ายต่อการทำความเข้าใจ โดยวิธีที่นำเสนอในงานวิจัยนี้ใช้หลักการของการแปลงวัตถุเป็นตารางความสัมพันธ์ (Object-Relational Mapping) เป็นหลักการหลักเพื่อใช้ช่วยระบุว่าโปรแกรมอ้างอิงถึงตาราง และคอลัมน์ใดในฐานข้อมูล ซึ่งมีรายละเอียดทฤษฎีที่เกี่ยวข้องดังนี้

2.1.1 การรีแฟคเตอร์ฐานข้อมูล (Database Refactoring)

การรีแฟคเตอร์ฐานข้อมูล เป็นแนวคิดที่สก๊อต แอมเบล (Scott W. Ambler) ได้นำเสนอไว้ [4] โดยการปรับเปลี่ยนเค้าร่าง (Schema) ของฐานข้อมูลที่มีอยู่แล้ว ให้มีประสิทธิภาพมากยิ่งขึ้น และให้ฐานข้อมูลง่ายต่อการอ่านทำความเข้าใจ

การรีแฟคเตอร์ฐานข้อมูล แตกต่างกับการรีแฟคเตอร์โค้ดโปรแกรมทั่วไป คือ การรีแฟคเตอร์ซอร์สโค้ด จะสนใจเพียงแต่ให้โปรแกรมสามารถทำงานได้เหมือนเดิม นั่นคือสนใจเพียงแค่พฤติกรรมการทำงานของโปรแกรม (Behavioral semantics) เช่นการเปลี่ยนชื่อฟังก์ชันของโปรแกรม ทุกๆส่วนของโปรแกรมที่เรียกใช้งานฟังก์ชันที่ชื่อเปลี่ยนไป ก็ต้องเปลี่ยนชื่อฟังก์ชันตามด้วย แต่การรีแฟคเตอร์ฐานข้อมูลจะยากกว่าการรีแฟคเตอร์ซอร์สโค้ดคือ จำเป็นต้องคงไว้ซึ่งความถูกต้องให้โปรแกรมสามารถทำงานได้เหมือนเดิม และข้อมูลเดิมที่มีจะตรงตามความหมายของข้อมูลเดิมด้วย (Informational semantics) ดังเช่นตัวอย่างที่ สก๊อต แอมเบล ยกตัวอย่างดังนี้ [4]

การแยกคอลัมน์ของฐานข้อมูลเป็นสองคอลัมน์ เดิมมีตาราง PERSON ซึ่งมีคอลัมน์ FIRSTDATE สำหรับเก็บข้อมูล 2 ความหมายด้วยกัน คือจะเก็บข้อมูลวันเกิดในกรณีที่เป็นลูกค้า และจะเก็บข้อมูลวันที่จ้างงานถ้าเป็นพนักงาน ดังภาพที่ 2.1

PERSON	
PK	ID
	FIRSTDATE

ภาพที่ 2.1 เอนติตีของฐานข้อมูลบุคคลก่อนรีแฟคเตอร์

ซึ่งต่อมาต้องการปรับปรุงประสิทธิภาพ และเพื่อให้ฐานข้อมูลง่ายต่อการอ่านทำความเข้าใจ จึงแยกคอลัมน์ FIRSTDATE นี้ออกเป็น 2 คอลัมน์ ได้แก่ คอลัมน์ BIRTHDATE ใช้สำหรับเก็บวันเกิดของลูกค้า และคอลัมน์ HIREDATE ใช้สำหรับเก็บข้อมูลวันที่จ้างงานของพนักงาน ดังแสดงในภาพที่ 2.2

PERSON	
PK	ID
	BIRTHDATE
	HIREDATE

ภาพที่ 2.2 เอนติตีของฐานข้อมูลบุคคลหลังจากรีแฟคเตอร์แล้ว

หลังรีแฟคเตอร์ฐานข้อมูลแล้ว จะเห็นว่าโปรแกรมจะไม่สามารถใช้งานได้เหมือนเดิม เนื่องจากคอลัมน์ FIRSTDATE ไม่มีให้โปรแกรมอ้างอิงถึง ผู้พัฒนาโปรแกรมจึงต้องปรับปรุงแก้ไขโปรแกรมใหม่ให้เปลี่ยนไปอ้างอิงคอลัมน์ BIRTHDATE และคอลัมน์ HIREDATE แทน โดยตรวจสอบเงื่อนไขว่าควรเลือกใช้คอลัมน์ใด การแก้ไขโปรแกรมให้ใช้งานได้เหมือนเดิม นั่นก็เป็นการคงไว้ซึ่งพฤติกรรมเดิม

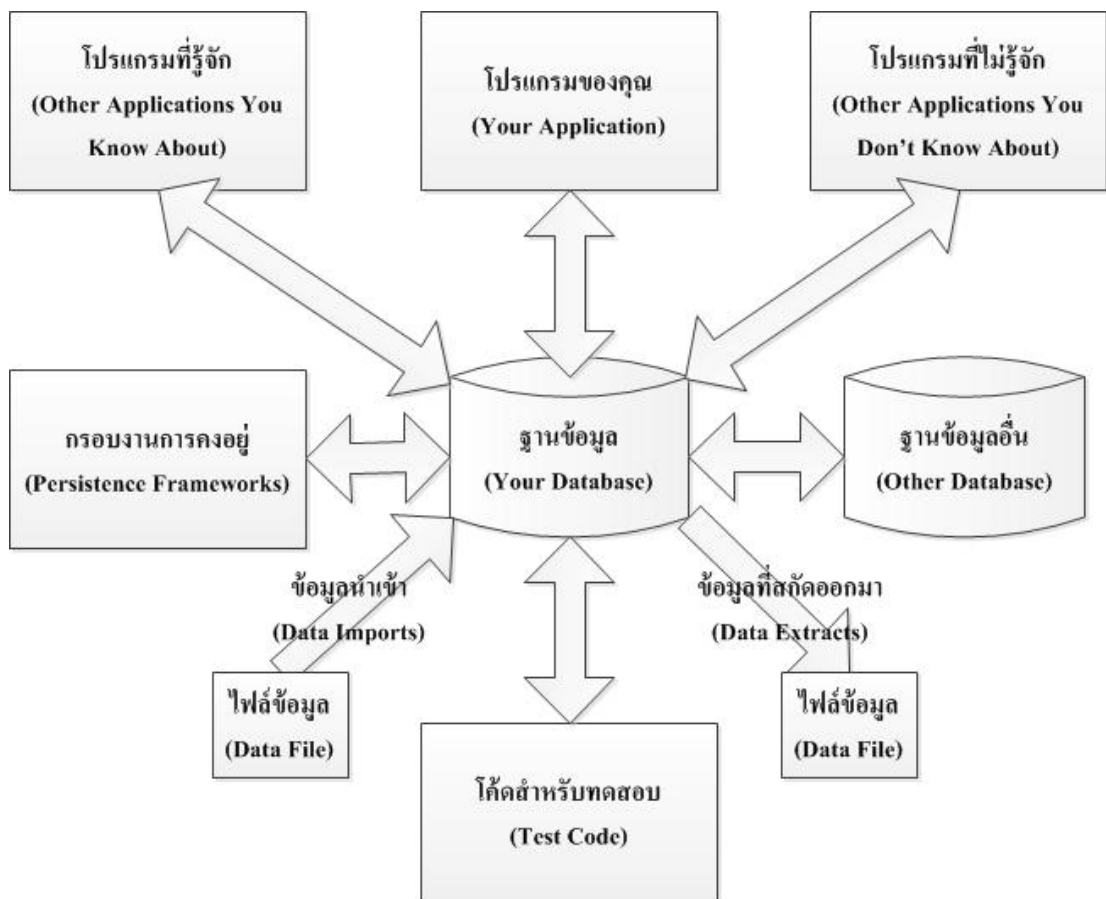
นอกจากนี้ข้อมูลเดิมที่มีอยู่ในคอลัมน์ FIRSTDATE จำเป็นต้องแปลงมาบันทึกลงในคอลัมน์ BIRTHDATE และคอลัมน์ HIREDATE แทน ซึ่งก่อนที่จะนำข้อมูลเดิมมาใส่ จำเป็นต้องดูเงื่อนไขก่อนว่าข้อมูลควรจะย้ายจากคอลัมน์ FIRSTDATE ลงมาเก็บที่คอลัมน์ BIRTHDATE หรือ คอลัมน์ HIREDATE อันเป็นการคงไว้ซึ่งความหมายของข้อมูล

จากตัวอย่างข้างต้นเป็นเพียงตัวอย่างของการรีแฟคเตอร์ฐานข้อมูล ซึ่งสก็อต แอมเบลล์ ได้แบ่งประเภทของการทำรีแฟคเตอร์ฐานข้อมูลไว้ด้วยกัน 5 ประเภท [4] ดังต่อไปนี้ โดยแสดงรายละเอียดประเภทการรีแฟคเตอร์ และคำสั่งดำเนินการในภาคผนวก ก

- 1) การรีแฟคเตอร์โครงสร้าง (Structural Refactoring) คือ การปรับเปลี่ยนโครงสร้างของ ตาราง, คอลัมน์ และทรรศนะ (View) เช่น การเปลี่ยนชื่อตาราง, การแยกคอลัมน์, การลบตาราง, การลบคอลัมน์ เป็นต้น
- 2) การรีแฟคเตอร์คุณภาพของข้อมูล (Data Quality Refactoring) คือ การปรับปรุงข้อมูลเดิมในฐานข้อมูลให้มีความต้องกันมากขึ้น และใช้ประเภทข้อมูล รวมถึงข้อมูลที่จัดเก็บให้เป็นมาตรฐานมากขึ้น เช่น การเก็บรหัสประเทศ ควรเก็บรหัสประเทศที่เป็นมาตรฐานที่ใช้กัน
- 3) การรีแฟคเตอร์การอ้างอิงของบูรณภาพ (Referential Integrity Refactoring) คือ การปรับปรุงการอ้างอิงถึงกันระหว่างตาราง เช่น การเพิ่มกุญแจนอก (Foreign Key) หรือ การกำหนดคิให้เมื่อข้อมูลแม่ถูกลบ ข้อมูลลูกๆ ที่เกี่ยวข้องก็ถูกลบออก (Cascading Delete) เป็นต้น
- 4) การรีแฟคเตอร์สถาปัตยกรรม (Architectural Refactoring) คือ การปรับปรุงลักษณะโดยรวมที่โปรแกรมภายนอกจะติดต่อกับฐานข้อมูล
- 5) การรีแฟคเตอร์เมธอดของฐานข้อมูล (Method Database Refactoring) คือ การพัฒนาคุณภาพของฟังก์ชันในฐานข้อมูล เช่น สตอร์โปรซีเจอร์ (Stored procedure) เป็นต้น

นอกจากนี้ สก็อต แอมเบลล์ยังได้กล่าวไว้อีกว่า การรีแฟคเตอร์ฐานข้อมูลยังเป็นวิธีการที่สนับสนุนการพัฒนาแบบอไจล์ (Agile) เนื่องจากอไจล์เป็นกระบวนการพัฒนาระบบที่มีการเปลี่ยนแปลงการออกแบบอยู่ตลอดเวลา [4, 14] ซึ่งทั้งตัวโปรแกรม และเค้าร่างฐานข้อมูลจะเปลี่ยนแปลงอยู่ตลอดเวลา

ในทางปฏิบัติการรีแฟกเตอร์ฐานข้อมูลนับว่าเป็นเรื่องยากที่จะกระทำ เนื่องจากไม่อาจทราบได้ว่าฐานข้อมูลตัวหนึ่งๆ จะมีโปรแกรมใดบ้างที่ใช้งานฐานข้อมูลเดียวกัน รวมไปถึงไม่ทราบว่าตาราง และคอลัมน์ในฐานข้อมูลมีโปรแกรมใดบ้างอ้างอิง ซึ่งโดยทั่วไปแล้วผู้พัฒนาโปรแกรม ก็จะทราบถึงตาราง และคอลัมน์ที่โปรแกรมอ้างอิงถึงฐานข้อมูลเฉพาะโปรแกรมที่ตนเอง หรือทีมตนเองพัฒนา ซึ่งอาจจะทราบไม่ทุกส่วนของโปรแกรม เนื่องจากอาจเป็นโปรแกรมที่รับช่วงต่อมาจากผู้พัฒนาคนอื่น จึงไม่ทราบถึงตาราง และคอลัมน์ทั้งหมดที่โปรแกรมอ้างอิง อีกทั้งยังมีระบบอื่นๆหลายระบบที่ใช้งานฐานข้อมูลเดียวกัน การรีแฟกเตอร์ฐานข้อมูลอาจส่งผลกระทบต่อระบบอื่นๆได้ ดังภาพที่ 2.3



ภาพที่ 2.3 กรณีของการรีแฟกเตอร์ฐานข้อมูลที่ยากที่สุด [4]

2.1.2 การแปลงวัตถุเป็นตารางความสัมพันธ์ (ORM : Object-Relational Mapping) [12, 13]

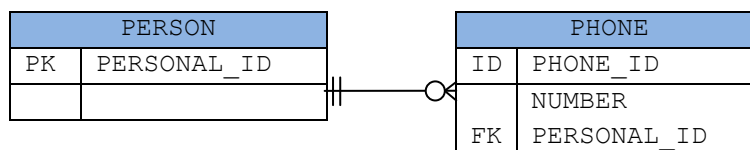
การพัฒนาโปรแกรมให้ติดต่อกับฐานข้อมูล สามารถพัฒนาได้หลายหลายวิธี เช่น การพัฒนาโดยเรียกใช้คลังคำสั่ง (Library) ที่มีมากับภาษาต่างๆ เขียนเชื่อมต่อกับฐานข้อมูลโดยตรง เช่นในภาษาจาวาใช้เจดีบีซี (JDBC) ในการเขียนโปรแกรมติดต่อกับฐานข้อมูล หรือจะพัฒนาโปรแกรมติดต่อกับฐานข้อมูลด้วยวิธีการของการแปลงวัตถุเป็นตารางความสัมพันธ์ ซึ่งเป็นเทคนิคที่ใช้กันอย่างแพร่หลายในปัจจุบัน เพราะง่ายต่อการพัฒนาโปรแกรม โดยผู้พัฒนาโปรแกรมไม่จำเป็นต้องทราบถึงโครงสร้างจริงของฐานข้อมูล ผู้พัฒนาโปรแกรมจะมองข้อมูลในฐานข้อมูลเป็นวัตถุในโปรแกรมเท่านั้น และนำวัตถุเหล่านี้ไปใช้งานต่อในส่วนอื่นๆของโปรแกรม ซึ่งการแปลงวัตถุเป็นตารางความสัมพันธ์ มีส่วนต่อประสานโปรแกรมประยุกต์ (API : Application Programming Interface) อยู่หลายตัวที่ทำการแปลงระหว่างตารางฐานข้อมูลเชิงสัมพันธ์และวัตถุของโปรแกรมให้โดยอัตโนมัติ หรือผู้พัฒนาสามารถเลือกที่จะพัฒนาส่วนการแปลงเองก็ได้

ข้อมูลที่เก็บในรูปของวัตถุสำหรับการพัฒนาโปรแกรมเชิงวัตถุ มักจะมีความสัมพันธ์กันระหว่างวัตถุ เช่น บุคคลแต่ละบุคคล อาจไม่มีหมายเลขโทรศัพท์ หรืออาจมีหมายเลขโทรศัพท์มากกว่า 1 หมายเลข ซึ่งผู้พัฒนาโปรแกรมสามารถเขียนคลาสสำหรับจัดเก็บข้อมูลบุคคลได้ โดยมีเพียงแค่คลาสเดียวโดยจัดเก็บหมายเลขโทรศัพท์ในลักษณะของรายการของข้อความ ดังภาพที่ 2.4

Person
- personId : String
- phoneNumber : List<String>

ภาพที่ 2.4 คลาสที่ใช้จัดเก็บข้อมูลบุคคล และหมายเลขโทรศัพท์

แต่ในมุมมองของฐานข้อมูลเชิงสัมพันธ์ จะจัดเก็บข้อมูลในรูปแบบของตารางซึ่งข้อมูลที่จัดเก็บได้จะอยู่ในรูปแบบของข้อความ หรือ ตัวเลขที่เป็นสเกลาร์ ซึ่งจากตัวอย่างดังกล่าว ตารางฐานข้อมูลเพียงตารางเดียวไม่สามารถจัดเก็บข้อมูลได้ตามที่คลาสต้องการเก็บ จึงต้องเพิ่มตารางสำหรับเก็บหมายเลขโทรศัพท์เพิ่มขึ้นมา และสร้างความสัมพันธ์กับตารางของบุคคลดังภาพที่ 2.5



ภาพที่ 2.5 ตารางฐานข้อมูลที่ใช้จัดเก็บข้อมูลบุคคล และหมายเลขโทรศัพท์

จากตัวอย่างดังกล่าวแสดงถึงหัวใจสำคัญของการแปลงวัตถุเป็นตารางความสัมพันธ์ ซึ่งจะต้องแปลงข้อมูลในรูปแบบของวัตถุที่มีความซับซ้อน ให้จัดเก็บข้อมูลลงในฐานข้อมูลเชิงสัมพันธ์ที่ไม่สามารถจัดเก็บข้อมูลแบบซับซ้อนได้ และต้องสามารถแปลงข้อมูลในฐานข้อมูลเชิงสัมพันธ์ กลับไปเป็นข้อมูลในรูปแบบของวัตถุได้เช่นเดียวกัน

จากเทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์นี้เองเมื่อพิจารณาจากตัวโปรแกรมก็จะทราบว่า โปรแกรมที่พิจารณาอ้างอิงถึงตาราง และคอลัมน์ใดบ้างของฐานข้อมูล ซึ่งสามารถนำความรู้ที่ได้จากทฤษฎีนี้ไปใช้ระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึง โดยเปรียบเทียบตาราง และคอลัมน์ที่โปรแกรมอ้างอิง กับตาราง และคอลัมน์ที่มีอยู่ในฐานข้อมูลจริง

หลักการของการแปลงวัตถุเป็นตารางความสัมพันธ์นี้มีกรอบงาน (Framework) ให้ใช้งานอยู่หลายๆ ภาษา โดย ภาษาจาวา (Java) มีกรอบงานเจพีเอ (JPA : Java Persistence API) ที่ทำงานตามแนวคิดของการแปลงวัตถุเป็นตารางความสัมพันธ์นี้

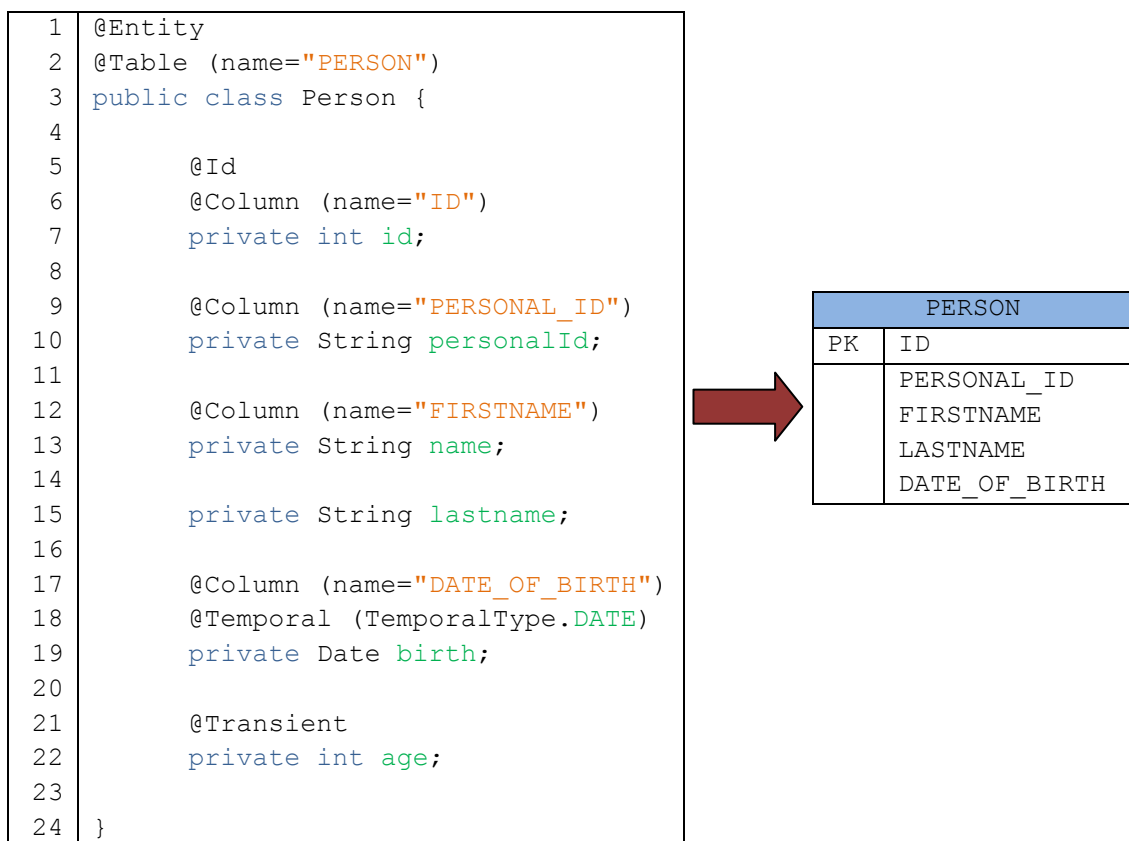
2.2 งานวิจัย และเครื่องมือที่เกี่ยวข้อง

งานวิจัยนี้จะนำเสนอวิธีการระบุคอลัมน์และตารางของฐานข้อมูลที่ไม่ถูกอ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ ผู้วิจัยจึงต้องพัฒนาโปรแกรมต้นแบบเพื่อใช้ทดสอบสมมติฐานว่าวิธีการที่นำเสนอ สามารถระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึงได้อย่างถูกต้อง ซึ่งผู้วิจัยได้เลือกเครื่องมือ และงานวิจัยที่เกี่ยวข้องเข้ามาช่วยในการออกแบบแนวคิด และพัฒนาโปรแกรมต้นแบบดังนี้

2.2.1 เจพีเอ (JPA : Java Persistence API) [13,15,16]

เจพีเอ เป็นกรอบงานภาษาจาวาที่ทำงานตามแนวคิดของการแปลงวัตถุเป็นตารางความสัมพันธ์ โดยตั้งแต่เวอร์ชันที่ 2 เป็นต้นไป จะรองรับความสามารถบรรณนิทัศน์ของภาษาจาวา (Java annotation) ซึ่งเป็นความสามารถของจาวาตั้งแต่เวอร์ชันที่ 5 เป็นต้นไป

วิธีการใช้งานเจพีเอ ผู้พัฒนาจะเขียนคลาสจาวาตามปกติเพื่อให้เก็บข้อมูลในวัตถุ และจะต้องระบุว่าต้องการให้คลาสใดบ้างที่จะเชื่อมโยงไปเป็นตารางของฐานข้อมูล ซึ่งจะระบุบรรณนิทัศน์เป็นเอนติตี้ (Entity) ให้แก่คลาสนั้นๆ และระบุบรรณนิทัศน์ให้แก่แอตทริบิวต์ (Attribute) ภายในคลาสจะเป็นคอลัมน์ที่ชื่ออะไรบ้างในตารางฐานข้อมูล ดังแสดงในภาพที่ 2.6 ซึ่งเป็นตัวอย่างการใช้งานเจพีเออย่างง่าย



ภาพที่ 2.6 การแปลงเอนติตี้คลาส เป็นตารางฐานข้อมูล ด้วยเจพีเอ

เจพีเอยังสามารถรองรับการใช้บรรณนิทัศน์ระบุความสัมพันธ์ระหว่างคลาสได้ ได้แก่ ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One Relationships) แบบหนึ่งต่อหลาย (One to Many Relationships) แบบหลายต่อหลาย (Many to Many Relationships) ดังแสดงตัวอย่างในภาพที่ 2.7

```

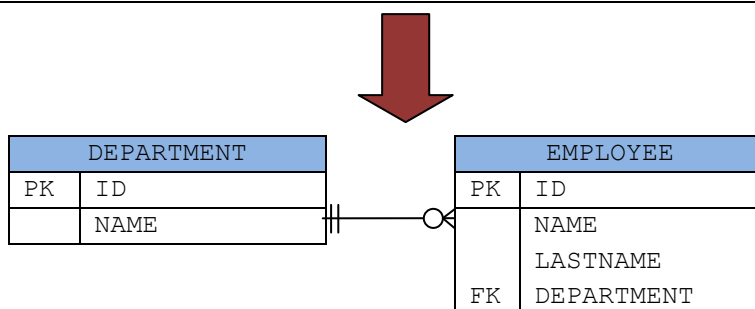
1  @Entity
2  @Table (name = "DEPARTMENT")
3  public class Department implements Serializable {
4      @Id
5      @GeneratedValue (strategy = GenerationType.AUTO)
6      private int id;
7
8      private String name;
9
10     @OneToMany (mappedBy = "dept")
11     private Collection<Employee> emp;
12 }

```

```

1  @Entity
2  @Table (name = "EMPLOYEE")
3  public class Employee implements Serializable {
4      @Id
5      @GeneratedValue (strategy = GenerationType.AUTO)
6      private int id;
7
8      private String name;
9      private String lastname;
10
11     @ManyToOne
12     @JoinColumn (name= "DEPARTMENT")
13     private Department dept;
14 }

```



ภาพที่ 2.7 การแปลงเอนติตี้คลาสที่มีความสัมพันธ์แบบหนึ่งต่อหลาย เป็นตารางฐานข้อมูลด้วยเจพีเอ

เจพีเอ ยังมีรายละเอียดการใช้งานบรรณนิทัศน์อื่นๆ อีกมาก เช่น @Column ไม่จำเป็นต้องระบุเสมอไป ถ้าผู้เขียนโปรแกรมไม่ต้องการเปลี่ยนชื่อคอลัมน์ หรือใช้ @Transient ระบุที่ขอบเขตที่ไม่ต้องการเก็บข้อมูลลงในฐานข้อมูล เป็นต้น ซึ่งรายละเอียดเพิ่มเติมสามารถศึกษาได้จากภาคผนวก ข หรือจากหนังสือ “Pro JPA 2: Mastering the Java Persistence API” [13]

กรอบงานเจพีเอที่ระบุบรรณนิทัศน์สำหรับการแปลงวัตถุเป็นตารางความสัมพันธ์นี้มีโปรแกรมประยุกต์หลายตัวที่สามารถรองรับที่อ่านบรรณนิทัศน์ของเจพีเอ และแปลงวัตถุเป็นตารางความสัมพันธ์ได้ โดยโปรแกรมประยุกต์เหล่านี้ เช่น ไฮเบอร์เนท (Hibernate) อีคลิปส์ลิงก์ (EclipseLink) โอเพ่นเจพีเอ (OpenJPA) และ ดาตานิวเคลียส (DataNucleus) เป็นต้น ซึ่งส่วนต่อประสานโปรแกรมประยุกต์เหล่านี้เป็นที่รู้จักกันดีสำหรับผู้พัฒนาภาษาจาวา ที่จะเชื่อมต่อฐานข้อมูลโดยใช้แนวคิดการแปลงวัตถุเป็นตารางความสัมพันธ์

การใช้งานเจพีเอในการติดต่อกับฐานข้อมูล ทำให้ทราบได้ว่าโปรแกรมที่พัฒนาเรียกใช้ตารางใด และคอลัมน์ใดบ้างจากฐานข้อมูล ซึ่งสามารถทำได้โดยพิจารณาเฉพาะคลาสที่ระบุบรรณนิทัศน์ที่เป็นเอนติตี้ และคูโครงสร้างของคลาสเอนติตี้เหล่านั้นก็จะทราบถึงตาราง และคอลัมน์ของฐานข้อมูลที่โปรแกรมอ้างอิงถึง อันจะเป็นการระบุขอบเขตตาราง และคอลัมน์ที่โปรแกรมอ้างอิงถึงได้แคบลง

แต่ทั้งนี้แม้ว่าจะทราบว่าโปรแกรมมีการอ้างอิงถึงตาราง และคอลัมน์ของฐานข้อมูลตัวใดบ้าง แต่ก็มิได้หมายความว่าส่วนที่ติดต่อกับฐานข้อมูลนั้นๆ จะถูกส่วนอื่นๆ ของโปรแกรมเรียกใช้ ซึ่งถ้าเอนติตี้คลาสนั้น หรือแอตทริบิวต์บางตัวในคลาส ไม่เคยถูกใช้งานเลยในโปรแกรม ตาราง และคอลัมน์เหล่านั้นก็สามารถลบออกจากฐานข้อมูลได้ และสามารถลบซอร์สโค้ดในส่วนที่เกี่ยวข้องเหล่านั้นได้

2.2.2 โค้ดที่เข้าไม่ถึง (Dead Code) [17,18]

โค้ดที่เข้าไม่ถึง คือ ซอร์สโค้ดของโปรแกรมที่ไม่เคยถูกใช้งานในโปรแกรมเลย ซึ่งแบ่งได้หลายระดับ เช่น คลาสทั้งคลาสไม่เคยถูกเรียกใช้งาน เมธอดไม่เคยถูกเรียกใช้งาน หรือบางคำสั่งในเมธอดที่ไม่เคยถูกเรียกใช้งาน เป็นต้น ซึ่งซอร์สโค้ดเหล่านี้ก็ไม่ควรจะมีอยู่ในโปรแกรม ซึ่งถ้าใช้หลักการนี้กับแอสทริวิวด์ในคลาสเอนติตี นั่นก็แสดงว่าคอลัมน์นั้นไม่จำเป็นต้องมีอยู่ในฐานข้อมูล รวมถึงไม่ควรจะมีแอสทริวิวด์นี้ในคลาสด้วย อีกทั้งถ้าไม่มีส่วนใดเลยในโปรแกรมที่ใช้งานเอนติตี คลาสบางตัว นั่นก็แสดงว่าในฐานข้อมูลไม่จำเป็นต้องมีตารางนี้ รวมไปถึงในซอร์สโค้ดก็ไม่จำเป็นต้องมีคลาสนี้ด้วยเช่นกัน

ถึงแม้ว่าการใช้เทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์ และโค้ดที่เข้าไม่ถึง สามารถระบุตาราง และคอลัมน์ของฐานข้อมูลที่สามารถลบได้ แต่ทั้งนี้คอลัมน์ฐานข้อมูลบางประเภทก็ไม่สามารถลบออกได้ ได้แก่ กุญแจหลัก (Primary Key) ของตารางฐานข้อมูล นอกจากนี้คอลัมน์ที่เป็นกุญแจนอก (Foreign Key) ก็เช่นเดียวกันที่ไม่สามารถลบได้ทันที แต่ถ้าวิเคราะห์แล้วพบว่าตารางที่กุญแจนอกอ้างอิงถึงสามารถลบได้ คอลัมน์ที่เป็นกุญแจนอกนี้ก็สามารถลบออกได้เช่นกัน

แม้ว่าเราจะสามารถระบุได้ว่าตาราง และคอลัมน์ใดบ้างที่สามารถลบออกจากฐานข้อมูลได้ แต่พบว่ามีตารางและคอลัมน์บางตัวของโปรแกรมถูกอ้างอิงถึง เพียงแต่ไม่เคยมีการใช้งานส่วนนั้นๆ ของโปรแกรมเลย งานวิจัยนี้จะถือว่าโปรแกรมไม่ได้อ้างอิงถึงคอลัมน์และตารางดังกล่าว เนื่องจากโปรแกรมไม่เคยมีการใช้งานตาราง และคอลัมน์เหล่านั้นเลย เมื่อลบตารางและคอลัมน์จากฐานข้อมูลแล้ว จะเกิดผลกระทบขึ้นกับตัวโปรแกรม ซึ่งจำเป็นต้องวิเคราะห์ถึงผลกระทบที่เกิดขึ้นตามมาด้วย เช่น โปรแกรมมีการใช้งานคำสั่งค้นคืน “SELECT A, B, C FROM T” แต่โปรแกรมใช้แค่คอลัมน์ A ไปแสดงผล นั่นคือคอลัมน์ B และ C จะถือว่าไม่ถูกใช้งานแต่เมื่อลบคอลัมน์ B และ C แล้วคำสั่งสืบค้นดังกล่าวจะไม่สามารถใช้งานได้ ส่งผลให้โปรแกรมผิดพลาดล้มเหลว

2.2.3 เครื่องมือหาเมธอดที่เข้าไม่ถึง (Dead Method)

เครื่องมือสำหรับใช้หาเมธอดที่เข้าไม่ถึง ที่ผู้วิจัยเลือกที่นำมาใช้งานได้แก่ เครื่องมือที่เป็น โอเพ่นซอร์ส (Open Source) ที่มีชื่อว่า com.mebigfatguy.deadmehods [19] ซึ่งเครื่องมือนี้มีส่วน นำเข้าของโปรแกรมคือพาธของโปรแกรมที่จะวิเคราะห์ ซึ่งผลลัพธ์ที่ได้คือ รายชื่อเมธอดทั้งหมด ในโปรแกรมที่ไม่มีส่วนอื่นๆของโปรแกรมเรียกใช้งาน ซึ่งผลลัพธ์ที่ได้ จะแสดงอยู่ในรูปของ ข้อความออกทางคอนโซล (Console) ไม่สามารถนำไปใช้ได้ แต่เนื่องจากเป็นเครื่องมือที่เป็น โอเพ่นซอร์ส ผู้วิจัยจึงนำซอร์สโค้ดของเครื่องมือนี้มาใช้เป็นส่วนหนึ่งของเครื่องมือต้นแบบในงานวิจัย โดยแก้ไขส่วนผลลัพธ์ให้เครื่องมือต้นแบบสามารถนำไปใช้ได้

2.2.4 การวิเคราะห์ผลกระทบเมื่อเค้าร่างฐานข้อมูลเปลี่ยนแปลง (Impact Analysis of Database Schema Changes) [20]

แอนดี้ มอล (Andy Maule) และคณะได้เสนอวิธีการวิเคราะห์ผลกระทบของโปรแกรมที่จะ เกิดขึ้นเมื่อเค้าร่างฐานข้อมูลเปลี่ยนแปลง โดยสนใจเค้าร่างของฐานข้อมูลว่ามีตาราง และคอลัมน์ใดบ้าง จากนั้นนำซอร์สโค้ดของโปรแกรมมาวิเคราะห์หาว่ามีการใช้คำสั่งสืบค้นใดบ้าง และคำสั่งสืบค้น เหล่านั้นมีการใช้งานตาราง และคอลัมน์ใดบ้างของฐานข้อมูล ซึ่งสามารถระบุผลกระทบใน โปรแกรมที่เกิดขึ้นได้ แต่วิธีการนี้สามารถใช้งานกับโปรแกรมที่เชื่อมต่อกับฐานข้อมูลโดยตรง เท่านั้น ไม่รองรับโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ติดต่อกับฐานข้อมูล

วิธีการที่ แอนดี้ มอล เสนอมานี้ สามารถนำแนวคิดมาประยุกต์ใช้กับโปรแกรมที่ใช้การแปลง วัตถุเป็นตารางความสัมพันธ์ติดต่อกับฐานข้อมูลได้ โดยวิเคราะห์ว่าในโปรแกรมมีการใช้คำสั่ง สืบค้นที่ใดบ้าง โดยกรอบงานเจพีเอ รองรับคำสั่งสืบค้นอยู่ 2 ภาษาได้แก่ ภาษาเจพีคิวแอล (JPQL) และ เอสคิวแอล (SQL) และหาความสัมพันธ์ว่าคำสั่งสืบค้นแต่ละคำสั่ง เกี่ยวข้องกับตาราง และ คอลัมน์ของฐานข้อมูลตัวใดบ้าง เพื่อใช้วิเคราะห์ผลกระทบที่จะเกิดขึ้นเมื่อ ตาราง และคอลัมน์ที่ โปรแกรมไม่ได้อ้างถึงถูกลบออกจากฐานข้อมูล

2.2.5 ภาษาสืบค้นข้อมูล (Query Language)

คำสั่งที่กระทำกับฐานข้อมูล เมื่อโปรแกรมใช้เจพีเอเป็นกรอบงานนั้น จะเป็นคำสั่งเลือกข้อมูลเท่านั้น (Select) เนื่องจากคำสั่งแทรก (Insert) เปลี่ยนแปลง (Update) และลบข้อมูล (Delete) ในเจพีเอ จะมีเมธอดสำหรับคำสั่งเหล่านี้ โดยส่งค่าวัตถุผ่านเข้าเมธอดได้โดยตรง ดังแสดงคำสั่งในภาพที่ 2.8 โดยไม่ต้องเขียนคำสั่งเพื่อกระทำกับฐานข้อมูลแต่อย่างใด การวิเคราะห์คำสั่งที่กระทำกับฐานข้อมูล จึงให้ความสนใจเฉพาะแต่คำสั่งสืบค้นข้อมูลเพียงอย่างเดียว ซึ่งคำสั่งสืบค้นที่ใช้งานร่วมกับ เจพีเอ มี 2 ภาษา ได้แก่ ภาษาเจพีคิวแอล และภาษาเอสคิวแอล

1	<code>entityManager.persist(object);</code>
2	<code>entityManager.merge(object);</code>
3	<code>entityManager.remove(object);</code>

ภาพที่ 2.8 การใช้คำสั่ง แทรก เปลี่ยนแปลง และลบข้อมูล ตามลำดับ

2.2.5.1 ภาษาเจพีคิวแอล (JPQL : Java Persistence Query Language)

เจพีคิวแอล เป็นภาษาใช้สำหรับสืบค้นข้อมูลที่ใช้ร่วมกับกรอบงานเจพีเอ ซึ่งมีลักษณะคล้ายกับภาษาเอสคิวแอล (SQL) แต่ต่างกันตรงที่ การอ้างถึงข้อมูล จะใช้ชื่อเอนทิตีคลาส และแอตทริบิวต์ แทนตารางและคอลัมน์แบบที่ภาษาเอสคิวแอลใช้ โดยคำสั่งสืบค้นนี้ ไม่จำเป็นต้องระบุถึงแอตทริบิวต์ที่จะสืบค้น เนื่องจากผลลัพธ์จะได้เป็นวัตถุที่สืบค้น นั่นคือจะสืบค้นครบทุกแอตทริบิวต์อยู่แล้ว เพียงแต่การใช้ชื่อแอตทริบิวต์ เป็นเพียงการใช้เงื่อนไข หรือใช้เป็นตัวเชื่อมเท่านั้น [15, 21] ซึ่งตัวอย่างการใช้งานภาษาเจพีคิวแอล เป็นไปดังภาพที่ 2.9 ในบรรทัดที่ 2-3 ส่วนในบรรทัดที่ 4 เป็นการกำหนดคำสั่งสืบค้น

1	<code>public List<Person> getPersonByName (String name) {</code>
2	<code>String jpql = "SELECT p FROM Person p WHERE "</code>
3	<code> + "p.name = :personName";</code>
4	<code>Query query = entityManager.createQuery(jpql);</code>
5	<code>Query.setParameter("personName", name);</code>
6	<code>return query.getResultList();</code>
7	<code>}</code>

ภาพที่ 2.9 ตัวอย่างการสืบค้นข้อมูลด้วยภาษาเจพีคิวแอล

2.2.5.2 ภาษาเอสคิวแอล (SQL : Structured Query Language)

เอสคิวแอล เป็นภาษามาตรฐานที่ใช้ติดต่อกับฐานข้อมูล ซึ่งสามารถใช้งานร่วมกับกรอบงานเจพีเอไค้ [21] แต่เนื่องจากผลลัพธ์ที่ได้จำเป็นต้องแปลงไปเป็นวัตถุของจาวา จึงจำเป็นต้องระบุถึงเอนติตี้ที่จะแปลงผลลัพธ์ให้เป็นวัตถุด้วยดังแสดงในภาพที่ 2.10 ในบรรทัดที่ 2-3 เป็นคำสั่งภาษาเอสคิวแอล ส่วนในบรรทัดที่ 4 เป็นการกำหนดคำสั่งสืบค้น และเอนติตี้คลาสที่จะแปลงผลลัพธ์ ซึ่งผลลัพธ์ที่ได้จากภาพที่ 2.9 และ 2.10 จะให้ผลลัพธ์เดียวกัน

```

1 public List<Person> getPersonByName (String name) {
2     String sql = "SELECT * FROM PERSON WHERE "
3         + "NAME = :personName";
4     Query query = entityManager.createNativeQuery(sql,
5         Person.class);
6     Query.setParameter("personName", name);
7     return query.getResultList();
8 }

```

ภาพที่ 2.10 ตัวอย่างการสืบค้นข้อมูลด้วยภาษาเอสคิวแอล

2.2.5.3 วิธีการใช้งานคำสั่งสืบค้น

เจพีเอจะใช้คำสั่งสืบค้นผ่านคลาสเอนติตี้เมเนเจอร์ (EntityManager) ซึ่งมีวิธีการเรียกใช้งานคำสั่งสืบค้นอยู่ด้วยกัน 3 วิธีดังแสดงในภาพที่ 2.11

```

1 entityManager.createQuery(jpql);
2 entityManager.createNativeNamedQuery(sql, class);
3 entityManager.createNamedQuery("named");

```

ภาพที่ 2.11 การใช้คำสั่งสืบค้นผ่าน เอนติตี้เมเนเจอร์

โดยในบรรทัดที่ 1 เป็นการใช้คำสั่งสืบค้น โดยใช้ภาษาเจพีคิวแอล ซึ่งผู้เขียนโปรแกรมจะส่งค่าคำสั่งสืบค้นภาษาเจพีคิวแอลเป็นประเภทข้อความเข้ามา

บรรทัดที่ 2 เป็นการใช้คำสั่งสืบค้น โดยใช้ภาษาเอสคิวแอล ซึ่งผู้เขียนโปรแกรมจะส่งค่าคำสั่งสืบค้นภาษาเอสคิวแอลเป็นข้อความ และคลาสที่ใช้สำหรับการแปลงผลลัพธ์เข้ามา ซึ่งผลลัพธ์ที่ได้จากการสืบค้น จะแปลงผลลัพธ์ไปเป็นรายการของวัตถุในคลาสที่ระบุ

บรรทัดที่ 3 เป็นการใช้อ้างอิงสืบทอด โดยระบุชื่อคำสั่งสืบทอด ซึ่งชื่อคำสั่งสืบทอดในโปรแกรมจะต้องมีไม่ซ้ำกัน และประกาศชื่อคำสั่งสืบทอด ไว้ที่แอนติคลาสที่เกี่ยวข้องกับผลลัพธ์ ดังแสดงในภาพที่ 2.12 เป็นการกำหนดชื่อคำสั่งสืบทอดให้แก่ภาษาเจพีคิวแอล และเอสคิวแอล ในบรรทัดที่ 3-7 และ 8-12 ตามลำดับ

```

1  @Entity
2  @Table (name = "PERSON")
3  @NamedQueries (
4      @NamedQuery ( name = "listPerson_jpql",
5                    query = "SELECT p FROM Person p"
6                    )
7  )
8  @NamedNativeQueries (
9      @NamedNativeQuery ( name = "listPerson_sql",
10                         query = "SELECT * FROM PERSON"
11                         )
12 )
13 public class Person implements Serializable {
14     //...
15 }

```

ภาพที่ 2.12 การตั้งชื่อให้แก่คำสั่งสืบทอดในคลาสแอนติ

2.2.6 เครื่องมือบีชีอีแอล (Byte Code Engineering Library: BCEL) [22]

เครื่องมือบีชีอีแอลเป็นเครื่องมือที่พัฒนาโดยบริษัทอาพาเซ่ (Apache) โดยบีชีอีแอลมีความสามารถในการอ่านไฟล์คลาสของภาษาจาวาซึ่งเป็นไฟล์ที่คอมไพล์แล้ว ให้มาอยู่ในรูปของรหัสคำสั่ง (Instruction Code) ของภาษาจาวา ดังแสดงในภาพที่ 2.13 ซึ่งเป็นซอร์สโค้ดก่อนที่จะคอมไพล์ และภาพที่ 2.14 แสดงรหัสคำสั่งที่ได้จากการอ่านไฟล์คลาสที่คอมไพล์แล้ว

```

1  public static final int fac(int n) {
2      return (n == 0)? 1 : n * fac(n - 1);
3  }

```

ภาพที่ 2.13 ซอร์สโค้ดก่อนคอมไพล์ [22]

1	0: iload_0	
2	1: ifne	#8
3	4: iconst_1	
4	5: goto	#16
5	8: iload_0	
6	9: iload_0	
7	10: iconst_1	
8	11: isub	
9	12: invokestatic	Factorial.fac (I)I (12)
10	15: imul	
11	16: ireturn	
12		
13	LocalVariable(start_pc = 0, length = 16, index = 0:int n)	

ภาพที่ 2.14 รหัสคำสั่งที่เครื่องมือบีชีอีแอลอ่านได้ จากไฟล์คลาสที่คอมไพล์แล้ว [22]

เครื่องมือบีชีอีแอลสามารถนำมาใช้ร่วมกับเครื่องมือต้นแบบของงานวิจัยได้ โดยนำมาใช้สำหรับอ่านคลาสไฟล์ของโปรแกรมทั้งหมด เพื่อหาว่าโปรแกรมมีการใช้คำสั่งสืบทอดใดบ้าง ซึ่งสังเกตได้จากคำสั่งสืบทอด จะต้องถูกใช้งานภายใต้คำสั่งที่แสดงในภาพที่ 2.11 โดยเครื่องมือบีชีอีแอล จะใช้รหัสคำสั่ง “invokeinterface” ในการเรียกใช้คำสั่งดังกล่าว ดังแสดงในภาพที่ 2.15 ถึง ภาพที่ 2.17

1	<code>public void testQuery(){</code>
2	<code> String jpql = "SELECT p FROM Person p";</code>
3	<code> Query query = entityManager.createQuery(jpql);</code>
4	<code>}</code>

1	0: ldc	"SELECT p FROM Person p" (2)
2	2: astore_1	
3	3: aload_0	
4	4: getfield	EM.entityManager
5		Ljavax/persistence/EntityManager; (3)
6	7: aload_1	
7	8: invokeinterface	javax.persistence.EntityManager.createQuery
8		(Ljava/lang/String;)
9		Ljavax/persistence/Query; (4) 2 0
10	13:astore_2	
11	14:return	

ภาพที่ 2.15 ตัวอย่างรหัสคำสั่งที่อ่านได้ จากการใช้คำสั่งสืบทอดภาษาเจฟคิวแอล

```

1 public void testNativeQuery(){
2     String sql = "SELECT * FROM PERSON";
3     Query query = entityManager.createNativeQuery(sql,
4                               Person.class);
5 }

```

```

1 0: ldc          "SELECT * FROM PERSON" (5)
2 2: astore_1
3 3: aload_0
4 4: getfield     EM.entityManager
5              Ljavax/persistence/EntityManager; (3)
6 7: aload_1
7 8: ldc_w       Person (6)
8 11: invokeinterface javax.persistence.EntityManager.
9                 createNativeQuery
10                (Ljava/lang/String;Ljava/lang/Class;)
11                Ljavax/persistence/Query; (7) 3 0
12 16: astore_2
13 17: return

