

เครื่องมือสำหรับการทดสอบเงื่อนไขในแคมเปญการตลาด

นางสาวฤกษ์ระวี พันธุ์สุภา

โครงการมหาบัณฑิตนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A Tool for Conditions Testing in Marketing Campaign

Miss Roekrawee Panluka

A Master Project 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 2016
Copyright of Chulalongkorn University

หัวข้อโครงการมหาบัณฑิต	เครื่องมือสำหรับการทดสอบเงื่อนไขในแคมเปญการตลาด
โดย	นางสาวฤกษ์ระวี พันธุ์สุภา
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษา	ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
อนุมัติให้รับโครงการมหาบัณฑิตฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์)

..... อาจารย์ที่ปรึกษา
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

นางสาวฤกษ์ระวี พันธุ์ลู่กา : เครื่องมือสำหรับการทดสอบเงื่อนไขในแคมเปญการตลาด. (A Tool for Conditions Testing in Marketing Campaign) อ. ที่ปรึกษา: ศาสตราจารย์ ดร. ประภาส จงสถิตย์วัฒนา จำนวนหน้า 43 หน้า.

ในการทดสอบเงื่อนไขแคมเปญที่ใช้ในการโฆษณาผลิตภัณฑ์นั้นเป็นการทำงานร่วมกันระหว่างฝ่ายการตลาดและฝ่ายสารสนเทศ ซึ่งมีข้อจำกัดทางด้านความรู้ในรายละเอียดงานของแต่ละฝ่าย อีกทั้งปัญหาด้านเวลาที่ใช้ในการทดสอบเงื่อนไข และปัญหาความผิดพลาดของเงื่อนไข เนื่องด้วยวิธีการทดสอบในปัจจุบันใช้การทดสอบโดยมนุษย์ และทดสอบบางกรณีเท่านั้น โครงการมหาบัณฑิตนี้จึงได้เสนอเครื่องมือสำหรับการทดสอบเงื่อนไขในแคมเปญการตลาดเพื่อช่วยลดความผิดพลาดและลดระยะเวลาในการทดสอบเงื่อนไขแคมเปญการตลาดดังกล่าว

โดยโครงการมหาบัณฑิตนี้ได้นำเสนอเครื่องมือสำหรับสร้างกรณีสอบของเงื่อนไขที่ได้จากการสร้างแคมเปญจากซอฟต์แวร์การจัดการแคมเปญ โดยเป็นการทดสอบแบบวิธีไวท์บ็อกซ์ ซึ่งเป็นการทดสอบทุกเงื่อนไขในแคมเปญ และใช้วิธีการหาค่าตัวแปรโดยใช้วิธีแยกชั้นสมมูล ผลจากการนำเครื่องมือไปทดสอบใช้งานพบว่า เครื่องมือที่ได้มีความถูกต้อง แสดงว่าเครื่องมือที่นำเสนอมีประสิทธิภาพในการทดสอบเงื่อนไขแคมเปญการตลาด

ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมซอฟต์แวร์

ลายมือชื่ออาจารย์ที่ปรึกษา

ปีการศึกษา 2559

5770956521 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: Testing / Test Case / Paser

ROEKRAWEE PANLUKA : เครื่องมือสำหรับการทดสอบเงื่อนไขในแคมเปญการตลาด. (A tool for Conditions Testing in Marketing Campaign) MASTER PROJECT
ADVISOR: Prof.Dr. PRABHAS CHONGSTITVATANA, 43 pp.

Campaign conditions testing for advertised products is a collaboration between the marketing and information technology department team. Each team has a limited knowledge of the details. It takes a lot of time to test many conditions. The conditions tested by a manual method are error prone. This master project offers a tool for testing conditions in marketing campaigns to help reduce errors and time to test the marketing campaign conditions.

This project presents a tool for test case generation from the conditions created by a campaign management software. It employs a white-box method which will generate all possible tests of conditions in a campaign. The tool identifies all variables and creates equivalent classes. The result of using the tool is correct. This tool is effective in testing conditions in the marketing campaign.