```

ภาพที่ 2.16 ตัวอย่างรหัสคำสั่งที่อ่านได้ จากการใช้คำสั่งสืบค้นภาษาเอสคิวแอล

```

1 public void testNamedQuery(){
2     Query query = entityManager.createNamedQuery
3                       ("listPerson");
4 }

```

```

1 0: aload_0
2 1: getfield     EM.entityManager
3              Ljavax/persistence/EntityManager; (3)
4 4: ldc          "listPerson" (8)
5 6: invokeinterface javax.persistence.EntityManager.
6                 createNamedQuery(Ljava/lang/String;)
7              Ljavax/persistence/Query; (9) 2 0
8 11: astore_1
9 12: return

```

ภาพที่ 2.17 ตัวอย่างรหัสคำสั่งที่อ่านได้ จากการใช้คำสั่งสืบค้นแบบระบุชื่อคำสั่งสืบค้น

จากภาพที่ 2.15 ถึง ภาพที่ 2.17 จะเห็นว่า ไม่ว่าจะเป็คำสั่งสืบค้นแบบใด เมื่ออ่านรหัสคำสั่งแล้วจะเป็นการใช้รหัสคำสั่ง “invokeinterface” ทั้งสิ้น ส่วนรายละเอียดหลังรหัสคำสั่งจะเป็น “javax.persistence.EntityManager” ทั้งสิ้น เนื่องจากคำสั่งสร้างคำสั่งสืบค้นไม่ว่าจะเป็นแบบใดล้วนเป็นคำสั่งของคลาสเอนติตีเมเนเจอร์ทั้งสิ้น

ลักษณะการทำงานของรหัสคำสั่งเครื่องของภาษาจาวา คือจำเป็นต้องโหลดค่าในตัวแปรทั้งหมดก่อนที่เรียกใช้รหัสคำสั่ง “invokeinterface” ด้วยเหตุนี้จึงสามารถไต่ย้อนหาค่าของคำสั่งสืบค้นที่โปรแกรมใช้งานได้ ซึ่งความหมายรหัสคำสั่งเครื่องของภาษาจาวาสามารถดูได้จากภาคผนวก ค หรือศึกษาได้จากหนังสือข้อกำหนดภาษาจาวาเครื่องเสมือน “The Java Virtual Machine Specification Java SE 7 Edition” [23]

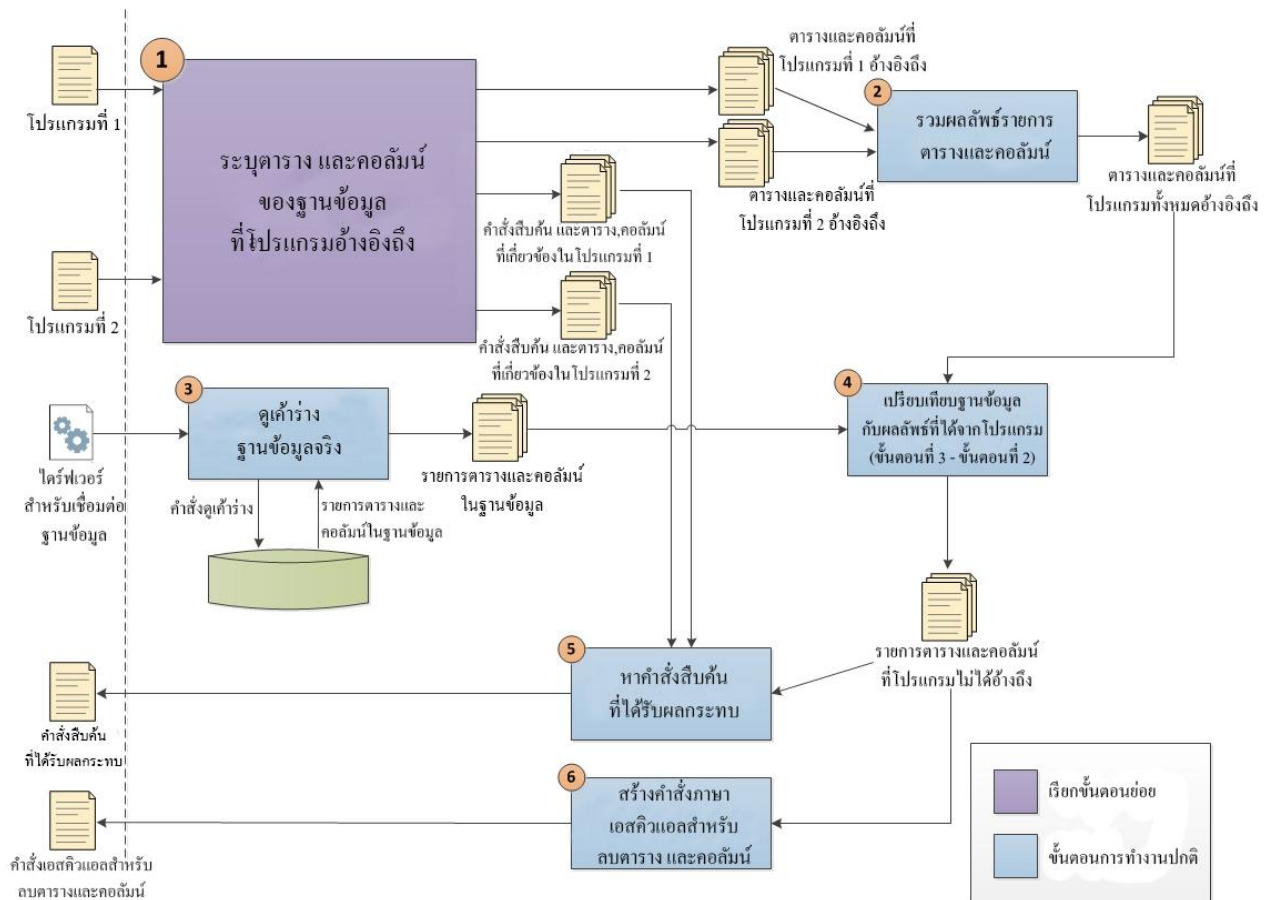
จากทฤษฎี และงานวิจัยที่เกี่ยวข้อง ผู้วิจัยนำความรู้ที่ได้ต่าง ๆ มารวมกันเพื่อนำเสนอวิธีใช้สำหรับระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึง และพัฒนาเครื่องมือต้นแบบสำหรับทดสอบแนวความคิดว่างานวิจัยที่นำเสนอให้ผลลัพธ์ที่สามารถนำไปใช้งานได้จริง โดยเครื่องมือต้นแบบ ผู้วิจัยเลือกที่จะพัฒนาด้วยภาษาจาวา ซึ่งเป็นภาษาที่ไม่มีค่าใช้จ่ายใดๆ และเป็นภาษาที่ได้รับความนิยมในปัจจุบัน เนื่องจากสามารถใช้งานได้กับหลายระบบปฏิบัติการ ซึ่งรายละเอียดแนวความคิดของงานวิจัย และแนวทางการพัฒนาเครื่องมือต้นแบบจะกล่าวถึงในบทที่ 3 ถัดไป

บทที่ 3

วิธีการดำเนินงานวิจัย

ในบทนี้ จะกล่าวถึงวิธีการดำเนินงานวิจัย โดยจะนำเสนอวิธีที่ใช้ระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้เข้าถึง โดยใช้ความรู้ที่ได้จากทฤษฎี และงานวิจัยที่เกี่ยวข้องมาประยุกต์ใช้ และนำเสนอวิธีการออกแบบสำหรับพัฒนาเครื่องมือต้นแบบสำหรับทดสอบแนวความคิด ซึ่งนำเสนอฟังก์ชันของเครื่องมือด้วยแผนภาพยูสเคส ขั้นตอนการทำงานของผู้ใช้งานกับระบบด้วยแผนภาพกิจกรรม แสดงโครงสร้างของเครื่องมือด้วยแผนภาพคลาส และเสนอพฤติกรรมการทำงานของฟังก์ชันด้วยแผนภาพลำดับ โดยมีรายละเอียดดังต่อไปนี้

3.1 ภาพรวมของงานวิจัย



ภาพที่ 3.1 ภาพรวมของการค้นหาโครงสร้างของฐานข้อมูลที่โปรแกรมไม่ได้ใช้งาน

งานวิจัยนี้ ใช้เทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์ และโค้ดที่เข้าไม่ถึง ซึ่งมีขั้นตอนทำงานดังแสดงในภาพที่ 3.1 โดยจะต้องมีข้อมูลนำเข้าคือพาธ (Path) ของโปรแกรมทุกๆโปรแกรมที่ใช้งานฐานข้อมูลเดียวกัน และไคร์ฟเวอร์สำหรับเชื่อมต่อฐานข้อมูล โดยแนวคิดของงานวิจัยนี้ประกอบไปด้วย 6 ขั้นตอนดังนี้

ขั้นตอนที่ 1 ระบุตาราง และคอลัมน์ของฐานข้อมูลที่โปรแกรมอ้างอิงถึง

ข้อมูลนำเข้า - พาธของโปรแกรม ทุกๆโปรแกรม

ข้อมูลนำออก - รายการตาราง และคอลัมน์ที่โปรแกรมแต่ละตัวอ้างอิงถึง

- คำสั่งสืบค้นที่ปรากฏในแต่ละโปรแกรม รวมถึงตาราง และคอลัมน์ที่เกี่ยวข้องกับแต่ละคำสั่ง

คำอธิบาย เป็นขั้นตอนวิเคราะห์โปรแกรมแต่ละโปรแกรม ว่าในแต่ละโปรแกรมมีซอร์สโค้ดที่อ้างอิงไปยังตารางใด และคอลัมน์ใด ของฐานข้อมูล อีกทั้งยังค้นหาคำสั่งสืบค้นต่างๆ ที่ปรากฏอยู่ในซอร์สโค้ดของ โปรแกรม และดูความสัมพันธ์ว่าแต่ละคำสั่งสืบค้นมีความเกี่ยวข้องกับตาราง และคอลัมน์ฐานข้อมูลใด

ขั้นตอนที่ 2 รวมผลลัพธ์รายการตารางและคอลัมน์

ข้อมูลนำเข้า - รายการตาราง และคอลัมน์ที่โปรแกรมแต่ละตัวอ้างอิงถึง
(ผลลัพธ์จากขั้นตอนที่ 1)

ข้อมูลนำออก - รายการตาราง และคอลัมน์ที่โปรแกรมทุกตัวที่ใช้งานฐานข้อมูลเดียวกันอ้างอิงถึง

คำอธิบาย รวมผลลัพธ์ รายการตาราง และคอลัมน์ที่โปรแกรมแต่ละโปรแกรมอ้างอิงถึงที่ได้จากขั้นตอนที่ 1 เข้าไว้ด้วยกันเป็นผลลัพธ์เดียว ซึ่งผลลัพธ์ที่ได้จากขั้นตอนนี้จะหมายถึงรายการตาราง และคอลัมน์ที่โปรแกรมทุกๆตัวที่ใช้งานฐานข้อมูลเดียวกันอ้างอิงถึง

ขั้นตอนที่ 3 คู่มือร่างฐานข้อมูลจริง

- ข้อมูลนำเข้า - ไรต์เฟออร์สำหรับใช้เชื่อมต่อกับฐานข้อมูล
- ข้อมูลสำหรับใช้เข้าถึงฐานข้อมูล เช่น ชื่อผู้ใช้ และรหัสผ่าน เพื่อเข้าสู่ฐานข้อมูล
- ข้อมูลนำออก - รายการตาราง และคอลัมน์ที่มีอยู่จริงในฐานข้อมูล
- คำอธิบาย คู่มือร่างฐานข้อมูลจริง โดยใช้ไรต์เฟออร์ของฐานข้อมูล เชื่อมต่อฐานข้อมูลโดยตรง แล้วใช้คำสั่งสำหรับคู่มือร่างฐานข้อมูล เพื่อหาว่าในฐานข้อมูลมีตาราง และคอลัมน์ใดบ้าง

ขั้นตอนที่ 4 เปรียบเทียบฐานข้อมูล กับผลลัพธ์ที่ได้จากโปรแกรม

- ข้อมูลนำเข้า - รายการตาราง และคอลัมน์ที่มีอยู่จริงในฐานข้อมูล (ผลลัพธ์จากขั้นตอนที่ 3)
- รายการตาราง และคอลัมน์ที่โปรแกรมทุกตัวที่ใช้งานฐานข้อมูลเดียวกัน อ้างอิงถึง (ผลลัพธ์จากขั้นตอนที่ 2)
- ข้อมูลนำออก - รายการตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึง
- คำอธิบาย เปรียบเทียบข้อมูลระหว่าง ตาราง และคอลัมน์ที่ฐานข้อมูลมีอยู่จริง เทียบกับ ตาราง และคอลัมน์ที่โปรแกรมอ้างอิงถึง ซึ่งจะทราบว่า ตาราง และคอลัมน์ใดบ้างในฐานข้อมูลที่โปรแกรมไม่ได้ใช้งาน (ผลลัพธ์ขั้นตอนที่ 3 ลบด้วยผลลัพธ์ขั้นตอนที่ 2) ซึ่งจะได้รายชื่อ ตาราง และคอลัมน์ของฐานข้อมูลที่ไม่มีโปรแกรมตัวใดเลยอ้างอิงถึง

ขั้นตอนที่ 5 หากำสั่งสืบค้นที่ได้รับผลกระทบ

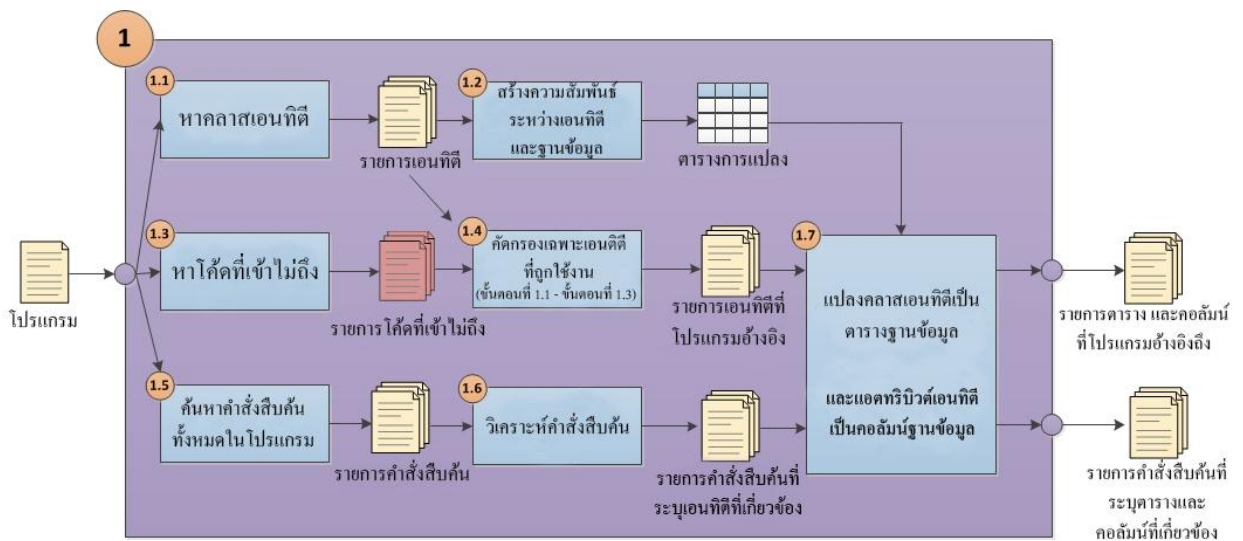
- ข้อมูลนำเข้า - คำสั่งสืบค้นที่ปรากฏในแต่ละโปรแกรม รวมถึงตาราง และคอลัมน์
ที่เกี่ยวข้องกับแต่ละคำสั่ง (ผลลัพธ์จากขั้นตอนที่ 1)
- รายการตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึง (ผลลัพธ์จากขั้นตอน
ที่ 4)
- ข้อมูลนำออก - รายการคำสั่งสืบค้นที่ได้รับผลกระทบจากการลบตาราง และคอลัมน์
- คำอธิบาย หากำสั่งสืบค้นทั้งหมดในหลายๆ โปรแกรมที่ได้รับผลกระทบ เนื่องจากบาง
คำสั่งอ้างอิงถึงทุกคอลัมน์ในตาราง แต่ในโปรแกรมเรียกใช้เพียงแคบบางคอลัมน์
แต่ถ้าคอลัมน์ที่โปรแกรมไม่เรียกใช้ถูกลบจากฐานข้อมูล คำสั่งสืบค้นก็จะ
ได้รับผลกระทบ จึงจำเป็นต้องหาว่ามีคำสั่งใดบ้างที่จะได้รับผลกระทบ โดย
ดูว่าในคำสั่งมีการอ้างอิงถึงตารางและคอลัมน์ใด ซึ่งคอลัมน์นี้รวมไปถึง
คอลัมน์ที่อยู่ในเงื่อนไขของคำสั่งด้วยเนื่องจากคอลัมน์ที่ใช้เป็นเงื่อนไข จะต้อง
มีอย่างน้อย 1 โปรแกรมที่เป็นตัวกำหนดค่าให้กับข้อมูลในคอลัมน์เงื่อนไข
มีเช่นนั้นแล้วเงื่อนไขในคอลัมน์ดังกล่าวจะไม่มีประโยชน์ แล้วเทียบกับ
ข้อมูลรายการตารางและคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึง ก็จะทราบว่า
คำสั่งใดบ้างที่จะได้รับผลกระทบ

ขั้นตอนที่ 6 สร้างคำสั่งภาษาเอสคิวแอลสำหรับลบตารางและคอลัมน์

- ข้อมูลนำเข้า - รายการตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึง (ผลลัพธ์จากขั้นตอน
ที่ 4)
- ข้อมูลนำออก - รายการคำสั่งภาษาเอสคิวแอลสำหรับลบตาราง และลบคอลัมน์ที่
โปรแกรมไม่ได้อ้างอิงถึง
- คำอธิบาย สร้างคำสั่ง เอสคิวแอล สำหรับการปรับปรุงเค้าร่างฐานข้อมูล ได้แก่ คำสั่ง
ลบตาราง และคอลัมน์ที่ไม่ได้ใช้งาน โดยใช้คำสั่ง “*DROP TABLE ชื่อ
ตาราง*” สำหรับลบตาราง และใช้คำสั่ง “*ALTER TABLE ชื่อตาราง DROP
COLUMN ชื่อคอลัมน์*” สำหรับลบคอลัมน์ใน ตารางที่ระบุอยู่ในคำสั่ง

3.1.1 ขั้นตอนการหาตาราง และคอลัมน์ที่โปรแกรมอ้างอิง

จากขั้นตอนการทำงานของแนวคิดการระบุตารางและคอลัมน์ของฐานข้อมูลที่โปรแกรมไม่ได้อ้างอิงถึงที่นำเสนอในภาพที่ 3.1 ในขั้นตอนที่ 1 ซึ่งเป็นขั้นตอนการหาว่าโปรแกรมแต่ละโปรแกรมมีการอ้างอิงถึงตาราง และคอลัมน์ใดบ้าง โดยจะใช้เทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์ และโค้ดที่เข้าไม่ถึงร่วมกันเพื่อระบุตาราง และคอลัมน์ที่โปรแกรมแต่ละโปรแกรมอ้างอิงถึง โดยมีขั้นตอนการทำงานดังภาพที่ 3.2 ซึ่งประกอบไปด้วย 7 ขั้นตอน โดยถ้ามีโปรแกรมมากกว่าหนึ่งโปรแกรม การวิเคราะห์โปรแกรมแต่ละโปรแกรมจะต้องทำขั้นตอนทั้ง 7 ขั้นตอนนี้เหมือนกัน ซึ่งแสดงรายละเอียดของแต่ละขั้นตอนดังนี้



ภาพที่ 3.2 ขั้นตอนการหาตาราง และคอลัมน์ที่โปรแกรมใช้งาน โดยใช้เทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์ และโค้ดที่เข้าไม่ถึง

1	@Entity
2	@Table (name = "PERSON")
3	public class Person implements Serializable {
4	//...
5	}

ภาพที่ 3.3 บรรณนิทศน์ @Entity ที่ระบุให้กับคลาสเอนติตี้

ขั้นตอนที่ 1.1 หาคลาสเอนติตี้

ข้อมูลนำเข้า - พาชของโปรแกรม 1 โปรแกรม

ข้อมูลนำออก - รายการคลาสเอนติตี้ทั้งหมดในโปรแกรม

คำอธิบาย ค้นหากลุคเอนติตี้ทั้งหมดในโปรแกรม ซึ่งเป็นกลุคที่จะเปลงไปเป็นตารางของฐานข้อมูล โดยหาได้จากบรรณนิทัศน์ของภาษาจาวา โดยกลุคเอนติตี้จะระบุด้วยบรรณนิทัศน์ @Entity อยู่ภายนอกกลุคดังแสดงภาพที่ 3.3 ซึ่งวิธีการหากลุคเอนติตี้ สามารถหาได้โดยใช้กรอบงานการสะท้อนของจาวา (Java Reflection) ซึ่งเป็นคลังคำสั่งมาตรฐานที่มาพร้อมกับจาวา ช่วยในการหาบรรณนิทัศน์ในกลุคได้ว่ากลุคที่สนใจมีบรรณนิทัศน์ใดบ้าง ซึ่งถ้านำกลุคทุกกลุคในโปรแกรมมาหา ก็จะได้กลุคที่เป็นกลุคเอนติตี้ทั้งหมดในโปรแกรม เมื่อหากลุคเอนติตี้ทั้งหมดในโปรแกรมได้แล้ว ก็จะทราบว่ในแต่ละกลุคเอนติตี้มีโครงสร้างของกลุคเป็นแบบใด

ขั้นตอนที่ 1.2 สร้างความสัมพันธ์ระหว่างเอนติตี้และฐานข้อมูล

ข้อมูลนำเข้า - รายการกลุคเอนติตี้ทั้งหมดในโปรแกรม (ผลลัพธ์จากขั้นตอนที่ 1.1)

ข้อมูลนำออก - ตารางระบุความสัมพันธ์ระหว่างเอนติตี้และฐานข้อมูล

คำอธิบาย หลังจากทีหากลุคเอนติตี้ทั้งหมดในโปรแกรมได้จากขั้นตอนที่ 1.1 แล้วถ้าพิจารณาโครงสร้างกลุคสรวมถึงบรรณนิทัศน์ต่างๆที่ใช้กำหนดวิธีการเปลงวัตถุเป็นตารางความสัมพันธ์ ดังทีกล่าวมาแล้วในบทที่ 2 หัวข้อที่ 2.2.1 ก็จะสามารถหาความสัมพันธ์การเปลงเอนติตี้ไปเป็นตาราง และคอลลัมน์ของฐานข้อมูลได้ โดยผลลัพธ์ของขั้นตอนนี้จะทราบว่ กลุคเอนติตี้หนึ่งๆ จะสัมพันธ์กับตารางฐานข้อมูลใด แอตทริบิวต์ของกลุคสัมพันธ์กับคอลลัมน์ใด รวมไปถึงสามารถบอกได้ว่า ตารางฐานข้อมูลหนึ่งสัมพันธ์กับเอนติตี้กลุคใด และคอลลัมน์ในตารางฐานข้อมูลสัมพันธ์กับ แอตทริบิวต์ใดในกลุคใด

ขั้นตอนที่ 1.3 หาโค้ดที่เข้าไม่ถึง

ข้อมูลนำเข้า - พาทของโปรแกรม 1 โปรแกรม

ข้อมูลนำออก - รายการคลาสและเมธอดในโปรแกรมที่ไม่ถูกใช้งาน

คำอธิบาย ค้นหาคلاس และเมธอด ภายในคลาสต่างๆ คลาส ว่ามีคลาส และเมธอดใดบ้างที่ไม่เคยถูกใช้งานในโปรแกรม โดยคลาสที่ไม่ถูกใช้งานสามารถหาได้โดยใช้คำสั่งของบีซีอีแอล [22] เข้าช่วยในการหา เนื่องจากบีซีอีแอลสามารถอ่านไฟล์คลาสของจาวาได้ และดูว่าคลาสใดบ้างที่ไม่ปรากฏอยู่ในส่วนอื่นๆ ของโปรแกรม นอกเหนือจากในคลาสตัวเอง ก็แสดงว่าคลาสเหล่านั้นเป็นคลาสที่ไม่ถูกใช้งาน และสามารถหาเมธอดที่ไม่ถูกใช้งานโดยใช้โอเพ่นซอร์ส `com.mebigfatguy.deadmethods` [19] ซึ่งจะสามารถบอกได้ว่ามีเมธอดใดบ้างที่ไม่ถูกใช้งาน และเมธอดดังกล่าวอยู่ภายใต้คลาสใด ดังนั้นผลลัพธ์ที่ได้จากขั้นตอนนี้จะมีรายการอยู่ 2 รายการ คือ รายการชื่อของคลาสที่ไม่ถูกใช้งานทั้งคลาส และรายการของเมธอดที่ไม่ถูกใช้งาน ซึ่งเมธอดที่ไม่ถูกใช้งานจะระบุด้วยว่าเป็นเมธอดของคลาสใด

ขั้นตอนที่ 1.4 คัดกรองเฉพาะเอนติตี้ที่ถูกใช้งาน

ข้อมูลนำเข้า - รายการคลาสเอนติตี้ทั้งหมดในโปรแกรม (ผลลัพธ์จากขั้นตอนที่ 1.1)

- รายการคลาสและเมธอดในโปรแกรมที่ไม่ถูกใช้งาน (ผลลัพธ์จากขั้นตอนที่ 1.3)

ข้อมูลนำออก - รายการเอนติตี้ที่โปรแกรมอ้างอิงถึง

คำอธิบาย คัดกรองเฉพาะคลาสเอนติตี้ แอตทริบิวต์ของเอนติตี้ที่ถูกใช้งาน โดยคลาสเอนติตี้ที่ถูกใช้งานสามารถหาได้จากรายการคลาสเอนติตี้ทั้งหมดที่ได้จากขั้นตอนที่ 1.1 ลบด้วยรายการคลาสที่ไม่ถูกใช้งานที่หาได้จากขั้นตอนที่ 1.3 ซึ่งจะเหลือเพียงแต่เอนติตี้คลาสที่มีการใช้งาน

ขั้นตอนที่ 1.4 คัดกรองเฉพาะเอนติตี้ที่ถูกใช้งาน (ต่อ)

คำอธิบาย(ต่อ) ส่วนการหาแอตทริบิวต์ของเอนติตี้ที่ถูกใช้งาน สามารถหาได้โดยพิจารณาจากเมธอดตั้งค่า (Getter Method) และเมธอดกำหนดค่า (Setter Method) เนื่องจากแนวคิดการห่อหุ้ม (Encapsulation) ทุกแอตทริบิวต์ของคลาสจะถูกเรียกใช้ผ่านเมธอดตั้งค่า และกำหนดค่าเท่านั้น นั่นคือแอตทริบิวต์ที่มีการใช้งานเมธอดตั้งค่า และกำหนดค่าต้องไม่อยู่ในรายการเมธอดที่ไม่ถูกใช้งานที่ได้จากขั้นตอนที่ 1.3 อย่างน้อย 1 เมธอด นั่นคือแอตทริบิวต์ที่ถูกใช้งาน ซึ่งผลลัพธ์ที่ได้จากขั้นตอนนี้ คือ สามารถบอกได้ว่าคลาสเอนติตี้ และแอตทริบิวต์ของเอนติตี้ตัวใดบ้างที่มีการ ใช้งานจริงในโปรแกรม ซึ่งสามารถนำไปหาตาราง และคอลัมน์ที่สัมพันธ์กับเอนติตี้เหล่านี้ได้ต่อไป

ขั้นตอนที่ 1.5 ค้นหาคำสั่งสืบค้นทั้งหมดในโปรแกรม

ข้อมูลนำเข้า - พารามิเตอร์ของโปรแกรม 1 โปรแกรม

ข้อมูลนำออก - รายการคำสั่งสืบค้น

คำอธิบาย ค้นหาคำสั่งสืบค้นทั้งหมดในโปรแกรม โดยเจพีเอจจะใช้คำสั่งสืบค้นผ่านคลาสเอนติตี้เมเนเจอร์ ดังที่กล่าวไปในบทที่ 2 หัวข้อที่ 2.2.5 โดยการหาคำสั่งสืบค้นในโปรแกรมจำเป็นต้องรู้การทำงานเข้าไปถึงระดับภายในเมธอด เพื่อหาบรรทัดคำสั่งของเอนติตี้เมเนเจอร์ที่ใช้งานคำสั่งสืบค้น โดยในขั้นตอนนี้สามารถใช้คลังคำสั่งของ บีซีอีแอล (BCEL) [22] เข้ามาช่วย เนื่องจากบีซีอีแอลเป็นคลังคำสั่งสำหรับอ่านโครงสร้างการทำงานของเมธอดภายในคลาส ดังที่กล่าวไปในบทที่ 2 หัวข้อที่ 2.2.6 ซึ่งสามารถนำมาใช้หาได้ว่าคำสั่งสืบค้นใดที่ใช้งานผ่านเอนติตี้เมเนเจอร์ ซึ่งในขั้นตอนนี้จะได้รายการคำสั่งสืบค้นทั้งหมดในโปรแกรม และสามารถจำแนกได้ว่าเป็นคำสั่งภาษาเจฟีกิวแอลหรือเอสคิวแอล เนื่องจากคำสั่งทั้ง 2 ภาษาจะใช้งานผ่านเมธอดของเอนติตี้เมเนเจอร์ที่ต่างกัน

ขั้นตอนที่ 1.6 วิเคราะห์คำสั่งสืบค้น

- ข้อมูลนำเข้า - รายการคำสั่งสืบค้น (ผลลัพธ์จากขั้นตอนที่ 1.5)
- ข้อมูลนำออก - รายการคำสั่งสืบค้น ที่มีคลาสเอนติตี และแอตทริบิวต์ที่เกี่ยวข้องด้วย
- คำอธิบาย - วิเคราะห์คำสั่งสืบค้นที่ได้จากขั้นตอนที่ 1.5 โดยวิเคราะห์ทีละคำสั่ง ในภาษา เจพีคิวแอลดูว่ามีคลาส และแอตทริบิวต์ใดบ้างที่ปรากฏในคำสั่ง ซึ่งจะได้ รายชื่อคลาสเอนติตี และแอตทริบิวต์ที่เกี่ยวข้องกับคำสั่งสืบค้น ส่วนคำสั่ง สืบค้นที่เป็นภาษาเอสคิวแอลจะดูว่ามีรายชื่อตาราง และคอลัมน์ที่ปรากฏใน คำสั่งบ้าง ซึ่งจะได้รายการตาราง และคอลัมน์ที่คำสั่งนั้นๆเกี่ยวข้องด้วย

ขั้นตอนที่ 1.7 แปลงเอนติตีเป็นตาราง และคอลัมน์ฐานข้อมูล

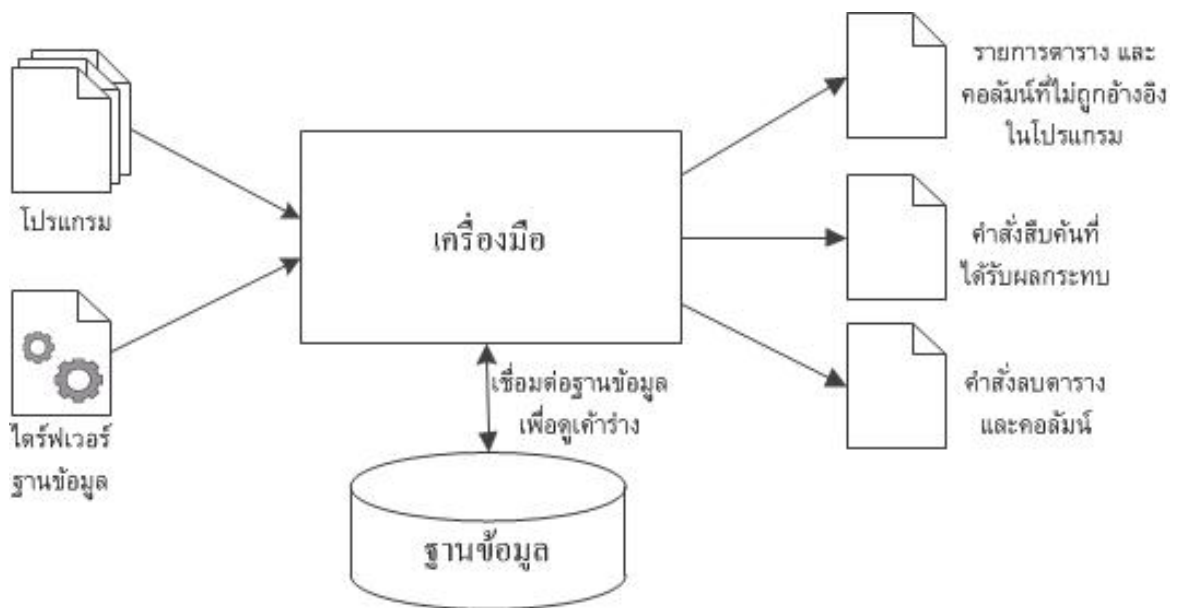
- ข้อมูลนำเข้า - ตารางระบุความสัมพันธ์ระหว่างเอนติตีและฐานข้อมูล (ผลลัพธ์จากขั้นตอน ที่ 1.2)
- รายการเอนติตีที่โปรแกรมอ้างอิงถึง (ผลลัพธ์จากขั้นตอนที่ 1.4)
- รายการคำสั่งสืบค้น ที่มีคลาสเอนติตี และแอตทริบิวต์ที่เกี่ยวข้องด้วย (ผลลัพธ์จากขั้นตอนที่ 1.6)
- ข้อมูลนำออก - รายการตาราง และคอลัมน์ที่โปรแกรมอ้างอิงถึง
- คำสั่งสืบค้นที่ปรากฏในโปรแกรม รวมถึงตาราง และคอลัมน์ที่เกี่ยวข้อง กับแต่ละคำสั่ง
- คำอธิบาย - แปลงรายชื่อคลาสเอนติตี และแอตทริบิวต์ที่โปรแกรมมีการใช้งานที่ได้จาก ขั้นตอนที่ 1.4 มาแปลงเป็นรายชื่อตารางและคอลัมน์ โดยใช้ตารางการแปลง ที่สร้างได้จากขั้นตอนที่ 1.2 ซึ่งผลลัพธ์ที่ได้คือรายชื่อตารางและคอลัมน์ของ ฐานข้อมูลที่โปรแกรมอ้างอิงถึง และแปลงคำสั่งสืบค้นภาษาเจพีคิวแอลที่ได้ จากขั้นตอนที่ 1.6 ให้เป็นชื่อตารางและคอลัมน์โดยใช้ตารางการแปลงที่สร้าง ได้จากขั้นตอนที่ 1.2 ซึ่งผลลัพธ์ที่ได้คือคำสั่งแต่ละคำสั่งมีความเกี่ยวข้องกับ ตารางและคอลัมน์ใดบ้าง

3.2 แนวทางการพัฒนาเครื่องมือต้นแบบ

แนวทางสำหรับการพัฒนาเครื่องมือต้นแบบ สำหรับระบุตาราง และคอลัมน์ที่ไม่ได้อ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ โดยผู้วิจัยได้แบ่งเป็น 2 ส่วนคือ ภาพรวมขั้นตอนการทำงานของเครื่องมือ และการออกแบบเครื่องมือ ซึ่งมีรายละเอียดดังนี้

3.2.1 ภาพรวมขั้นตอนการทำงานของเครื่องมือ

เครื่องมือต้นแบบของงานวิจัย ผู้วิจัยสร้างขึ้นเพื่อใช้ทดสอบแนวความคิดที่นำเสนอว่าแนวความคิดดังกล่าว เมื่อนำไปใช้จริงแล้วจะได้ผลอย่างไร โดยเครื่องมือต้นแบบที่ออกแบบแสดงในภาพที่ 3.4 ซึ่งประกอบไปด้วย 3 ส่วนได้แก่



ภาพที่ 3.4 ภาพรวมการทำงานของเครื่องมือต้นแบบ

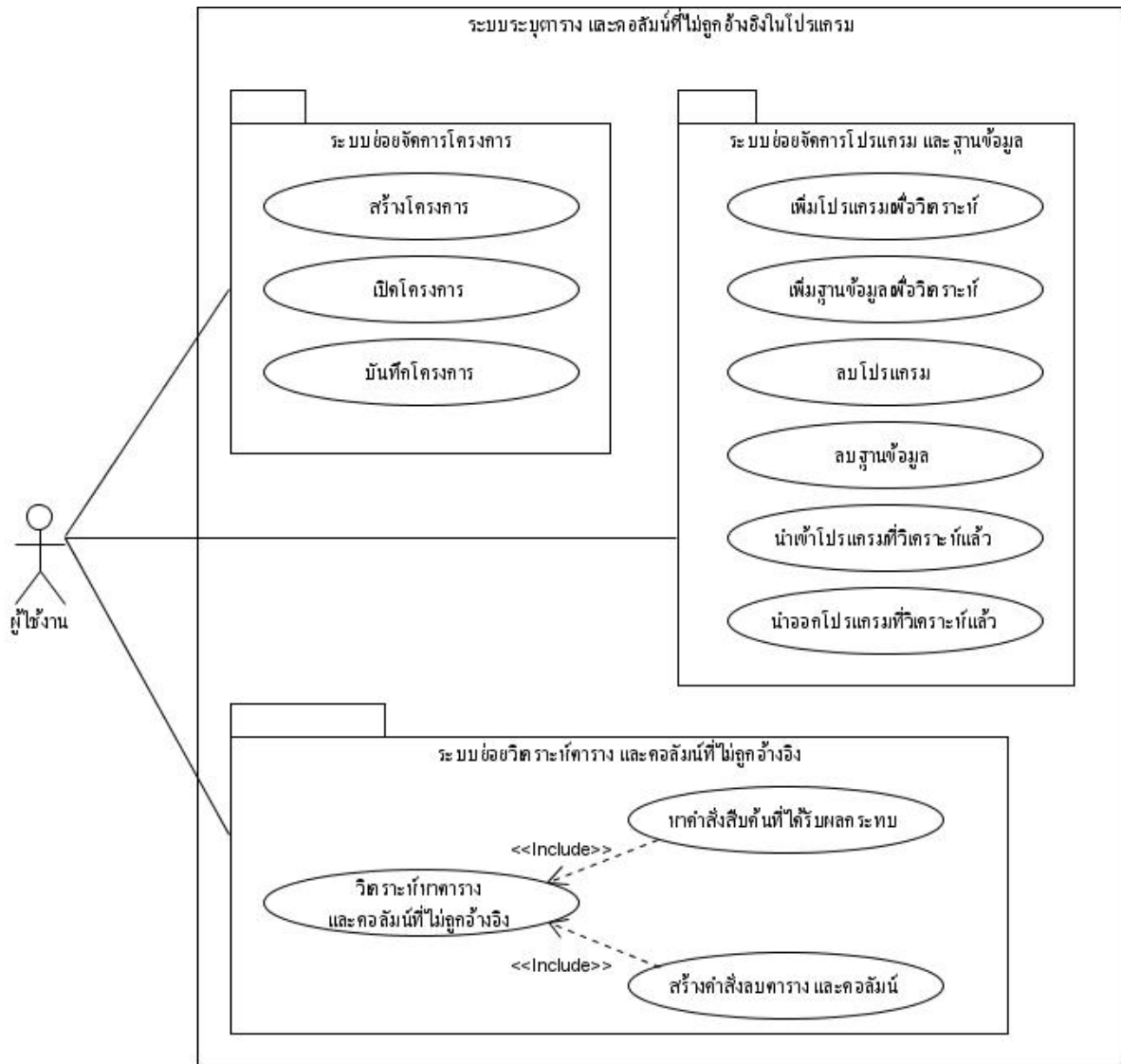
1. ส่วนนำเข้า สำหรับให้เครื่องมือนำไปวิเคราะห์ ได้แก่ โปรแกรมต่างๆ ที่ต้องการให้วิเคราะห์ว่ามีการอ้างอิงตาราง และคอลัมน์ฐานข้อมูลใดบ้าง และไดร์ฟเวอร์สำหรับใช้ติดต่อกับฐานข้อมูล รวมถึงค่าต่างๆที่กำหนด เพื่อให้เครื่องมือสามารถเชื่อมต่อกับฐานข้อมูลได้

2. ส่วนการทำงานของเครื่องมือ ได้แก่ การวิเคราะห์หาการอ้างอิงตาราง และคอลัมน์ฐานข้อมูลแต่ละโปรแกรม การค้นคืนตาราง และคอลัมน์จากฐานข้อมูลจริง การวิเคราะห์เพื่อหาตาราง และคอลัมน์ที่ไม่มีโปรแกรมอ้างอิงถึง วิเคราะห์ผลกระทบของคำสั่งสืบค้นในโปรแกรมในกรณีตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงถูกลบจากฐานข้อมูล รวมไปถึงสร้างคำสั่งภาษาเอสคิวแอลมาตรฐานสำหรับลบตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงออก
3. ส่วนผลลัพธ์ของเครื่องมือ ซึ่งจะแสดงรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม คำสั่งสืบค้นที่ได้รับผลกระทบ และคำสั่งภาษาเอสคิวแอลสำหรับลบตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง

3.2.2 ฟังก์ชันของเครื่องมือ

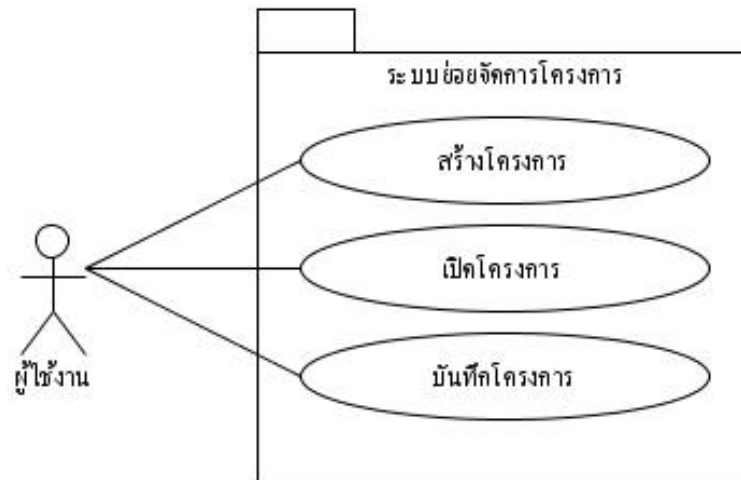
เครื่องมือต้นแบบของงานวิจัยนี้มีฟังก์ชันต่างๆ ในมุมมองของผู้ใช้งานรวมทั้งหมด 12 ฟังก์ชันงานซึ่งเป็นความต้องการของเครื่องมือ ดังแสดงในแผนภาพยูสเคสภาพที่ 3.5 ซึ่งประกอบไปด้วย 3 ระบบย่อย ได้แก่

1. ระบบย่อยจัดการโครงการ คือระบบย่อยสำหรับจัดการไฟล์ของโครงการ ได้แก่ การสร้างไฟล์โครงการ การเปิดไฟล์โครงการ และการบันทึกไฟล์โครงการ
2. ระบบย่อยจัดการโปรแกรม และฐานข้อมูล คือระบบย่อยสำหรับเพิ่ม และลบโปรแกรม และฐานข้อมูลออกจากโครงการ
3. ระบบย่อยวิเคราะห์ตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง คือระบบย่อยส่วนหลักของเครื่องมือ ที่ใช้สำหรับวิเคราะห์โปรแกรมต่างๆ และหารายการตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิง



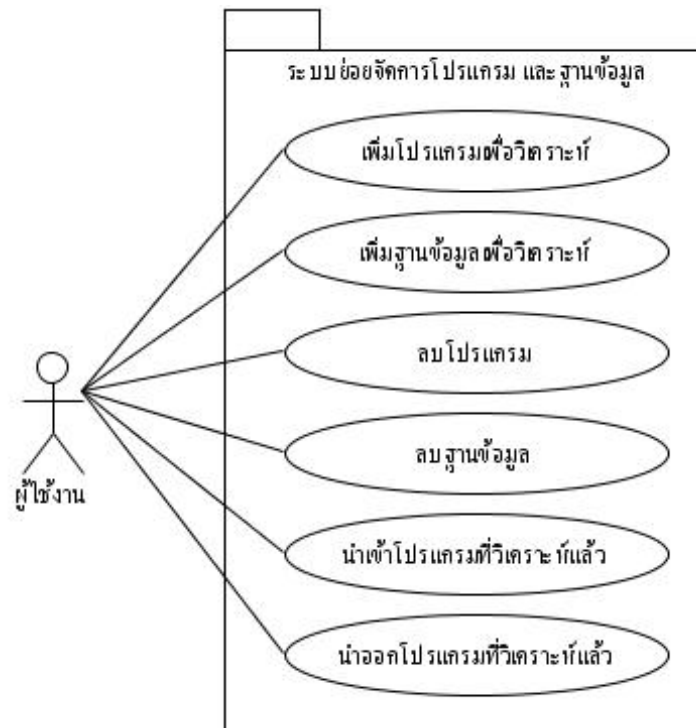
ภาพที่ 3.5 แผนภาพยูสเคสฟังก์ชันงานของเครื่องมือ

สำหรับระบบย่อยจัดการโครงการ ประกอบไปด้วย 3 ยูสเคส ได้แก่ ยูสเคสการสร้างโครงการ ยูสเคสการเปิดโครงการ และยูสเคสการบันทึกโครงการ ดังภาพที่ 3.6 และคำบรรยายรายละเอียดของแต่ละยูสเคสในภาคผนวก ง ตารางที่ ง.1 – ง.3



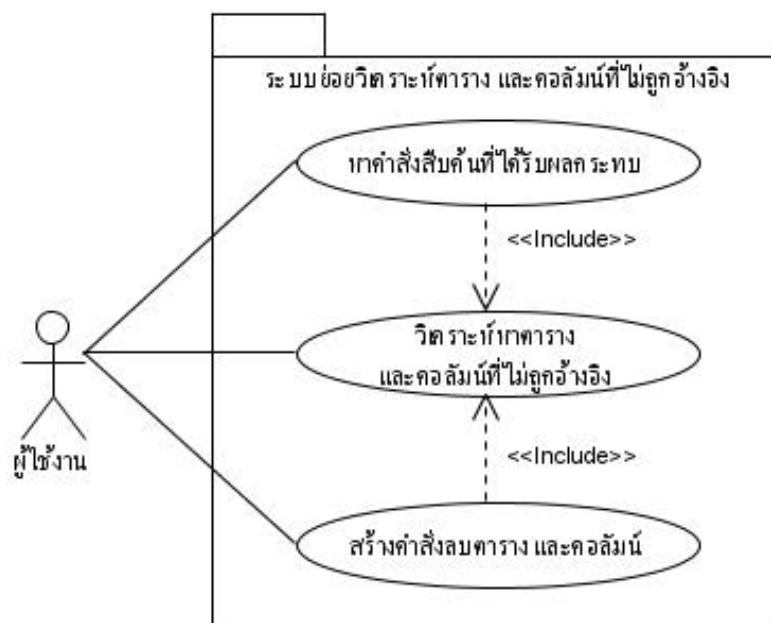
ภาพที่ 3.6 แผนภาพยูสเคสระบบย่อยจัดการ โครงการ