Department	<u>Computer Engineering</u>	Student's signature	_____
Field of study	<u>Software Engineering</u>	Advisor's signature	_____
Academic year	<u>2016</u>		

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ ศ. ดร.ประภาส จงสถิตย์วัฒนา อาจารย์ที่ปรึกษาโครงการนมหบัณฑิต เป็นอย่างยิ่งที่ได้ให้คำแนะนำ แนวทางแก้ไขปัญหา ข้อเสนอแนะต่าง ๆ ในการศึกษาและพัฒนาโครงการนมหบัณฑิต นอกจากนี้แล้วอาจารย์ยังเสียสละเวลาและกำลังกายเพื่อมาพบและคอยให้คำปรึกษา พร้อมทั้งชี้แนะแนวทางในการดำเนินชีวิตประจำวัน

ขอกราบขอบพระคุณ ผศ.นครทิพย์ พร้อมพูล อาจารย์ประจำภาควิชาวิศวกรรมซอฟต์แวร์ที่คอยถามไถ่ ให้ความเป็นห่วงเป็นใยแก่นิสิตทุกคน และกรุณาแนะนำแนวทาง รวมถึงการตรวจสอบและแก้ไขโครงการนมหบัณฑิต

ขอกราบขอบพระคุณ รศ.นงลักษณ์ โค้ววิสารัช คณะกรรมการคุมสอบโครงการนมหบัณฑิตเป็นอย่างยิ่ง ที่ได้กรุณาแนะนำแนวทาง รวมถึงการตรวจสอบและแก้ไขโครงการนมหบัณฑิต

ขอกราบขอบพระคุณ ผศ. ดร. ทวีติย์ เสนีวงศ์ ณ อยุธยา อาจารย์ผู้ประสานงานโครงการนมหบัณฑิตที่ได้สละเวลาให้ความช่วยเหลือแก่นิสิตที่ทาโครงการนทุกคน

ขอขอบคุณ พรรค์มี จตุณราพิทย์ น.ส.โซษิตา พัฒนธีระเดช น.ส.ปารวี และนายธนธาร ภัทรกวิน สำหรับการช่วยเหลือในเรื่องของการเรียนในห้องเรียน การให้คำปรึกษา ตลอดจนการช่วยจัดทำโครงการนมหบัณฑิต และคอยเป็นที่ปรึกษาในทุก ๆ ด้าน

ขอขอบคุณ พี่และเพื่อนๆ นิสิตร่วมรุ่น ในหลักสูตรวิศวกรรมซอฟต์แวร์ สำหรับข้อเสนอแนะ คำแนะนำและกำลังใจในการจัดทำโครงการนมหบัณฑิต

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดา รวมถึงสมาชิกในครอบครัวที่ให้การสนับสนุน และให้กำลังใจที่ตีเสมอมา

ฤกษ์ระวี พันธุ์ฤกา

สารบัญ

	หน้า
สารบัญ.....	๗
สารบัญรูป.....	๘
สารบัญตาราง.....	๙
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ.....	3
1.3 ขอบเขตของโครงการ.....	3
1.4 ขั้นตอนและวิธีการดำเนินโครงการ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 ผลงานตีพิมพ์.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1.1 การทดสอบซอฟต์แวร์	4
2.1.2 การทดสอบแบบชั้นสมมูล.....	4
2.1.3 การทดสอบแบบไวท์บ็อกซ์	7
2.1.4 ภาษาเอสคิวแอล.....	7
2.1.5 ภาษาพีแอลเอสคิวแอล.....	7
2.1.6 ซอฟต์แวร์การจัดการแคมเปญ.....	8
2.2 งานวิจัยที่เกี่ยวข้อง	8
2.2.1 การใช้การวัดความครอบคลุมเอสคิวแอลสำหรับแอปพลิเคชันการทดสอบ ฐานข้อมูล.....	8
2.2.2 แนวทางปฏิบัติต่อการทดสอบภาษาเอสคิวแอลแบบไวท์บ็อกซ์	9
2.2.3 การสร้างเอสคิวแอลคิวรีอัตโนมัติ สำหรับระบบการทดสอบของเครื่องประมวลผล ฐานข้อมูล.....	9
2.2.4 การสร้างฐานข้อมูลเสถียรผ่านการดำเนินการสัญลักษณ์แบบไดนามิกสำหรับเกณฑ์ ความครอบคลุม.....	10