ระบบย่อยจัดการ โปรแกรม และฐานข้อมูล ประกอบไปด้วย 6 ยูสเคส ได้แก่ ยูสเคสเพิ่มโปรแกรมเพื่อวิเคราะห์ ยูสเคสเพิ่มฐานข้อมูลเพื่อวิเคราะห์ ยูสเคสลบโปรแกรม ยูสเคสลบฐานข้อมูล ยูสเคสนำเข้าโปรแกรมที่วิเคราะห์แล้ว และยูสเคสนำออกโปรแกรมที่วิเคราะห์แล้ว ดังภาพที่ 3.7 และคำบรรยายรายละเอียดของแต่ละยูสเคสในภาคผนวก ง ตารางที่ ง.4 – ง.9



ภาพที่ 3.7 แผนภาพยูสเคสระบบย่อยจัดการ โปรแกรม และฐานข้อมูล

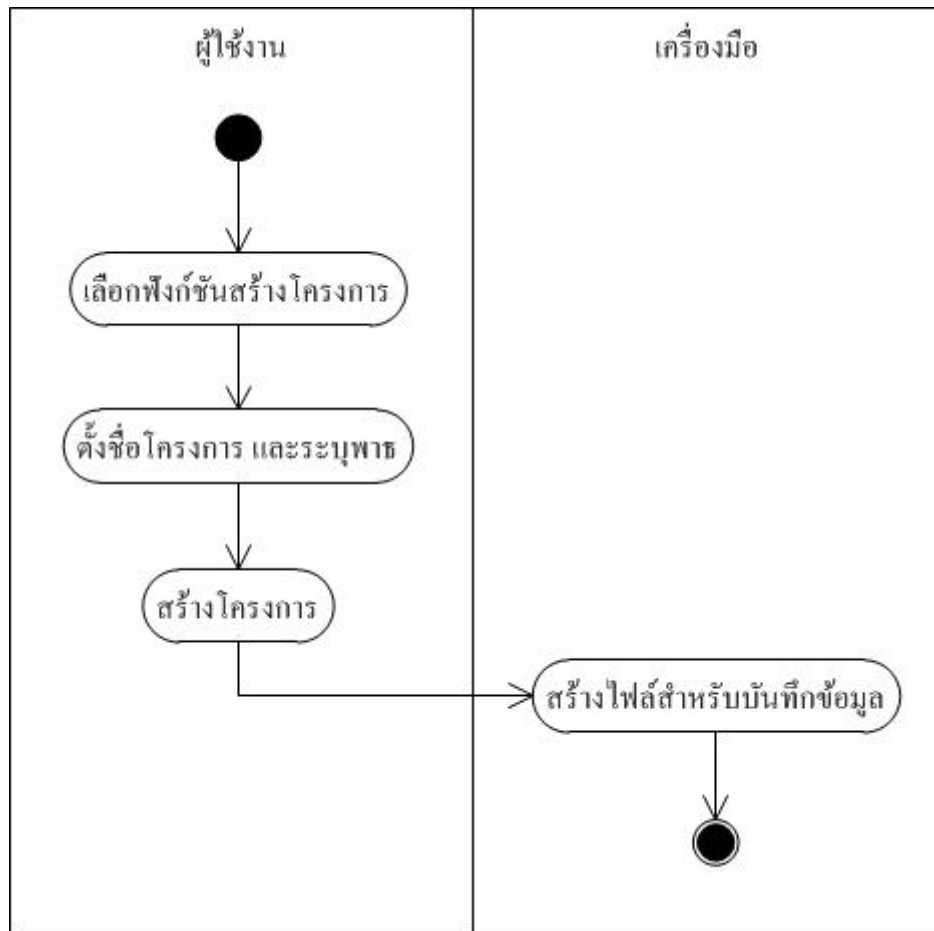
ระบบย่อยจัดการวิเคราะห์ตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง ประกอบไปด้วย 3 ยูสเคส ได้แก่ ยูสเคสวิเคราะห์หาตารางและคอลัมน์ที่ไม่ถูกอ้างอิง ยูสเคสหาคำสั่งสืบค้นที่ได้รับผลกระทบ และ ยูสเคสสร้างคำสั่งลบตาราง และคอลัมน์ ดังภาพที่ 3.8 และคำบรรยายรายละเอียดของแต่ละยูสเคส ในภาคผนวก ง ตารางที่ ง.10 – ง.12



ภาพที่ 3.8 แผนภาพยูสเคสระบบย่อยวิเคราะห์ตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง

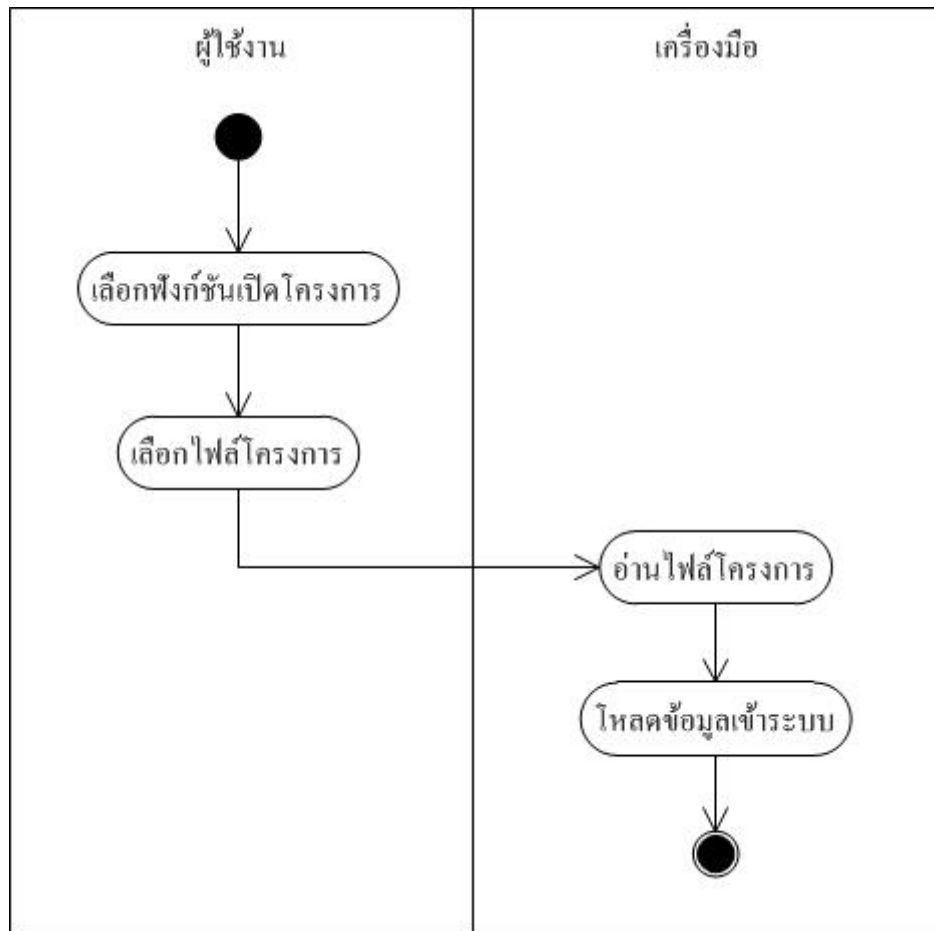
จะเห็นได้ว่ายูสเคสหาคำสั่งสืบค้นที่ได้รับผลกระทบ และยูสเคสสร้างคำสั่งลบตาราง และคอลัมน์ จำเป็นต้องเรียกใช้ยูสเคสวิเคราะห์หาตารางและคอลัมน์ที่ไม่ถูกอ้างอิงด้วย ถ้าผู้ใช้งานต้องการใช้ยูสเคสหาคำสั่งสืบค้นที่ได้รับผลกระทบ และยูสเคสสร้างคำสั่งลบตาราง และคอลัมน์ จะต้องใช้ยูสเคสวิเคราะห์หาตารางและคอลัมน์ที่ไม่ถูกอ้างอิงด้วย เพื่อให้เครื่องมือหารายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรมเสียก่อน

กิจกรรมต่างๆ ระหว่างผู้ใช้งานกับเครื่องมือ ในฟังก์ชันต่างๆ ทั้ง 12 ฟังก์ชัน มีลำดับกิจกรรมต่างๆ ระหว่างผู้ใช้งานกับเครื่องมือ โดยแบ่งตามฟังก์ชันงานต่างๆ ดังแสดงในแผนภาพกิจกรรมในภาพที่ 3.9 ถึง ภาพที่ 3.20 ดังนี้



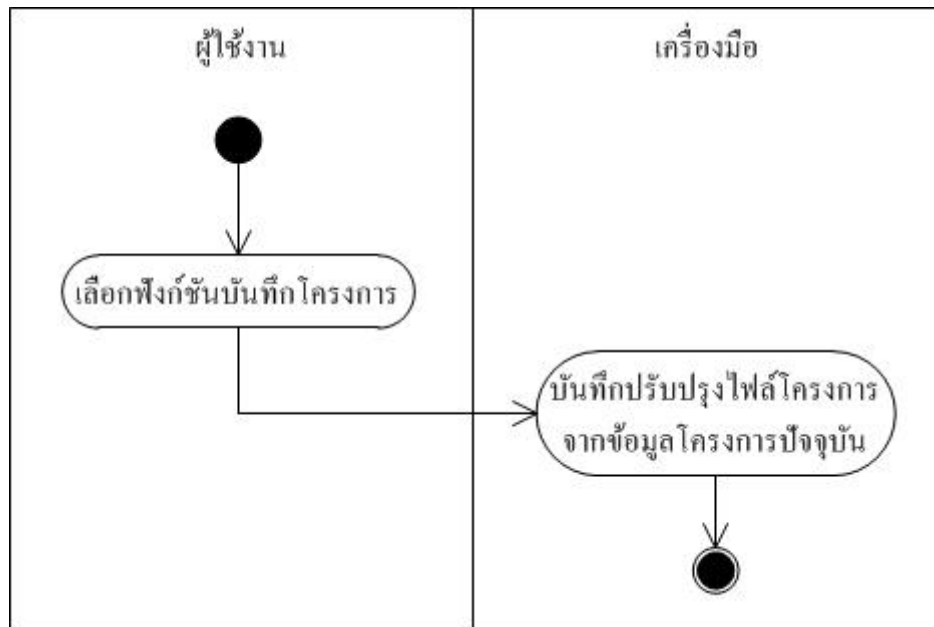
ภาพที่ 3.9 แผนภาพกิจกรรมขั้นตอนการสร้างโครงการ

ในภาพที่ 3.9 แสดงขั้นตอนการสร้างโครงการสำหรับใช้งานเครื่องมือ โดยผู้ใช้งานจะต้องเลือกฟังก์ชันสร้างโครงการจากเครื่องมือ จากนั้นตั้งชื่อโครงการ และระบุพาธสำหรับจัดเก็บไฟล์โครงการ และเลือกคำสั่งสร้างโครงการ เพื่อให้เครื่องมือสร้างไฟล์สำหรับจัดเก็บข้อมูลของไฟล์โครงการขึ้น และสร้างสภาพแวดล้อมให้แก่ผู้ใช้งานพร้อมที่จะใช้งานเครื่องมือต่อไป โดยรายละเอียดของไฟล์โครงการจะแสดงโครงสร้างของไฟล์ในบทที่ 4 ต่อไป



ภาพที่ 3.10 แผนภาพกิจกรรมขั้นตอนการเปิดโครงการ

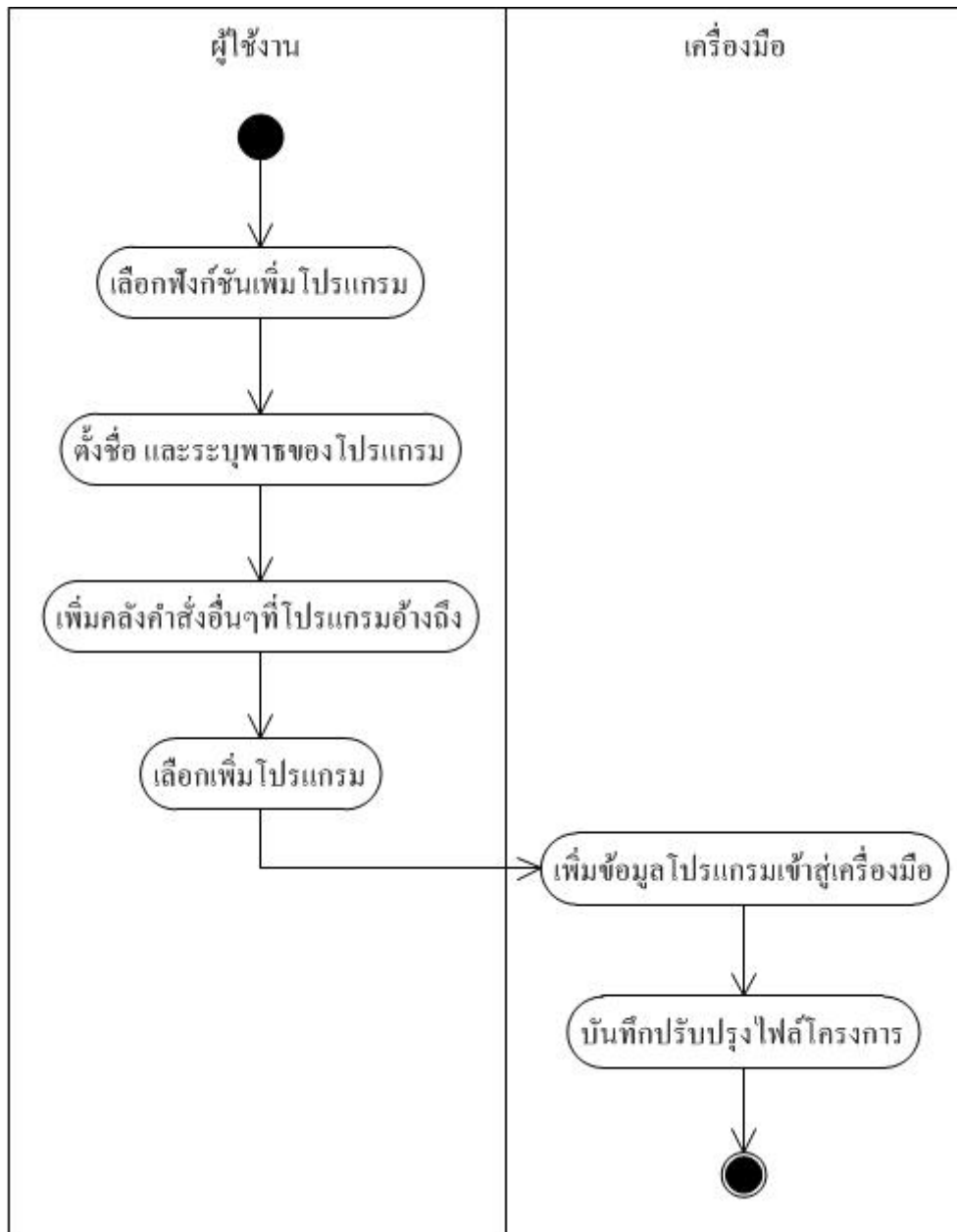
ในภาพที่ 3.10 แสดงขั้นตอนการเปิดไฟล์โครงการ โดยผู้ใช้งานจะเลือกฟังก์ชันเพื่อเปิดไฟล์โครงการ จากนั้นผู้ใช้งานจะต้องเลือกไฟล์โครงการที่เคยบันทึกไว้ เพื่อให้เครื่องมือเปิดไฟล์โครงการที่เคยบันทึกไว้ โดยเครื่องมือจะอ่านไฟล์โครงการ และสร้างสภาพแวดล้อมของเครื่องมือให้เหมือนกับสภาพแวดล้อมของโครงการในไฟล์ที่บันทึก เพื่อให้ผู้ใช้งานพร้อมที่จะใช้งานเครื่องมือ ต่อจากที่เคยบันทึกไว้



ภาพที่ 3.11 แผนภาพกิจกรรมขั้นตอนการบันทึกโครงการ

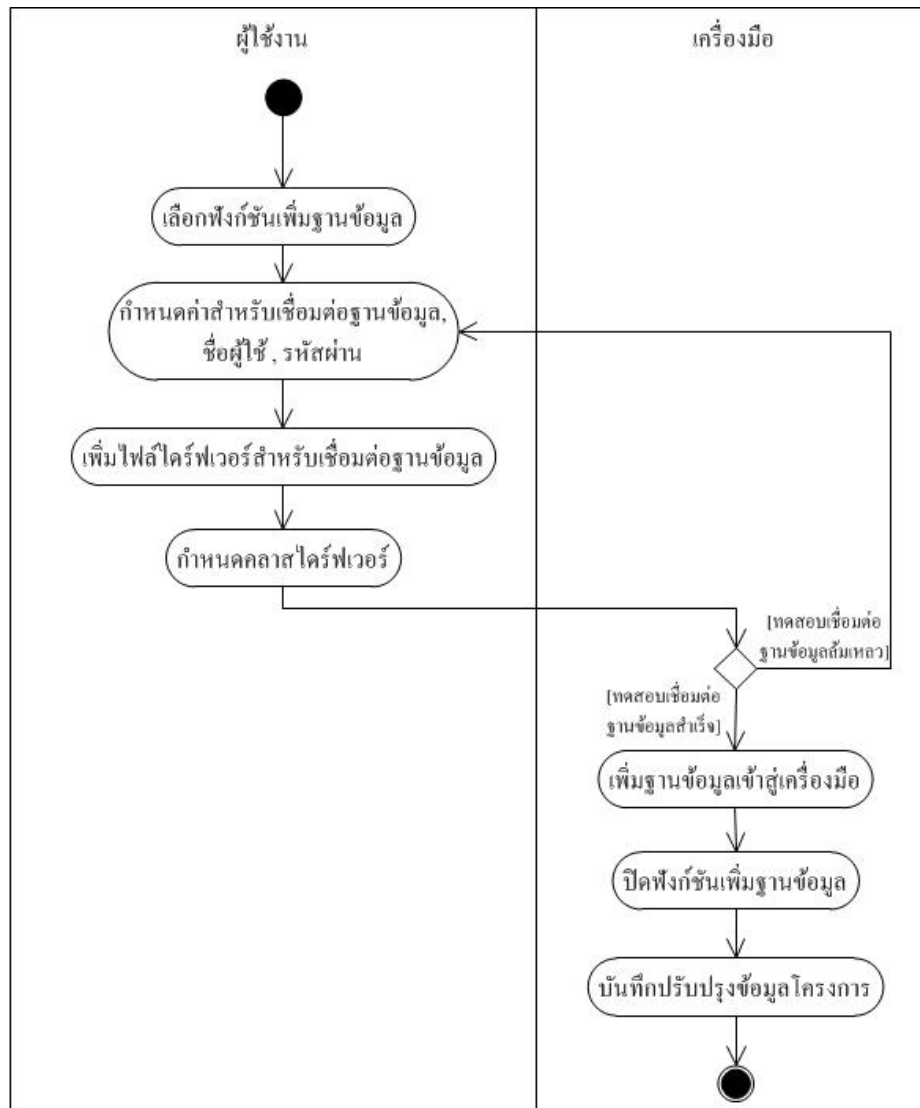
ในภาพที่ 3.11 แสดงขั้นตอนการบันทึกไฟล์โครงการ โดยผู้ใช้งานจะเลือกฟังก์ชันเพื่อ บันทึกไฟล์โครงการ จากนั้นเครื่องมือจะบันทึกข้อมูลใหม่ของโครงการลงไปในไฟล์

เมื่อเครื่องมือมีส่วนของโครงการสำหรับให้ผู้ใช้สามารถทำงานได้แล้ว ส่วนต่อมาก็คือขั้นตอน วิธีการนำเข้าข้อมูลสำหรับให้เครื่องมือนำไปวิเคราะห์ ได้แก่ นำเข้าโปรแกรม และนำเข้าฐานข้อมูล ซึ่งผู้วิจัยออกแบบให้รองรับการลบโปรแกรม และลบฐานข้อมูลจากโครงการด้วย ในกรณีที่ ผู้ใช้งานไม่ต้องการวิเคราะห์โปรแกรมที่นำเข้า หรือต้องการเปลี่ยนฐานข้อมูลที่จะวิเคราะห์ โดยใน แต่ละโครงการ การนำเข้าโปรแกรมจะสามารถนำเข้าได้หลายโปรแกรม ส่วนฐานข้อมูลจะอนุญาต ให้นำเข้าได้เพียงฐานข้อมูลเดียว ซึ่งแสดงรายละเอียดของแต่ละขั้นตอนดังในภาพที่ 3.12 – 3.15 ดังต่อไปนี้



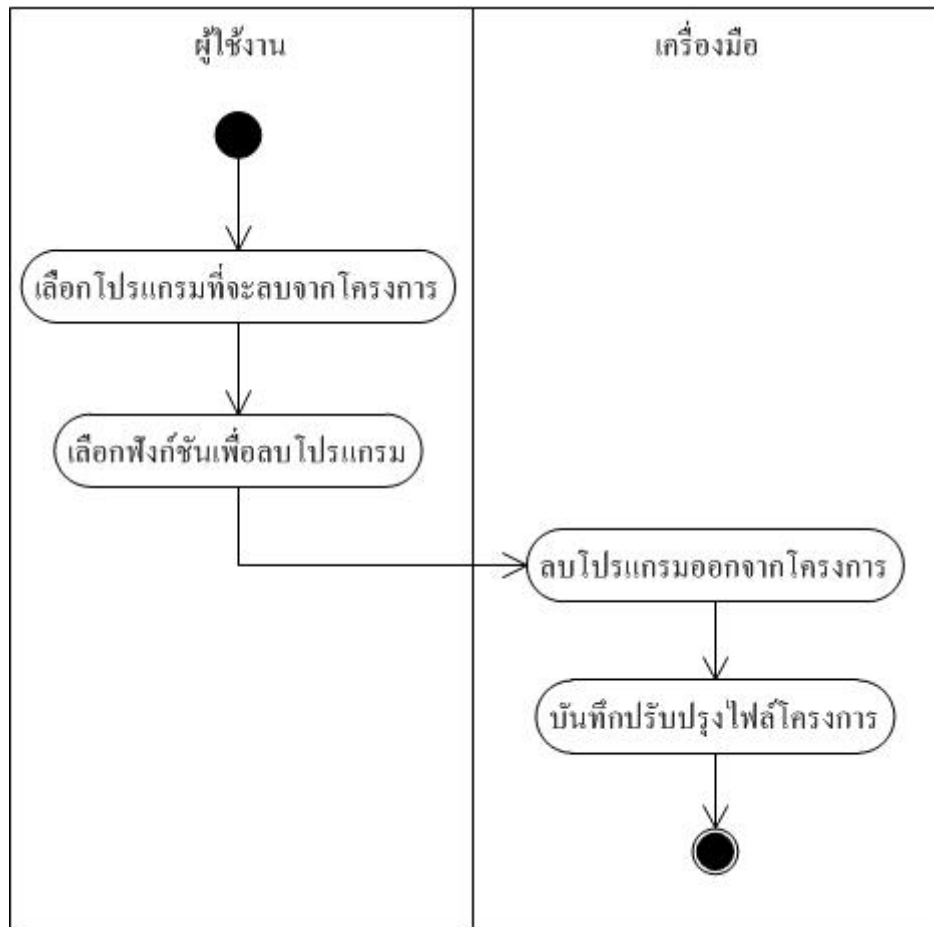
ภาพที่ 3.12 แผนภาพกิจกรรมขั้นตอนการเพิ่มโปรแกรมเข้าในโครงการ

ในภาพที่ 3.12 แสดงขั้นตอนการเพิ่มโปรแกรมเข้าสู่โครงการ โดยผู้ใช้งานจะต้องเลือกฟังก์ชันสำหรับเพิ่มโปรแกรม และจะต้องตั้งชื่อ โปรแกรม ระบุพารามิเตอร์ของโปรแกรม อีกทั้งต้องเพิ่มคำสั่งคำสั่งอื่นๆที่โปรแกรมใช้อ้างอิง เพื่อให้เครื่องมืออ่านและวิเคราะห์โปรแกรมได้ จากนั้นผู้ใช้งานใช้คำสั่งเพื่อเพิ่มโปรแกรม เครื่องมือจะเพิ่มข้อมูลโปรแกรมเข้าสู่โครงการ ได้แก่ พารามิเตอร์ของโปรแกรม ชื่อโปรแกรม คำสั่งคำสั่งอื่นๆที่โปรแกรมอ้างอิง และบันทึกปรับปรุงไฟล์โครงการ



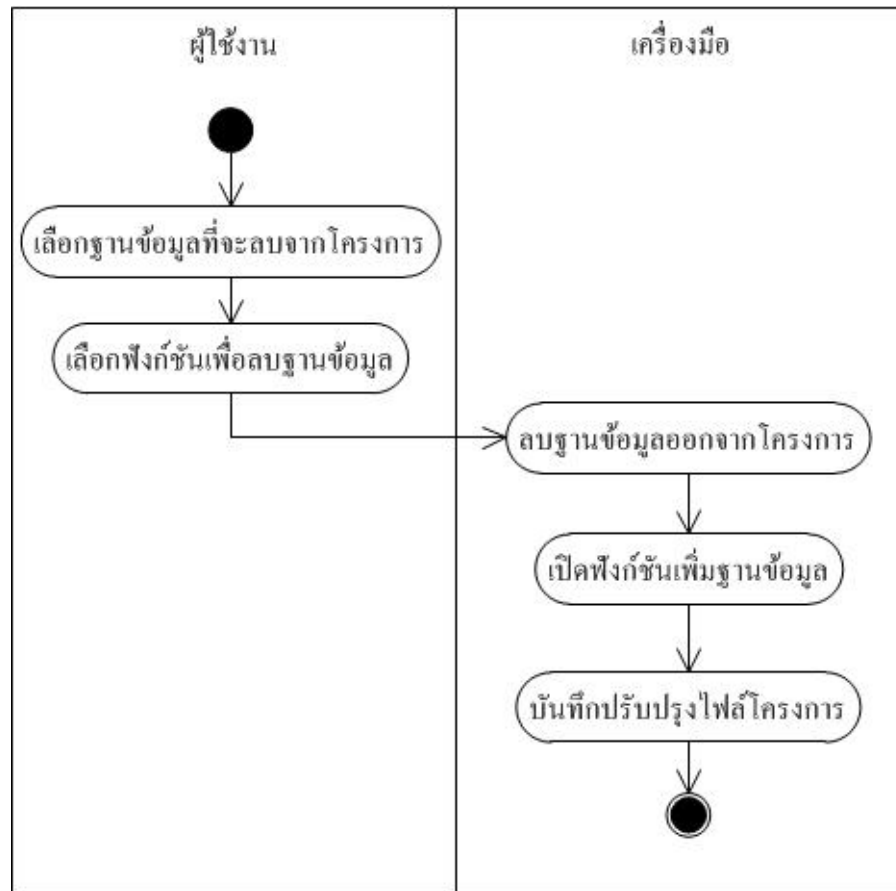
ภาพที่ 3.13 แผนภาพกิจกรรมขั้นตอนการเพิ่มฐานข้อมูลเข้าในโครงการ

ในภาพที่ 3.13 แสดงขั้นตอนการเพิ่มฐานข้อมูลเข้าสู่โครงการ กิจกรรมนี้จะสามารถทำได้ก็ต่อเมื่อโครงการยังไม่มีฐานข้อมูลที่ถูกเพิ่มเข้ามาเลย โดยผู้ใช้งานจะต้องเลือกฟังก์ชันสำหรับเพิ่มฐานข้อมูล และระบุค่าต่างๆ เช่น ชื่อผู้ใช้ รหัสผ่าน ชื่อคลาสไดรฟ์เวอร์ เป็นต้น อีกทั้งต้องระบุไฟล์ไดรฟ์เวอร์สำหรับเชื่อมต่อฐานข้อมูลด้วย จากนั้นจะต้องทดสอบการกำหนดค่าต่างๆ ว่าสามารถเชื่อมต่อฐานข้อมูลได้จริงหรือไม่ ซึ่งถ้าสามารถเชื่อมต่อฐานข้อมูลได้ เครื่องมือจะเพิ่มรายละเอียดฐานข้อมูล เช่น ชื่อคลาสไดรฟ์เวอร์ พาทไฟล์ไดรฟ์เวอร์ ข้อมูลต่างๆ สำหรับเชื่อมต่อฐานข้อมูลเข้าไปในโครงการ และบันทึกปรับปรุงไฟล์โครงการ



ภาพที่ 3.14 แผนภาพกิจกรรมขั้นตอนการลบโปรแกรมออกจากโครงการ

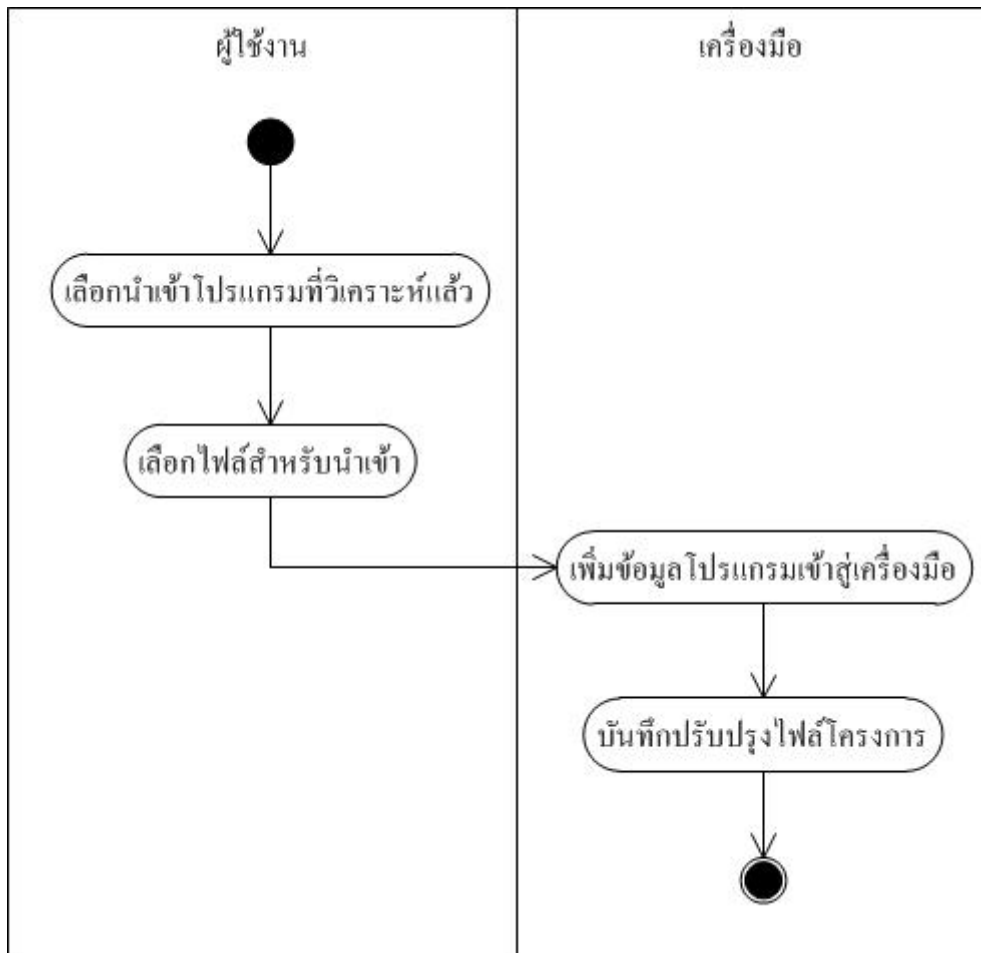
ในภาพที่ 3.14 แสดงขั้นตอนการลบโปรแกรมออกจากโครงการ ถ้าผู้ใช้งานไม่ต้องการที่จะวิเคราะห์โปรแกรม สามารถเลือกโปรแกรมที่ต้องการจะลบออกจากโครงการ และเลือกใช้คำสั่งเพื่อลบโปรแกรมออกจากโครงการ จากนั้นเครื่องมือจะลบพารามิเตอร์ของโปรแกรม รายละเอียดของโปรแกรม รวมถึงพารามิเตอร์คำสั่งต่างๆของโปรแกรมที่ผู้ใช้งานเลือกออกจากโครงการ และบันทึกปรับปรุงไฟล์ข้อมูลโครงการ



ภาพที่ 3.15 แผนภาพกิจกรรมขั้นตอนการลบฐานข้อมูลออกจากโครงการ

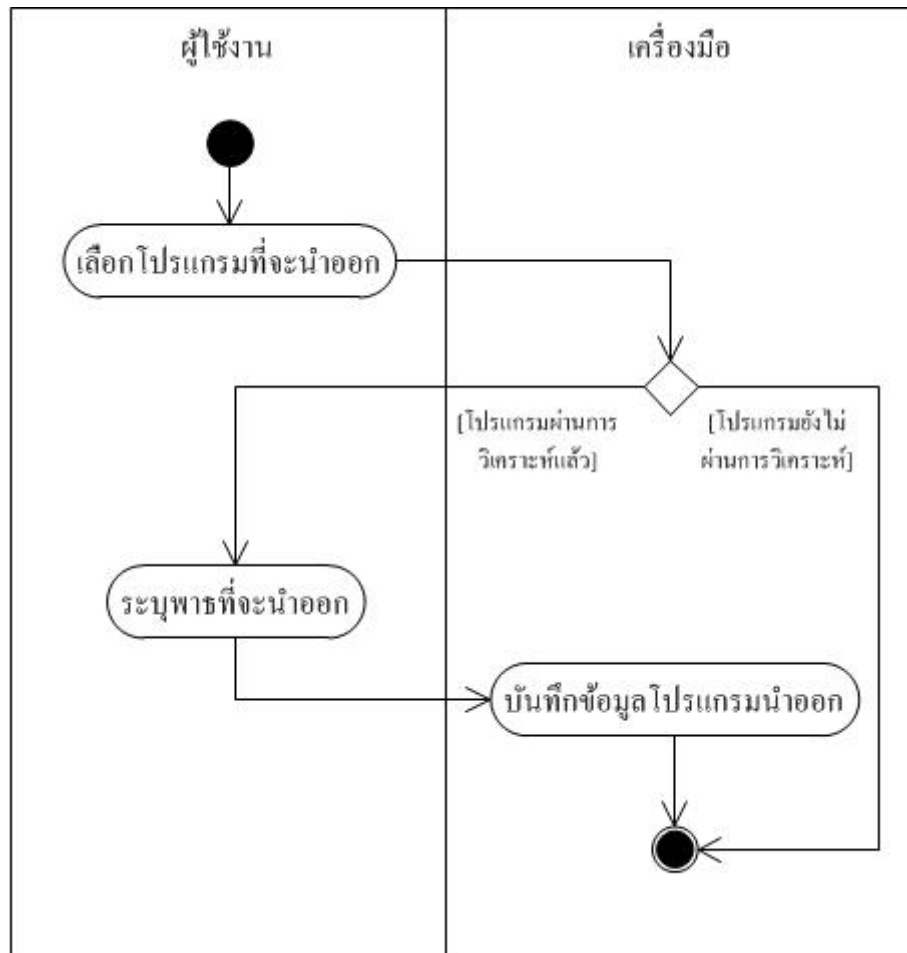
ในภาพที่ 3.15 แสดงขั้นตอนการลบฐานข้อมูลออกจากโครงการ ต้องการเปลี่ยนฐานข้อมูลใหม่ จะต้องทำการลบฐานข้อมูลเดิมจาก โครงการ ซึ่งสามารถเลือกฐานข้อมูลที่ต้องการจะลบ และเลือกใช้คำสั่งเพื่อลบฐานข้อมูลออกจาก โครงการ จากนั้นเครื่องมือจะลบฐานข้อมูลออกจากโครงการ แล้วจึงเปิดฟังก์ชันเพิ่มฐานข้อมูลให้ผู้ใช้สามารถเพิ่มฐานข้อมูลได้ใหม่ และบันทึกปรับปรุงไฟล์ข้อมูลโครงการ

นอกจากนี้ผู้วิจัยยังออกแบบให้เครื่องมือรองรับที่จะนำเข้าโปรแกรมเพิ่มเติมอีกวิธีหนึ่ง คือการนำเข้าโปรแกรมที่ผ่านการวิเคราะห์แล้ว เนื่องจากโปรแกรมเหล่านี้ถ้าไม่มีการเปลี่ยนแปลงซอร์สโค้ด ตาราง และคอมเมนต์ที่โปรแกรมอ้างอิงถึง จะไม่เปลี่ยนแปลง จึงไม่จำเป็นต้องวิเคราะห์โปรแกรมใหม่ ผู้วิจัยจึงออกแบบเครื่องมือ ให้รองรับที่จะนำเข้า และนำออกโปรแกรมที่ผ่านการวิเคราะห์แล้ว ซึ่งแสดงรายละเอียดการทำงานดังแผนภาพกิจกรรมภาพที่ 3.16 และ 3.17



ภาพที่ 3.16 แผนภาพกิจกรรมขั้นตอนการนำเข้าโปรแกรมที่วิเคราะห์แล้ว

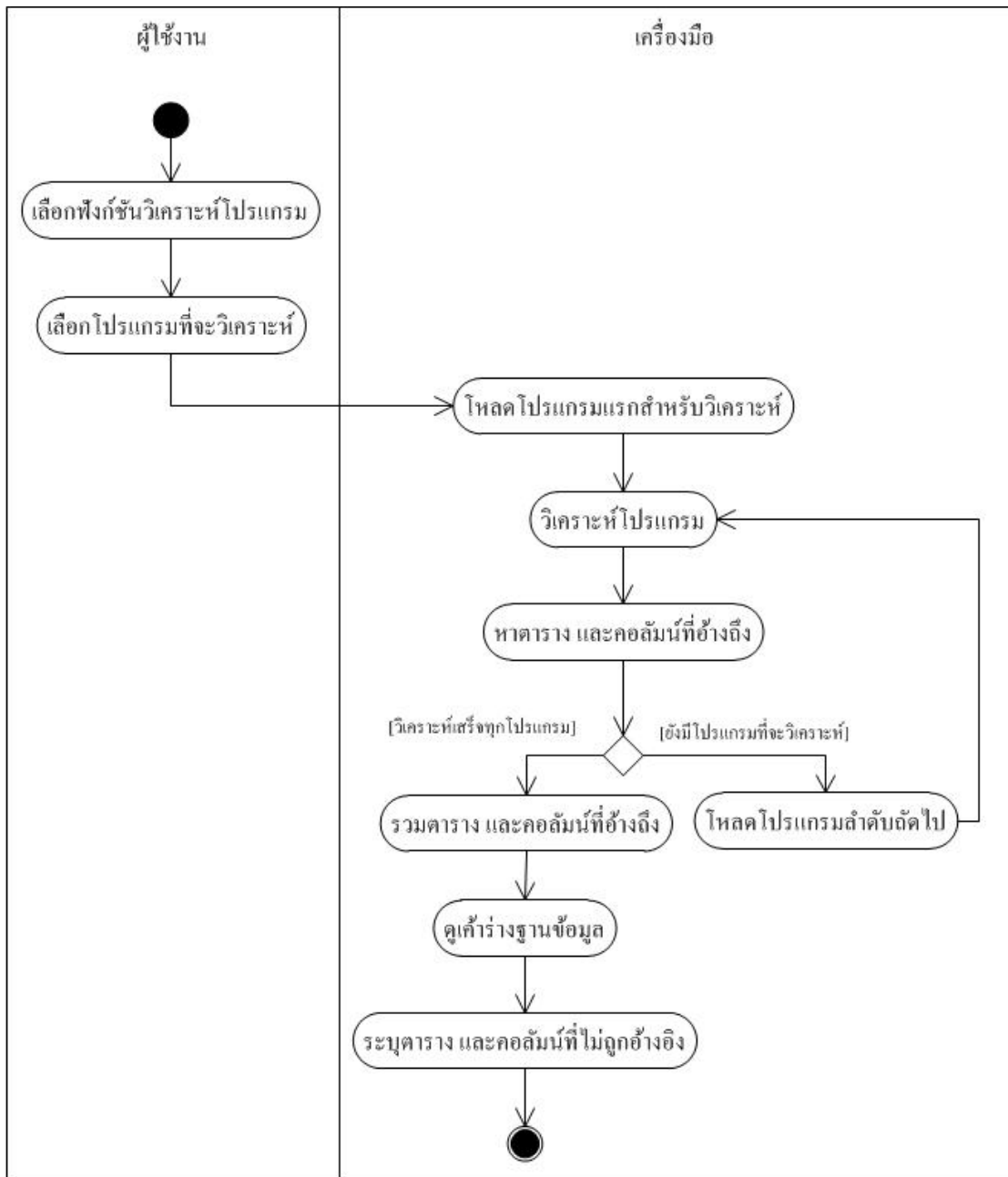
ในภาพที่ 3.16 แสดงขั้นตอนการนำเข้าโปรแกรมที่วิเคราะห์แล้วเข้าสู่โครงการ โดยผู้ใช้งานจะต้องเลือกฟังก์ชันสำหรับนำเข้าโปรแกรมที่ผ่านการวิเคราะห์แล้ว จากนั้นเลือกไฟล์ของโปรแกรมที่ผ่านการวิเคราะห์แล้ว จากนั้นเครื่องมือจะอ่านค่าไฟล์ดังกล่าว และเพิ่มเป็นโปรแกรมเข้าสู่โครงการ และบันทึกปรับปรุงไฟล์ข้อมูลโครงการ



ภาพที่ 3.17 แผนภาพกิจกรรมขั้นตอนการนำออกโปรแกรมที่วิเคราะห์แล้ว

ในภาพที่ 3.17 แสดงขั้นตอนการนำออกโปรแกรมที่วิเคราะห์เพื่อนำไปใช้ในโครงการอื่นๆ โดยผู้ใช้งานจะเลือกโปรแกรมที่ผ่านการวิเคราะห์แล้ว และระบุพารสำหรับจัดเก็บไฟล์ผลลัพธ์ของโปรแกรมที่วิเคราะห์แล้ว จากนั้นระบบจะสร้างไฟล์ของโปรแกรมที่ผ่านการวิเคราะห์ไปยังพารที่ผู้ใช้งานระบุไว้

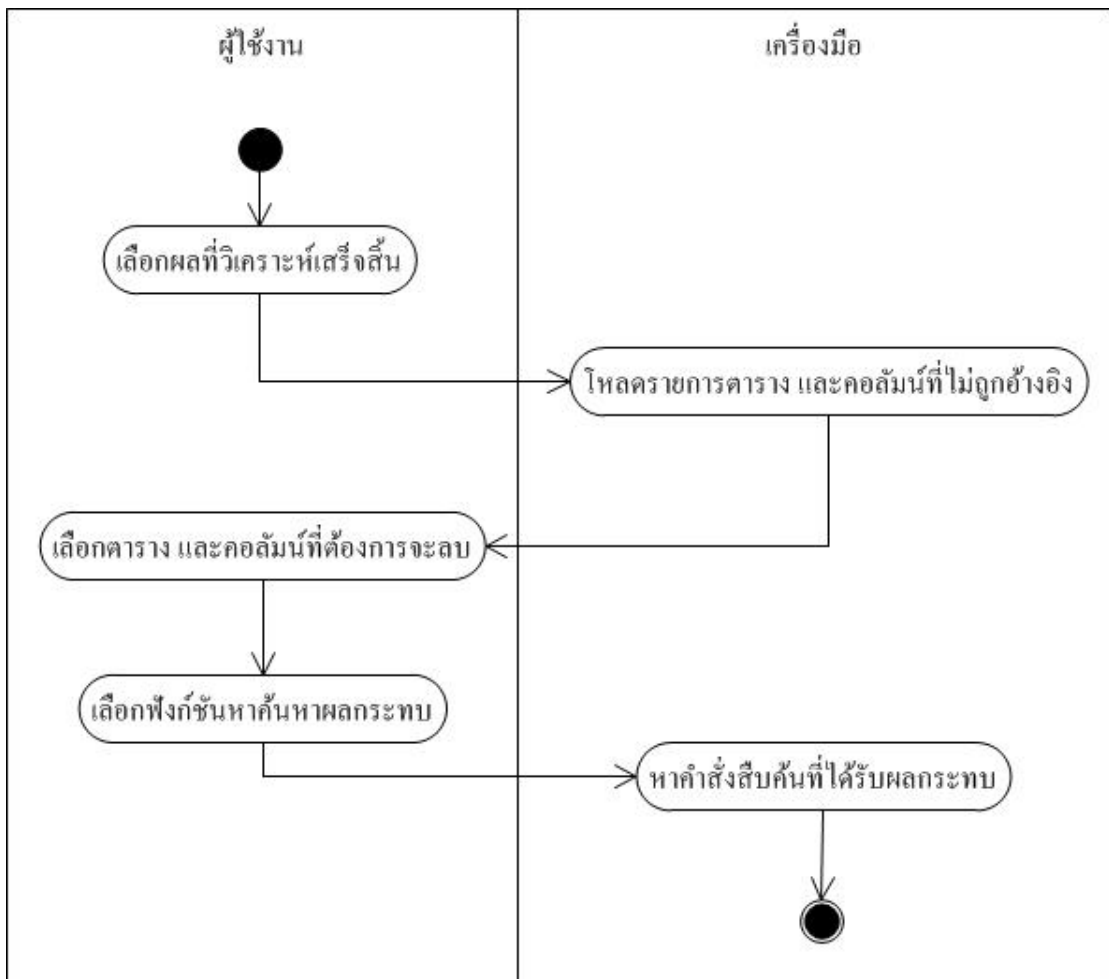
ส่วนการทำงานหลักของเครื่องมืออันประกอบไปด้วย การวิเคราะห์เพื่อระบุตาราง และคอลัมน์ที่โปรแกรมต่างๆ ไม่ได้อ้างอิง การวิเคราะห์ผลกระทบที่จะเกิดขึ้นกับคำสั่งสืบค้นในแต่ละโปรแกรม และการสร้างคำสั่งภาษาแอสควิเอลมาตรฐาน สำหรับลบบตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิง ซึ่งแสดงรายละเอียดขั้นตอนในแผนภาพกิจกรรมดังภาพที่ 3.18 – 3.20 ดังนี้



ภาพที่ 3.18 แผนภาพกิจกรรมขั้นตอนการวิเคราะห์เพื่อระบุ
ตารางและคอลลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม

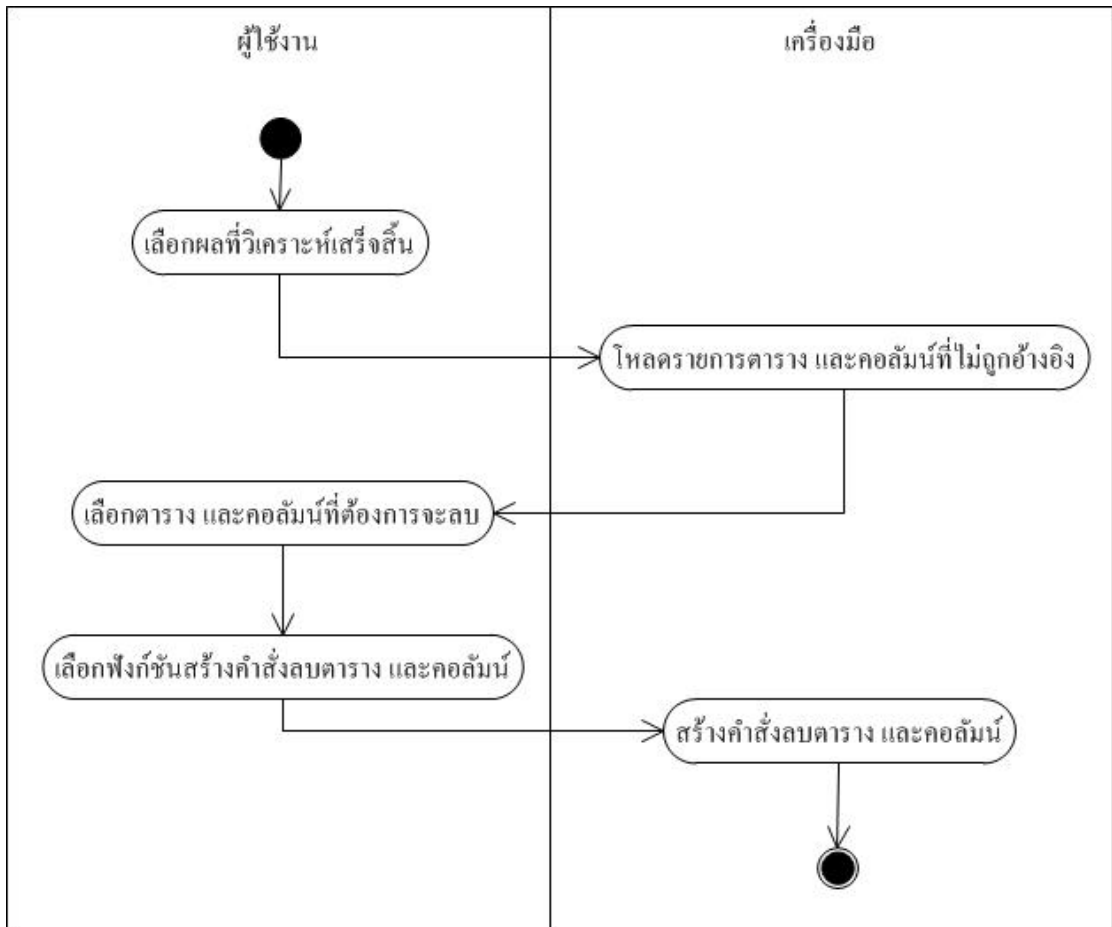
ในภาพที่ 3.18 แสดงขั้นตอนการวิเคราะห์เพื่อระบุตาราง และคอลลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม โดยผู้ใช้งานจะต้องเลือกโปรแกรมจากรายการโปรแกรมทั้งหมดในโครงการที่จะวิเคราะห์ จากนั้นเครื่องมือจะวิเคราะห์โปรแกรมทีละโปรแกรมจากรายการที่ผู้ใช้เลือก เพื่อหาว่า

แต่ละโปรแกรมมีการอ้างอิงถึงตาราง และคอลัมน์ใดบ้าง ซึ่งเมื่อวิเคราะห์เสร็จ เครื่องมือจะรวมผลลัพธ์ที่ได้เข้าไว้ด้วยกัน คือรายการตาราง และคอลัมน์ที่โปรแกรมทั้งหมดอ้างอิง จากนั้นเครื่องมือจะดูเค้าร่างฐานข้อมูลจริง และเปรียบเทียบกับผลที่ได้จากการวิเคราะห์โปรแกรม ว่ามีความต่างกันอย่างไร เพื่อระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิงถึง



ภาพที่ 3.19 แผนภาพกิจกรรมขั้นตอนการวิเคราะห์หาค่าตั้งสืบค้นที่ได้รับผลกระทบ

ในภาพที่ 3.19 แสดงขั้นตอนการวิเคราะห์หาค่าตั้งสืบค้นที่ได้รับผลกระทบ ซึ่งกิจกรรมนี้เป็นกิจกรรมที่เกิดขึ้นหลังจากวิเคราะห์เพื่อระบุตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม โดยผู้ใช้งานจะเลือกรายการตาราง และคอลัมน์จากรายการที่เครื่องมือแสดง ว่าไม่มีโปรแกรมอ้างอิง โดยรายการที่เลือกคือตาราง และคอลัมน์ที่ต้องการที่จะลบออกจากฐานข้อมูล จากนั้นผู้ใช้งานเลือกใช้งานฟังก์ชันค้นหาผลกระทบ ระบบจะหาค่าตั้งสืบค้นในที่ได้รับผลกระทบ

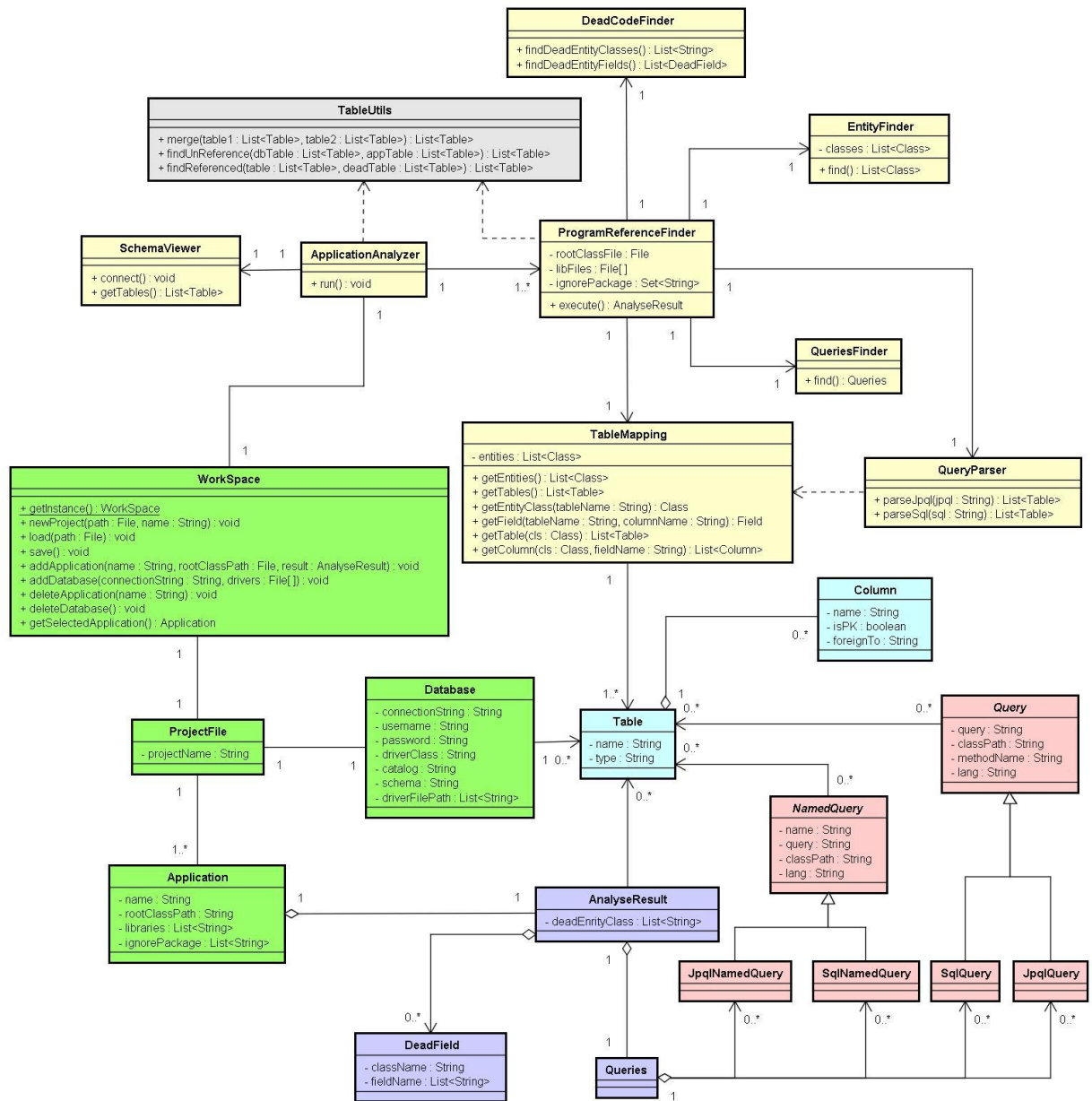


ภาพที่ 3.20 แผนภาพกิจกรรมขั้นตอนการสร้างคำสั่งสำหรับลบตารางและคอลัมน์

ในภาพที่ 3.20 แสดงขั้นตอนการสร้างคำสั่งสำหรับลบตารางและคอลัมน์ ซึ่งกิจกรรมนี้เป็นกิจกรรมที่เกิดขึ้นหลังจากวิเคราะห์เพื่อระบุตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม โดยผู้ใช้งานจะเลือกรายการตาราง และคอลัมน์จากรายการที่เครื่องมือแสดง ว่าไม่มีโปรแกรมอ้างอิง โดยรายการที่เลือกคือตาราง และคอลัมน์ที่ต้องการที่จะลบออกจากฐานข้อมูล จากนั้นผู้ใช้งานเลือกใช้งานฟังก์ชันสร้างคำสั่งลบตาราง และคอลัมน์ ระบบจะนำรายการตาราง และคอลัมน์จากที่ผู้ใช้เลือกมาสร้างเป็นคำสั่งภาษาเอสควิลมาตรฐานสำหรับลบตาราง และคอลัมน์

3.2.3 โครงสร้างของเครื่องมือ

เครื่องมือต้นแบบของงานวิจัยมีโครงสร้างเครื่องมือดังแผนภาพคลาสในภาพที่ 3.21 โดยในแต่ละคลาส จะแสดงถึงแอตทริบิวต์ และการดำเนินการ (Operation) ที่สำคัญเท่านั้น ซึ่งหน้าที่ของคลาสในเครื่องมือ มีหน้าที่ดังตารางที่ 3.1



ภาพที่ 3.21 แผนภาพคลาสแสดงโครงสร้างของเครื่องมือ

ตารางที่ 3.1 หน้าที่ของคลาสในเครื่องมือ

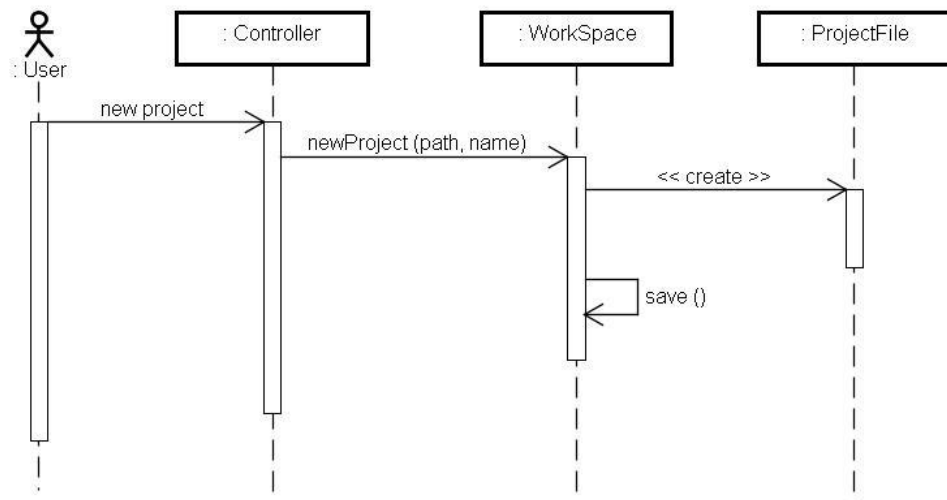
ชื่อคลาส	หน้าที่
Table	ใช้เก็บชื่อตาราง และประเภทของตาราง
Column	ใช้เก็บชื่อคอลัมน์ คอลัมน์นี้เป็นคีย์หลักหรือไม่ หรือเป็นคีย์นอกไปยังตารางใด
Query	เป็นคลาสนามธรรม ซึ่งใช้เก็บข้อมูลคำสั่งสืบค้น คำสั่งสืบค้นปรากฏที่คลาส และเมธอดใดของโปรแกรม และตารางรวมไปถึงคอลัมน์ที่คำสั่งนั้นๆเกี่ยวข้อง
JpqlJuery	สืบทอดมาจากคลาส Query เพื่อใช้เก็บคำสั่งภาษาเจพีกิวแอล
SqlQuery	สืบทอดมาจากคลาส Query เพื่อใช้เก็บคำสั่งภาษาเอสคิวแอล
NamedQuery	เป็นคลาสนามธรรม ซึ่งใช้เก็บข้อมูลคำสั่งสืบค้นแบบระบุชื่อคำสั่งสืบค้น โดยเก็บชื่อของคำสั่ง และเอนติตี้คลาสของโปรแกรมที่ประกาศคำสั่ง รวม ไปถึงตาราง และคอลัมน์ที่คำสั่งเกี่ยวข้อง
JpqlNamedQuery	สืบทอดมาจากคลาส NamedQuery เพื่อใช้เก็บคำสั่งภาษาเจพีกิวแอล
SqlNamedQuery	สืบทอดมาจากคลาส NamedQuery เพื่อใช้เก็บคำสั่งภาษาเอสคิวแอล
AnalyseResult	เก็บผลการวิเคราะห์โปรแกรม โดยแต่ละโปรแกรมจะมีวัตถุนี้เป็นของตัวเองหลังจากที่วิเคราะห์โปรแกรมเสร็จ
DeadFiled	เป็นส่วนหนึ่งของคลาส AnalyseResult ซึ่งจะเก็บข้อมูลของแอตทริบิวต์ในคลาสต่างๆที่เข้าไม่ถึง
Queries	เป็นส่วนหนึ่งของคลาส AnalyseResult ซึ่งจะเก็บว่าในแต่ละโปรแกรมมีคำสั่งสืบค้นใดบ้าง
WorkSpace	เป็นคลาสที่มีฟังก์ชันต่างๆ และมีความสัมพันธ์กับคลาสต่างๆในโปรแกรม ทั้งทางตรง และทางอ้อม สำหรับให้ผู้ใช้งานเรียกทำงานฟังก์ชันต่างๆได้โดยผ่านคลาสนี้
ProjectFile	เก็บข้อมูลของโครงการ ฐานข้อมูล และรายการของโปรแกรม
Database	เก็บรายละเอียดของฐานข้อมูล
Application	เก็บรายละเอียดของโปรแกรมแต่ละโปรแกรม

ตารางที่ 3.1 หน้าทีของคลาสในเครื่องมือ (ต่อ)

ชื่อคลาส	หน้าที่
ApplicationAnalyzer	เป็นคลาสหลักสำหรับฟังก์ชันวิเคราะห์ตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม โดยคลาสนี้จะเรียกใช้งานคลาสต่างๆ ที่เกี่ยวข้องเพื่อระบุตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม
ProgramReferenceFinder	หาตาราง และคอลัมน์ที่โปรแกรมแต่ละโปรแกรมอ้างอิง
EntityFinder	หาคลาสเอนติตี้ในโปรแกรม
TableMapping	สร้างความสัมพันธ์ระหว่างคลาสเอนติตี้ กับตารางและคอลัมน์ของฐานข้อมูล โดยสร้างความสัมพันธ์จากบรรณนิทัศน์
DeadCodeFinder	หาเอนติตี้คลาส และแอตทริบิวต์ของเอนติตี้ ที่ไม่ถูกเรียกใช้งานในส่วนอื่นๆของโปรแกรม
QueriesFinder	หาคำสั่งสืบค้นทุกรูปแบบ ที่ปรากฏในโปรแกรม
QueryParser	หาความสัมพันธ์ของคำสั่งสืบค้นทั้งภาษาเจพีคิวแอล และเอสคิวเอลว่าเกี่ยวข้องกับตาราง และคอลัมน์ฐานข้อมูลใดบ้าง
SchemeViewer	เชื่อมต่อฐานข้อมูลจริง เพื่อดูเค้าร่าง ตาราง และคอลัมน์จริงในฐานข้อมูล
TableUtils	เป็นคลาสที่ช่วยรวมผลลัพธ์รายการตาราง และหาผลต่างของรายการตาราง เพื่อช่วยในการระบุตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม

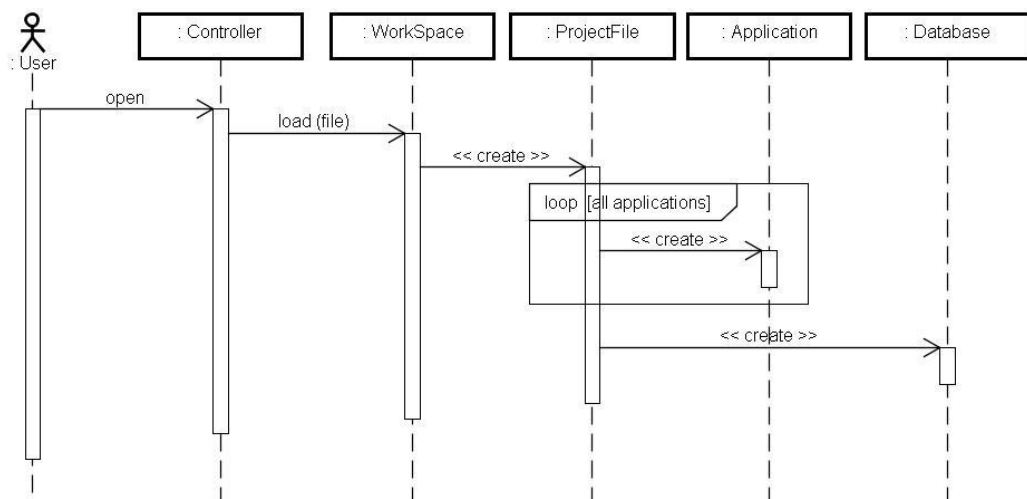
3.2.4 พฤติกรรมของเครื่องมือ

เครื่องมือต้นแบบที่พัฒนาจะต้องมีฟังก์ชันการทำงาน รวม 12 ฟังก์ชันดังที่ปรากฏในยูสเคส ซึ่งแต่ละฟังก์ชันงาน เครื่องมือจะมีขั้นตอนการทำงานที่แตกต่างกันออกไป ในงานวิจัยนี้ ผู้วิจัยจึงนำเสนอพฤติกรรมการทำงานของเครื่องมือในรูปแบบของแผนภาพลำดับในภาพที่ 3.22 – 3.33 โดยในแผนภาพลำดับจะแสดงถึงความสัมพันธ์ระหว่างวัตถุของเครื่องมือ ว่าพฤติกรรมการทำงานของฟังก์ชันต่างๆ วัตถุแต่ละวัตถุมีการเรียกใช้กันอย่างไร



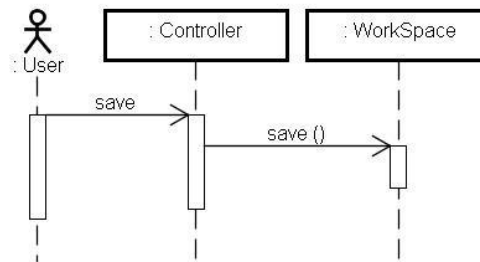
ภาพที่ 3.22 แผนภาพลำดับพฤติกรรมเครื่องมือขณะสร้างโครงการ

ในภาพที่ 3.22 เมื่อมีการสร้างโครงการ เครื่องมือจะรับค่าชื่อโครงการ และพาธสำหรับเก็บไฟล์โครงการ แล้วสร้างวัตถุของไฟล์โครงการขึ้นในระบบ จากนั้นจะบันทึกไฟล์โครงการลงในพาธของโครงการ



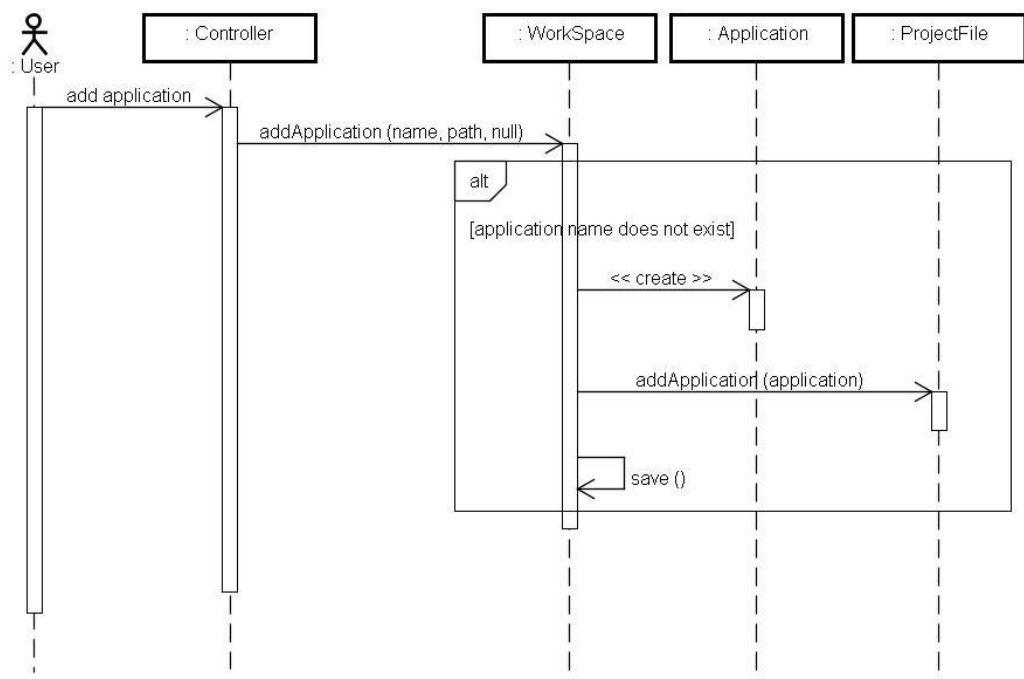
ภาพที่ 3.23 แผนภาพลำดับพฤติกรรมเครื่องมือขณะเปิดโครงการ

ในภาพที่ 3.23 เมื่อมีการเปิดโครงการ เครื่องมือจะสร้างวัตถุของไฟล์โครงการสำหรับเตรียมเก็บข้อมูลต่างๆของโครงการ จากนั้นจะวนสร้างวัตถุโปรแกรมต่างๆ ที่มีในไฟล์โครงการเก่า และสุดท้ายคือสร้างวัตถุของฐานข้อมูลเพื่อเก็บข้อมูลในหน่วยความจำ และพร้อมที่จะให้ผู้ใช้ทำงาน



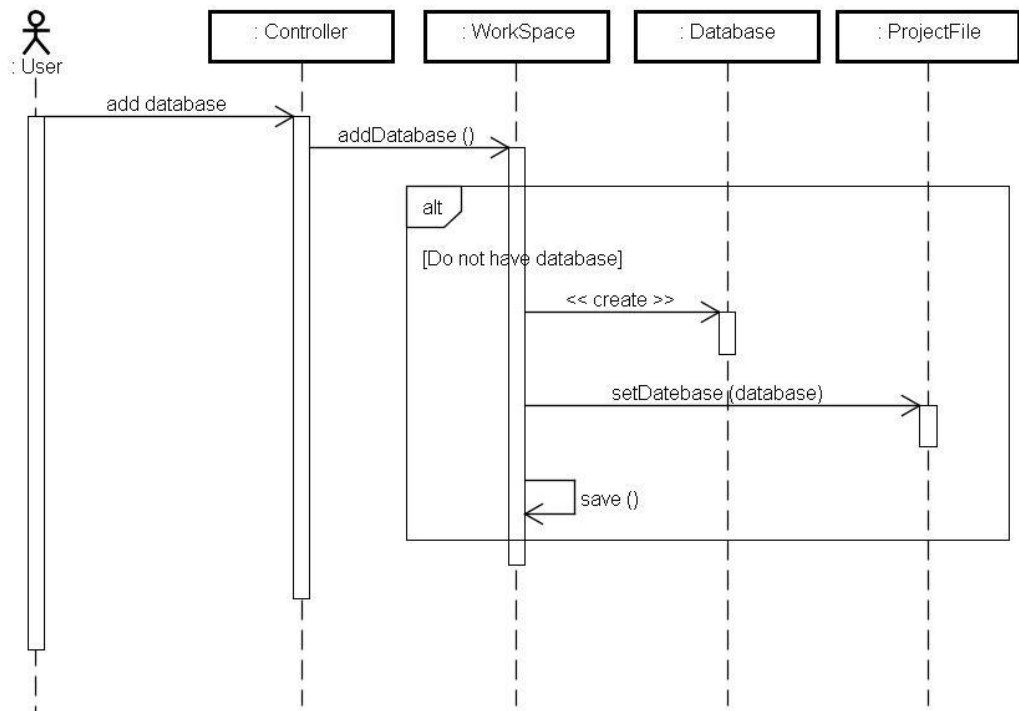
ภาพที่ 3.24 แผนภาพลำดับพฤติกรรมเครื่องมือขณะบันทึกโครงการ

ในภาพที่ 3.24 เมื่อมีการบันทึกโครงการ เครื่องมือจะนำข้อมูลที่มีอยู่ในวัตถุของโครงการขณะที่ผู้ใช้งานต้องการบันทึก เขียนทับไฟล์โครงการเดิม



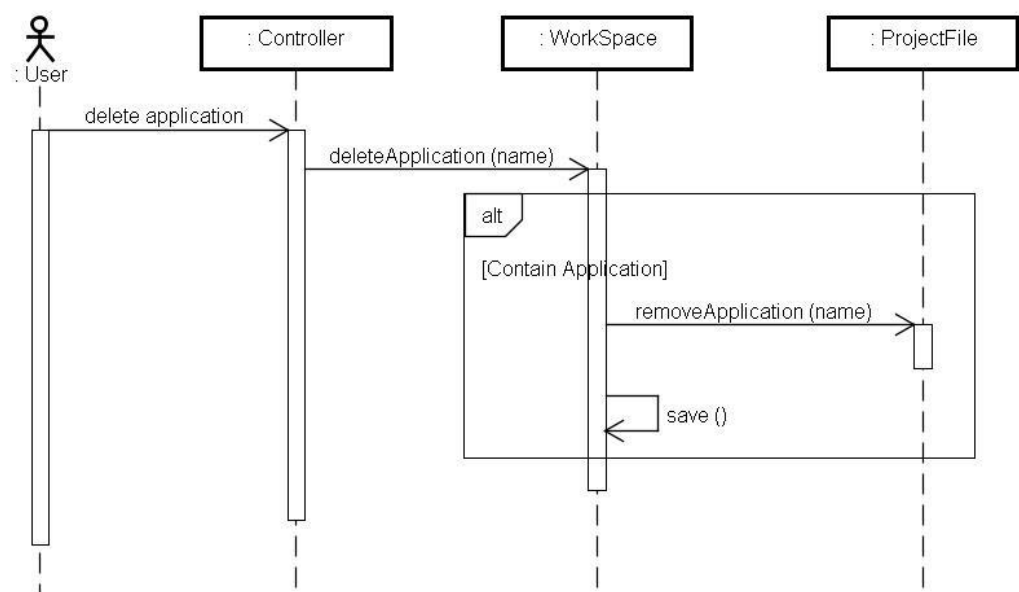
ภาพที่ 3.25 แผนภาพลำดับพฤติกรรมเครื่องมือขณะเพิ่ม โปรแกรมเพื่อวิเคราะห์

ในภาพที่ 3.25 เมื่อมีการเพิ่มโปรแกรมเพื่อวิเคราะห์ เครื่องมือรับค่าชื่อโปรแกรม และพารามิเตอร์ของโปรแกรมจากเมธอด “addApplication” แต่ค่าตัวสุดท้ายคือค่าว่าง เนื่องจากโปรแกรมที่เพิ่งเพิ่มเข้ามาใหม่ยังไม่มีผลการวิเคราะห์ โปรแกรม จากนั้นเครื่องมือตรวจสอบก่อนว่าชื่อโปรแกรมมีอยู่แล้วในโครงการหรือไม่ ซึ่งถ้าไม่มีเครื่องมือจะสร้างวัตถุของโปรแกรมตามค่าที่ได้รับ และเพิ่มโปรแกรมเข้าไปในวัตถุไฟล์โครงการ และบันทึกการเปลี่ยนแปลงของวัตถุ



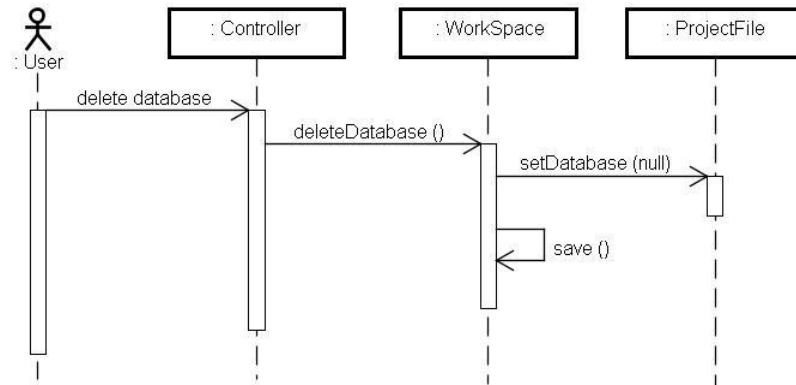
ภาพที่ 3.26 แผนภาพลำดับพฤติกรรมเครื่องมือขณะเพิ่มฐานข้อมูลเพื่อวิเคราะห์

ในภาพที่ 3.26 เมื่อมีการเพิ่มฐานข้อมูลเพื่อวิเคราะห์ เครื่องมือรับค่าต่าง ๆ ที่ใช้กำหนดการเชื่อมต่อฐานข้อมูล เช่นชื่อผู้ใช้ และรหัสผ่านเป็นต้น จากนั้นเครื่องมือจะตรวจสอบก่อนว่าในโครงการมีวัตถุฐานข้อมูลอยู่แล้วหรือไม่ ถ้าไม่มีเครื่องมือจะสร้างวัตถุฐานข้อมูลเพื่อเก็บค่าฐานข้อมูล และเพิ่มวัตถุฐานข้อมูลเข้าไปในวัตถุไฟล์โครงการ จากนั้นจึงบันทึกการเปลี่ยนแปลง



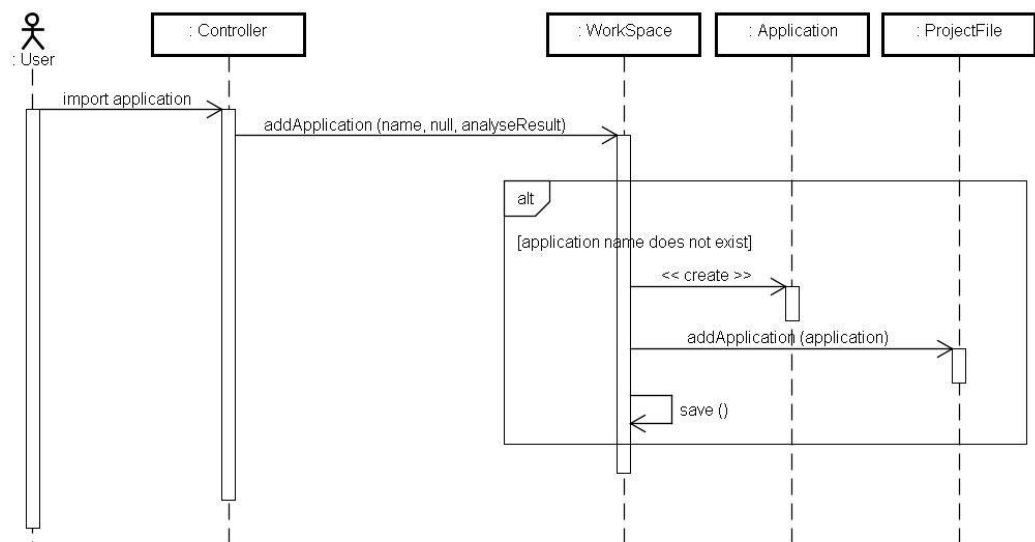
ภาพที่ 3.27 แผนภาพลำดับแสดงพฤติกรรมเครื่องมือขณะลบโปรแกรม

ในภาพที่ 3.27 เมื่อมีการลบโปรแกรม เครื่องมือรับค่าชื่อของโปรแกรมที่ต้องการจะลบ จากนั้นจะตรวจสอบจากวัตถุของโปรแกรมว่ามีโปรแกรมชื่อดังกล่าวหรือไม่ ถ้ามีโปรแกรมชื่อดังกล่าว จะลบโปรแกรมชื่อนั้นออกจากรายการ โปรแกรมของวัตถุไฟล์โครงการ จากนั้นบันทึกการเปลี่ยนแปลงของไฟล์วัตถุ



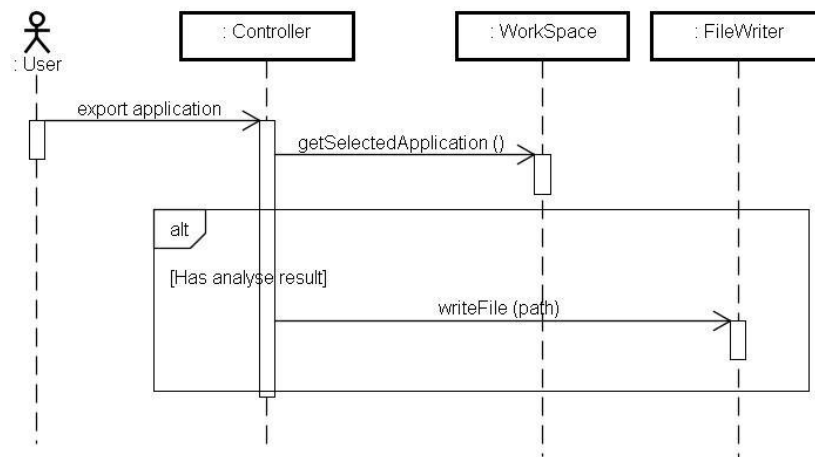
ภาพที่ 3.28 แผนภาพลำดับพฤติกรรมเครื่องมือขณะลบฐานข้อมูล

ในภาพที่ 3.28 เมื่อมีการลบฐานข้อมูล เครื่องมือจะตั้งค่าฐานข้อมูลในวัตถุของไฟล์โครงการเป็นค่าว่าง นั่นคือกำหนดให้วัตถุของไฟล์โครงการไม่มีวัตถุของฐานข้อมูล หรือคือการลบวัตถุของฐานข้อมูลออกไปจากไฟล์โครงการ จากนั้นบันทึกการเปลี่ยนแปลงไฟล์วัตถุ



ภาพที่ 3.29 แผนภาพลำดับพฤติกรรมเครื่องมือขณะนำเข้าโปรแกรมที่วิเคราะห์แล้ว

ในภาพที่ 3.29 เมื่อมีการนำเข้าโปรแกรมที่วิเคราะห์แล้ว เครื่องมือจะมีพฤติกรรมการทำงาน คล้ายกับการเพิ่ม โปรแกรมเพื่อวิเคราะห์ แต่จะต่างกันตรงที่ข้อมูลที่รับเข้ามาจะเป็นชื่อ โปรแกรม และผลของโปรแกรมที่ผ่านการวิเคราะห์แล้ว ส่วนพาชของโปรแกรมนั้นจะเป็นค่าว่าง เนื่องจากไม่ จำเป็นต้องวิเคราะห์โปรแกรมนี้แล้ว สามารถนำผลการวิเคราะห์ไปใช้ต่อได้ เมื่อรับข้อมูลดังกล่าว แล้ว เครื่องมือตรวจสอบก่อนว่าชื่อ โปรแกรมมีอยู่แล้วใน โครงการหรือไม่ ซึ่งถ้าไม่มีเครื่องมือจะ สร้างวัตถุของโปรแกรมตามค่าที่ได้รับ และเพิ่ม โปรแกรมเข้าไปในวัตถุไฟล์โครงการ และบันทึก การเปลี่ยนแปลงของวัตถุ

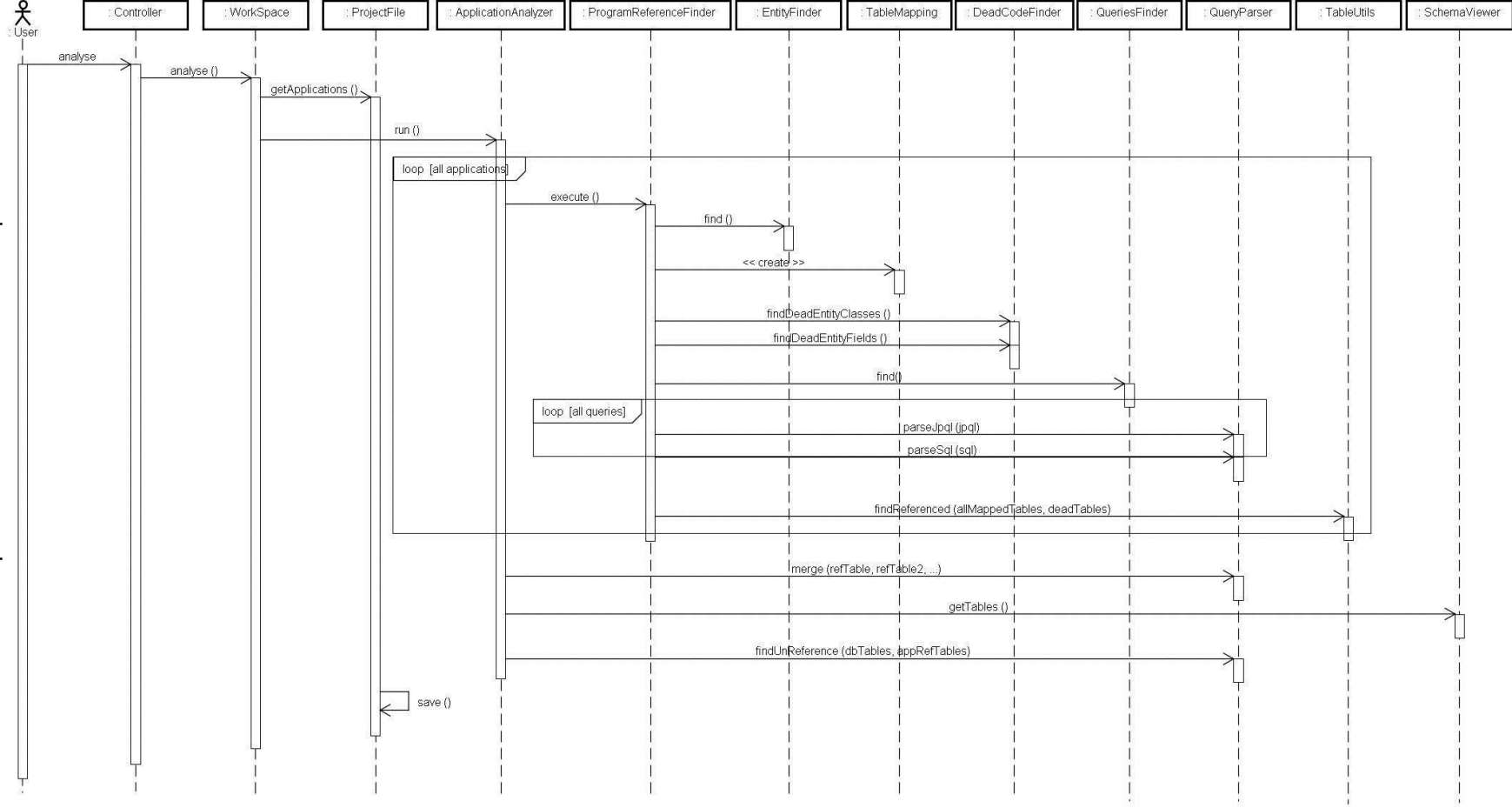


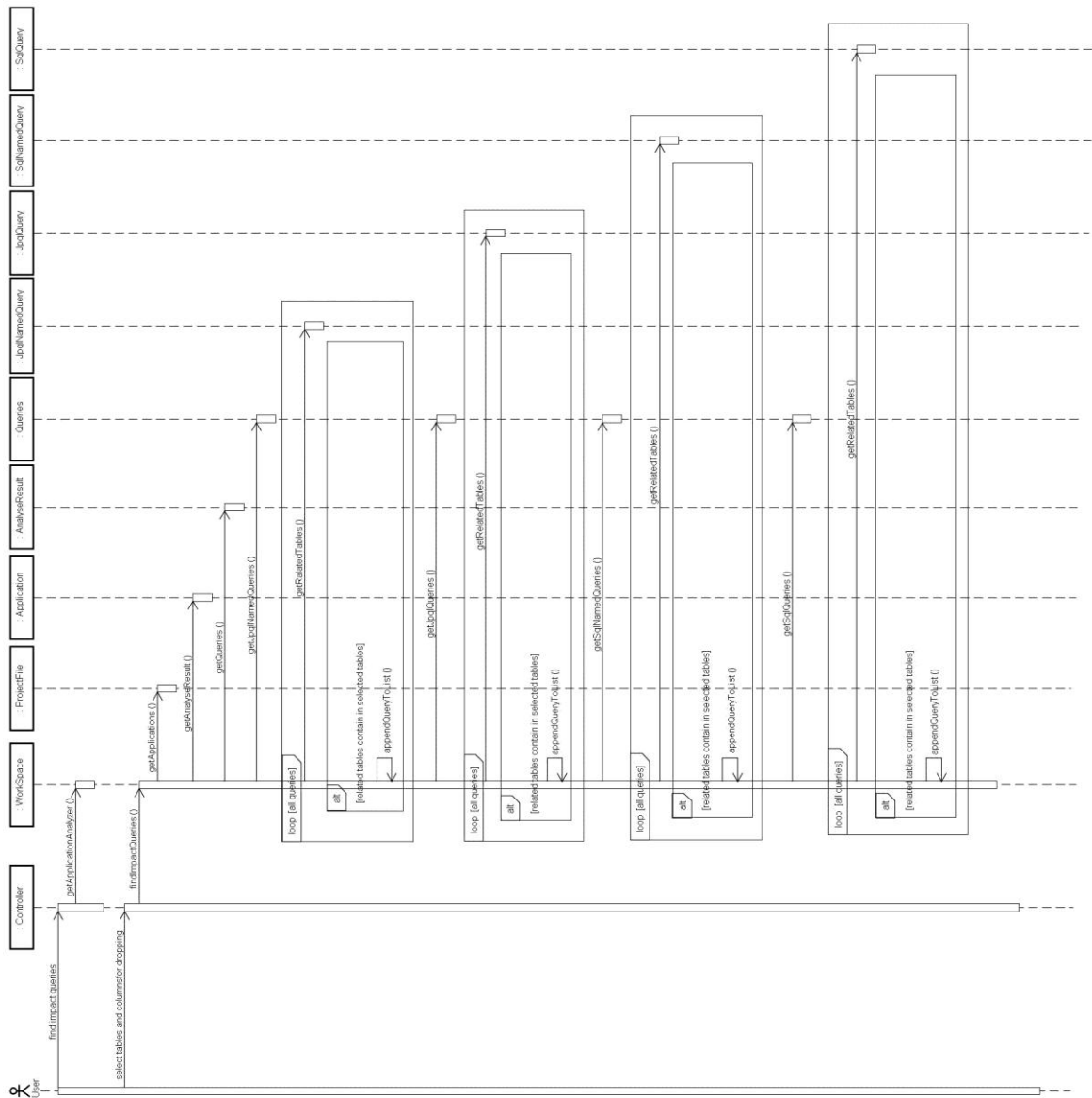
ภาพที่ 3.30 แผนภาพลำดับพฤติกรรมเครื่องมือขณะนำออกโปรแกรมที่วิเคราะห์แล้ว

ในภาพที่ 3.30 เมื่อมีการนำออกโปรแกรมที่วิเคราะห์แล้ว เครื่องมือจะโหลดวัตถุ โปรแกรมที่ถูกเลือกอยู่ เพื่อเตรียมจะนำออก จากนั้นจะตรวจสอบวัตถุโปรแกรมก่อนว่ามีผลการวิเคราะห์แล้วหรือไม่ ซึ่งถ้ามีผลการวิเคราะห์แล้วก็จะเขียนไฟล์นำออกไปยังในพาชที่ผู้ใช้งานระบุ

แผนภาพลำดับ ในภาพที่ 3.31 จะแสดงพฤติกรรมการทำงานหลักของเครื่องมือ โดยเครื่องมือจะเรียกวัตถุวิเคราะห์โปรแกรม ซึ่งจะวิเคราะห์แยกทีละโปรแกรม ว่าแต่ละโปรแกรม มีการอ้างถึงตาราง และคอลัมน์ใดบ้าง จากนั้นจะรวมผลลัพธ์ที่ได้เข้าด้วยกันเป็นวัตถุรายการตารางที่โปรแกรมทั้งหมดอ้างถึง จากนั้นเครื่องมือจะดูเค้าร่างจริงของฐานข้อมูล ซึ่งจะได้อายการวัตถุของรายการตารางเช่นเดียวกัน และสุดท้ายคือเครื่องมือจะเปรียบเทียบวัตถุรายการตารางที่ได้จากฐานข้อมูลกับวัตถุรายการตารางที่โปรแกรมอ้างถึง เพื่อหารายการตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างถึง

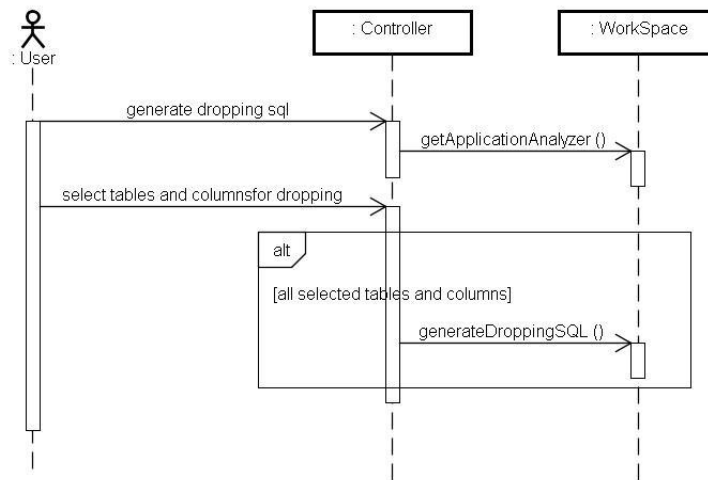
ภาพที่ 3.31 แผนภาพลำดับเหตุการณ์เครื่องมือของ
วิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง





ภาพที่ 3.32 แผนภาพลำดับพฤติกรรมเครื่องมือขณะหาคำสั่งสืบทอดที่ได้รับผลกระทบ

ในภาพที่ 3.32 พฤติกรรมของโปรแกรมที่จะหาคำสั่งสืบทอดที่ได้รับผลกระทบ จะเกิดขึ้นได้เมื่อเกิดพฤติกรรมการวิเคราะห์หาราย และคอลัมน์ที่ไม่ถูกต้องดังแผนภาพลำดับ ภาพที่ 3.31 เสียก่อน โดยผลการวิเคราะห์โปรแกรมแต่ละโปรแกรมจะมีวัตถุของคำสั่งสืบทอดอยู่ด้วย ซึ่งวัตถุดังกล่าวจะเก็บคำสั่งสืบทอดภาษาต่าง ๆ ได้แก่ ภาษาเจพีคิวแอล และภาษาเอสคิวแอล ทั้งแบบใช้ในเมธอดโดยตรง และแบบใช้ผ่านชื่อคำสั่งสืบทอด โดยในวัตถุคำสั่งสืบทอดแต่ละภาษา จะบอกว่าแต่ละคำสั่งสัมพันธ์กับตาราง และคอลัมน์ใดในฐานข้อมูล ซึ่งเครื่องมือจะนำรายการตาราง และคอลัมน์ที่ผู้ใช้งานเลือกมาเปรียบเทียบกับวัตถุคำสั่งสืบทอด เพื่อหาคำสั่งที่ได้รับผลกระทบ



ภาพที่ 3.33 แผนภาพลำดับพฤติกรรมเครื่องมือขณะสร้างคำสั่งลบตาราง และคอลัมน์

ในภาพที่ 3.33 พฤติกรรมของโปรแกรมที่สร้างคำสั่งลบตาราง และคอลัมน์ จะเกิดขึ้นได้เมื่อเกิดพฤติกรรมการวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง ดังแผนภาพลำดับ ภาพที่ 3.31 เสียก่อน เนื่องจากผู้ใช้งานจะต้องระบุรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรมว่าจะลบตาราง และคอลัมน์ใด เมื่อผู้ใช้งานเลือกรายการตาราง และคอลัมน์แล้ว เครื่องมือจะอ่านวัตถุที่ผู้ใช้งานเลือกมาสร้างคำสั่งสำหรับลบตาราง และคอลัมน์ โดยสร้างคำสั่งดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 คำสั่งภาษาเอสคิวแอลสำหรับลบตาราง และคอลัมน์

คำสั่ง	ความหมาย
DROP TABLE ชื่อตาราง	ลบตารางชื่อที่มีชื่อตามที่ระบุออกจากฐานข้อมูล
ALTER TABLE ชื่อตาราง DROP COLUMN ชื่อคอลัมน์	ลบคอลัมน์ตามชื่อที่ระบุออกจากตารางชื่อที่ระบุ

บทที่ 4

การพัฒนาเครื่องมือ

ในบทนี้ จะกล่าวถึงสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือต้นแบบ และหน้าจอสวนต่อประสานกับผู้ใช้ของเครื่องมือสำหรับระบุตาราง และคอลัมน์ที่ไม่ได้ถูกอ้างอิงใน โปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือต้นแบบสำหรับระบุตาราง และคอลัมน์ที่ไม่ได้ถูกอ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ มีรายละเอียดดังนี้

1) ฮาร์ดแวร์ (Hardware)

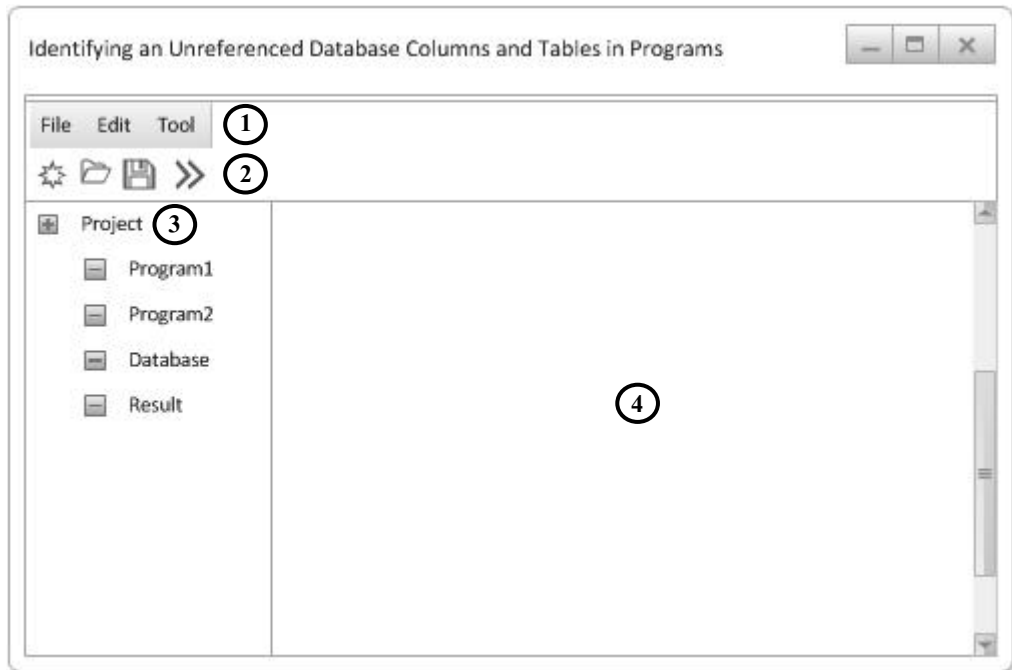
- เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ (Personal Computer)
- หน่วยประมวลผล อินเทลคอร์ไอเจ็ด 2.76 กิกะเฮิร์ตซ์ (Intel Core i7 2.76 GHz)
- หน่วยความจำหลัก (RAM) 5 กิกะไบต์ (5 GB)
- ฮาร์ดดิส (Harddisk) 2 เทระไบต์ (2 TB)

2) ซอฟต์แวร์ (Software)

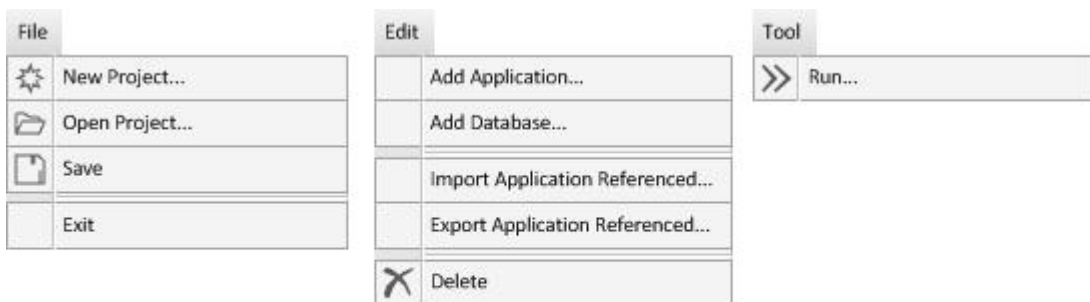
- ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์เจ็ด (Microsoft Windows 7)
- จาวา เวอร์ชัน 7 (Java 7)
- โปรแกรมเน็ทเบิน เวอร์ชัน 7.1.2 (Netbean 7.1.2)

4.2 ส่วนต่อประสานกับผู้ใช้ของเครื่องมือ

ส่วนต่อประสานกับผู้ใช้ของเครื่องมือสำหรับระบุตาราง และคอลัมน์ที่ไม่ได้ถูกอ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ ประกอบไปด้วย 4 ส่วนหลัก ๆ ดังแสดงในภาพที่ 4.1 ได้แก่



ภาพที่ 4.1 ส่วนต่อประสานกับผู้ใช้ของเครื่องมือ



(ก)

(ข)

(ค)

ภาพที่ 4.2 เมนูต่าง ๆ ในเครื่องมือ

- 1) ส่วนที่ 1 คือ เมนู ซึ่งประกอบไปด้วย 3 เมนู ได้แก่
 - เมนูเพิ่ม (File) จะประกอบไปด้วยฟังก์ชันต่างๆ เกี่ยวกับโครงการ ได้แก่ ฟังก์ชันสร้างโครงการใหม่ ฟังก์ชันเปิดโครงการ ฟังก์ชันบันทึกโครงการ และออกจากเครื่องมือ ดังแสดงในภาพที่ 4.2 (ก)
 - เมนูแก้ไข (Edit) เป็นเมนูสำหรับเพิ่ม และลบ โปรแกรม และฐานข้อมูล สำหรับเป็นข้อมูลให้เครื่องมือใช้วิเคราะห์ ประกอบไปด้วย ฟังก์ชันเพิ่มโปรแกรมเพื่อวิเคราะห์ ฟังก์ชันเพิ่มฐานข้อมูลเพื่อวิเคราะห์ ฟังก์ชันลบโปรแกรม ฟังก์ชันลบฐานข้อมูล ฟังก์ชันนำเข้าโปรแกรมที่วิเคราะห์แล้ว และฟังก์ชันนำออกโปรแกรมที่วิเคราะห์แล้ว ดังแสดงในภาพที่ 4.2 (ข)
 - เมนูเครื่องมือ (Tool) เป็นเมนูที่เกี่ยวข้องกับการทำงานของเครื่องมือ ซึ่งได้แก่ ฟังก์ชันวิเคราะห์ตารางและคอลัมน์ที่ไม่ถูกอ้างอิง ดังแสดงในภาพที่ 4.2 (ค)
- 2) ส่วนที่ 2 คือ แถบเครื่องมือ ซึ่งแสดงฟังก์ชันที่ใช้งานบ่อย ๆ ได้แก่ ฟังก์ชันสร้างโครงการใหม่ ฟังก์ชันเปิดโครงการ ฟังก์ชันบันทึกโครงการ และฟังก์ชันวิเคราะห์ตารางและคอลัมน์ที่ไม่ถูกอ้างอิง ตามลำดับ
- 3) ส่วนที่ 3 คือ ส่วนเมื่อนำทาง โดยในแต่ละโครงการ จะประกอบไปด้วย 3 ส่วน ได้แก่
 - โปรแกรมต่างๆ ซึ่งอาจมีมากกว่า 1 โปรแกรม เพื่อให้เครื่องมือวิเคราะห์
 - ฐานข้อมูล สำหรับให้เครื่องมือดูเค้าร่างฐานข้อมูล
 - ผลการวิเคราะห์ ตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม
 โดยเมื่อผู้ใช้งานเลือกที่เมนูใด รายละเอียดจะปรากฏในส่วนที่ 4
- 4) ส่วนที่ 4 คือ ส่วนแสดงรายละเอียดต่าง ๆ ได้แก่
 - แสดงรายละเอียดของโปรแกรม ว่าในโปรแกรมมีการอ้างอิงตาราง และคอลัมน์ฐานข้อมูลใด รวมถึงมีคำสั่งสืบค้นใด และแต่ละคำสั่งอ้างอิงตาราง และคอลัมน์ฐานข้อมูลใด
 - แสดงรายละเอียดของผลการวิเคราะห์ ซึ่งจะแสดงรายการตาราง และคอลัมน์ที่โปรแกรมต่าง ๆ ไม่ได้อ้างอิง

4.3 โครงสร้างไฟล์โครงการสำหรับเก็บข้อมูล

ผู้วิจัยได้ออกแบบโครงสร้างไฟล์โครงการสำหรับจัดเก็บข้อมูลต่างๆของเครื่องมืออยู่ในรูปของไฟล์เอ็กซ์เอ็มแอล โดยข้อมูลทุกๆข้อมูลจะจัดเก็บลงในไฟล์เดียว โดยใช้ป้ายระบุ (Tag) เป็นตัวกำหนดข้อมูลแต่ละประเภท ซึ่งมีรายละเอียดของโครงสร้างไฟล์ดังตารางที่ 4.1 ดังนี้

ตารางที่ 4.1 โครงสร้างไฟล์โครงการ

โครงสร้าง และป้ายระบุ สำหรับเก็บข้อมูล	ความหมาย	มีข้อมูลได้มากกว่า 1 ชุด
1. <ProjectPath>	พาธของไฟล์โครงการ	
2. <Applications>	รายละเอียดโปรแกรมแต่ละโปรแกรม	✓
2.1 <ApplicationName>	ชื่อโปรแกรม	
2.2 <RootClassPath>	คลาสพาธของโปรแกรม	
2.3 <Library>	พาธคำสั่งอื่นๆ นอกเหนือจากมาตรฐานจาวา	✓
2.4 <IgnorePackage>	ชื่อแพ็คเกจในโปรแกรมที่ไม่ต้องการให้วิเคราะห์	✓
2.5 <Result>	ผลลัพธ์การวิเคราะห์โปรแกรม	
2.5.1 <Ref_Tables Name="ชื่อตาราง">	ตารางที่โปรแกรมอ้างอิง	✓
2.5.1.1 <Columns PK="เป็น PK หรือไม่" Name="ชื่อคอลัมน์">	คอลัมน์ในตารางที่โปรแกรมอ้างอิง	✓
2.5.2 <DeadClass>	ชื่อคลาสที่เข้าไม่ถึงในโปรแกรม	✓
2.5.3 <DeadField Class="ชื่อคลาส">	เก็บรายการเขตข้อมูลที่เข้าไม่ถึงที่อยู่ในคลาสที่ระบุ	✓
2.5.3.1 <Field>	ชื่อเขตข้อมูลที่เข้าไม่ถึง	✓
2.5.4 <Queries>	รายการคำสั่งสืบค้นในโปรแกรม	

ตารางที่ 4.1 โครงสร้างไฟล์โครงการ (ต่อ)

โครงสร้าง และป้ายระบุ สำหรับเก็บข้อมูล	ความหมาย	มีข้อมูลได้ มากกว่า 1 ชุด
2.5.4.1 <NameQuery lang="JPQL หรือ SQL">	คำสั่งสืบค้นที่เป็นแบบระบุชื่อ โดยระบุว่าเป็นภาษาเจพีคิว แอล หรือ เอสคิวแอล	✓
2.5.4.1.1 <Name>	ชื่อคำสั่งสืบค้น	
2.5.4.1.2 <Query>	คำสั่งสืบค้น	
2.5.4.1.3 <Class>	ชื่อคลาสที่ประกาศคำสั่งสืบค้น นี้	
2.5.4.1.4 <RelatedTables Name= "ชื่อตาราง">	ตารางที่เกี่ยวข้องกับคำสั่ง สืบค้นนี้	✓
2.5.4.1.4.1 <Columns Name="ชื่อคอลัมน์">	ชื่อคอลัมน์ในตาราง ที่เกี่ยวข้อง กับคำสั่งสืบค้นนี้	✓
2.5.4.2 <Query lang="JPQL หรือ SQL">	คำสั่งสืบค้น โดยระบุว่าเป็น ภาษาเจพีคิวแอล หรือ เอสคิว แอล	✓
2.5.4.2.1 <Query>	คำสั่งสืบค้น	
2.5.4.2.2 <Class>	ชื่อคลาสที่มีการใช้คำสั่งสืบค้น	
2.5.4.2.3 <Method>	ชื่อเมธอดที่มีการใช้คำสั่ง สืบค้น	
2.5.4.2.4 <RelatedTables Name= "ชื่อตาราง">	ตารางที่เกี่ยวข้องกับคำสั่ง สืบค้นนี้	✓
2.5.4.2.4.1 <Columns Name="ชื่อคอลัมน์">	ชื่อคอลัมน์ในตาราง ที่เกี่ยวข้อง กับคำสั่งสืบค้นนี้	✓

ตารางที่ 4.1 โครงสร้างไฟล์โครงการ (ต่อ)

โครงสร้าง และปัยระนะ สำหรับเก็บข้อมูล	ความหมาย	มีข้อมูลได้มากกว่า 1 ชุด
3. <Database>	รายละเอียดฐานข้อมูลที่เชื่อมต่อ	
3.1 <ConnectionString>	สายข้อความสำหรับเชื่อมต่อกับฐานข้อมูล	
3.2 <Username>	ชื่อผู้ใช้ที่สามารถเชื่อมต่อฐานข้อมูลได้	
3.3 <Password>	รหัสผ่านของชื่อผู้ใช้	
3.4 <DriverClass>	ชื่อคลาสไดรฟ์เวอร์ฐานข้อมูล	
3.5 <Schema>	ชื่อเค้าร่างฐานข้อมูล (บางฐานข้อมูลมีการใช้งานเค้าร่าง)	
3.6 <Catalog>	ชื่อคาสตาล็อกฐานข้อมูล (บางฐานข้อมูลมีการใช้งานคาสตาล็อก)	
3.7 < DriverPath >	พาธไฟล์ไดรฟ์เวอร์ฐานข้อมูล	✓
4. <UnreferenceTables Name="ชื่อตาราง">	ตารางที่ไม่ถูกอ้างอิงในโปรแกรม	✓
4.1 <Columns PK="เป็น PK หรือไม่" Name="ชื่อคอลัมน์">	ชื่อคอลัมน์ในตารางที่ไม่ถูกอ้างอิงในโปรแกรม	✓
5. <ImpactedQueries>	คำสั่งสืบค้นที่ได้รับผลกระทบซึ่งมีรายละเอียดเหมือนกับข้อ 2.5.4	
6. <DroppingSQL>	คำสั่งลบตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง	✓

บทที่ 5

การทดสอบเครื่องมือ

การทดสอบแนวความคิดวิธีที่ใช้ระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้เข้าถึง โดยใช้หลักการของการแปลงวัตถุ สามารถทดสอบความถูกต้องของแนวความคิดที่ได้โดยทดสอบเครื่องมือต้นแบบที่พัฒนาขึ้น โดยใช้แนวความคิดดังกล่าว โดยมีรายละเอียดการทดสอบดังนี้

5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือต้นแบบสำหรับระบุตาราง และคอลัมน์ที่ไม่ได้ถูกอ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ มีรายละเอียดดังนี้

1) ฮาร์ดแวร์ (Hardware)

- เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ (Personal Computer)
- หน่วยประมวลผล อินเทลคอร์ทูดูโอ 3 กิกะเฮิร์ตซ์ (Intel Core2Duo 3 GHz)
- หน่วยความจำหลัก (RAM) 2 กิกะไบต์ (2 GB)
- ฮาร์ดดิส (Harddisk) 300 กิกะไบต์ (300 GB)

2) ซอฟต์แวร์ (Software)

- ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์เอ็กซ์พี (Microsoft Windows XP)
- จาวา เวอร์ชัน 6 (Java 6)
- ดิบีทูไคลเอนต์ (DB2 Client)

5.2 การทดสอบเครื่องมือ

การทดสอบเครื่องมือต้นแบบนี้ เพื่อตรวจสอบความถูกต้องของการทำงานของเครื่องมือว่าสามารถให้ผลลัพธ์รายการตาราง และคอลัมน์ที่โปรแกรมไม่ได้เข้าถึง ได้ถูกต้องหรือไม่ โดยการทดสอบเครื่องมือต้นแบบนี้ ได้ทำการแบ่งการทดสอบออกเป็น 2 การทดสอบ คือการทดสอบความถูกต้องของเครื่องมือโดยทดสอบกับกรณีทดสอบ และการทดสอบในการใช้งานจริงกับโปรแกรมของธนาคารในประเทศไทยแห่งหนึ่ง โดยมีรายละเอียดดังนี้

5.2.1 ทดสอบเครื่องมือกับกรณีทดสอบ

สำหรับการทดสอบความถูกต้องของเครื่องมือต้นแบบสำหรับ ระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้เข้าถึงนี้ ผู้วิจัยได้ทำการสร้างกรณีทดสอบตามส่วนการทำงานหลักของเครื่องมือ ซึ่งแบ่งออกเป็น 6 ส่วน โดยในแต่ละส่วนจะมีจำนวนกรณีทดสอบที่แตกต่างกันออกไปขึ้นกับความซับซ้อนของแต่ละการทำงาน ซึ่งสามารถดูรายละเอียดกรณีทดสอบเพิ่มเติมได้จากภาคผนวก จ โดยการทดสอบการทำงานทั้ง 6 ส่วนมีรายละเอียดดังนี้

ส่วนที่ 1 การทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์ โดยการใช้บรรณนิทัศน์ในรูปแบบต่างๆ

ส่วนที่ 2 การทดสอบการหาโค้ดที่เข้าไม่ถึงในโปรแกรม โดยทดสอบหาคลาสเอนติตี และแอตทริบิวต์ของเอนติตีที่ไม่ถูกอ้างในที่ต่างๆของโปรแกรมทดสอบ

ส่วนที่ 3 การทดสอบการหาคำสั่งสืบทอดในโปรแกรม โดยทดสอบหาคำสั่งสืบทอดภาษาเจฟคิวแอล และเอสคิวแอล ในโปรแกรมทดสอบ

ส่วนที่ 4 การทดสอบหาตาราง และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบทอดในภาษาเจฟคิวแอล และเอสคิวแอล

ส่วนที่ 5 การทดสอบการรวมผลลัพธ์รายการตาราง 2 รายการเข้าด้วยกัน

ส่วนที่ 6 การทดสอบหารายการตารางที่ไม่ถูกอ้างอิง

5.2.2 ทดสอบเครื่องมือกับองค์กรจริง

การทดสอบในส่วนนี้จะเป็นการนำเครื่องมือต้นแบบสำหรับระบบตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างถึงไปทดสอบหาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในองค์กรจริง ซึ่งนำไปทดสอบกับโปรแกรมของธนาคารในประเทศไทยแห่งหนึ่ง โดยเลือกโปรแกรม 2 โปรแกรมที่อ้างอิงไปที่ฐานข้อมูลเดียวกัน และเป็นโปรแกรมที่พัฒนาโดยใช้การเชื่อมต่อฐานข้อมูลโดยใช้เจพีเอ โดยมีรายละเอียดของโปรแกรมต่างๆ และฐานข้อมูลดังนี้

โปรแกรมที่ 1 เป็นโปรแกรมกระแสนงาน (Workflow) ซึ่งตัวโปรแกรมมีเฉพาะฟังก์ชันต่างๆที่เกี่ยวข้องกับตัวกระแสนงานเพียงอย่างเดียว โดยโปรแกรมนี้พัฒนาด้วยภาษาจาวา ใช้โปรแกรมประยุกต์ไฮเบอร์เนท ในการเชื่อมต่อฐานข้อมูล โดยตั้งค่าให้ไฮเบอร์เนทอ่านการระบุความสัมพันธ์ระหว่างคลาสเอนติตี้กับตารางฐานข้อมูลผ่านบรรณนิทัศน์ของเจพีเอ โดยโปรแกรมนี้อ้างอิงตารางฐานข้อมูลอยู่ 23 ตาราง และคอลัมน์ทั้งหมด 108 คอลัมน์

โปรแกรมที่ 2 เป็นส่วนของโปรแกรมการอนุมัติสินเชื่อบ้าน (Housing Loan Approval) โดยโปรแกรมส่วนนี้จะมีเฉพาะส่วนการติดต่อกับโปรแกรมอื่นๆในธนาคาร และส่วนการเก็บข้อมูลความลับของลูกค้า ซึ่งเป็นข้อมูลที่จำเป็นต้องเก็บไว้ภายในธนาคาร ส่วนอื่นๆของโปรแกรม จะอยู่บนคลาวด์ (Cloud) โดยติดต่อกันผ่านเว็บเซอร์วิส (Web Service) โดยโปรแกรมในส่วนที่เก็บข้อมูลความลับลูกค้าพัฒนาด้วยภาษาจาวา บนกรอบงานสปริงซ์ (Spring Framework) โดยใช้โปรแกรมประยุกต์อีคลิปส์ลิงค์ ในการเชื่อมต่อฐานข้อมูล โดยตั้งค่าให้อีคลิปส์ลิงค์อ่านการระบุความสัมพันธ์ระหว่างคลาสเอนติตี้กับตารางฐานข้อมูลผ่านบรรณนิทัศน์ของเจพีเอ โดยโปรแกรมนี้อ้างอิงตารางฐานข้อมูลอยู่ 12 ตาราง และคอลัมน์ทั้งหมด 217 คอลัมน์

ฐานข้อมูล เนื่องจากฐานข้อมูลจริงเป็นความลับของลูกค้า ฐานข้อมูลที่น่ามาทดสอบจึงต้องใช้เป็นฐานข้อมูลสำหรับทดสอบโปรแกรม โดยจะมีเค้าร่างเหมือนฐานข้อมูลจริง แต่ข้อมูลเป็นข้อมูลทดสอบ ซึ่งฐานข้อมูลที่น่ามาทดสอบเป็นฐานข้อมูลของไอบีเอ็ม ดีบีทู (IBM DB2) โดยมีตารางทั้งหมด 378 ตาราง รวม 4344 คอลัมน์

5.3 ผลการทดสอบ

หลังจากทดสอบเครื่องมือต้นแบบกับกรณีทดสอบต่างๆ และนำไปทดสอบกับองค์กรจริง ได้ผลการทดสอบดังต่อไปนี้

5.3.1 ผลการทดสอบเครื่องมือกับกรณีทดสอบ

เมื่อนำเครื่องมือต้นแบบไปทดสอบการทำงานกับกรณีทดสอบต่างๆ ได้ผลการทดสอบดังตารางที่ 5.1 โดยสามารถดูกรณีทดสอบย่อยโดยละเอียดได้จากภาคผนวก จ

ตารางที่ 5.1 ผลการทดสอบเครื่องมือกับกรณีทดสอบ

กรณีทดสอบ	ผลการทดสอบ
การแปลงระหว่างวัตถุและตารางความสัมพันธ์ <ul style="list-style-type: none"> - การแปลงแบบง่าย - การแปลงแบบใช้วัตถุแบบฝังตัว - การแปลงแบบมีความสัมพันธ์ระหว่างคลาส - การแปลงที่สัมพันธ์มากกว่า 1 ตาราง - การแปลงความสัมพันธ์แบบสืบทอดคุณสมบัติ 	ผ่าน
การทดสอบหาโค้ดที่เข้าไม่ถึง <ul style="list-style-type: none"> - หาคลาสเอนติตี้ที่เข้าไม่ถึง - หาแอตทริบิวต์ของเอนติตี้ที่เข้าไม่ถึง 	ผ่าน
การทดสอบหาคำสั่งสืบทอดในโปรแกรม <ul style="list-style-type: none"> - หาคำสั่งสืบทอดแบบระบุชื่อคำสั่ง - หาคำสั่งสืบทอดภาษาเจฟีกิวแอล - หาคำสั่งสืบทอดภาษาเอสคิวแอล 	ผ่าน
การทดสอบหาราย และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบทอด <ul style="list-style-type: none"> - หาราย และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบทอดภาษาเจฟีกิวแอล - หาราย และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบทอดภาษาเอสคิวแอล 	ผ่าน

ตารางที่ 5.1 ผลการทดสอบเครื่องมือกับกรณีทดสอบ (ต่อ)

กรณีทดสอบ	ผลการทดสอบ
การทดสอบการรวมผลลัพธ์รายการตาราง 2 รายการ	ผ่าน
การทดสอบหารายการตารางที่ไม่ถูกอ้างอิง	ผ่าน

จากผลการทดสอบพบว่าการทำงานหลักๆในแต่ละส่วนของเครื่องมือ สามารถทำงานได้อย่างถูกต้องครบตามกรณีทดสอบ คือในแต่ละส่วนของเครื่องมือสามารถทำงานได้อย่างถูกต้อง แต่ยังคงทดสอบเครื่องมือกับโปรแกรมจริงตั้งแต่เริ่มจนจบการทำงาน เพื่อทดสอบแนวความคิดว่าแนวความคิดที่นำเสนอ สามารถนำไปใช้งานได้จริง

5.3.2 ผลการทดสอบเครื่องมือกับองค์กรจริง

เมื่อนำเครื่องมือต้นแบบไปทดสอบกับโปรแกรมจริงของธนาคารแห่งหนึ่ง โดยมีข้อมูลนำเข้าดังแสดงในตารางที่ 5.2 โดยโปรแกรมที่นำเข้าทั้ง 2 โปรแกรมไม่มีการอ้างอิงตาราง หรือคอลัมน์เดียวกัน ซึ่งหลังจากเครื่องมือวิเคราะห์เสร็จสิ้นแล้วได้ผลลัพธ์ที่คาดหวัง และผลลัพธ์จริงดังแสดงในตารางที่ 5.3 ดังนี้

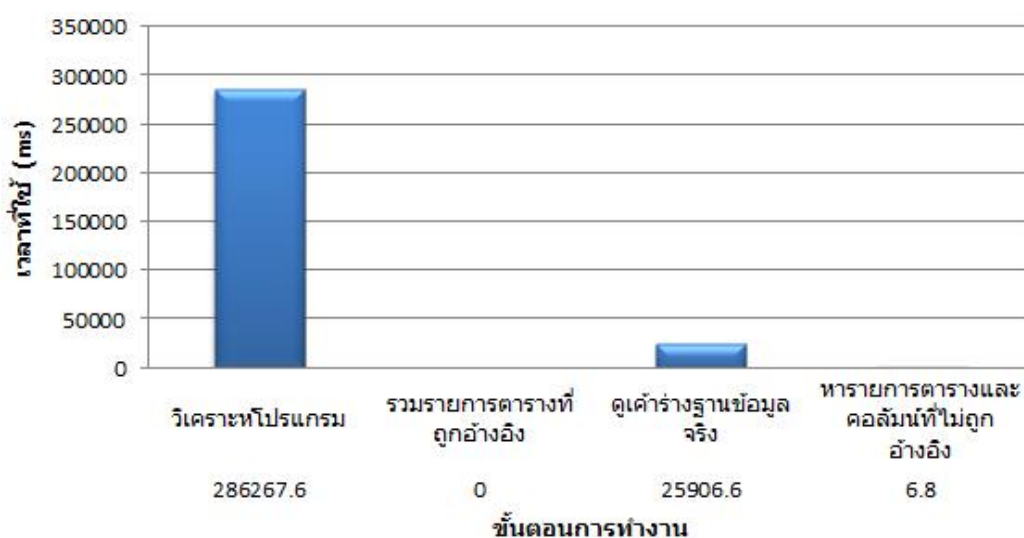
ตารางที่ 5.2 ข้อมูลนำเข้าของเครื่องมือ

ข้อมูลนำเข้า	จำนวนตารางที่อ้างอิง	จำนวนคอลัมน์รวมที่อ้างอิง
โปรแกรมที่ 1	23	108
โปรแกรมที่ 2	12	217
ฐานข้อมูล	378	4344

ตารางที่ 5.3 ผลการทดสอบเครื่องมือกับองค์กรจริง

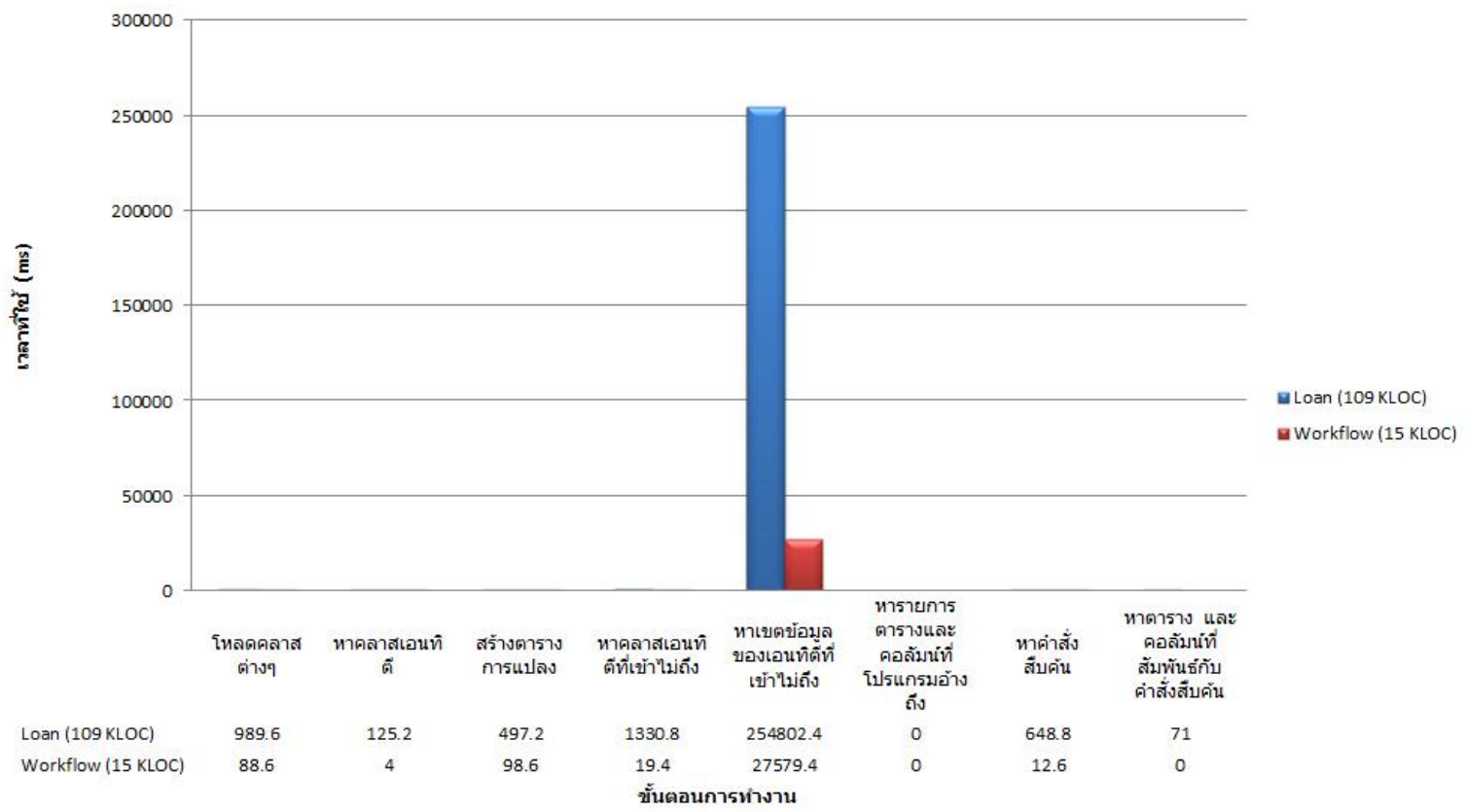
กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
จำนวนตารางที่ 2 โปรแกรมอ้างอิง	35	35
จำนวนคอลัมน์รวมที่ 2 โปรแกรมอ้างอิง	325	325
จำนวนตารางที่ไม่ถูกอ้างอิงทั้งตาราง	343	343
จำนวนตารางที่มีคอลัมน์ที่ไม่ถูกอ้างอิง	8	8
จำนวนคอลัมน์ที่ไม่ถูกอ้างอิง (ไม่รวมคอลัมน์ในตารางที่ไม่ถูกอ้างอิงทั้งตาราง)	24	24

นอกจากนี้ผู้วิจัยเก็บสถิติเวลาที่เครื่องมือใช้ในการทำงานเพื่อนำมาวิเคราะห์ถึงขั้นตอนการทำงานที่ควรจะพัฒนาต่อยอดเพื่อให้ประสิทธิภาพการทำงานของเครื่องมือดีขึ้น โดยผู้วิจัยได้ทดสอบเครื่องมือเป็นจำนวน 5 ครั้ง แล้วเฉลี่ยหาเวลาที่เครื่องมือใช้งาน โดยเวลาที่เครื่องมือใช้ในแต่ละขั้นตอนแสดงในภาพที่ 5.1 และเปรียบเทียบเวลาที่ใช้ในการวิเคราะห์โปรแกรม ระหว่างโปรแกรมขนาดเล็ก และโปรแกรมขนาดใหญ่ในภาพที่ 5.2



ภาพที่ 5.1 เวลาที่เครื่องมือใช้งานในแต่ละขั้นตอน

ภาพที่ 5.2 เปรียบเทียบเวลาที่ใช้ในการวิเคราะห์โปรแกรม ระหว่างโปรแกรมขนาดเล็ก และโปรแกรมขนาดใหญ่



จากผลการทดสอบเครื่องมือต้นแบบ พบว่าผลลัพธ์การระบุตาราง และคอลัมน์ที่โปรแกรมไม่ได้อ้างอิง สามารถหารายการตาราง และคอลัมน์ได้อย่างถูกต้อง นั่นสะท้อนให้เห็นว่าแนวความคิดที่ได้นำเสนอไป สามารถนำไปใช้ได้จริงในทางปฏิบัติ เพื่อสามารถลดปริมาณข้อมูลที่เกิดจากข้อมูลในตาราง และคอลัมน์ที่ไม่มีการอ้างอิงได้

นอกจากนี้ผลการทดลองเรื่องเวลาที่เครื่องมือใช้ทำงานในภาพที่ 5.1 จะเห็นว่าเวลาของขั้นตอนการวิเคราะห์โปรแกรม เป็นขั้นตอนที่ใช้เวลานานที่สุด ซึ่งเมื่อดูขั้นตอนย่อยๆ ในภาพที่ 5.2 ก็จะพบว่าที่ใช้เวลานาน เป็นเพราะขั้นตอนหาแอตทริบิวต์ของเอนติตี้ที่เข้าไม่ถึง ซึ่งในเครื่องมือต้นแบบผู้วิจัยได้นำเครื่องมือที่เป็นโอเพ่นซอร์ส ชื่อว่า “com.mebigfatguy.deadmethods” [19] เข้ามาใช้ ทั้งนี้หากต้องการลดเวลาการทำงานของเครื่องมือ อาจหาวิธีการอื่นๆ ในการหาแอตทริบิวต์ของเอนติตี้ที่เข้าไม่ถึงแทน นอกจากนี้ในภาพที่ 5.2 ยังแสดงให้เห็นว่าโปรแกรมที่มีขนาดเล็กย่อมใช้เวลาวิเคราะห์น้อยกว่าโปรแกรมที่มีขนาดใหญ่

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

จากการศึกษา นำเสนอแนวคิด และสร้างเครื่องมือต้นแบบสำหรับการระบุคอล์มน์และตารางของฐานข้อมูลที่ไม่ถูกอ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ สามารถสรุปผลการวิจัย ข้อจำกัดของวิธีที่นำเสนอ ข้อจำกัดของเครื่องมือต้นแบบ และแนวทางการพัฒนาต่อ ซึ่งมีรายละเอียดดังนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอวิธีการระบุคอล์มน์และตารางของฐานข้อมูลที่ไม่ถูกอ้างอิงในโปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ โดยพัฒนาเครื่องมือต้นแบบสำหรับวิเคราะห์โปรแกรมภาษาจาวา ที่ใช้ส่วนต่อประสานโปรแกรมประยุกต์เจพีเอ ที่ทำงานตามแนวคิดการแปลงวัตถุเป็นตารางความสัมพันธ์ และนำเครื่องมือไปทดสอบกับโปรแกรม และฐานข้อมูลจริงที่ใช้งานในองค์กร ซึ่งผลที่ได้คือเครื่องมือสามารถระบุตาราง และคอล์มน์ที่โปรแกรมไม่ได้อ้างอิงถึงได้อย่างถูกต้อง นั่นคือวิธีการที่นำเสนอสามารถใช้ระบุคอล์มน์และตารางของฐานข้อมูลที่ไม่ถูกอ้างอิงในโปรแกรมได้จริง

นอกจากนี้แนวคิดนี้ยังสามารถลดระยะเวลาในการบำรุงรักษาฐานข้อมูล เนื่องจากสามารถระบุตาราง และคอล์มน์ที่โปรแกรมไม่ได้อ้างอิงถึงได้ในเวลาอันสั้น เพื่อที่สามารถลบตาราง และคอล์มน์ดังกล่าวออก อันเป็นการช่วยจัดข้อมูลให้เป็นระเบียบ และลดปริมาณข้อมูลที่ไม่จำเป็นออกจากฐานข้อมูล อีกทั้งยังช่วยสนับสนุนการพัฒนาซอฟต์แวร์แบบอไบเจต์ เนื่องจากการพัฒนาซอฟต์แวร์แบบอไบเจต์จะมีการปรับปรุงเปลี่ยนแปลงโปรแกรม รวมไปถึงเค้าร่างของฐานข้อมูลอยู่ตลอดเวลา ซึ่งวิธีการที่นำเสนอนี้สามารถช่วยระบุตาราง และคอล์มน์ที่ไม่มีความจำเป็นได้อีกด้วย

6.2 ข้อจำกัดของวิธีที่ทำเสนอ

1. วิธีการที่นำเสนอสามารถใช้วิเคราะห์โปรแกรมที่ใช้การแปลงวัตถุเป็นตารางความสัมพันธ์ติดต่อกับฐานข้อมูลเท่านั้น
2. ฐานข้อมูลที่โปรแกรมเชื่อมต่ออยู่ต้องเป็นฐานข้อมูลเชิงสัมพันธ์เท่านั้น เนื่องจากเทคนิคการแปลงวัตถุเป็นตารางความสัมพันธ์ จะแปลงวัตถุในคลาสของโปรแกรม เป็นตาราง และคอลัมน์ของฐานข้อมูลเชิงสัมพันธ์เท่านั้น
3. ไม่สามารถวิเคราะห์โปรแกรมในส่วนที่ใช้งานสตอร์โพรซีเยอร์ได้

6.3 ข้อจำกัดของเครื่องมือต้นแบบ

1. เครื่องมือที่พัฒนาสามารถวิเคราะห์โปรแกรมภาษาจาวา ที่ใช้ เจพีเอติดต่อกับฐานข้อมูล เท่านั้น
2. เครื่องมือรองรับที่จะอ่านค่าการแปลงระหว่างวัตถุ กับฐานข้อมูล โดยใช้บรรณนิทัศน์ของจาวาเท่านั้น
3. เครื่องมือรองรับการหาคำสั่งสืบลำดับที่เป็นคำสั่งสืบลำดับแบบคงที่ ที่ใช้งานกับเมธอดของเอนติตีเมเนเจอร์เท่านั้น คือคำสั่งจะต้องเป็นข้อมูลประเภทสายอักขระ (String) ที่มีข้อความคงที่เพียงหนึ่งข้อความ ไม่มีการเชื่อมข้อความมากกว่าหนึ่งข้อความเข้าด้วยกัน (Concatenation String) แต่รองรับคำสั่งสืบลำดับที่มีตัวแปรได้
4. เครื่องมือจะวิเคราะห์โปรแกรมในรูปแบบไฟล์คลาส (*.class) เท่านั้น นั่นคือจะไม่พิจารณาโค้ดที่อยู่ในหน้าเว็บ อาทิ ไฟล์เจเอสพี (*.jsp) เป็นต้น เนื่องจากไฟล์คลาส เป็นไฟล์ที่ผ่านการคอมไพล์แล้ว จึงไม่มีปัญหาเรื่องผิดวากยสัมพันธ์ (Syntax Error) อีกทั้งยังสามารถใช้ไฟล์ของโปรแกรมที่ใช้งานอยู่จริงมาวิเคราะห์ได้ทันที โดยไม่ต้องหาซอร์สโค้ด
5. เครื่องมือที่พัฒนาขึ้นไม่สามารถวิเคราะห์โปรแกรมที่ใช้งานตารางแบบทรศนะ (View)
6. เครื่องมือที่พัฒนาขึ้นไม่มีตัวไคร์ฟเวอร์ สำหรับเชื่อมต่อฐานข้อมูลให้ ผู้ใช้งานเครื่องมือจำเป็นต้องมีไคร์ฟเวอร์ไว้ใช้สำหรับเชื่อมต่อฐานข้อมูลเอง และผู้ใช้งานเครื่องมือจำเป็นต้องทราบวิธีเขียนสายอักขระสำหรับเชื่อมต่อฐานข้อมูลแต่ละฐานข้อมูลเอง (Connection String)

7. ผู้ใช้งานจำเป็นต้องรวบรวมโปรแกรมที่อ้างอิงฐานข้อมูลเดียวกันให้ครบทุกโปรแกรม และนำเข้าโปรแกรมต่างๆ เพื่อใช้ในการวิเคราะห์ครั้งเดียวกัน เนื่องจากเครื่องมือจะรวมผลลัพธ์การอ้างอิงฐานข้อมูลของทุกโปรแกรมเข้าด้วยกัน แล้วจึงหาตาราง และคอลัมน์ที่ไม่มีโปรแกรมอ้างอิง ถ้านำเข้าข้อมูลโปรแกรมทีละโปรแกรมแล้ววิเคราะห์ จะทำให้ข้อมูลการอ้างอิงฐานข้อมูลของ โปรแกรมมีไม่ครบ ซึ่งจะทำให้ผลลัพธ์ผิดพลาดได้

8. เครื่องมือจะวิเคราะห์โค้ดที่เข้าไม่ถึงเฉพาะเอนติตีเท่านั้น นั่นคือแม้ว่าจะมีคำสั่งสืบค้นที่เข้าไม่ถึง เครื่องมือก็จะมองว่าคำสั่งดังกล่าวอาจถูกเรียกใช้งานได้จากเหตุการณ์ (Event) ในหน้าเว็บ เนื่องด้วยขอบเขตงานวิจัยจะวิเคราะห์เฉพาะไฟล์คลาสของจาวาเท่านั้น เครื่องมือจึงพิจารณา คำสั่งสืบค้นในเมธอดที่เข้าไม่ถึงด้วย

9. เครื่องมือรองรับที่จะวิเคราะห์โปรแกรมที่ใช้ฐานข้อมูลเพียงฐานข้อมูลเดียวเท่านั้น

6.4 แนวทางการพัฒนาต่อ

1. วิธีการที่นำเสนอนี้สามารถวิเคราะห์โปรแกรม ที่ใช้การแปลงวัตถุเป็นตาราง ความสัมพันธ์เชื่อมต่อกับฐานข้อมูลเท่านั้น แต่ยังมีวิธีเขียนโปรแกรมติดต่อกับฐานข้อมูลอยู่อีกหลายวิธี เช่น การเขียนโปรแกรมต่อตรงกับฐานข้อมูลโดยใช้คำสั่งพื้นฐานที่มีมาให้ในแต่ละภาษา เช่น ในภาษาจาวาใช้เจดีบีซี (JDBC) แนวทางการพัฒนาต่อยอดอีกแนวทางหนึ่ง คือพัฒนาให้ เครื่องมือรองรับที่จะวิเคราะห์โปรแกรมที่เชื่อมต่อกับฐานข้อมูลในวิธีที่หลากหลาย

2. วิธีการที่นำเสนอยังไม่รองรับที่จะวิเคราะห์โปรแกรมที่เรียกใช้งานสตอร์โพรซีเจอร์ หนึ่งในแนวทางการพัฒนาต่อ คือต่อยอดวิธีการนี้ให้รองรับที่จะวิเคราะห์โปรแกรมที่ใช้งาน สตอร์โพรซีเจอร์

3. เครื่องมือที่พัฒนานี้รองรับที่จะวิเคราะห์เฉพาะ โปรแกรมภาษาจาวาที่ใช้ เจพีเอในการติดต่อกับฐานข้อมูล และใช้บรรณนิทัศน์ของจาวาในการระบุความสัมพันธ์ระหว่างคลาส กับตารางฐานข้อมูลเท่านั้น แต่ในความเป็นจริงมีวิธีการระบุความสัมพันธ์ระหว่างคลาสกับตาราง ฐานข้อมูลอยู่หลายวิธี เช่น ใช้ไฟล์เอ็กซ์เอ็มแอล ในการระบุความสัมพันธ์นี้ ซึ่งสามารถพัฒนาต่อยอดให้เครื่องมือรองรับการใช้ไฟล์เอ็กซ์เอ็มแอลในการระบุความสัมพันธ์ระหว่างคลาสกับตาราง ฐานข้อมูล หรือพัฒนาต่อยอดให้เครื่องมือรองรับที่จะวิเคราะห์โปรแกรมภาษาอื่นๆ ได้

4. ในขั้นตอนการหาแอตทริบิวต์ของเอนดีตีที่เข้าไม่ถึง ซึ่งผู้วิจัยเลือกใช้เครื่องมือที่เป็นโอเพ่นซอร์ส ชื่อว่า “com.mebigfatguy.deadmethods” [19] ปรากฏว่าขั้นตอนนี้เป็นขั้นตอนที่ใช้เวลามากที่สุด ซึ่งแนวทางการพัฒนาต่ออีกแนวทางหนึ่ง คือ ใช้เครื่องมืออื่นๆ หรือแนวทางอื่นในการหาแอตทริบิวต์ของเอนดีตีที่เข้าไม่ถึง เพื่อเพิ่มประสิทธิภาพการทำงานของเครื่องมือให้ดียิ่งขึ้น

รายการอ้างอิง

- [1] Gallagher, J.M. Ph.D. The Data Asset: Databases, Business Intelligence, and Competitive Advantage. 2009 (Unpublished Manuscript)
- [2] Fairley, J. The Growth Trend in Data Storage. Information Management [Online]. 2012. Available from : <http://www.information-management.com/news/data-storage-data-center-Digital-Realty-10022155-1.html> [2012, July 27]
- [3] Safar M. ,and Al-Najjar A. Data Growth in Banking Sector. DEXA '06 Proceedings of the 17th International Conference on Database and Expert Systems Applications (September 2006) : 667-674.
- [4] Amber, S.W. ,and Sadalage, P.J. Refactoring Database: Evolution Database Design. Boston : Addison Wesley, 2006.
- [5] IBM, Corporation. Database size, database reorganization, and performance considerations for Tivoli Storage Manager Version 6 servers. IBM Support [Online]. 2012. Available from : <http://www01.ibm.com/support/docview.wss?uid=swg21452146> [2012, July 27]
- [6] Li Y. ,and Lü K. Performance Issues of a Web Database. DEXA '00 Proceedings of the 11th International Conference on Database and Expert Systems Applications (September 2000) : 825-834
- [7] Mcgehee, B. Performance Tuning Tips for SQL Server Backup and Restore. SQL Server Performance [Online]. 2009. Available from : internet: <http://www.sql-server-performance.com/2007/backup-restore-tuning/> [2012, July 27]
- [8] Nelson, R. How to estimate your real backup requirements. Next Paradigm Systems [Online]. 1999. Available from : <http://nextparadigmsystems.com/EstimateYourBackup.htm> [2012, July 27]
- [9] Creative Commons, Organization. Defining Refactoring. Source Making Teaching IT Professionals [Online]. 2009. Available from : <http://sourcemaking.com/refactoring/defining-refactoring> [2012, July 27]
- [10] Burleson D.K. Find unused columns with dba_unused_col_tabs. Burleson Consulting [Online]. 2010. Available from : http://www.dba-oracle.com/t_dba_unused_col_tabs_columns.htm [2012, July 27]

- [11] Menard B. Mysql-Find unused tables and columns. StackOverflow [Online]. 2009.
Available from : <http://stackoverflow.com/questions/1391712/mysql-find-unused-tables-and-columns> [2012, July 27]
- [12] Ambler, S.W. ,and Line M. Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. US : IBM Press, 2012.
- [13] Keith M. ,and Schincariol M. Pro JPA 2: Mastering the Java Persistence API. US : Apress, 2009.
- [14] Ambler S.W. Agile Database Techniques: Effective Strategies the Agile Software Developer. US : Wiley Publishing, 2003.
- [15] Oracle, Corporation. The JavaEE 6 Tutorial. California : Oracle Corporation, 2011.
- [16] DeMichiel L. JSR 317: Java Persistence API, Version 2.0. California : Sun Microsystems, 2009.
- [17] Debray, S.K. ,and Evans W. Compiler Techniques for Code Compaction. ACM Transactions on Programming Languages and Systems (TOPLAS) 22 (March 2000) : 378-415.
- [18] Salste T. Dead code detection and removal. Aivosto [Online]. 2010. Available from : <http://www.aivosto.com/vbtips/deadcode.html>, [2012, July 27]
- [19] Brosius D. Deadmethods. GitHub [Online]. 2009. Available from : <https://github.com/mebigfatguy/deadmethods> [2012, July 27]
- [20] Maule A., Emmerich W. ,and Rosenblum D.S. Impact Analysis of Database Schema Changes. ICSE '08 Proceedings of the 30th international conference on Software engineering (May 2008) : 451-460.
- [21] Vasiliev Y. Querying JPA Entities with JPQL and Native SQL. Oracle Technology Network [Online]. 2008. Available from : <http://www.oracle.com/technetwork/articles/vasiliev-jpql-087123.html> [2012, July 27]
- [22] Brosius D. BCEL: Byte Code Engineering Library. Apache Commons [Online]. 2011.
Available from : <http://commons.apache.org/bcel/manual.html> [2012, July 27]
- [23] Lindholm, T. ,Yellin F. ,Bracha G. ,and Buckley A. The Java Virtual Machine Specification Java SE 7 Edition. California : Oracle Corporation, 2011.