2.2.5 เครื่องมือสร้างกรณีทดสอบสำหรับโปรแกรมประยุกต์บนเว็บบนพื้นฐานสคีมาของฐานข้อมูล.....	10
บทที่ 3 แนวคิดและวิธีดำเนินการ.....	12
3.1 ภาพรวมการทำงานของระบบการสร้างกรณีทดสอบ.....	12
3.2 ภาพรวมการทำงาน	13
3.2.1 ตัวแฉส่วน	13
3.2.2 การวิเคราะห์หาค่าของตัวแปร	14
3.2.3 การสร้างกรณีทดสอบ	16
3.2.4 ขั้นตอนการสร้างกรณีทดสอบ	16
3.3 การออกแบบส่วนต่อประสานผู้ใช้งาน.....	19
บทที่ 4 การทดลอง.....	22
4.1 สภาพแวดล้อมที่ใช้ทดสอบ	22
4.2 ข้อมูลที่ใช้ในการทดลอง	22
4.3 ผลการทดสอบ.....	23
4.4 ผลการทดสอบเครื่องมือ	31
บทที่ 5 บทสรุปโครงการและข้อเสนอแนะ	32
5.1 สรุปผลโครงการมหาบัณฑิต.....	32
5.2 ข้อเสนอแนะและข้อปรับปรุงงานวิจัยในอนาคต.....	32
รายการอ้างอิง.....	33
ภาคผนวก	35
ภาคผนวก ก รหัสเงินตรา.....	36
ภาคผนวก ข รหัสประเภทร้านค้า.....	39
ประวัติผู้เขียนโครงการมหาบัณฑิต.....	43

สารบัญรูป

รูปที่ 1.1	แผนภาพแสดงลำดับเหตุการณ์การทำงานในการออกแคมเปญ	2
รูปที่ 2.1	กรณีการทดสอบขั้นสมมูลแบบวิกนอร์มอล	5
รูปที่ 2.2	กรณีการทดสอบขั้นสมมูลแบบสตรองนอร์มอล	6
รูปที่ 2.3	กรณีการทดสอบขั้นสมมูลแบบวิกโรบัส	6
รูปที่ 2.4	กรณีการทดสอบขั้นสมมูลแบบสตรองโรบัส	7
รูปที่ 2.5	ตัวอย่างซอฟต์แวร์การจัดการอะโดบีแคมเปญ (Adobe Campaign)	8
รูปที่ 2.6	ตัวอย่างข้อมูลในการสร้างข้อสอบถาม (Query)	9
รูปที่ 2.7	ผลลัพธ์ที่จากการสร้างแบบอัตโนมัติผ่านชุดเครื่องมือ	10
รูปที่ 3.1	แผนภาพแสดงการไหลของระบบงาน	12
รูปที่ 3.2	แผนภาพกิจกรรมของการทำงาน	13
รูปที่ 3.3	ไวยากรณ์ภาษา SQL	14
รูปที่ 3.4	แผนภาพต้นไม้ที่จากการวิเคราะห์	17
รูปที่ 3.5	พารามิเตอร์และค่าของพารามิเตอร์จากโปรแกรม	18
รูปที่ 3.6	เงื่อนไขที่ใช้เป็นส่วนนำเข้าของข้อมูล	19
รูปที่ 3.7	หน้าจอแสดงผลพีธีในการสร้างกรณีทดสอบ	20
รูปที่ 3.8	ตัวอย่างการเรียกใช้งานเครื่องมือ	20

สารบัญตาราง

ตารางที่ 2.1 ตัวอย่างชั้นสมมูลที่สามารถแบ่งได้.....	5
ตารางที่ 3.1 ผลลัพธ์ที่จากการวิเคราะห์ไวยากรณ์ของภาษาเอสคิวแอลเมื่อใช้ตัวดำเนินการ AND..	15
ตารางที่ 3.2 ผลลัพธ์ที่จากการวิเคราะห์ไวยากรณ์ของภาษาเอสคิวแอลเมื่อใช้ตัวดำเนินการ OR....	15
ตารางที่ 3.3 ผลลัพธ์ที่จากการวิเคราะห์ไวยากรณ์ของภาษาเอสคิวแอล.....	15
ตารางที่ 3.4 แสดงตัวอย่างกรณีที่ต้องและไม่ถูกต้อง.....	16
ตารางที่ 3.5 ตัวอย่างการสร้างกรณีทดสอบย่อยของพารามิเตอร์ที่ 1	18
ตารางที่ 3.6 ตัวอย่างการสร้างกรณีทดสอบย่อยของพารามิเตอร์ที่ 2	18
ตารางที่ 3. 7 ตัวอย่างการสร้างกรณีทดสอบชั้นสมมูลแบบสตรองโรบัส	19
ตารางที่ 3. 8 ตารางตรรกศาสตร์ที่ได้จากการวิเคราะห์ของเครื่องมือ	21
ตารางที่ 4.1 ผลการทดลองของ Campaign ID : CM001.....	23
ตารางที่ 4.2 ผลการทดลองของ Campaign ID : CM002.....	24
ตารางที่ 4.3 ผลการทดลองของ Campaign ID : CM003.....	25
ตารางที่ 4.4 ผลการทดลองของ Campaign ID : CM004.....	25
ตารางที่ 4.5 ผลการทดลองของ Campaign ID : CM005.....	26
ตารางที่ 4.6 ผลการทดลองของ Campaign ID : CM006.....	27
ตารางที่ 4.7 ผลการทดลองของ Campaign ID : CM007.....	28
ตารางที่ 4.8 ผลการทดลองของ Campaign ID: CM008.....	29
ตารางที่ 4.9 ผลการทดลองของ Campaign ID : CM009.....	30
ตารางที่ 4.10 ผลการทดลองของ Campaign ID: CM010.....	30