ภาคผนวก

ภาคผนวก ก

รายละเอียดประเภทการรีแฟคเตอร์ และคำสั่งดำเนินการ

การรีแฟคเตอร์ฐานข้อมูลเป็นแนวคิดของสก็อต แอมเบลอ เพื่อใช้ปรับปรุงฐานข้อมูลให้มีประสิทธิภาพมากยิ่งขึ้น และง่ายต่อการอ่านทำความเข้าใจ ซึ่งสก็อต แอมเบลอได้แบ่งประเภทของการรีแฟคเตอร์ฐานข้อมูลไว้ด้วยกัน 5 ประเภท โดยมีคำสั่งในแต่ละประเภทดังแสดงในตารางที่ ก.1 ดังนี้

ตารางที่ ก.1 ประเภทของการรีแฟคเตอร์ฐานข้อมูล และคำสั่งดำเนินการที่เกี่ยวข้องกับการรีแฟคเตอร์ฐานข้อมูลแต่ละประเภท

ประเภท	คำสั่งที่เกี่ยวข้อง
1. การรีแฟคเตอร์โครงสร้าง (Structural Refactoring)	Drop Column Drop Table Drop View Introduce Calculated Column Introduce Surrogate Key Merge Columns Merge Tables Move Column Rename Column Rename Table Rename View Replace LOB With Table Replace Column Replace One-to-Many With Associative Table Replace Surrogate Key with Natural Key

ตารางที่ ก.1 ประเภทของการรีแฟคเตอร์ฐานข้อมูล และคำสั่งดำเนินการที่เกี่ยวข้องกับการ
รีแฟคเตอร์ฐานข้อมูลแต่ละประเภท (ต่อ)

ประเภท	คำสั่งที่เกี่ยวข้อง
1. การรีแฟคเตอร์โครงสร้าง (Structural Refactoring)	Split Column Split Table
2. การรีแฟคเตอร์คุณภาพของข้อมูล (Data Quality Refactoring)	Add Lookup Table Apply Standard Codes Apply Standard Type Consolidate Key Strategy Drop Column Constraint Drop Default Value Drop Non-Nullable Constraint Introduce Column Constraint Introduce Common Format Introduce Default Value Make Column Non-Nullable Move Data Replace Type Code With Property Flags
3. การรีแฟคเตอร์อ้างอิงของบูรณภาพ (Referential Integrity Refactoring)	Add Foreign Key Constraint Add Trigger for Calculated Column Drop Foreign Key Constraint Introduce Cascading Delete Introduce Hard Delete Introduce Soft Delete

ตารางที่ ก.1 ประเภทของการรีแฟคเตอร์ฐานข้อมูล และคำสั่งดำเนินการที่เกี่ยวข้องกับการ
รีแฟคเตอร์ฐานข้อมูลแต่ละประเภท (ต่อ)

ประเภท	คำสั่งที่เกี่ยวข้อง
3. การรีแฟคเตอร์อ้างอิงของบรรณภาพ (Referential Integrity Refactoring)	Introduce Trigger for History
4. การรีแฟคเตอร์สถาปัตยกรรม (Architectural Refactoring)	Add CRUD Methods Add Mirror Table Add Read Method Encapsulate Table With View Introduce Calculation Method Introduce Index Introduce Read Only Table Migrate Method From Database Migrate Method to Database Replace Method(s) With View Replace View With Method(s) Use Official Data Source
5. การรีแฟคเตอร์เมธอดของฐานข้อมูล (Method Database Refactoring)	Add Parameter Consolidate Conditional Expression Decompose Conditional Extract Method Introduce Variable Parameterize Methods Remove Control Flag

ตารางที่ ก.1 ประเภทของการรีแฟกเตอร์ฐานข้อมูล และคำสั่งดำเนินการที่เกี่ยวข้องกับการ
รีแฟกเตอร์ฐานข้อมูลแต่ละประเภท (ต่อ)

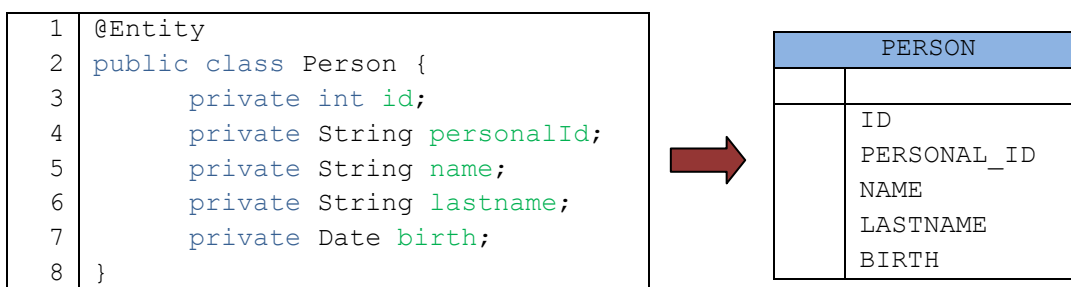
ประเภท	คำสั่งที่เกี่ยวข้อง
5. การรีแฟกเตอร์เมธอดของฐานข้อมูล (Method Database Refactoring)	Remove Middleman Remove Parameter Rename Method Reorder Parameters Replace Literal With Table Lookup Replace Nested Expression With Guard Clauses Replace Parameter With Specific Methods Split Temporary Variable Substitute Algorithm

ภาคผนวก ข

วิธีการใช้บรรณนิทัศน์เจพีเอระบุความสัมพันธ์ การแปลงวัตถุเป็นตารางความสัมพันธ์

1. การใช้บรรณนิทัศน์ระบุคลาสจาวาให้เป็นเอนติตีคลาส

วิธีการใช้บรรณนิทัศน์ระบุว่าคลาสใดบ้างที่จะเป็นเอนติตีคลาส สามารถทำได้โดยใช้บรรณนิทัศน์ `@Entity` ระบุไว้ที่ส่วนหัวของคลาส ดังภาพที่ ข.1 โดยแอตทริบิวต์ในคลาสเอนติตีทุกตัวจะถูกแปลงไปเป็นคอลัมน์ในตารางฐานข้อมูลโดยอัตโนมัติ



ภาพที่ ข.1 การใช้บรรณนิทัศน์ระบุคลาสให้เป็นเอนติตีคลาส

2. การกำหนดแอตทริบิวต์ที่ไม่ต้องการให้บันทึกลงในฐานข้อมูล

หากไม่ต้องการให้แอตทริบิวต์บันทึกลงในคอลัมน์ของตารางฐานข้อมูลสามารถทำได้ 2 วิธี คือ กำหนดแอตทริบิวต์นั้นให้เป็นประเภทชั่วคราว (Transient) และการใช้บรรณนิทัศน์ `@Transient` ระบุที่แอตทริบิวต์ ดังภาพที่ ข.2 บรรทัดที่ 8 - 9 ซึ่งตารางฐานข้อมูลที่แปลงจะมีลักษณะเหมือนกับแผนภาพอีอาร์ในภาพที่ ข.1

1	<code>@Entity</code>
2	<code>public class Person {</code>
3	<code>private int id;</code>
4	<code>private String personalId;</code>
5	<code>private String name;</code>
6	<code>private String lastname;</code>
7	<code>private Date birth;</code>
8	<code>private transient int age;</code>
9	<code>@Transient private String fullname;</code>
10	<code>}</code>

ภาพที่ ข.2 การใช้บรรณนิทัศน์กำหนดแอตทริบิวต์ที่ไม่ต้องการจัดเก็บลงฐานข้อมูล

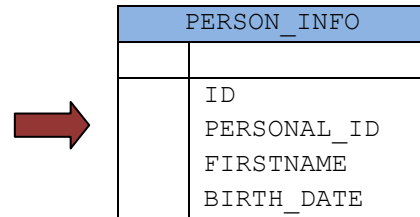
3. การเปลี่ยนกำหนดชื่อตาราง และคอลัมน์ด้วยตัวเอง

โดยปกติแล้วคลาสเอนติตี้ จะใช้ชื่อคลาส และชื่อของแอตทริบิวต์เป็นชื่อตาราง และคอลัมน์ ในฐานข้อมูล ตามลำดับ แต่ถ้าต้องการกำหนดชื่อตาราง และคอลัมน์เอง สามารถทำได้โดยใช้ บรรณนิทศน์ `@Table` บนคลาสเพื่อเปลี่ยนชื่อตาราง และ `@Column` บนแอตทริบิวต์เพื่อเปลี่ยนชื่อคอลัมน์ ดังแสดงในภาพที่ ข.3 ในบรรทัดที่ 2, 6 และ 8

```

1 @Entity
2 @Table (name="PERSON_INFO")
3 public class Person {
4     private int id;
5     private String personalId;
6     @Column (name="FIRSTNAME")
7     private String name;
8     @Column (name="BIRTH_DATE")
9     private Date birth;
10 }

```



ภาพที่ ข.3 การใช้บรรณนิทศน์กำหนดชื่อตาราง และคอลัมน์ฐานข้อมูล

4. การใช้วัตถุแบบฝังตัว (Embedded Object)

การเขียนโปรแกรมเชิงวัตถุ สามารถนำคลาสที่เขียนขึ้นไปใช้ได้หลายๆที่ โดยไม่ต้องเขียนใหม่ ซึ่งเจทีเอก็รองรับการใช้วัตถุแบบฝังตัวด้วย เช่น สร้างคลาสที่อยู่ขึ้น โดยคลาสที่อยู่เป็นแอตทริบิวต์ภายในคลาสพนักงาน และบริษัท ดังภาพที่ ข.4

```

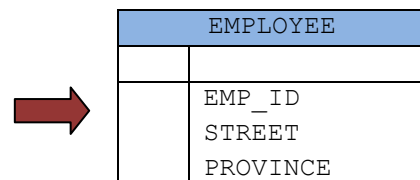
1 @Embeddable
2 public class Address {
3     private String street;
4     private String province;
5 }

```

```

1 @Entity
2 public class Employee {
3     private String emp_id;
4     @Embedded
5     private Address address;
6 }

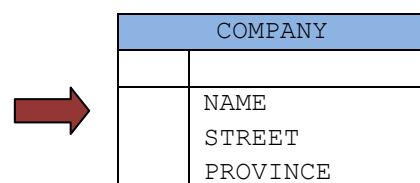
```



```

1 @Entity
2 public class Company {
3     private String name;
4     @Embedded
5     private Address address;
6 }

```



ภาพที่ ข.4 การใช้บรรณนิทศน์ระบุการใช้แอตทริบิวต์แบบวัตถุแบบฝังตัว

โดยในคลาสที่อยู่จะต้องใช้บรรณนิทัศน์ `@Embeddable` เพื่อบอกว่าคลาสนี้สามารถนำไปใช้เป็นแอตทริบิวต์ในเอนติตีคลาสอื่นๆได้ โดยจากตัวอย่าง นำคลาสที่อยู่ไปใช้เป็นแอตทริบิวต์ในคลาสพนักงาน และบริษัทอีกทีหนึ่ง โดยจะต้องประกาศบรรณนิทัศน์ `@Embedded` เพื่อบอกว่าเรียกใช้คลาสที่อยู่เป็นส่วนหนึ่งในคลาส คอลัมน์ถนน และจังหวัดจึงปรากฏอยู่ในตารางพนักงาน และบริษัทด้วย

5. การกำหนดชื่อคอลัมน์ใหม่ให้แก่วัตถุแบบฝังตัว

จากภาพตัวอย่างที่ ข.4 จะเห็นว่าตารางพนักงาน และตารางบริษัทมีชื่อคอลัมน์ถนน และจังหวัดเหมือนกัน ซึ่งถ้าต้องการเปลี่ยนชื่อคอลัมน์ดังกล่าวให้แต่ละตารางมีชื่อที่ต่างกัน สามารถทำได้โดยใช้บรรณนิทัศน์ `@AttributeOverrides` และ `@AttributeOverride` โดย `@AttributeOverrides` จะใช้ระบุชื่อคอลัมน์ใหม่หลายๆคอลัมน์ดังภาพที่ ข.5 บรรทัดที่ 5 - 10 ส่วน `@AttributeOverride` จะสามารถกำหนดได้แค่คอลัมน์เดียว ดังภาพที่ ข.6 บรรทัดที่ 5 - 7

```

1  @Entity
2  public class Employee {
3      private String emp_id;
4      @Embedded
5      @AttributeOverrides({
6          @AttributeOverride(name="street",
7                          column=@Column(name="road")),
8          @AttributeOverride(name="province",
9                          column=@Column(name="area"))
10     })
11     private Address address;
12 }

```



EMPLOYEE	
	EMP_ID
	ROAD
	AREA

ภาพที่ ข.5 การใช้บรรณนิทัศน์ `@AttributeOverrides` กำหนดชื่อคอลัมน์ใหม่ให้วัตถุแบบฝังตัว


```

1 @Entity
2 public class Company {
3     private String name;
4     @Embedded
5     @AttributeOverride(name="province",
6                         column=@Column(name="region"))
7     )
8     private Address address;
9 }

```



COMPANY	
	NAME
	STREET
	REGION

ภาพที่ ข.6 การใช้บรรณนิทัศน์ `@AttributeOverride` กำหนดชื่อคอลัมน์ใหม่ให้วัตถุแบบฝังตัว

6. การกำหนดคอลัมน์ให้เป็นคีย์หลัก

การกำหนดคอลัมน์ให้กลายเป็นคีย์หลักของตาราง สามารถทำได้โดยใช้บรรณนิทัศน์ `@Id` ระบุที่คอลัมน์ดังภาพที่ ข.7 บรรทัดที่ 3

```

1 @Entity
2 public class Person {
3     @Id private int id;
4     private String personalId;
5     private String name;
6     private String lastname;
7     private Date birth;
8 }

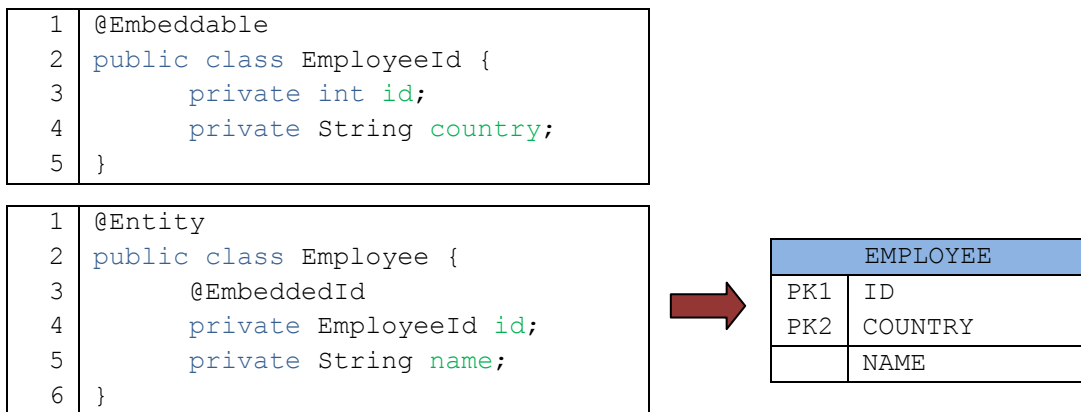
```



PERSON	
PK	ID
	PERSONAL_ID
	NAME
	LASTNAME
	BIRTH

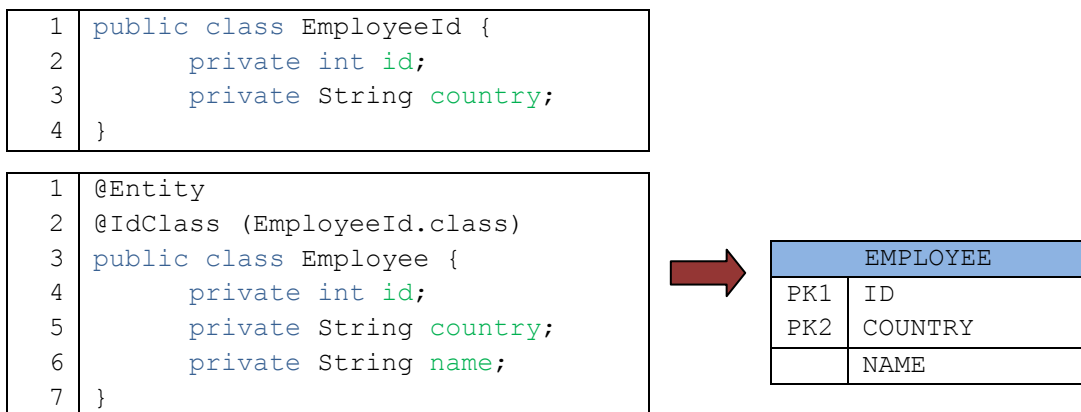
ภาพที่ ข.7 การใช้บรรณนิทัศน์ระบุคีย์หลัก

ถ้าต้องการระบุคีย์ผสม (Composite Key) จะมีวิธีการประกาศที่แตกต่างออกไป โดยจะต้องสร้างคลาสที่เป็นคีย์ผสมแยกต่างหากเป็นแบบวัตถุแบบฝังตัว ส่วนคลาสที่ใช้งานจะต้องระบุบรรณนิทัศน์ `@EmbeddId` ดังภาพที่ ข.8



ภาพที่ ข.8 การใช้บรรณนิทัศน์ระบุบุคคิยผสมโดยใช้ @EmbeddedId

นอกจากนี้ ยังมีวิธีการระบุบุคคิยผสมอีกวิธีหนึ่ง ซึ่งจะต้องสร้างคลาสทัวๆไป โดยมีแอตทริบิวต์ต่างๆ ที่ต้องการใช้เป็นคิย ส่วนในคลาสเอนติตีจะต้องระบุบรรณนิทัศน์เพิ่มเติมโดยใช้ @IdClass แล้วระบุคลาสที่สร้างขึ้นเพื่อเป็นคิย อีกทั้งยังต้องสร้างแอตทริบิวต์ต่างๆให้เหมือนกับคลาสที่เป็นคิยด้วย ดังแสดงในภาพที่ ข.9



ภาพที่ ข.9 การใช้บรรณนิทัศน์ระบุบุคคิยผสมโดยใช้ @IdClass

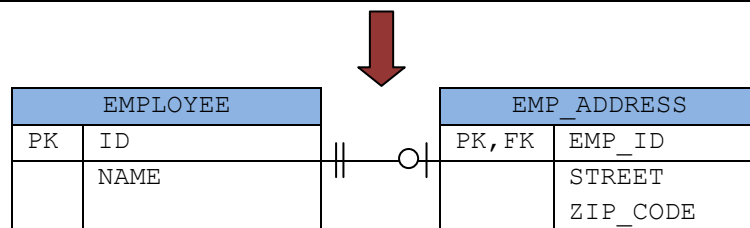
7. การกำหนดเอนติตีคลาส ให้สัมพันธ์กับตารางฐานข้อมูลมากกว่า 1 ตาราง

นอกจากความสัมพันธ์ระหว่างเอนติตีคลาส กับตารางฐานข้อมูลจะเป็นแบบ 1 คลาสสัมพันธ์กับ 1 ตารางฐานข้อมูลดังในตัวอย่างก่อนหน้าที่กล่าวมาแล้ว เจพีเอ ยังสามารถกำหนดความสัมพันธ์ให้แก่เอนติตีคลาส ให้สัมพันธ์กับตารางฐานข้อมูลได้มากกว่า 1 ตาราง โดยใช้บรรณนิทัศน์ @SecondaryTables ในกรณีที่ต้องการระบุตารางอื่นๆมากกว่า 1 ตาราง และใช้ @SecondaryTable กำหนดรายละเอียดของแต่ละตาราง ซึ่งแอตทริบิวต์ในเอนติตีคลาส ถ้าไม่ใช่คอลัมน์ในตารางหลักก็จำเป็นต้องบอกชื่อตารางด้วย ดังภาพที่ ข.10

```

1  @Entity
2  @Table (name="EMP")
3  @SecondaryTable (name="EMP_ADDRESS",
4      pkJoinColumns=@PrimaryKeyJoinColumn (name="EMP_ID"))
5  public class Employee {
6      @Id private int id;
7      private String name;
8      @Column (table="EMP_ADDRESS")
9      private String street;
10     @Column (name="ZIP_CODE", table="EMP_ADDRESS")
11     private String zip;
12 }

```



ภาพที่ ข.10 การกำหนดเอนติตีคลาสให้สัมพันธ์กับตารางมากกว่า 1 ตาราง

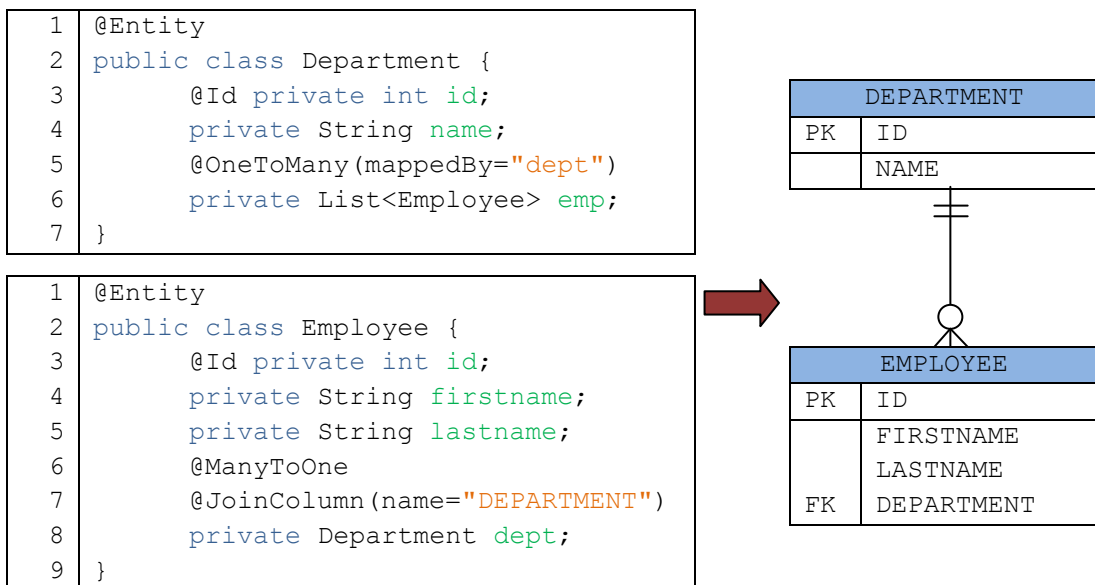
8. การกำหนดความสัมพันธ์แก่เอนติตีคลาส

บรรณนิทัศน์ของเจพีเอ สามารถระบุความสัมพันธ์ระหว่างตารางฐานข้อมูล ไม่ว่าจะเป็นความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One Relationships) แบบหนึ่งต่อหลาย (One to Many Relationships) และแบบหลายต่อหลาย (Many to Many Relationships) โดยมีรายละเอียดดังนี้

8.1 ความสัมพันธ์แบบหนึ่งต่อหลาย

ถ้ามองในมุมมองของการออกแบบฐานข้อมูลให้สามารถเก็บข้อมูลที่มีความสัมพันธ์แบบหนึ่งต่อหลาย ก็จะต้องมีตารางหนึ่งที่มีคอลลัมน์สำหรับเก็บคีย์นอกเพื่อชี้มายังอีกตารางหนึ่ง โดยฝั่งที่เก็บคีย์นอกจะต้องเป็นตารางที่มีข้อมูลเป็นแบบกลุ่ม คือมีข้อมูลได้หลายตัว ส่วนตารางอีกฝั่งหนึ่งไม่จำเป็นต้องมีคอลลัมน์ใดๆ ดังภาพที่ ข.11 ซึ่งแสดงความสัมพันธ์แบบหนึ่งต่อหลาย

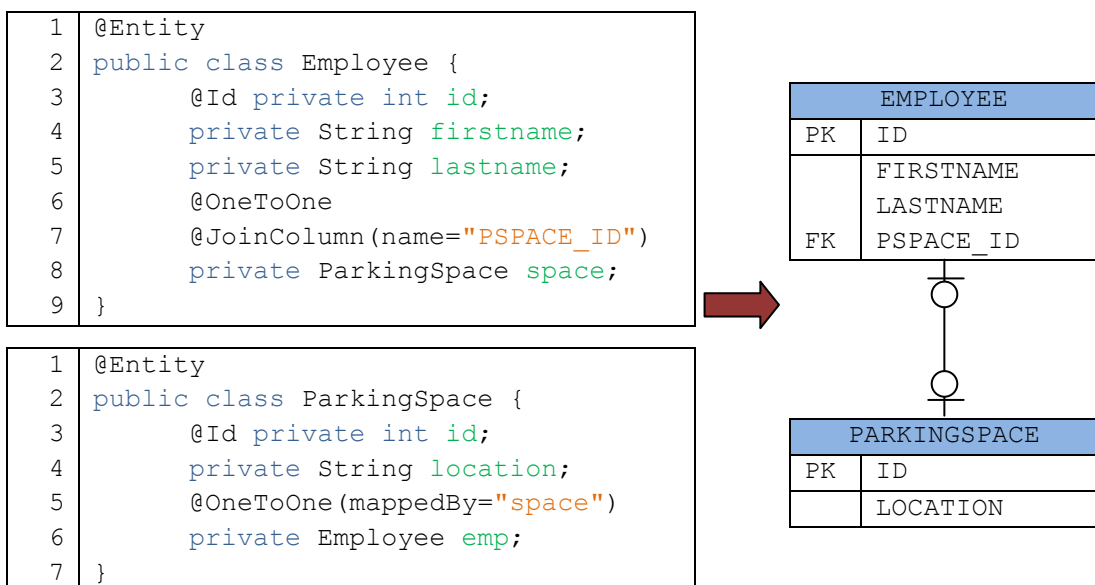
โดยจากตัวอย่างจะเห็นได้ว่า แผนกหนึ่งจะมีพนักงานได้หลายคน นั่นคือแผนกมีความสัมพันธ์กับพนักงานแบบหนึ่งต่อหลาย ส่วนพนักงานจะสังกัดได้เพียง 1 แผนก นั่นคือถ้ามองในมุมมองของพนักงาน จะมีความสัมพันธ์แบบหลายต่อหนึ่ง กับแผนก ดังนั้นตารางของพนักงานจึงต้องเก็บคีย์นอกเพื่อสร้างความสัมพันธ์กับแผนก โดยใช้บรรณนิทัศน์ @JoinColumn กำหนดว่าเป็นคีย์นอกไปยังตารางแผนก



ภาพที่ ข.11 การสร้างความสัมพันธ์แบบหนึ่งต่อหลาย

8.2 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

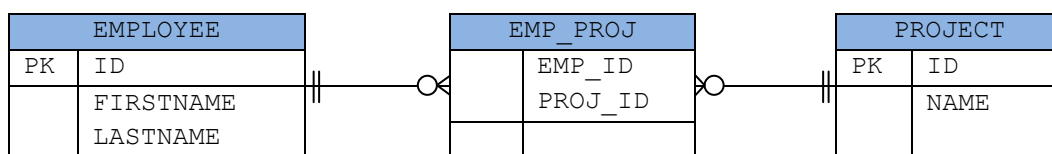
ความสัมพันธ์แบบหนึ่งต่อหนึ่งนั้นมีความคล้ายคลึงกับความสัมพันธ์แบบหนึ่งต่อหลาย นั่นคือจะต้องมีตารางหนึ่งที่มีคอล์มสำหรับเก็บคีย์นอกเพื่อชี้ไปยังอีกตารางหนึ่ง โดยจะใช้ตารางใดก็ได้เป็นตารางที่เก็บคีย์นอกไปยังอีกตารางหนึ่ง ดังแดงในภาพที่ ข.12



ภาพที่ ข.12 การสร้างความสัมพันธ์แบบหนึ่งต่อหนึ่ง

8.3 ความสัมพันธ์แบบหลายต่อหลาย

การสร้างตารางเพื่อเก็บข้อมูลที่มีความสัมพันธ์แบบหลายต่อหลายนั้น ไม่สามารถสร้างได้โดยใช้ตารางเพียง 2 ตาราง จะต้องมีการเพิ่มอีก 1 ตารางการรวมกลุ่ม (Association Table) สำหรับช่วยเก็บข้อมูลที่มีความสัมพันธ์แบบหลายต่อหลาย เช่น พนักงานกับโครงการ มีความสัมพันธ์กันแบบหลายต่อหลาย โดยพนักงานสามารถทำงานได้หลายโครงการ และในโครงการก็มีพนักงานหลายคนทำงาน ซึ่งการออกแบบฐานข้อมูลจะต้องมีตารางแทรกตรงกลางเพื่อช่วยสร้างความสัมพันธ์ของตารางพนักงาน กับโครงการให้สัมพันธ์กันแบบหลายต่อหลาย ดังแสดงในภาพที่ ข.13



ภาพที่ ข.13 แผนภาพอีอาร์ตารางสำหรับเก็บข้อมูลที่มีความสัมพันธ์แบบหลายต่อหลาย

ดังนั้นการระบุบรรณนิทัศน์เจพีเอ จึงจำเป็นต้องกำหนดค่าของตารางการรวมกลุ่มที่เพิ่มเข้ามา โดยใช้บรรณนิทัศน์ @JoinTable สำหรับกำหนดค่าให้แก่ตารางการรวมกลุ่ม ดังภาพที่ ข.14 โดยจะต้องกำหนดคุณลักษณะที่สำคัญ 2 คุณลักษณะ คือ คอลัมน์สำหรับการเชื่อม (joinColumns) และคอลัมน์สำหรับการเชื่อมในทิศตรงกันข้าม (inverseJoinColumns) โดยคอลัมน์สำหรับการเชื่อม จะกำหนดให้ตารางการรวมกลุ่มสร้างคีย์นอกขึ้นมายังตารางของคลาสที่ใช้บรรณนิทัศน์ ส่วนคอลัมน์สำหรับการเชื่อมในทิศตรงกันข้าม จะกำหนดคีย์นอกให้ชี้ไปยังตารางอีกฝั่งหนึ่งของความสัมพันธ์แบบหลายต่อหลาย

```
1 @Entity
2 public class Employee {
3     @Id private int id;
4     private String firstname;
5     private String lastname;
6     @ManyToMany
7     @JoinTable(name="EMP_PROJ",
8         joinColumns=@JoinColumn(name="EMP_ID"),
9         inverseJoinColumns=@JoinColumn(name="PROJ_ID")
10    )
11    private List<Project> projects;
12 }
```

```
1 @Entity
2 public class Project {
3     @Id private int id;
4     private String name;
5     @ManyToMany(mappedBy="projects")
6     private List< Employee> employees;
7 }
```

ภาพที่ ข.14 การสร้างความสัมพันธ์แบบหลายต่อหลาย

ภาคผนวก ค

รายการรหัสคำสั่งเครื่องของภาษาจาวา

รหัสคำสั่งเครื่องของภาษาจาวา เป็นรหัสคำสั่งที่อยู่ในไฟล์คลาสของจาวา ซึ่งเมื่ออ่านไฟล์คลาสของจาวาแล้วจะพบคำสั่งต่างๆดังแสดงในตารางที่ ค.1 โดยแต่ละคำสั่งมีความหมายดังต่อไปนี้

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา

รหัสคำสั่ง	คำอธิบาย
aaload	load onto the stack a reference from an array
aastore	store into a reference in an array
aconst_null	push a null reference onto the stack
aload	load a reference onto the stack from a local variable #index
aload_0	load a reference onto the stack from local variable 0
aload_1	load a reference onto the stack from local variable 1
aload_2	load a reference onto the stack from local variable 2
aload_3	load a reference onto the stack from local variable 3
anewarray	create a new array of references of length count and component type identified by the class reference index (indexbyte1 << 8 + indexbyte2) in the constant pool
areturn	return a reference from a method
arraylength	get the length of an array
astore	store a reference into a local variable #index
astore_0	store a reference into local variable 0
astore_1	store a reference into local variable 1
astore_2	store a reference into local variable 2
astore_3	store a reference into local variable 3
athrow	throws an error or exception (notice that the rest of the stack is cleared, leaving only a reference to the Throwable)

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
baload	load a byte or Boolean value from an array
bastore	store a byte or Boolean value into an array
bipush	push a byte onto the stack as an integer value
caload	load a char from an array
castore	store a char into an array
checkcast	checks whether an objectref is of a certain type, the class reference of which is in the constant pool at index (indexbyte1 << 8 + indexbyte2)
dadd	add two doubles
daload	load a double from an array
dastore	store a double into an array
dcmpg	compare two doubles
dcmpl	compare two doubles
dconst_0	push the constant 0.0 onto the stack
dconst_1	push the constant 1.0 onto the stack
ddiv	divide two doubles
dload	load a double value from a local variable #index
dload_0	load a double from local variable 0
dload_1	load a double from local variable 1
dload_2	load a double from local variable 2
dload_3	load a double from local variable 3
dmul	multiply two doubles
dneg	negate a double
drem	get the remainder from a division between two doubles
dreturn	return a double from a method
dstore	store a double value into a local variable #index
dstore_0	store a double into local variable 0
dstore_1	store a double into local variable 1

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
dstore_2	store a double into local variable 2
dstore_3	store a double into local variable 3
dsub	subtract a double from another
dup	duplicate the value on top of the stack
dup_x1	insert a copy of the top value into the stack two values from the top. value1 and value2 must not be of the type double or long.
dup_x2	insert a copy of the top value into the stack two (if value2 is double or long it takes up the entry of value3, too) or three values (if value2 is neither double nor long) from the top
dup2	duplicate top two stack words (two values, if value1 is not double nor long; a single value, if value1 is double or long)
dup2_x1	duplicate two words and insert beneath third word (see explanation above)
dup2_x2	duplicate two words and insert beneath fourth word
fadd	add two floats
faload	load a float from an array
fastore	store a float in an array
fcmpg	compare two floats
fcmpl	compare two floats
fconst_0	push 0.0f on the stack
fconst_1	push 1.0f on the stack
fconst_2	push 2.0f on the stack
fdiv	divide two floats
fload	load a float value from a local variable #index
fload_0	load a float value from local variable 0
fload_1	load a float value from local variable 1
fload_2	load a float value from local variable 2
fload_3	load a float value from local variable 3

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
fmul	multiply two floats
fneg	negate a float
frem	get the remainder from a division between two floats
freturn	return a float
fstore	store a float value into a local variable #index
fstore_0	store a float value into local variable 0
fstore_1	store a float value into local variable 1
fstore_2	store a float value into local variable 2
fstore_3	store a float value into local variable 3
fsub	subtract two floats
getfield	get a field value of an object objectref, where the field is identified by field reference in the constant pool index ($\text{index1} \ll 8 + \text{index2}$)
getstatic	get a static field value of a class, where the field is identified by field reference in the constant pool index ($\text{index1} \ll 8 + \text{index2}$)
goto	goes to another instruction at branchoffset (signed short constructed from unsigned bytes $\text{branchbyte1} \ll 8 + \text{branchbyte2}$)
goto_w	goes to another instruction at branchoffset (signed int constructed from unsigned bytes $\text{branchbyte1} \ll 24 + \text{branchbyte2} \ll 16 + \text{branchbyte3} \ll 8 + \text{branchbyte4}$)
i2l	convert an int into a long
i2f	convert an int into a float
i2d	convert an int into a double
l2i	convert a long to a int
l2f	convert a long to a float
l2d	convert a long to a double
f2i	convert a float to an int
f2l	convert a float to a long
f2d	convert a float to a double

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
d2i	convert a double to an int
d2l	convert a double to a long
d2f	convert a double to a float
i2b	convert an int into a byte
i2c	convert an int into a character
i2s	convert an int into a short
iadd	add two ints
iaload	load an int from an array
iand	perform a bitwise and on two integers
iastore	store an int into an array
iconst_m1	load the int value -1 onto the stack
iconst_0	load the int value 0 onto the stack
iconst_1	load the int value 1 onto the stack
iconst_2	load the int value 2 onto the stack
iconst_3	load the int value 3 onto the stack
iconst_4	load the int value 4 onto the stack
iconst_5	load the int value 5 onto the stack
idiv	divide two integers
if_acmpeq	if references are equal, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
if_acmpne	if references are not equal, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
if_icmpeq	if ints are equal, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
if_icmpne	if ints are not equal, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
if_icmplt	if value1 is less than value2, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
if_icmpge	if value1 is greater than or equal to value2, branch to instruction at branchoffset(signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
if_icmpgt	if value1 is greater than value2, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
if_icmple	if value1 is less than or equal to value2, branch to instruction at branchoffset(signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
ifeq	if value is 0, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
ifne	if value is not 0, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
iflt	if value is less than 0, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
ifge	if value is greater than or equal to 0, branch to instruction at branchoffset(signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
ifgt	if value is greater than 0, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
ifle	if value is less than or equal to 0, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
ifnonnull	if value is not null, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
ifnull	if value is null, branch to instruction at branchoffset (signed short constructed from unsigned bytes branchbyte1 << 8 + branchbyte2)
iinc	increment local variable #index by signed byte const
iload	load an int value from a local variable #index
iload_0	load an int value from local variable 0

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
iload_1	load an int value from local variable 1
iload_2	load an int value from local variable 2
iload_3	load an int value from local variable 3
imul	multiply two integers
ineg	negate int
instanceof	determines if an object objectref is of a given type, identified by class referenceindex in constant pool (indexbyte1 << 8 + indexbyte2)
invokedynamic	invokes a dynamic method identified by method reference index in constant pool (indexbyte1 << 8 + indexbyte2)
invokeinterface	invokes an interface method on object objectref, where the interface method is identified by method reference index in constant pool (indexbyte1 << 8 + indexbyte2)
invokespecial	invoke instance method on object objectref, where the method is identified by method reference index in constant pool (indexbyte1 << 8 + indexbyte2)
invokestatic	invoke a static method, where the method is identified by method referenceindex in constant pool (indexbyte1 << 8 + indexbyte2)
invokevirtual	invoke virtual method on object objectref, where the method is identified by method reference index in constant pool (indexbyte1 << 8 + indexbyte2)
ior	bitwise int or
irem	logical int remainder
ireturn	return an integer from a method
ishl	int shift left
ishr	int arithmetic shift right
istore	store int value into variable #index
istore_0	store int value into variable 0
istore_1	store int value into variable 1
istore_2	store int value into variable 2
istore_3	store int value into variable 3

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
isub	int subtract
iushr	int logical shift right
ixor	int xor
jsr	jump to subroutine at branchoffset (signed short constructed from unsigned bytes $\text{branchbyte1} \ll 8 + \text{branchbyte2}$) and place the return address on the stack
jsr_w	jump to subroutine at branchoffset (signed int constructed from unsigned bytes $\text{branchbyte1} \ll 24 + \text{branchbyte2} \ll 16 + \text{branchbyte3} \ll 8 + \text{branchbyte4}$) and place the return address on the stack
ladd	add two longs
laload	load a long from an array
land	bitwise and of two longs
lastore	store a long to an array
lcmp	compare two longs values
lconst_0	push the long 0 onto the stack
lconst_1	push the long 1 onto the stack
ldc	push a constant #index from a constant pool (String, int or float) onto the stack
ldc_w	push a constant #index from a constant pool (String, int or float) onto the stack (wide index is constructed as $\text{indexbyte1} \ll 8 + \text{indexbyte2}$)
ldc2_w	push a constant #index from a constant pool (double or long) onto the stack (wide index is constructed as $\text{indexbyte1} \ll 8 + \text{indexbyte2}$)
ldiv	divide two longs
lload	load a long value from a local variable #index
lload_0	load a long value from a local variable 0
lload_1	load a long value from a local variable 1
lload_2	load a long value from a local variable 2
lload_3	load a long value from a local variable 3
lmul	multiply two longs

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
lneg	negate a long
lookupswitch	a target address is looked up from a table using a key and execution continues from the instruction at that address
lor	bitwise or of two longs
lrem	remainder of division of two longs
lreturn	return a long value
lshl	bitwise shift left of a long value1 by value2 positions
lshr	bitwise shift right of a long value1 by value2 positions
lstore	store a long value in a local variable #index
lstore_0	store a long value in a local variable 0
lstore_1	store a long value in a local variable 1
lstore_2	store a long value in a local variable 2
lstore_3	store a long value in a local variable 3
lsub	subtract two longs
lushr	bitwise shift right of a long value1 by value2 positions, unsigned
lxor	bitwise exclusive or of two longs
monitorenter	enter monitor for object ("grab the lock" - start of synchronized() section)
monitorexit	exit monitor for object ("release the lock" - end of synchronized() section)
multianewarray	create a new array of dimensions dimensions with elements of type identified by class reference in constant pool index (indexbyte1 << 8 + indexbyte2); the sizes of each dimension is identified by count1, [count2, etc.]
new	create new object of type identified by class reference in constant pool index(indexbyte1 << 8 + indexbyte2)
newarray	create new array with count elements of primitive type identified by atype
nop	perform no operation
pop	discard the top value on the stack
pop2	discard the top two values on the stack (or one value, if it is a double or long)

ตารางที่ ค.1 รายการรหัสคำสั่ง และคำอธิบายรหัสคำสั่งเครื่องของภาษาจาวา (ต่อ)

รหัสคำสั่ง	คำอธิบาย
putfield	set field to value in an object objectref, where the field is identified by a field reference index in constant pool ($\text{indexbyte1} \ll 8 + \text{indexbyte2}$)
putstatic	set static field to value in a class, where the field is identified by a field reference index in constant pool ($\text{indexbyte1} \ll 8 + \text{indexbyte2}$)
ret	continue execution from address taken from a local variable #index (the asymmetry with jsr is intentional)
return	return void from method
saload	load short from array
sastore	store short to array
sipush	push a short onto the stack
swap	swaps two top words on the stack (note that value1 and value2 must not be double or long)
tableswitch	continue execution from an address in the table at offset index
wide	execute opcode, where opcode is either iload, fload, aload, lload, dload, istore, fstore, astore, lstore, dstore, or ret, but assume the index is 16 bit; or execute iinc, where the index is 16 bits and the constant to increment by is a signed 16 bit short \square
breakpoint	reserved for breakpoints in Java debuggers; should not appear in any class file
impdep1	reserved for implementation-dependent operations within debuggers; should not appear in any class file
impdep2	reserved for implementation-dependent operations within debuggers; should not appear in any class file

ภาคผนวก ง

คำบรรยายยูสเคส

คำบรรยายยูสเคสแสดงรายละเอียดและขั้นตอนการทำงานของแต่ละยูสเคส โดยในแต่ละยูสเคสจะบอกถึงลำดับขั้นตอนต่างๆ ทั้งทางฝั่งผู้ใช้งาน และระบบ ว่ามีลำดับขั้นตอนการทำงานในยูสเคสอย่างไร โดยมีรายละเอียดคำบรรยายแต่ละยูสเคสดังนี้

ตารางที่ ง.1 คำบรรยายยูสเคสการสร้างโครงการ

ชื่อยูสเคส	สร้างโครงการ	ระดับความสำคัญ	ปานกลาง
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	สร้างโครงการสำหรับการทำงานของเครื่องมือ		
เงื่อนไขก่อนหน้า	ไม่มี		
ขั้นตอนการทำงานปกติ	ผู้กระทำ	ระบบ	
	1. เลือกสร้างโครงการใหม่	2. ระบบแสดงรายละเอียดของโครงการให้กรอกข้อมูล	
	3. ระบุชื่อโครงการ		
	4. ระบุพารสำหรับเก็บโครงการ		
	5. กดปุ่มสร้างโครงการ	6. ระบบบันทึกไฟล์โครงการในพารที่กำหนด	
		7. ระบบแสดงหน้าจอในเครื่องมือเป็นสภาพแวดล้อมของโครงการใหม่	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์โครงการในพารที่ระบุ และมีสภาพแวดล้อมการทำงานเป็นของโครงการใหม่		

ตารางที่ ง.2 คำบรรยายยูสเคสการเปิดโครงการ

ชื่อยูสเคส	เปิดโครงการ	ระดับความสำคัญ	ปานกลาง
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	เปิดโครงการที่เคยทำงานค้างไว้เพื่อทำงานต่อจากเดิม		
เงื่อนไขก่อนหน้า	ไม่มี		
ขั้นตอนการทำงาน ปกติ	ผู้กระทำ	ระบบ	
	1. เลือกเปิดโครงการ		
	2. เลือกไฟล์โครงการที่ต้องการเปิด	3. ระบบอ่านไฟล์โครงการ	
		4. สร้างสภาพแวดล้อมให้เป็นของโครงการที่เปิด	
เงื่อนไขภายหลัง	ระบบแสดงสภาพแวดล้อมของไฟล์โครงการที่เปิด ให้ผู้กระทำพร้อมใช้งาน		

ตารางที่ ง.3 คำบรรยายยูสเคสการบันทึก

ชื่อยูสเคส	บันทึกโครงการ	ระดับความสำคัญ	ปานกลาง
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	บันทึกโครงการที่ทำงานอยู่ปัจจุบัน		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่		
ขั้นตอนการทำงาน ปกติ	ผู้กระทำ	ระบบ	
	1. เลือกบันทึกโครงการ	2. บันทึกโครงการลงในไฟล์ไปยังพารของโครงการ	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์โครงการลงในพารของไฟล์โครงการ		

ตารางที่ ง.4 คำบรรยายยุทธศาสตร์เพิ่มโปรแกรมเพื่อวิเคราะห์

ชื่อยุทธศาสตร์	เพิ่มโปรแกรมเพื่อวิเคราะห์	ระดับความสำคัญ	มาก
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	เพิ่มโปรแกรมเข้าในโครงการ เพื่อเป็นข้อมูลให้เครื่องมือวิเคราะห์ตาราง และ คอลัมน์ที่โปรแกรมอ้างอิง		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่		
ขั้นตอนการทำงาน ปกติ	ผู้กระทำ	ระบบ	
	1. เลือกเพิ่มโปรแกรมเข้าในโครงการ เพื่อวิเคราะห์	2. แสดงรายละเอียดของ โปรแกรมให้ กรอก	
	3. กรอกชื่อ โปรแกรม		
	4. กรอกคลาสพาธหลักของ โปรแกรมที่จะเพิ่ม		
	5. เพิ่มพารคั้งข้อมูลอื่นๆ ถ้า โปรแกรมมีการอ้างอิงถึงคำสั่ง ภายนอก		
	6. เพิ่มแพ็กเกจในคลาสจาวาที่ไม่ ต้องการให้เครื่องมือวิเคราะห์	8. เพิ่มข้อมูล โปรแกรมเข้าโครงการ	
	7. กดปุ่มเพิ่ม โปรแกรม		
		9. บันทึกโครงการ	
	ขั้นตอนการทำงาน ทางเลือก	5.1 ถ้าโปรแกรมไม่ได้อ้างอิงถึงคำสั่งภายนอก ไม่ต้องทำขั้นตอนนี้ 6.1 ถ้าต้องการให้เครื่องมือวิเคราะห์ทุกแพ็กเกจ ไม่ต้องทำขั้นตอนนี้ 7.1 ถ้าชื่อโปรแกรมซ้ำจะไม่อนุญาตให้เพิ่ม โปรแกรม และสิ้นสุดการทำงาน ยุทธศาสตร์	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์โครงการที่ข้อมูลโปรแกรมถูกเพิ่มในโครงการแล้ว		

ตารางที่ ง.5 คำบรรยายยูกเสศการเพิ่มฐานข้อมูลเพื่อวิเคราะห์

ชื่อยูกเสศ	เพิ่มฐานข้อมูลเพื่อวิเคราะห์	ระดับความสำคัญ	มาก
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	เพิ่มฐานข้อมูลเข้าในโครงการ เพื่อเป็นข้อมูลให้เครื่องมือวิเคราะห์ดูเค้าร่างฐานข้อมูลที่โปรแกรมต่างๆอ้างถึง		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่ - โครงการต้องยังไม่มีฐานข้อมูลที่ถูกเพิ่มเข้าไป		
ขั้นตอนการทำงานปกติ	ผู้กระทำ	ระบบ	
	1. เลือกเพิ่มฐานข้อมูล	2. แสดงรายละเอียดให้กรอก	
	3. กรอกข้อความเชื่อมต่อ สำหรับเชื่อมต่อฐานข้อมูล		
	4. กรอกชื่อผู้ใช้ และรหัสผ่านของฐานข้อมูล		
	5. กรอกชื่อคลาสไคร์ฟเวอร์		
	6. กรอกชื่อคาตาลอกสำหรับให้เครื่องมือวิเคราะห์เฉพาะคาตาลอกที่ระบุ		
	7. กรอกชื่อเค้าร่างสำหรับให้เครื่องมือวิเคราะห์เฉพาะเค้าร่างที่ระบุ		
	8. ระบุไฟล์ไคร์ฟเวอร์สำหรับให้เชื่อมต่อฐานข้อมูล		
	9. ทดสอบการเชื่อมต่อ		
	10. กดปุ่มเพิ่มฐานข้อมูล	11. เพิ่มข้อมูลฐานข้อมูลเข้าโครงการ	
		12. บันทึกโครงการ	
	ขั้นตอนการทำงานทางเลือก	7.1 ถ้าฐานข้อมูลไม่มีคาตาลอก ไม่จำเป็นต้องทำขั้นตอนนี้ 7.1 ถ้าฐานข้อมูลไม่มีเค้าร่าง ไม่จำเป็นต้องทำขั้นตอนนี้ 9.1 ถ้าการทดสอบเชื่อมต่อล้มเหลว ผู้กระทำต้องแก้ไขข้อมูลใหม่	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์โครงการที่ข้อมูลของฐานข้อมูลถูกเพิ่มในโครงการแล้ว		

ตารางที่ ง.6 คำบรรยายยูสเคสการลบโปรแกรม

ชื่อยูสเคส	ลบโปรแกรม	ระดับความสำคัญ	ปานกลาง
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	ลบโปรแกรมออกจากโครงการ		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่ - โครงการต้องมีโปรแกรมอย่างน้อย 1 โปรแกรม		
ขั้นตอนการทำงาน ปกติ	ผู้กระทำ	ระบบ	
	1. เลือกโปรแกรมที่ต้องการลบ		
	2. กดปุ่มลบโปรแกรม	3. ลบโปรแกรมออกจากโครงการ	
		4. บันทึกโครงการ	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์โครงการที่ข้อมูลโปรแกรมถูกลบออกจากโครงการแล้ว		

ตารางที่ ง.7 คำบรรยายยูสเคสการลบฐานข้อมูล

ชื่อยูสเคส	ลบฐานข้อมูล	ระดับความสำคัญ	ปานกลาง
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	ลบฐานข้อมูลออกจากโครงการ		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่ - โครงการต้องมีฐานข้อมูลอยู่แล้ว		
ขั้นตอนการทำงาน ปกติ	ผู้กระทำ	ระบบ	
	1. เลือกฐานข้อมูล		
	2. กดปุ่มลบฐานข้อมูล	3. ลบฐานข้อมูลออกจากโครงการ	
		4. บันทึกโครงการ	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์โครงการที่ไม่มีข้อมูลของฐานข้อมูล		

ตารางที่ ง.8 คำบรรยายยูสเคสการนำเข้าโปรแกรมที่วิเคราะห์แล้ว

ชื่อยูสเคส	นำเข้าโปรแกรมที่วิเคราะห์แล้ว	ระดับความสำคัญ	ปานกลาง
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	นำเข้าโปรแกรมที่วิเคราะห์แล้ว เข้ามาในโครงการ เพื่อใช้วิเคราะห์หาตารางและคอลัมน์ที่ไม่ถูกอ้างอิง โดยข้อมูลของ โปรแกรมที่วิเคราะห์แล้วจะมีรายการตาราง และคอลัมน์ที่โปรแกรมอ้างอิงอยู่แล้ว		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่		
ขั้นตอนการทำงานปกติ	ผู้กระทำ	ระบบ	
	1. เลือกนำเข้าโปรแกรม		
	2. เลือกไฟล์โปรแกรมที่วิเคราะห์แล้ว ที่จะนำเข้า	3. ระบบอ่านไฟล์นำเข้า	
		4. เพิ่มโปรแกรมที่นำเข้ลงในโครงการ	
		5. บันทึกโครงการ	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์โครงการที่มีข้อมูลโปรแกรมที่นำเข้แล้ว		

ตารางที่ ง.9 คำบรรยายยูสเคสการนำออกโปรแกรมที่วิเคราะห์แล้ว

ชื่อยูสเคส	นำออกโปรแกรมที่วิเคราะห์แล้ว	ระดับความสำคัญ	ปานกลาง
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	นำออกโปรแกรมที่วิเคราะห์แล้ว เพื่อนำไปใช้ในโครงการอื่นๆ ในกรณีที่โปรแกรมไม่มีการเปลี่ยนแปลงการอ้างอิงฐานข้อมูล		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่ - โปรแกรมที่นำออกต้องมีผลการวิเคราะห์ตาราง และคอลัมน์ที่อ้างอิงแล้ว		
ขั้นตอนการทำงานปกติ	ผู้กระทำ	ระบบ	
	1. เลือกโปรแกรมที่จะนำออก		
	2. เลือกพาธที่ต้องการบันทึกโปรแกรมที่นำออก		
	3. กดปุ่มนำออก	4. ระบบบันทึกโปรแกรมที่นำออกไปยังพาธที่ระบุ	
เงื่อนไขภายหลัง	ระบบบันทึกไฟล์ข้อมูลของโปรแกรมที่ผ่านการวิเคราะห์ไปยังพาธที่นำออก		

ตารางที่ ง.10 คำบรรยายยูสเคสการวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง

ชื่อยูสเคส	วิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง	ระดับความสำคัญ	มาก
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	วิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงใน โปรแกรมต่างๆในโครงการ โดยเปรียบเทียบตาราง และคอลัมน์จริงในฐานข้อมูล เปรียบเทียบกับตาราง และคอลัมน์ที่โปรแกรมต่างๆอ้างอิง		
เงื่อนไขก่อนหน้า	<ul style="list-style-type: none"> - ต้องมีการทำงานในโครงการอยู่ - โครงการต้องมีฐานข้อมูลอยู่ก่อนแล้ว - โครงการต้องมีโปรแกรมสำหรับนำมาวิเคราะห์อย่างน้อย 1 โปรแกรม 		
ขั้นตอนการทำงานปกติ	ผู้กระทำ	ระบบ	
	1. เลือกฟังก์ชันวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง		
	2. เลือกโปรแกรมที่จะนำมาวิเคราะห์		
	3. กดปุ่มวิเคราะห์	4. ทหารายการตาราง และคอลัมน์ที่โปรแกรมแต่ละโปรแกรมอ้างอิง	
		5. รวมรายการตาราง และคอลัมน์ที่แต่ละโปรแกรมอ้างอิงเข้าด้วยกัน	
		6. ดูเค้าร่างจริงในฐานข้อมูล	
		7. เปรียบเทียบผลที่ได้จากฐานข้อมูลกับรายการที่โปรแกรมอ้างอิง	
		8. แสดงผลลัพธ์การวิเคราะห์	
		9. บันทึกโครงการ	
เงื่อนไขภายหลัง	ระบบบันทึกผลลัพธ์การวิเคราะห์ในโครงการ		

ตารางที่ ง.11 คำบรรยายยวดยุสเคสการหาคำสั่งสืบค้นที่ได้รับผลกระทบ

ชื่อยวดยุสเคส	หาคำสั่งสืบค้นที่ได้รับผลกระทบ	ระดับความสำคัญ	มาก
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	หาคำสั่งสืบค้นที่ได้รับผลกระทบ จากรายการตาราง และคอลัมน์จากที่ผู้กระทำเลือก		
เงื่อนไขก่อนหน้า	- ต้องมีการทำงานในโครงการอยู่ - โครงการต้องผ่านการวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงก่อนแล้ว		
ขั้นตอนการทำงานปกติ	ผู้กระทำ	ระบบ	
	1. เลือกฟังก์ชันหาคำสั่งสืบค้นที่ได้รับผลกระทบ	2. แสดงรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง	
	3. เลือกตาราง และคอลัมน์จากในรายการที่จะลบ		
	4. กดปุ่มหาผลกระทบ	5. ระบบนำคำสั่งสืบค้นของแต่ละโปรแกรมมาหาตาราง และคอลัมน์ที่เกี่ยวข้อง	
		6. เปรียบเทียบตารางและคอลัมน์ที่เกี่ยวข้อง กับรายการที่ผู้กระทำเลือก	
		7. แสดงรายการคำสั่งสืบค้นที่ได้รับผลกระทบ	
เงื่อนไขภายหลัง	รายการคำสั่งสืบค้นที่ได้รับผลกระทบถูกเรียกดูโดยผู้กระทำ		

ตารางที่ ง.12 คำบรรยายยวดยุสเคสการสร้างคำสั่งลบบตาราง และคอลัมน์

ชื่อยวดยุสเคส	สร้างคำสั่งลบบตาราง และคอลัมน์	ระดับความสำคัญ	มาก
ผู้กระทำหลัก	ผู้ใช้งาน		
คำอธิบาย	สร้างคำสั่งภาษาเอสคิวแอลสำหรับลบบตาราง และคอลัมน์		
เงื่อนไขก่อนหน้า	<ul style="list-style-type: none"> - ต้องมีการทำงานในโครงการอยู่ - โครงการต้องผ่านการวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงก่อนแล้ว 		
ขั้นตอนการทำงานปกติ	ผู้กระทำ	ระบบ	
	1. เลือกฟังก์ชันหาคำสั่งสืบค้นที่ได้รับผลกระทบ	2. แสดงรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง	
	3. เลือกตาราง และคอลัมน์จากในรายการที่จะลบ		
	4. กดปุ่มสร้างคำสั่งเอสคิวแอล	5. ระบบสร้างคำสั่งสำหรับลบบตาราง และคอลัมน์	
		6. แสดงรายการคำสั่งเอสคิวแอลที่ใช้ลบบตาราง และคอลัมน์	
เงื่อนไขภายหลัง	รายการคำสั่งถูกผู้กระทำเรียกดู		

ภาคผนวก จ

ผลการทดสอบเครื่องมือ

ผู้วิจัยนำเครื่องมือที่พัฒนาเสร็จแล้วไปทดสอบระดับหน่วย และทดสอบกับโปรแกรมในองค์กรจริงซึ่งมีรายละเอียดกรณีทดสอบต่างๆ และผลของการทดสอบดังต่อไปนี้

ตารางที่ จ.1 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์แบบง่าย

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
จำนวนตารางที่สัมพันธ์กับคลาส test.entity1.Person	1	1
ชื่อตารางที่สัมพันธ์กับคลาส test.entity1.Person	PERSON	PERSON
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ userId	USERID	USERID
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ birthDate	BIRTH	BIRTH
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ name	F_NAME	F_NAME
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ type	TYPE	TYPE
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ age	ไม่มี	ไม่มี
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ dayOfWeek	ไม่มี	ไม่มี
แอตทริบิวต์ userId ไม่ต้องเก็บลงฐานข้อมูล	False	False
แอตทริบิวต์ birthDate ไม่ต้องเก็บลงฐานข้อมูล	False	False
แอตทริบิวต์ name ไม่ต้องเก็บลงฐานข้อมูล	False	False
แอตทริบิวต์ type ไม่ต้องเก็บลงฐานข้อมูล	False	False
แอตทริบิวต์ age ไม่ต้องเก็บลงฐานข้อมูล	True	True
แอตทริบิวต์ dayOfWeek ไม่ต้องเก็บลงฐานข้อมูล	True	True
ชื่อคลาสที่สัมพันธ์กับตาราง PERSON	test.entity1.Person	test.entity1.Person
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ USERID	userId	userId
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ BIRTH	birthDate	birthDate
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ F_NAME	name	name
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ TYPE	type	type

ตารางที่ จ.2 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์
แบบใช้วัตถุแบบฝังตัว

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ชื่อตารางที่สัมพันธ์กับคลาส test.entity2.Customer	CUSTOMER	CUSTOMER
ชื่อตารางที่สัมพันธ์กับคลาส test.entity2.Employee	EMPLOYEE	EMPLOYEE
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ addNo ของ คลาส test.entity2.Address ซึ่งเป็นคลาสที่คลาส อื่นๆเรียกใช้แบบฝังตัว	ADDRESS_NO ของ ตาราง CUTOMER และ ADDRESS_NO ของ ตาราง EMPLOYEE	ADDRESS_NO ของ ตาราง CUTOMER และ ADDRESS_NO ของ ตาราง EMPLOYEE
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ street ของ คลาส test.entity2.Address ซึ่งเป็นคลาสที่คลาส อื่นๆเรียกใช้แบบฝังตัว	STREET ของตาราง CUTOMER และ ROAD ของตาราง EMPLOYEE	STREET ของตาราง CUTOMER และ ROAD ของตาราง EMPLOYEE
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ prefix ของ คลาส test.entity2.ClassId ซึ่งเป็นคลาสที่คลาส อื่นๆเรียกใช้แบบฝังตัวแบบคีย์หลัก	PREFIX ของตาราง EMPLOYEE	PREFIX ของตาราง EMPLOYEE
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ number ของ คลาส test.entity2.ClassId ซึ่งเป็นคลาสที่คลาส อื่นๆเรียกใช้แบบฝังตัวแบบคีย์หลัก	NUMBER ของตาราง EMPLOYEE	NUMBER ของ ตาราง EMPLOYEE
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ NUMBER ของตาราง EMPLOYEE	number ของคลาส test.entity2.ClassId	number ของคลาส test.entity2.ClassId
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ PREFIX ของตาราง EMPLOYEE	prefix ของคลาส test.entity2.ClassId	prefix ของคลาส test.entity2.ClassId
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ ADDRESS_NO ของตาราง EMPLOYEE	addNo ของคลาส test.entity2.Address	addNo ของคลาส test.entity2.Address
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ ROAD ของ ตาราง EMPLOYEE	street ของคลาส test.entity2.Address	street ของคลาส test.entity2.Address

ตารางที่ จ.2 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์
แบบใช้วัตถุแบบฝังตัว (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ ADDRESS_NO ของตาราง CUSTOMER	addNo ของคลาส test.entity2.Address	addNo ของคลาส test.entity2.Address
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ STREET ของตาราง CUSTOMER	street ของคลาส test.entity2.Address	street ของคลาส test.entity2.Address

ตารางที่ จ.3 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์
แบบที่มีความสัมพันธ์ระหว่างคลาส

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ทดสอบความสัมพันธ์แบบ 1:1 ระหว่างคลาส Employee กับ ParkingArea	ตาราง EMPLOYEE มี คอลัมน์ที่เป็นคีย์นอก ชื่อ PARK ซึ่งไปยัง ตาราง PARKINGAREA	ตาราง EMPLOYEE มี คอลัมน์ที่เป็นคีย์นอก ชื่อ PARK ซึ่งไปยัง ตาราง PARKINGAREA
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ parking ของคลาส test.entity3.Employee	PARK ของตาราง EMPLOYEE	PARK ของตาราง EMPLOYEE
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ owner ของคลาส test.entity3. ParkingArea	ไม่มี	ไม่มี
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ PARK ของตาราง EMPLOYEE	parking ของคลาส test.entity3.Employee	parking ของคลาส test.entity3.Employee
ทดสอบความสัมพันธ์แบบ 1:m และ m:1 ระหว่างคลาส Employee และ Department	ตาราง EMPLOYEE มี คอลัมน์ที่เป็นคีย์นอก ชื่อ DEPARTMENT ซึ่งไปยังตาราง DEPARTMENT	ตาราง EMPLOYEE มี คอลัมน์ที่เป็นคีย์นอก ชื่อ DEPARTMENT ซึ่ง ไปยังตาราง DEPARTMENT
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ dept ของ คลาส test.entity3.Employee	DEPARTMENT ของ ตาราง EMPLOYEE	DEPARTMENT ของ ตาราง EMPLOYEE

ตารางที่ จ.3 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์
แบบที่มีความสัมพันธ์ระหว่างคลาส (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ employees ของคลาส test.entity3. Department	ไม่มี	ไม่มี
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ DEPARTMENT ของตาราง EMPLOYEE	dept ของคลาส test.entity3.Employee	dept ของคลาส test.entity3.Employee
ทดสอบความสัมพันธ์แบบ m:n ระหว่างคลาส Employee และ Project	ตาราง EMP_PROJ มี คอลัมน์EMP_ID เป็น คีย์นอกชี้ไปตาราง EMPLOYEE และ คอลัมน์ PROJ_ID เป็นคีย์นอกชี้ไปยัง ตาราง PROJECT	ตาราง EMP_PROJ มี คอลัมน์EMP_ID เป็น คีย์นอกชี้ไปตาราง EMPLOYEE และ คอลัมน์ PROJ_ID เป็น คีย์นอกชี้ไปยังตาราง PROJECT

ตารางที่ จ.4 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์
แบบที่สัมพันธ์มากกว่า 1 ตาราง

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ชื่อตารางที่สัมพันธ์กับคลาส test.entity4.Employee	EMPLOYEE และ EMP_ADDRESS	EMPLOYEE และ EMP_ADDRESS
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ addrNo ซึ่งเป็นแอตทริบิวต์ที่อ้างไปยังตารางที่ 2	ADDRNO ของ ตาราง EMP_ADDRESS	ADDRNO ของ ตาราง EMP_ADDRESS
ชื่อคลาสที่สัมพันธ์กับตาราง EMPLOYEE	test.entity4.Employee	test.entity4.Employee
ชื่อคลาสที่สัมพันธ์กับตาราง EMP_ADDRESS	test.entity4.Employee	test.entity4.Employee
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ EMP_ID ของตาราง EMPLOYEE	id ของคลาส test.entity4.Employee	id ของคลาส test.entity4.Employee
ชื่อแอตทริบิวต์ที่สัมพันธ์กับคอลัมน์ ADDRNO ของตาราง EMP_ADDRESS	addrNo ของคลาส test.entity4.Employee	addrNo ของคลาส test.entity4.Employee

ตารางที่ จ.5 รายการกรณีทดสอบการแปลงระหว่างวัตถุและตารางความสัมพันธ์
แบบความสัมพันธ์สืบทอดคุณสมบัติ

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ชื่อตารางที่สัมพันธ์กับคลาส test.entity4.SalaryEmployee	SALARYEMPLOYEE	SALARYEMPLOYEE
ชื่อตารางที่สัมพันธ์กับคลาส test.entity4.FixEmployee	FIXEMPLOYEE	FIXEMPLOYEE
ชื่อคอลัมน์ในตาราง SALARYEMPLOYEE	EMP_ID ,EMP_USER ,NAME , SARALY	EMP_ID ,EMP_USER ,NAME , SARALY
ชื่อคอลัมน์ในตาราง FIXEMPLOYEE	EMP_ID ,EMP_USER ,NAME , FIX	EMP_ID ,EMP_USER ,NAME , FIX
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ empId ของคลาส test.entity4.Employee ซึ่งเป็นคลาส แม่	EMP_ID ของตาราง SALARYEMPLOYEE และ EMP_ID ของตาราง FIXEMPLOYEE	EMP_ID ของตาราง SALARYEMPLOYEE และ EMP_ID ของตาราง FIXEMPLOYEE
ชื่อคอลัมน์ที่สัมพันธ์กับแอตทริบิวต์ name ของคลาส test.entity4.Employee ซึ่งเป็นคลาส แม่	NAME ของตาราง SALARYEMPLOYEE และ NAME ของตาราง FIXEMPLOYEE	NAME ของตาราง SALARYEMPLOYEE และ NAME ของตาราง FIXEMPLOYEE

ตารางที่ จ.6 รายการกรณีทดสอบการหาคลาสเอนติตี้ และแอตทริบิวต์ของเอนติตี้ที่เข้าไม่ถึง

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ชื่อคลาสเอนติตี้ทั้งหมดในโปรแกรม	entity.Department , entity.Employee , entity.ParkingArea , entity.Project	entity.Department , entity.Employee , entity.ParkingArea , entity.Project
ชื่อคลาสเอนติตี้ที่เข้าไม่ถึง	entity.ParkingArea	entity.ParkingArea

ตารางที่ จ.6 รายการกรณีทดสอบการหาคลาสเอนติตี้ และแอตทริบิวต์ของเอนติตี้ที่เข้าไม่ถึง (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ชื่อแอตทริบิวต์ที่เข้าไม่ถึงของคลาส entity.Project	id	id
ชื่อแอตทริบิวต์ที่เข้าไม่ถึงของคลาส entity.Employee	id , parking , nickname	id , parking , nickname
ชื่อแอตทริบิวต์ที่เข้าไม่ถึงของคลาส entity.ParkingArea	id , location , owner	id , location , owner
ชื่อแอตทริบิวต์ที่เข้าไม่ถึงของคลาส entity.Department	id , employees	id , employees

ตารางที่ จ.7 รายการกรณีทดสอบการหาคำสั่งสืบค้นในโปรแกรม

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
คำสั่งสืบค้นแบบระบุชื่อ	<p>ชื่อ findParkingByOwner คำสั่ง Name คำสั่ง "SELECT p FROM ParkingArea p WHERE p.owner.name LIKE :name"</p> <p>ชื่อ findProjectByName คำสั่ง "SELECT p FROM Project p WHERE p.name LIKE :name"</p> <p>ชื่อ findAllDepartment คำสั่ง "SELECT d FROM Department d"</p> <p>ชื่อ findAllEmployee คำสั่ง "SELECT e FROM Employee e"</p>	<p>ชื่อ findParkingByOwner คำสั่ง Name คำสั่ง "SELECT p FROM ParkingArea p WHERE p.owner.name LIKE :name"</p> <p>ชื่อ findProjectByName คำสั่ง "SELECT p FROM Project p WHERE p.name LIKE :name"</p> <p>ชื่อ findAllDepartment คำสั่ง "SELECT d FROM Department d"</p> <p>ชื่อ findAllEmployee คำสั่ง "SELECT e FROM Employee e"</p>

ตารางที่ จ.7 รายการกรณีทดสอบการหาคำสั่งสืบค้นในโปรแกรม (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
คำสั่งสืบค้นแบบระบุชื่อ (ต่อ)	<p>ชื่อ findEmployeeBy NameAndSurname คำสั่ง “SELECT e FROM Employee e WHERE e.name LIKE :name AND e.surname LIKE :surname”</p> <p>ชื่อ findByLocation คำสั่ง “SELECT p FROM ParkingArea p WHERE p.location LIKE :location”</p> <p>ชื่อ findAllParking คำสั่ง “SELECT p FROM ParkingArea p”</p> <p>ชื่อ findAllProject คำสั่ง “SELECT p FROM Project p”</p> <p>ชื่อ findDepartmentByName คำสั่ง “SELECT d FROM Department d WHERE d.name LIKE :name”</p> <p>ชื่อ findEmployeeByName คำสั่ง “SELECT e FROM Employee e WHERE e.name LIKE :name”</p>	<p>ชื่อ findEmployeeBy NameAndSurname คำสั่ง “SELECT e FROM Employee e WHERE e.name LIKE :name AND e.surname LIKE :surname”</p> <p>ชื่อ findByLocation คำสั่ง “SELECT p FROM ParkingArea p WHERE p.location LIKE :location”</p> <p>ชื่อ findAllParking คำสั่ง “SELECT p FROM ParkingArea p”</p> <p>ชื่อ findAllProject คำสั่ง “SELECT p FROM Project p”</p> <p>ชื่อ findDepartmentByName คำสั่ง “SELECT d FROM Department d WHERE d.name LIKE :name”</p> <p>ชื่อ findEmployeeByName คำสั่ง “SELECT e FROM Employee e WHERE e.name LIKE :name”</p>

ตารางที่ จ.7 รายการกรณีทดสอบการหาคำสั่งสืบค้นในโปรแกรม (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
คำสั่งสืบค้นภาษาเจพีคิวแอล	<pre>"SELECT e FROM Employee e WHERE e.nickname LIKE :nickname" "SELECT e FROM Employee e WHERE e.dept.name LIKE :name"</pre>	<pre>"SELECT e FROM Employee e WHERE e.nickname LIKE :nickname" "SELECT e FROM Employee e WHERE e.dept.name LIKE :name"</pre>
คำสั่งสืบค้นภาษาเอสคิวแอล	<pre>"SELECT * FROM EMPLOYEE AS e WHERE e.ID IN (SELECT EMP_ID FROM EMP_PROJ WHERE PROJ_ID IN (SELECT ID FROM PROJECT WHERE NAME = :projectName))" "SELECT * FROM PROJECT AS p WHERE p.ID IN (SELECT PROJ_ID FROM EMP_PROJ WHERE EMP_ID IN (SELECT ID FROM EMPLOYEE WHERE NAME LIKE : empName))"</pre>	<pre>"SELECT * FROM EMPLOYEE AS e WHERE e.ID IN (SELECT EMP_ID FROM EMP_PROJ WHERE PROJ_ID IN (SELECT ID FROM PROJECT WHERE NAME = :projectName))" "SELECT * FROM PROJECT AS p WHERE p.ID IN (SELECT PROJ_ID FROM EMP_PROJ WHERE EMP_ID IN (SELECT ID FROM EMPLOYEE WHERE NAME LIKE : empName))"</pre>

ตารางที่ จ.8 รายการกรณีทดสอบการหาตาราง และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบค้น

ภาษาเจพีคิวแอล

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
คำสั่ง "SELECT p FROM Party p"	ตาราง Party	ตาราง Party
คำสั่ง "SELECT p.name, p.surname FROM Party p"	ตาราง Party คอลัมน์ name , surname	ตาราง Party คอลัมน์ name , surname
คำสั่ง "SELECT p.name,p.ref.no FROM Party p"	ตาราง Party คอลัมน์ name , ref ตาราง Reference คอลัมน์ no	ตาราง Party คอลัมน์ name , ref ตาราง Reference คอลัมน์ no
คำสั่ง "SELECT p.name,r.no FROM Party p JOIN p.ref r JOIN p.ref.country c WHERE p.name=?1 order by r.country.country"	ตาราง Country คอลัมน์ country ตาราง Party คอลัมน์ name , ref ตาราง Reference คอลัมน์ no	ตาราง Country คอลัมน์ country ตาราง Party คอลัมน์ name , ref ตาราง Reference คอลัมน์ no
คำสั่ง "SELECT p FROM Party p Order by p.id"	ตาราง Party คอลัมน์ id	ตาราง Party คอลัมน์ id

ตารางที่ จ.9 รายการกรณีทดสอบการหาตาราง และคอลัมน์ที่สัมพันธ์กับคำสั่งสืบค้น

ภาษาเอสคิวแอล

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
คำสั่ง "SELECT * FROM Employee"	ตาราง Employee	ตาราง Employee
คำสั่ง "SELECT e.name,e.lastname FROM Employee AS e"	ตาราง Employee คอลัมน์ name , lastname	ตาราง Employee คอลัมน์ name , lastname
คำสั่ง "SELECT EMPLOYEE.*, d.name FROM Employee , Department AS d"	ตาราง Employee ตาราง Department คอลัมน์ name	ตาราง Employee ตาราง Department คอลัมน์ name
คำสั่ง "SELECT e.name, d.name FROM Employee e, Department AS d inner join Address a on a.no = e.address"	ตาราง Employee คอลัมน์ name , address ตาราง Department คอลัมน์ name ตาราง Address คอลัมน์ no	ตาราง Employee คอลัมน์ name , address ตาราง Department คอลัมน์ name ตาราง Address คอลัมน์ no
คำสั่ง "SELECT name FROM Employee Order by id"	ตาราง Employee คอลัมน์ name , id	ตาราง Employee คอลัมน์ name , id
คำสั่ง "SELECT e.name,a.no,d.location FROM Employee e,Address a,(SELECT id,name FROM Department) d"	ตาราง Employee คอลัมน์ name ตาราง Address คอลัมน์ no ตาราง Department คอลัมน์ id , name , location	ตาราง Employee คอลัมน์ name ตาราง Address คอลัมน์ no ตาราง Department คอลัมน์ id , name , location

ตารางที่ จ.10 รายการกรณีทดสอบการรวมผลลัพธ์รายการตาราง 2 รายการ

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
รายการตารางที่ 1 - ตาราง Table_A คอลัมน์ A1, A2, A3 - ตาราง Table_B คอลัมน์ B1, B2 รายการตารางที่ 2 - ตาราง Table_C คอลัมน์ C1 - ตาราง Table_D คอลัมน์ D1, D2, D3, D4	ตาราง Table_A คอลัมน์ A1, A2, A3 ตาราง Table_B คอลัมน์ B1, B2 ตาราง Table_C คอลัมน์ C1 ตาราง Table_D คอลัมน์ D1, D2, D3, D4	ตาราง Table_A คอลัมน์ A1, A2, A3 ตาราง Table_B คอลัมน์ B1, B2 ตาราง Table_C คอลัมน์ C1 ตาราง Table_D คอลัมน์ D1, D2, D3, D4
รายการตารางที่ 1 - ตาราง Table_A คอลัมน์ A1, A2, A3 - ตาราง Table_B คอลัมน์ B1, B2 - ตาราง Table_C คอลัมน์ C1 รายการตารางที่ 2 - ตาราง Table_C คอลัมน์ C1 - ตาราง Table_D คอลัมน์ D1, D2, D3, D4, D5	ตาราง Table_A คอลัมน์ A1, A2, A3 ตาราง Table_B คอลัมน์ B1, B2 ตาราง Table_C คอลัมน์ C1 ตาราง Table_D คอลัมน์ D1, D2, D3, D4, D5	ตาราง Table_A คอลัมน์ A1, A2, A3 ตาราง Table_B คอลัมน์ B1, B2 ตาราง Table_C คอลัมน์ C1 ตาราง Table_D คอลัมน์ D1, D2, D3, D4, D5
รายการตารางที่ 1 - ตาราง Table_A คอลัมน์ A1, A2, A3 - ตาราง Table_B คอลัมน์ B1, B2 ไม่มีรายการตารางที่ 2	ตาราง Table_A คอลัมน์ A1, A2, A3 ตาราง Table_B คอลัมน์ B1, B2	ตาราง Table_A คอลัมน์ A1, A2, A3 ตาราง Table_B คอลัมน์ B1, B2

ตารางที่ จ.10 รายการกรณีทดสอบการรวมผลลัพธ์รายการตาราง 2 รายการ (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
ไม่มีรายการตารางที่ 1 รายการตารางที่ 2 - ตาราง Table_A คอลลัมน์ A1, A2, A3 - ตาราง Table_B คอลลัมน์ B1, B2	ตาราง Table_A คอลลัมน์ A1, A2, A3 ตาราง Table_B คอลลัมน์ B1, B2	ตาราง Table_A คอลลัมน์ A1, A2, A3 ตาราง Table_B คอลลัมน์ B1, B2

ตารางที่ จ.11 รายการกรณีทดสอบการหารายการตารางที่ไม่ถูกอ้างอิง

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
รายการตารางที่ 1 - ตาราง Table_A คอลลัมน์ A1[PK], A2, A3 - ตาราง Table_B คอลลัมน์ B1[PK], B2 - ตาราง Table_C คอลลัมน์ C1[PK] - ตาราง Table_D คอลลัมน์ D1[PK], D2, D3, D4 รายการตารางที่ 2 - ตาราง Table_C คอลลัมน์ C1[PK] - ตาราง Table_D คอลลัมน์ D1[PK], D2, D3, D4	ตาราง Table_A และ ตาราง Table_B ทั้งตาราง	ตาราง Table_A และ ตาราง Table_B ทั้งตาราง

ตารางที่ จ.11 รายการกรณีทดสอบการหารายการตารางที่ไม่ถูกอ้างอิง (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
<p>รายการตารางที่ 1</p> <ul style="list-style-type: none"> - ตาราง Table_A คอลลัมน์ A1[PK], A2, A3 - ตาราง Table_B คอลลัมน์ B1[PK], B2 - ตาราง Table_C คอลลัมน์ C1[PK] - ตาราง Table_D คอลลัมน์ D1[PK], D2, D3, D4 <p>รายการตารางที่ 2</p> <ul style="list-style-type: none"> - ตาราง Table_C คอลลัมน์ C1[PK] - ตาราง Table_D คอลลัมน์ D1, D2, D4 	<p>ตาราง Table_A และ ตาราง Table_B ทั้งตาราง</p> <p>คอลลัมน์ D3 ในตาราง Table_D</p>	<p>ตาราง Table_A และ ตาราง Table_B ทั้งตาราง</p> <p>คอลลัมน์ D3 ในตาราง Table_D</p>
<p>รายการตารางที่ 1</p> <ul style="list-style-type: none"> - ตาราง Table_A คอลลัมน์ A1[PK], A2, A3 - ตาราง Table_B คอลลัมน์ B1[PK], B2 - ตาราง Table_C คอลลัมน์ C1[PK] - ตาราง Table_D คอลลัมน์ D1[PK], D2, D3, D4 <p>รายการตารางที่ 2</p> <ul style="list-style-type: none"> - ตาราง Table_D คอลลัมน์ D4 	<p>ตาราง Table_A , ตาราง Table_B และตาราง Table_C ทั้งตาราง</p> <p>คอลลัมน์ D2 และ D3 ในตาราง Table_D</p>	<p>ตาราง Table_A , ตาราง Table_B และตาราง Table_C ทั้งตาราง</p> <p>คอลลัมน์ D2 และ D3 ในตาราง Table_D</p>

ตารางที่ จ.11 รายการกรณีทดสอบการหารายการตารางที่ไม่ถูกอ้างอิง (ต่อ)

กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ได้จริง
<p>รายการตารางที่ 1</p> <ul style="list-style-type: none"> - ตาราง Table_A คอลลัมน์ A1[PK], A2, A3 - ตาราง Table_B คอลลัมน์ B1[PK], B2 - ตาราง Table_C คอลลัมน์ C1[PK] - ตาราง Table_D คอลลัมน์ D1[PK], D2, D3, D4 <p>รายการตารางที่ 2</p> <ul style="list-style-type: none"> - ตาราง Table_A คอลลัมน์ A1[PK], A2, A3 - ตาราง Table_B คอลลัมน์ B1[PK], B2 - ตาราง Table_C คอลลัมน์ C1[PK] - ตาราง Table_D คอลลัมน์ D1[PK], D2, D3, D4 	ไม่มี	ไม่มี

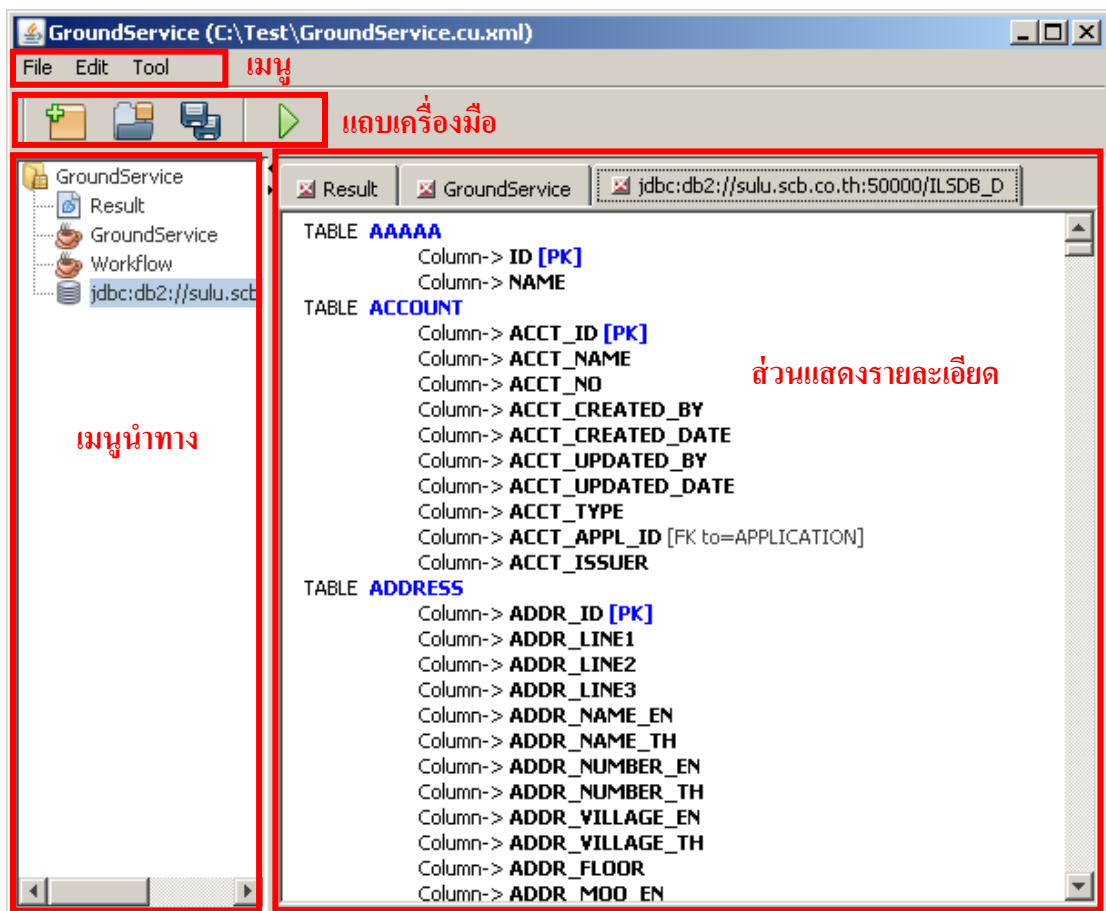
ภาคผนวก ฉ

คู่มือการใช้งานเครื่องมือระบุตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม

การทำงานของเครื่องมือต้นแบบสำหรับระบุตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม แสดงเป็นตัวอย่างหน้าจอการทำงานตามฟังก์ชันงาน ซึ่งมีรายละเอียดดังนี้

1. ส่วนประกอบในหน้าจอของเครื่องมือ

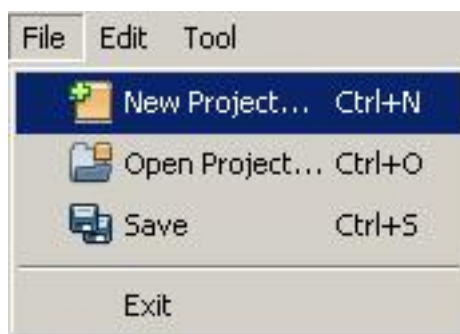
หน้าจอของเครื่องมือถูกออกแบบให้มีลักษณะคล้ายกับโปรแกรมจัดการข้อมูล (Editor) ทั่วไป โดยจะมีเมนูต่างๆทางด้านบนของเครื่องมือ ถัดลงมาจะเป็นแถบเครื่องมือที่รวบรวมฟังก์ชันที่ใช้งานบ่อย ทางซ้ายมือจะเป็นเมนูนำทางไปยังข้อมูลต่างๆในโครงการ และทางขวามือจะเป็นส่วนสำหรับแสดงรายละเอียดของไฟล์ข้อมูลในโครงการ ดังแสดงในภาพที่ ฉ.1



ภาพที่ ฉ.1 ส่วนประกอบต่างๆของเครื่องมือ

2. การสร้างโครงการใหม่ให้แก่เครื่องมือ

ก่อนใช้งานเครื่องมือ จำเป็นต้องสร้างโครงการให้แก่เครื่องมือเสียก่อน เพื่อตั้งค่าให้เครื่องมือมีสภาพแวดล้อมที่พร้อมจะทำงาน โดยสามารถสร้างโครงการได้โดยเลือกจากเมนู “File” แล้วเลือก “New Project...” ดังภาพที่ น.2 (ก) หรือจากแถบเครื่องมือ ดังภาพที่ น.2 (ข) จากนั้นเครื่องมือจะแสดงหน้าจอให้ผู้ใช้งานระบุค่าต่างๆของโครงการ ได้แก่ชื่อโครงการ และโฟลเดอร์ที่ต้องการจัดเก็บไฟล์โครงการ ซึ่งโปรแกรมจะแสดงผลพารของไฟล์โครงการเองอัตโนมัติ ดังแสดงในภาพที่ น.3 จากบนลงล่าง