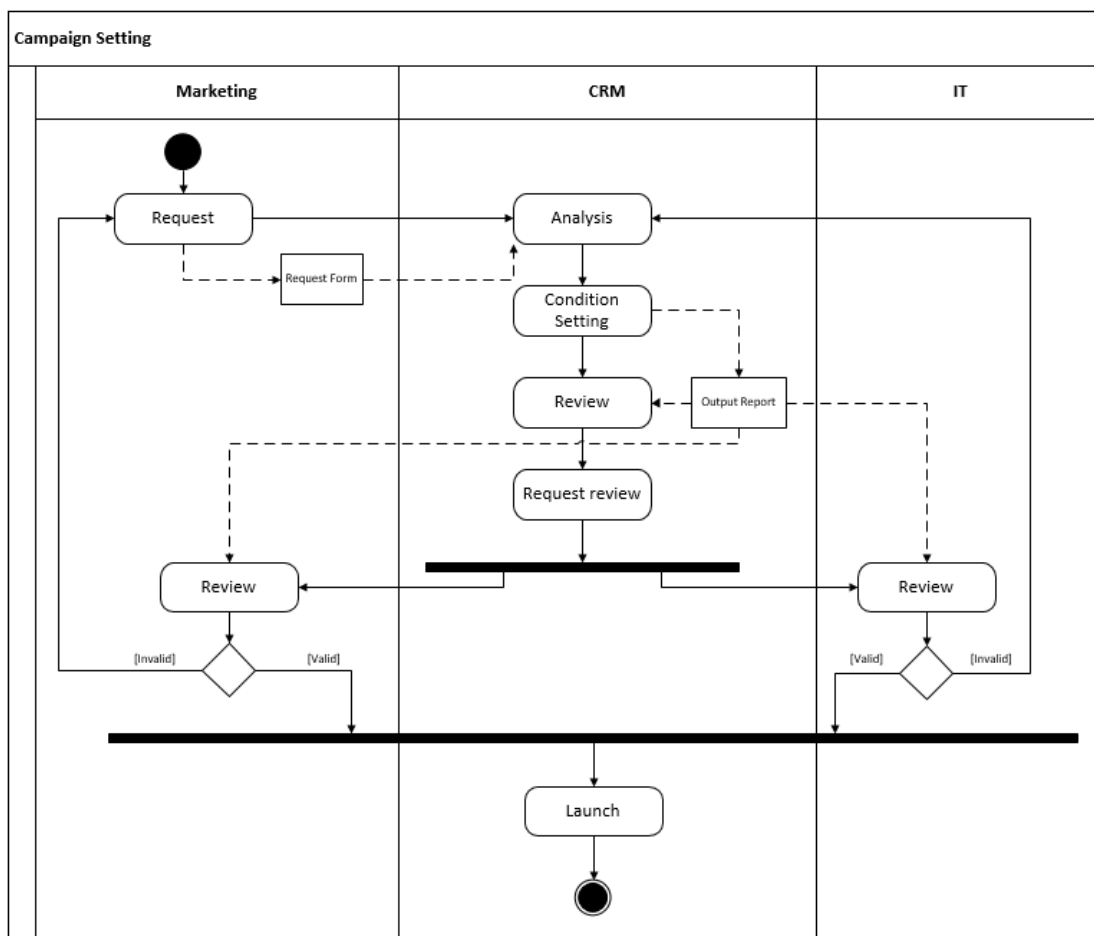
บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการสร้างความสัมพันธ์กับลูกค้าเพื่อให้ลูกค้ารู้สึกผูกพันกับสินค้าถือเป็นสิ่งจำเป็นอย่างหนึ่งที่หลายธุรกิจต้องการเพราะเป็นการรักษาฐานของลูกค้า แต่การที่จะทำให้ลูกค้ารู้สึกผูกพันกับสินค้าได้ต้องมีการสร้างปฏิสัมพันธ์กับกลุ่มลูกค้า โดยส่วนใหญ่ภาคธุรกิจจะออกการรณรงค์หรือแคมเปญ (Campaign) ให้กับกลุ่มลูกค้า เพื่อให้ลูกค้ามีปฏิสัมพันธ์กับสินค้า ซึ่งก่อนการออกแคมเปญแต่ละครั้งนักการตลาดจะทำการวิเคราะห์พฤติกรรมของกลุ่มลูกค้ากับสินค้าเพื่อสร้างข้อเสนอที่เหมาะสมกับกลุ่มลูกค้าและสร้างแรงจูงใจในการเข้าร่วมแคมเปญ ปัจจุบันมีซอฟต์แวร์ที่ช่วยในการจัดการแคมเปญซึ่งทำหน้าที่ในการสร้างแคมเปญเพื่อใช้ในการออกแคมเปญไปยังกลุ่มลูกค้าโดยผ่านช่องทางการสื่อสารได้หลายช่องทาง เช่น การส่งข้อความ การส่งไปรษณีย์ อิเล็กทรอนิกส์ (E-mail) และการโทรไปเสนอแคมเปญแก่ลูกค้าโดยตรงจากกลุ่มงานโทรศัพท์ เป็นต้น นอกจากนี้ซอฟต์แวร์ยังมีการจัดเก็บข้อมูลของลูกค้าที่ทำการเข้าร่วมแคมเปญ เพื่อนำข้อมูลไปวิเคราะห์ต่อไป