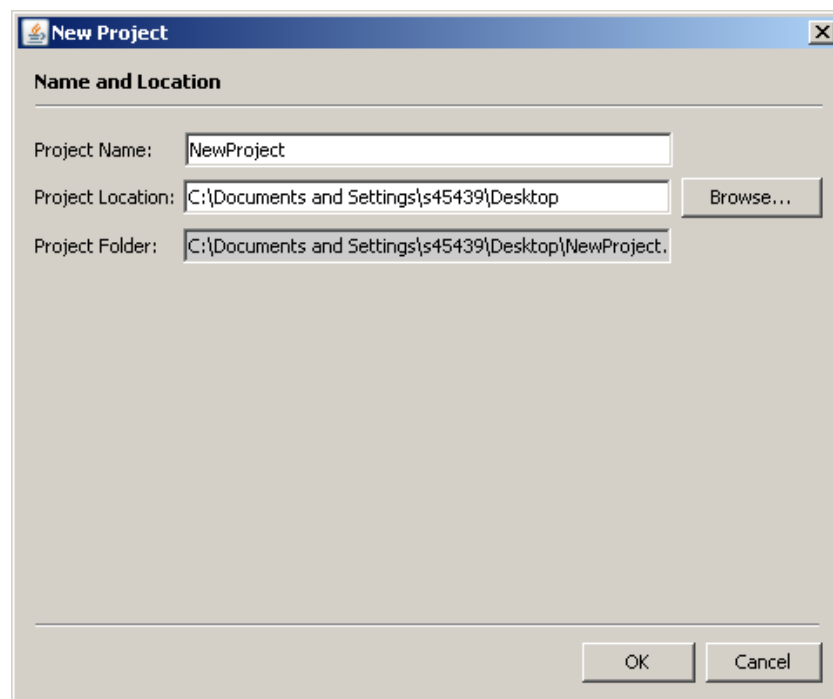


(ก)



(ข)

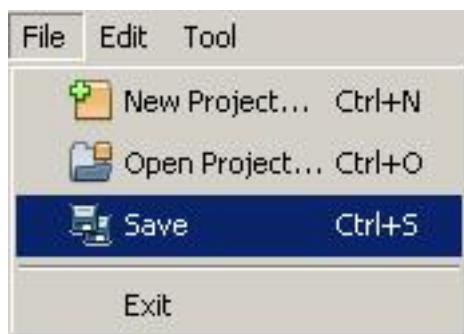
ภาพที่ น.2 วิธีการเรียกใช้ฟังก์ชันสร้างโครงการ



ภาพที่ น.3 หน้าจอรายละเอียดข้อมูลโครงการที่จะสร้าง

3. บันทึกไฟล์โครงการ

ในกรณีที่ผู้ใช้ต้องการบันทึกการเปลี่ยนแปลงของโครงการ สามารถทำได้โดยเรียกใช้ฟังก์ชันบันทึกโครงการ ซึ่งสามารถเรียกใช้ได้จากเมนู “File” แล้วเลือก “Save” ดังภาพที่ น.4 (ก) หรือจากแถบเครื่องมือ ดังแสดงในภาพที่ น.4 (ข)



(ก)

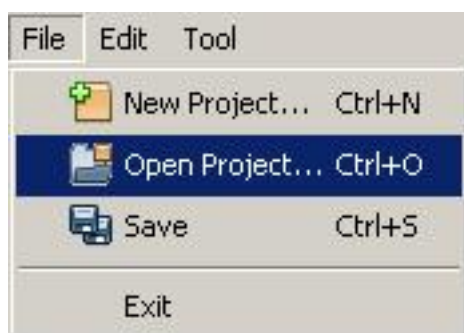


(ข)

ภาพที่ น.4 วิธีการเรียกใช้ฟังก์ชันบันทึกโครงการ

4. เปิดไฟล์โครงการ

หากผู้ใช้มีไฟล์โครงการที่เคยทำงานอยู่ก่อนแล้ว สามารถเปิดไฟล์โครงการเก่าขึ้นมาใหม่ได้ โดยเรียกใช้ฟังก์ชันเปิดไฟล์โครงการ โดยสามารถเรียกได้จากเมนู “File” แล้วเลือก “Open Project...” ดังภาพที่ น.5 (ก) หรือจากแถบเครื่องมือ ดังภาพที่ น.5 (ข) จากนั้นเครื่องมือจะแสดงหน้าจอให้ผู้ใช้งานเลือกไฟล์โครงการ เมื่อผู้ใช้เลือกไฟล์โครงการแล้ว ระบบจะโหลดไฟล์โครงการเพื่อเตรียมสภาพแวดล้อมให้ผู้ใช้งานใช้งานเครื่องมือ



(ก)



(ข)

ภาพที่ น.5 วิธีการเรียกใช้ฟังก์ชันเปิดโครงการ

5. เพิ่มโปรแกรมเพื่อวิเคราะห์

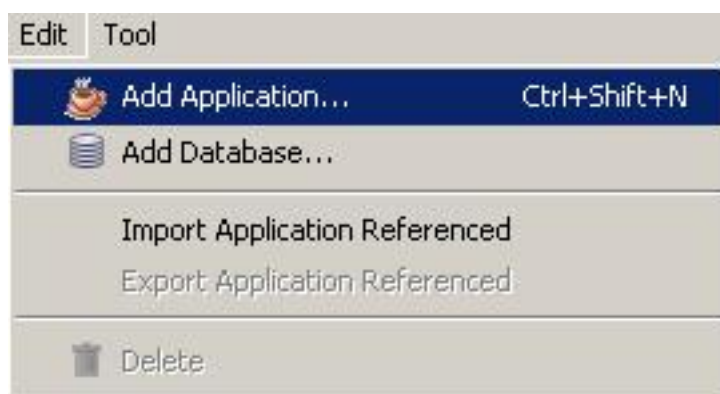
ผู้ใช้งานสามารถนำเข้าโปรแกรมต่างๆ สู่โครงการ เพื่อให้เครื่องมือวิเคราะห์ ได้มากกว่า 1 โปรแกรม โดยผู้ใช้งานจะต้องเลือกที่เมนู “Edit” แล้วเลือก “Add Application...” ดังแสดงในภาพที่ ๖.6 จากนั้นเครื่องมือจะแสดงหน้าจอให้ผู้ใส่ระบุข้อมูลของโปรแกรมที่จะนำเข้า ดังแสดงในภาพที่ ๖.7 โดยมีรายละเอียดข้อมูลนำเข้าดังนี้

1. Application Name คือชื่อของโปรแกรมซึ่งผู้ใช้งานจะกำหนดเป็นชื่ออะไรก็ได้ แต่ในโครงการเดียวกันห้ามมีชื่อโปรแกรมซ้ำกัน

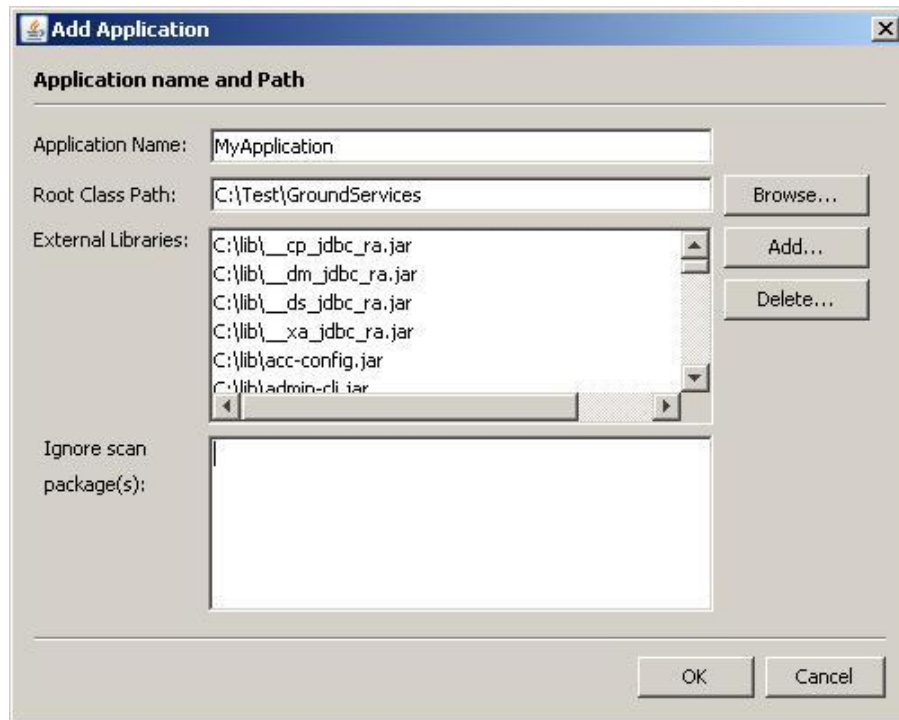
2. Root Class Path คือพาธของโปรแกรมที่ผ่านการแปลโปรแกรมแล้ว (Compile) โดยผู้ใช้งานจะต้องเลือกไปยังโฟลเดอร์นอกสุดที่เก็บไฟล์คลาสต่างๆไว้

3. External Libraries คือคลังคำสั่งอื่นๆที่โปรแกรมที่จะนำเข้าอ้างอิงถึง ซึ่งเป็นคลังคำสั่งนอกเหนือจากคลังคำสั่งมาตรฐานของจาวา โดยในการวิเคราะห์เครื่องมือจะต้องอ้างอิงไปยังคลังคำสั่งเหล่านี้ด้วย

4. Ignore Scan package(s) คือรายชื่อแพ็คเกจของโปรแกรม ที่ไม่ต้องการให้โปรแกรมวิเคราะห์



ภาพที่ ๖.6 วิธีการเรียกใช้ฟังก์ชันเพิ่มโปรแกรมเพื่อวิเคราะห์

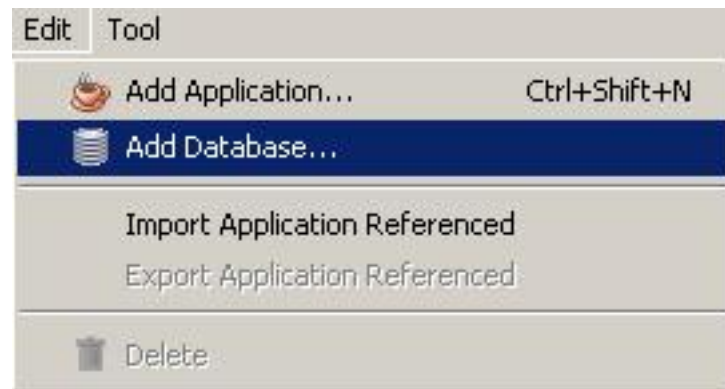


ภาพที่ ๓.7 หน้าจอรายละเอียดข้อมูล โปรแกรมที่จะเพิ่มเข้าสู่โครงการ

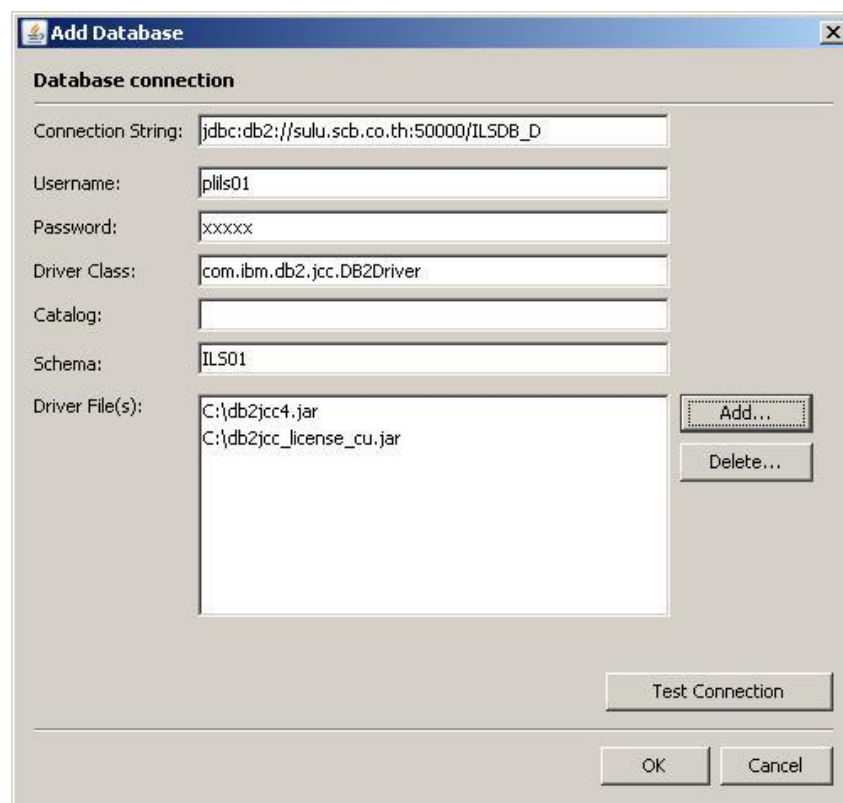
6. เพิ่มฐานข้อมูลเพื่อวิเคราะห์

ผู้ใช้งานสามารถนำเข้าฐานข้อมูลสู่โครงการ เพื่อให้เครื่องมือวิเคราะห์ โดยเลือกที่เมนู “Edit” แล้วเลือก “Add Database...” ดังแสดงในภาพที่ ๓.8 จากนั้นเครื่องมือจะแสดงหน้าจอให้ผู้ใส่ระบุข้อมูลของฐานข้อมูลที่จะนำเข้า ดังแสดงในภาพที่ ๓.9 โดยมีรายละเอียดข้อมูลนำเข้าดังนี้

1. Connection String คือข้อความสำหรับระบุการเชื่อมต่อไปยังฐานข้อมูล โดยแต่ละฐานข้อมูลจะมีข้อความการเชื่อมต่อที่ไม่เหมือนกัน
2. Username คือชื่อผู้ใช้งานของฐานข้อมูล
3. Password คือรหัสผ่านของชื่อผู้ใช้งานของฐานข้อมูล
4. Driver Class คือชื่อคลาสที่เป็นไดรฟ์เวอร์ของฐานข้อมูล
5. Catalog คือชื่อแคตตาล็อก ของฐานข้อมูลที่ต้องการให้เครื่องมืออ่านรายการตาราง และคอลัมน์ สำหรับฐานข้อมูลที่มีการใช้งานแคตตาล็อก
6. Schema คือชื่อเค้าร่าง ของฐานข้อมูลที่ต้องการให้เครื่องมืออ่านรายการตาราง และคอลัมน์ สำหรับฐานข้อมูลที่มีการใช้งานเค้าร่าง
7. Driver File(s) คือรายการพาธของไฟล์ไดรฟ์เวอร์สำหรับเชื่อมต่อฐานข้อมูล



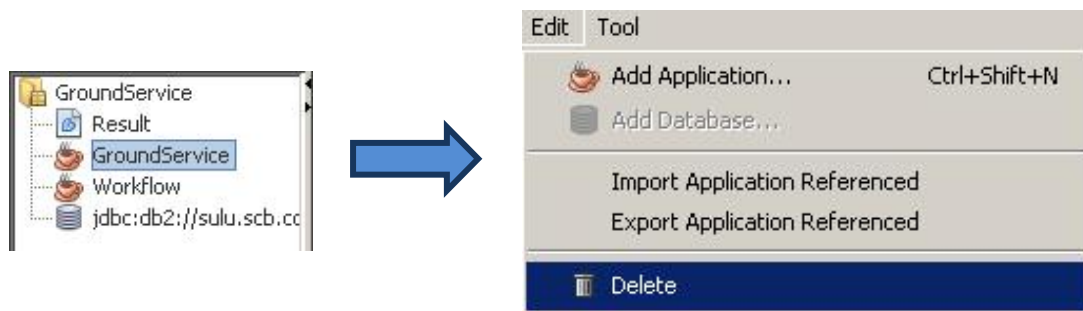
ภาพที่ ๘.8 วิธีการเรียกใช้ฟังก์ชันเพิ่มฐานข้อมูลเพื่อวิเคราะห์



ภาพที่ ๘.9 หน้าจอรายละเอียดข้อมูลฐานข้อมูลที่จะเพิ่มเข้าสู่โครงการ

7. ลบโปรแกรม หรือฐานข้อมูลออกจากโครงการ

ถ้าผู้ใช้งานไม่ต้องการอ้างอิงโปรแกรม หรือฐานข้อมูลจากโครงการแล้ว สามารถลบโปรแกรม หรือฐานข้อมูลออกจากโครงการได้โดย เลือกที่โปรแกรม หรือฐานข้อมูลจากเมนูนำทาง แล้วเลือกเมนู “Edit” จากนั้นเลือกที่ “Delete” เครื่องมือจะลบโปรแกรม หรือฐานข้อมูลที่เลือกออกจากโครงการ ดังแสดงในภาพที่ ๘.10

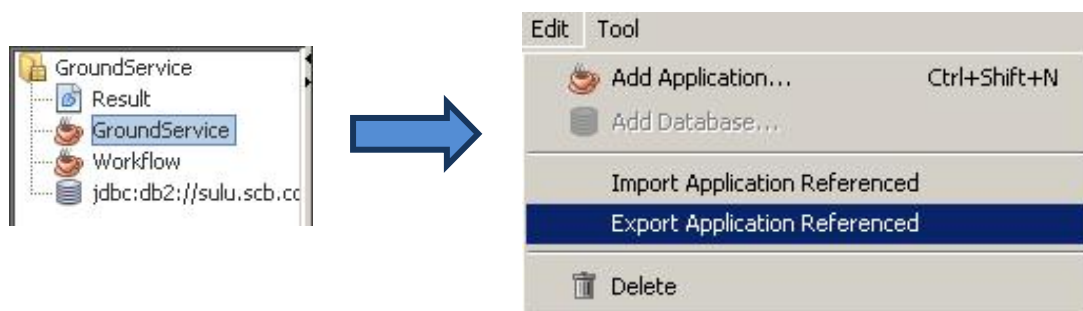


ภาพที่ ฉ.10 วิธีการเรียกใช้ฟังก์ชันลบโปรแกรม หรือฐานข้อมูล

8. นำออกโปรแกรมที่วิเคราะห์แล้ว

หลังจากเครื่องมือได้วิเคราะห์เพื่อระบุตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรมเสร็จเรียบร้อยแล้ว โปรแกรมต่างๆ จะถูกผ่านการวิเคราะห์หามาแล้ว ซึ่งจะทราบว่าแต่ละโปรแกรมมีการอ้างอิงตาราง และคอลัมน์ฐานข้อมูลใดบ้าง อีกทั้งยังทราบว่าในแต่ละโปรแกรมมีคำสั่งสืบค้นใดบ้าง และแต่ละคำสั่งสัมพันธ์กับตาราง และคอลัมน์ฐานข้อมูลใด

ถ้าโปรแกรมที่วิเคราะห์แล้ว ไม่มีการเปลี่ยนแปลงซอร์สโค้ด ผลลัพธ์ของการวิเคราะห์ก็ย่อมได้ผลลัพธ์ที่เหมือนเดิม เครื่องมือนี้จึงถูกออกแบบให้สามารถนำออกข้อมูลของโปรแกรมที่วิเคราะห์แล้ว เพื่อนำไปใช้ในโครงการอื่นๆต่อไป โดยเครื่องมือไม่ต้องเสียเวลาสำหรับวิเคราะห์โปรแกรมใหม่ โดยผู้ใช้งานสามารถนำออกโปรแกรมที่วิเคราะห์แล้วโดย เลือกที่ชื่อโปรแกรมที่ต้องการจะนำออกจากเมนูนำทาง จากนั้นเลือกที่เมนู “Edit” แล้วจึงเลือก “Export Application Referenced...” โดยเครื่องมือจะแสดงหน้าจอให้ผู้ใช้งานเลือกพารสำหรับนำออกผลลัพธ์ ดังแสดงในภาพที่ ฉ.11

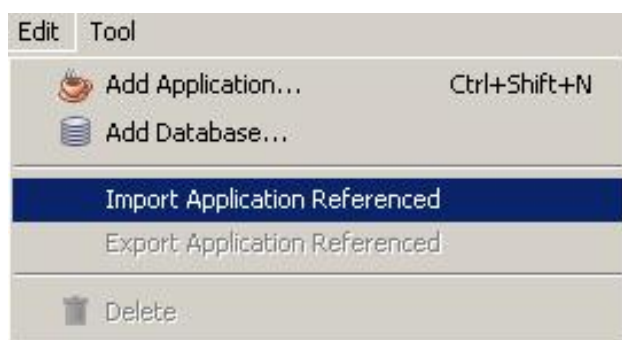


ภาพที่ ฉ.11 วิธีการเรียกใช้ฟังก์ชันนำออกโปรแกรมที่วิเคราะห์แล้ว

9. นำเข้าโปรแกรมที่วิเคราะห์แล้ว

ผู้ใช้งานสามารถนำเข้าโปรแกรมที่ผ่านการวิเคราะห์แล้วเข้าสู่โครงการได้โดยเลือกที่เมนู “Edit” แล้วจึงเลือก “Import Application Referenced...” โดยเครื่องมือจะแสดงหน้าจอให้ผู้เลือกไฟล์โปรแกรมที่จะนำเข้า ดังแสดงในภาพที่ น.12

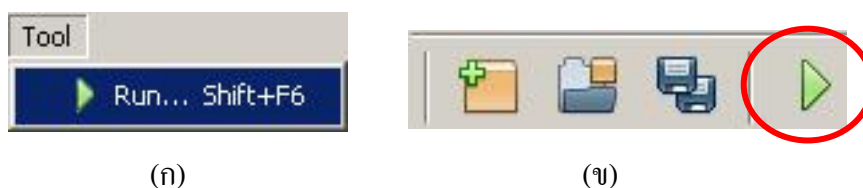
การนำเข้าโปรแกรมที่ผ่านการวิเคราะห์แล้ว มีข้อดีคือเครื่องมือไม่ต้องเสียเวลาวิเคราะห์โปรแกรมดังกล่าวใหม่ เนื่องจากไฟล์ที่นำเข้ามีผลการวิเคราะห์ทุกอย่างเรียบร้อยแล้ว



ภาพที่ น.12 วิธีการเรียกใช้ฟังก์ชันนำเข้าโปรแกรมที่วิเคราะห์แล้ว

10. วิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง

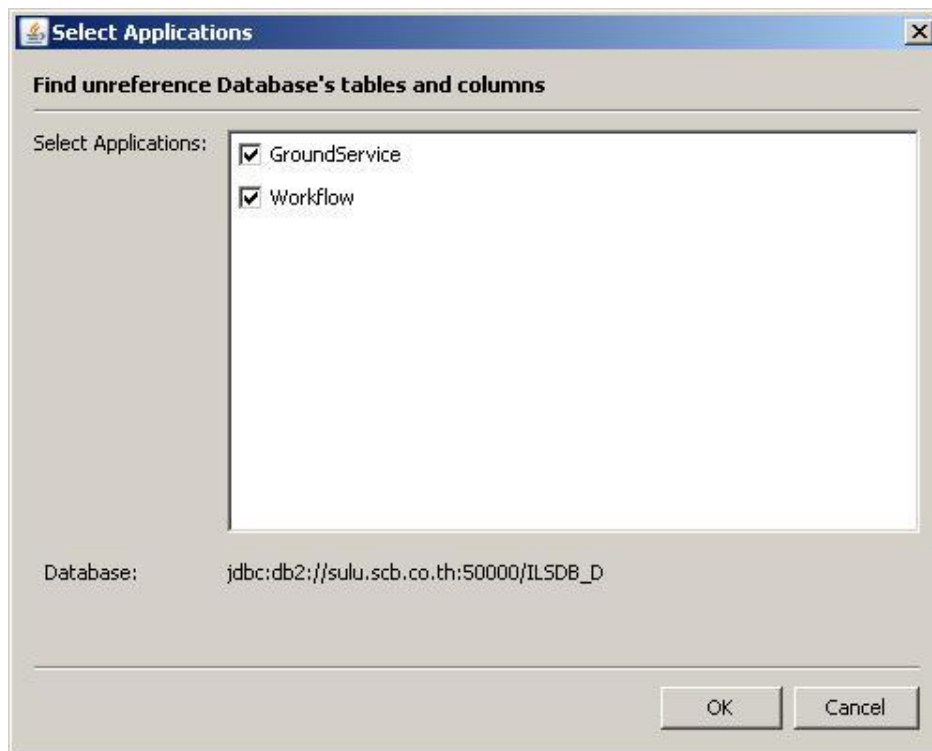
เมื่อผู้ใช้งานเพิ่มโปรแกรม และฐานข้อมูลเข้าสู่โครงการเรียบร้อยแล้ว ผู้ใช้งานสามารถใช้ฟังก์ชันวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง ได้จากเมนู “Tool” แล้วเลือก “Run...” ดังภาพที่ น.13 (ก) หรือเรียกใช้ได้จากแถบเครื่องมือ ดังแสดงในภาพที่ น.13 (ข) จากนั้นเครื่องมือจะแสดงหน้าจอให้ผู้เลือกรายการโปรแกรมที่จะนำไปวิเคราะห์โดยเครื่องมือจะแสดงรายการโปรแกรมทั้งหมดในไฟล์โครงการ แต่ผู้ใช้สามารถเลือกที่จะวิเคราะห์โปรแกรมแค่บางโปรแกรมได้ ดังแสดงภาพหน้าจอในภาพที่ น.14 จากนั้นกดปุ่ม “OK” เพื่อให้เครื่องมือเริ่มวิเคราะห์ ซึ่งขั้นตอนนี้จะใช้เวลาค่อนข้างนาน ขึ้นกับขนาดของโปรแกรมต่างๆที่นำมาวิเคราะห์ โดยระหว่างการวิเคราะห์เครื่องมือจะแสดงความคืบหน้าว่าวิเคราะห์ถึงขั้นตอนใดแล้ว ดังแสดงในภาพที่ น.15



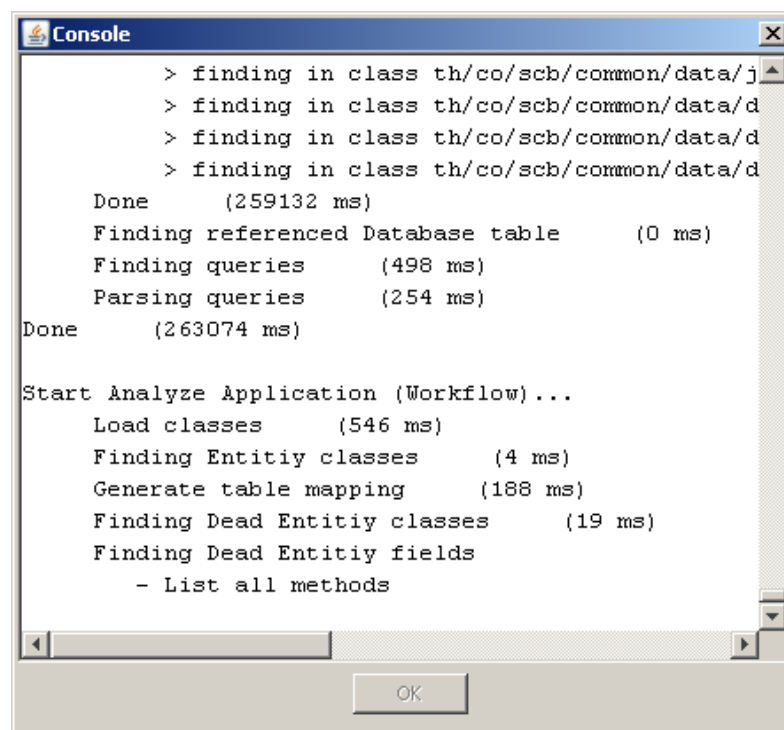
(ก)

(ข)

ภาพที่ น.13 วิธีการเรียกใช้ฟังก์ชันวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิง



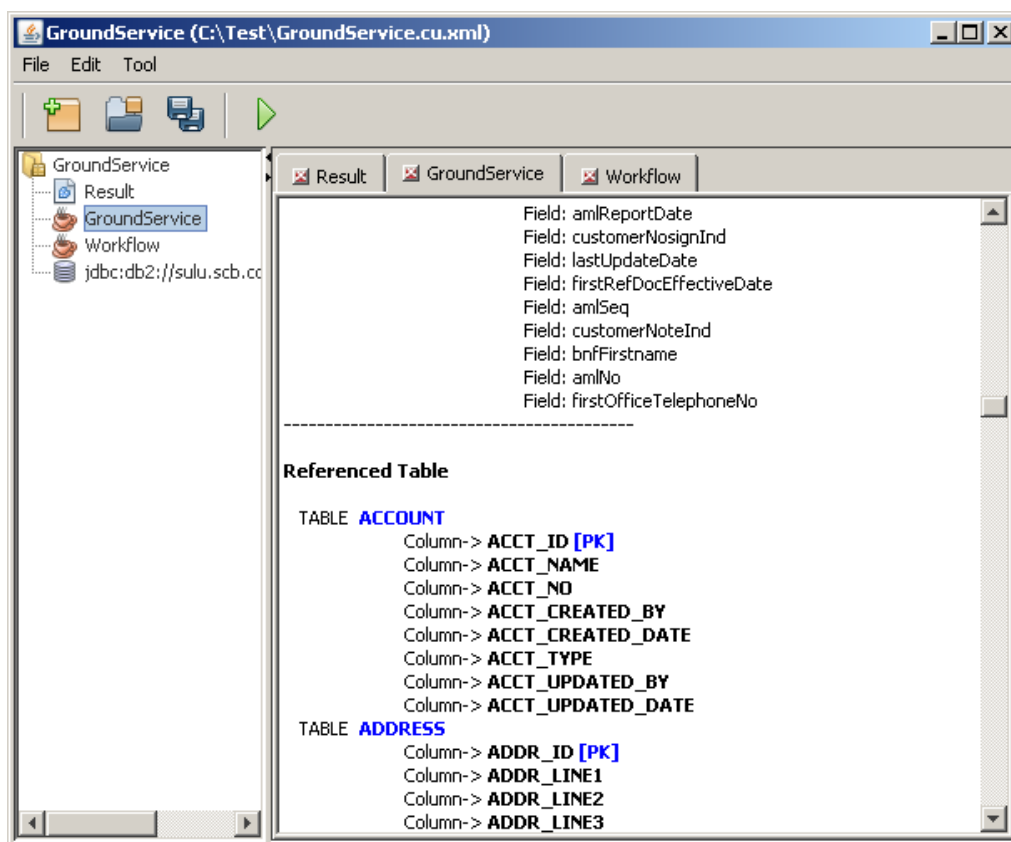
ภาพที่ ๑.14 หน้าจอแสดงรายการโปรแกรมให้ผู้ใช้เลือกก่อนทำการวิเคราะห์



ภาพที่ ๑.15 หน้าจอแสดงความคืบหน้าในระหว่างการวิเคราะห์

11. รายละเอียดผลการวิเคราะห์โปรแกรม

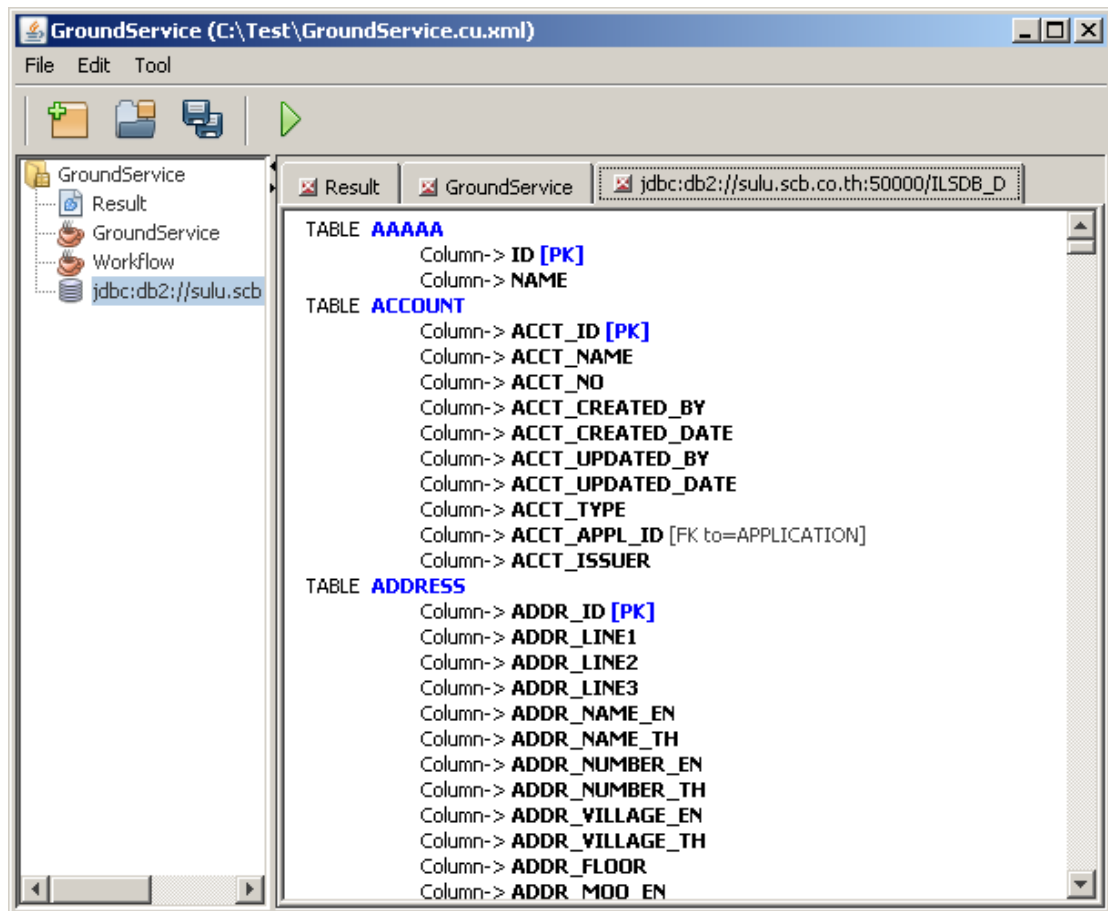
หลังจากวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงเสร็จเรียบร้อยแล้ว ผู้ใช้งานสามารถเรียกดูรายละเอียดของแต่ละโปรแกรมที่มีการอ้างอิงฐานข้อมูลอย่างไร โดยดับเบิลคลิกที่โปรแกรมจากเมนูนำทาง แล้วรายละเอียดจะปรากฏเป็นแถบใหม่ทางด้านขวาของเครื่องมือ ดังแสดงในภาพที่ น.16 โดยมีรายละเอียดที่แสดงผล คือรายการโค้ดที่เข้าไม่ถึง และรายการตาราง คอลัมน์ที่โปรแกรมอ้างอิง



ภาพที่ น.16 หน้าจอแสดงรายละเอียดผลลัพธ์ที่ได้จากการวิเคราะห์โปรแกรม

12. รายละเอียดรายการตาราง และคอลัมน์ในฐานข้อมูล

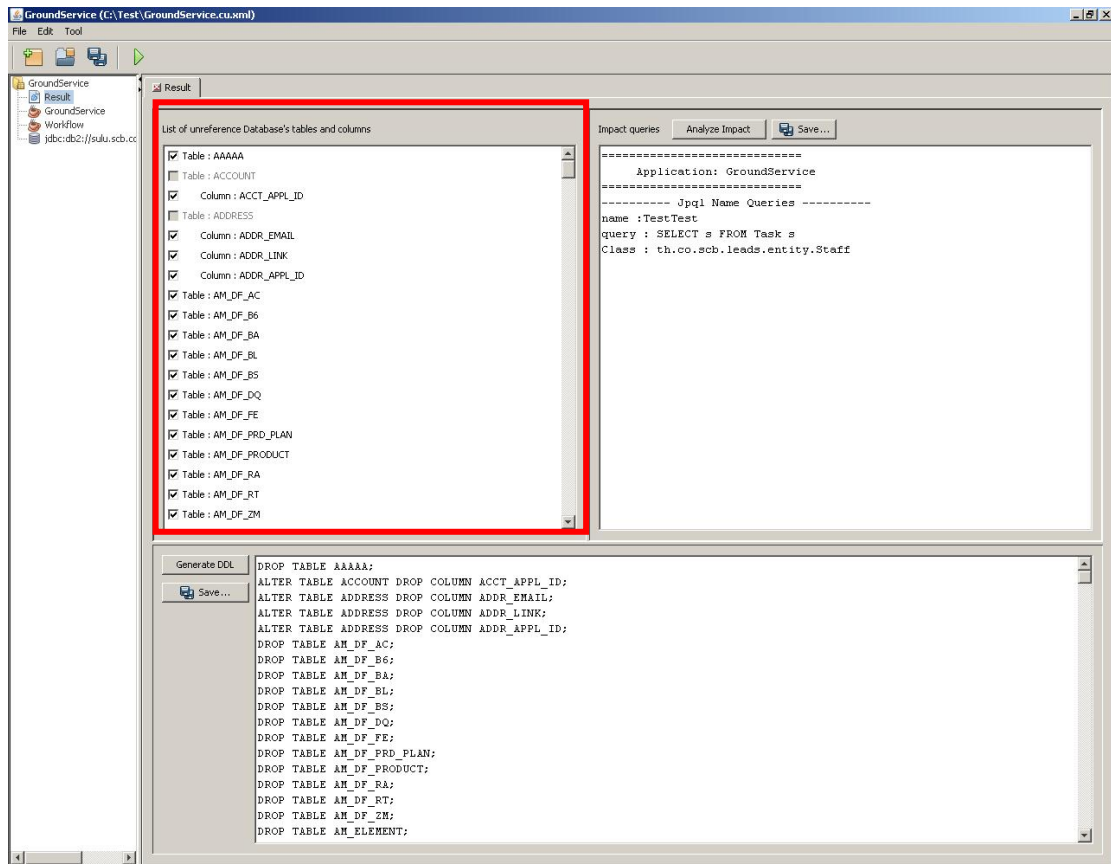
หลังจากวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงเสร็จเรียบร้อยแล้ว แสดงว่าเครื่องมือจะต้องดูแลร่างของฐานข้อมูลเพื่อใช้วิเคราะห์ผลลัพธ์แล้ว คือเครื่องมือจะทราบรายการตาราง และคอลัมน์ที่มีในฐานข้อมูล ซึ่งผู้ใช้งานสามารถดูรายการตาราง และคอลัมน์ของฐานข้อมูล ได้โดยการดับเบิลคลิกที่ฐานข้อมูลจากเมนูนำทาง ซึ่งรายการตาราง และคอลัมน์ที่อยู่ในฐานข้อมูลจะแสดงดังภาพที่ น.17



ภาพที่ ๑.17 หน้าจอแสดงรายละเอียดรายการตาราง และคอลัมน์ของฐานข้อมูล

13. รายละเอียดรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรม

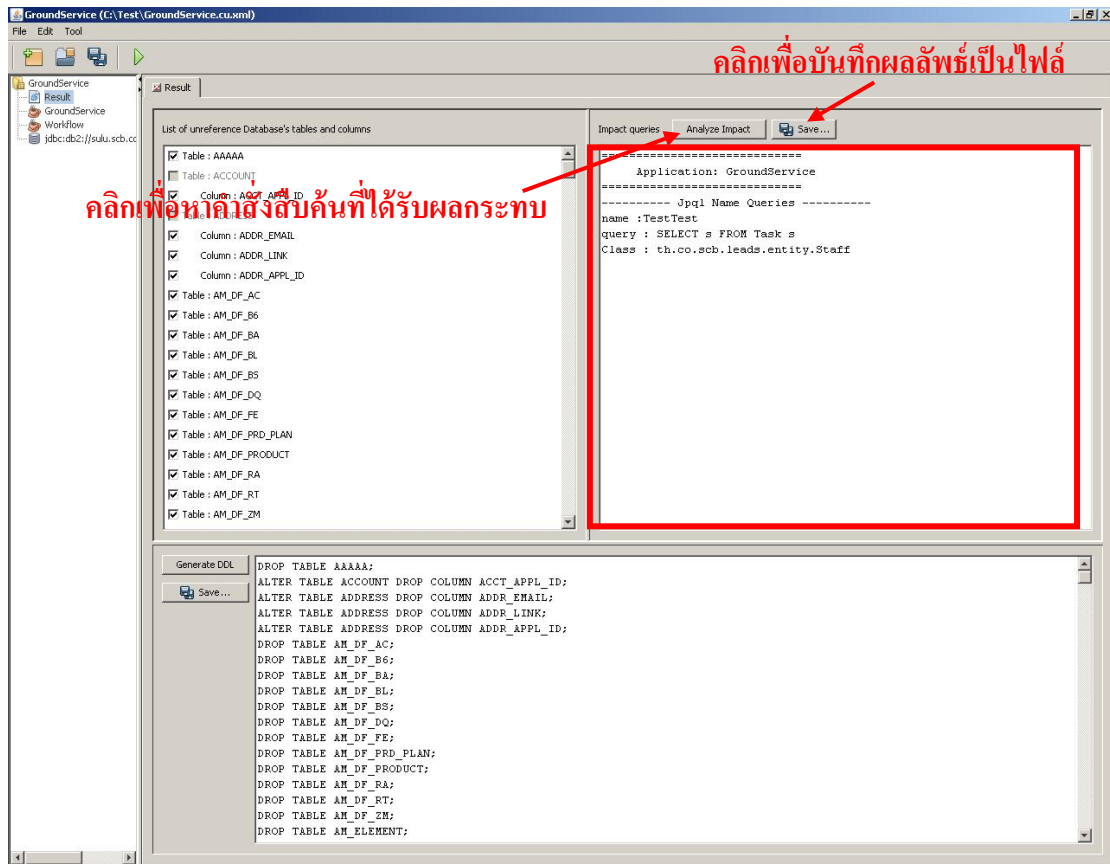
หลังจากวิเคราะห์หาตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงเสร็จเรียบร้อยแล้ว ผู้ใช้งานสามารถเรียกดูรายละเอียดรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรมได้โดยการดับเบิลคลิกที่ “Result” จากเมนูนำทาง ซึ่งรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรมจะแสดงดังภาพที่ ๑.18



ภาพที่ จ.18 หน้าจอแสดงรายละเอียดรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงใน โปรแกรม

13.1 วิเคราะห์หาคำสั่งสืบค้นที่ได้รับผลกระทบ

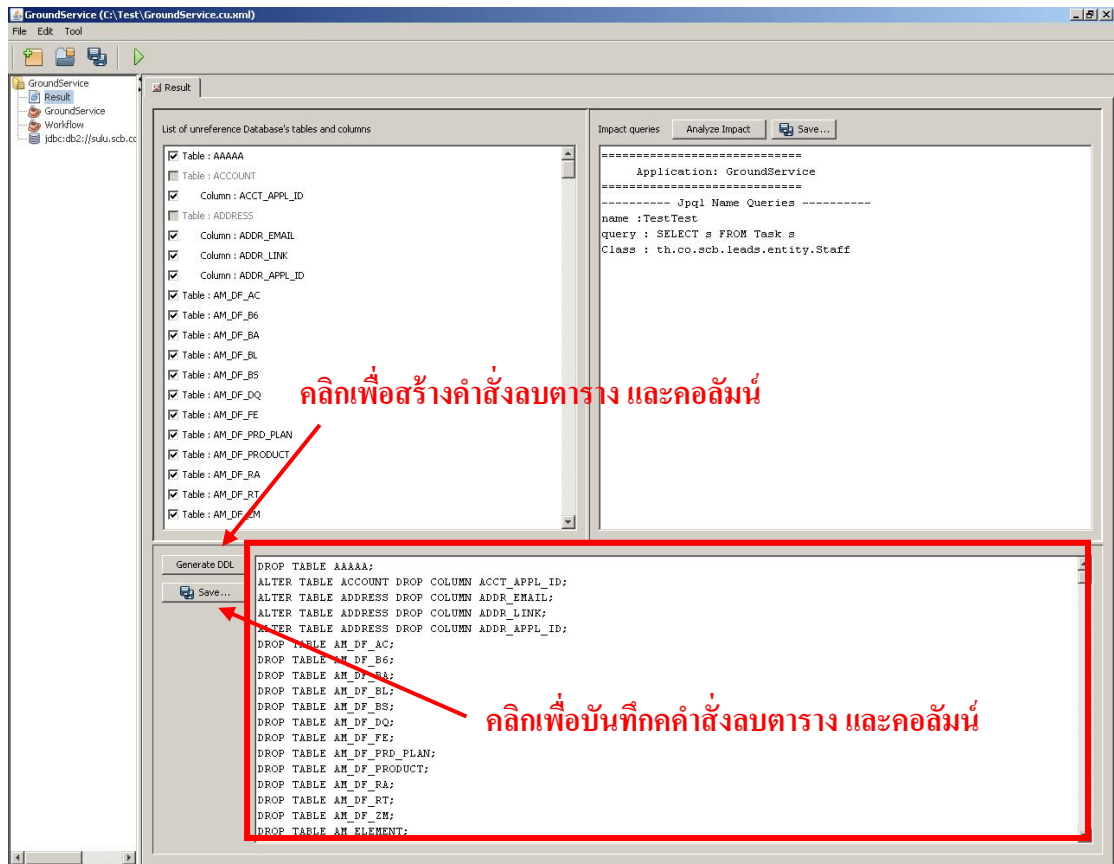
หลังจากผู้ใช้งานเปิดหน้าจอที่แสดงรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงใน โปรแกรมแล้ว หน้าจอจะแสดงรายการตาราง และคอลัมน์ให้ผู้ใช้งานเลือกว่าต้องการลบตาราง หรือคอลัมน์ใดบ้าง หลังจากผู้ใช้งานเลือกรายการตาราง และคอลัมน์ที่ต้องการจะลบแล้ว สามารถหาผลกระทบที่จะเกิดขึ้นกับคำสั่งสืบค้นในโปรแกรมได้ โดยคลิกที่ปุ่ม “Analyze Impact” แล้ว เครื่องมือจะหาคำสั่งสืบค้นที่ได้รับผลกระทบเมื่อรายการตาราง และคอลัมน์ที่ผู้ใช้งานเลือกถูกลบ ออกจากฐานข้อมูล ซึ่งผลลัพธ์ที่ได้จะแสดงรายการคำสั่งสืบค้น และบอกว่าคำสั่งดังกล่าวอยู่ใน คลาส และเมธอดใด ดังแสดงตัวอย่างในภาพที่ จ.19



ภาพที่ น.19 หน้าจอคำสั่งสืบค้นที่ได้รับผลกระทบ

13.2 สร้างคำสั่งลบตาราง และคอลัมน์

หลังจากผู้ใช้งานเปิดหน้าจอที่แสดงรายการตาราง และคอลัมน์ที่ไม่ถูกอ้างอิงในโปรแกรมแล้ว หน้าจอจะแสดงรายการตาราง และคอลัมน์ให้ผู้ใช้งานเลือกว่าต้องการลบตารางหรือคอลัมน์ใดบ้าง หลังจากผู้ใช้งานเลือกรายการตาราง และคอลัมน์ที่ต้องการจะลบแล้ว สามารถสร้างคำสั่งสำหรับลบตาราง และคอลัมน์ที่เลือก โดยการกดปุ่ม “Generate DDL” จากนั้นเครื่องมือจะนำรายการตาราง และคอลัมน์ที่ผู้ใช้งานเลือกมาสร้างเป็นคำสั่ง ดังแสดงในภาพที่ น.20



ภาพที่ น.20 หน้าจอแสดงคำสั่งเอสคิวแอลมาตรฐานสำหรับลบตาราง และคอลัมน์

ประวัติผู้เขียนวิทยานิพนธ์

นายบริรักษ์ หัตถา เกิดวันพุธที่ 21 ตุลาคม พ.ศ. 2530 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2552 ทำงานเป็นนักเขียนโปรแกรมของธนาคารไทยพาณิชย์ จำกัด มหาชน ตั้งแต่ปี 2553 จนถึงปัจจุบัน และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ภาคนอกเวลาราชการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554