การออกแคมเปญแต่ละครั้ง ฝ่ายการตลาด (Marketing) และฝ่ายการบริหารความสัมพันธ์กับลูกค้า (CRM: Customer Relationship Management) จะทำการวางแผนร่วมกัน โดยฝ่ายการตลาดจะทำหน้าที่ส่งความต้องการสำหรับการออกแคมเปญมายังฝ่ายการบริหารความสัมพันธ์กับลูกค้า (CRM) เพื่อสร้างเงื่อนไขที่ใช้ในการออกแคมเปญตามความต้องการ เมื่อมีการจัดทำแคมเปญเรียบร้อยแล้วจะทำการแจ้งกลับไปยังฝ่ายการตลาดเพื่อตรวจสอบเงื่อนไขกับเอกสารความต้องการในขณะเดียวกันก็จะส่งคำร้องขอให้ฝ่ายไอทีทำการตรวจสอบเงื่อนไขของแคมเปญในเรื่องความถูกต้องและความครบถ้วนของเอกสารความต้องการ หากพบข้อผิดพลาดจากการออกแคมเปญจะทำการตอบกลับไปยังฝ่ายการบริหารความสัมพันธ์กับลูกค้า (CRM) และทำการทวนสอบซ้ำกับฝ่ายการตลาดอีกครั้ง โดยสามารถอธิบายการทำงานนี้ตามแผนภาพกิจกรรมดังรูปที่ 1.1



รูปที่ 1.1 แผนภาพแสดงลำดับเหตุการณ์การทำงานในการออกแคมเปญ

จากรูปที่ 1.1 หลังกระบวนการสร้างแคมเปญจะมีมีกระบวนการตรวจสอบแคมเปญ ซึ่งกระบวนการตรวจสอบแคมเปญในแต่ละครั้งจะใช้วิธีตรวจสอบด้วยมนุษย์ โดยกระบวนการในปัจจุบันผู้ตรวจสอบจะมีการนำเงื่อนไขไปทำประมวลผลเงื่อนไขของแคมเปญ (Executed SQL Condition) ในลักษณะการทดสอบแบบแบล็คบ็อกซ์ (Black-Box) คือมีการตรวจสอบโดยการนำกรณีทดสอบมาตรวจสอบผลลัพธ์โดยไม่คำนึงถึงขั้นตอนและเงื่อนไขของแคมเปญ เพื่อดูว่ามีการแสดงข้อผิดพลาดออกมาทางหน้าจอหรือไม่ โดยการทดสอบในลักษณะนี้จะไม่เป็นการทดสอบเงื่อนไขในทุกกรณี จึงทำให้ไม่พบข้อผิดพลาดที่อาจเกิดขึ้นภายในเงื่อนไขของแคมเปญ

นอกจากปัญหาที่กล่าวมาข้างต้นระยะเวลาที่ใช้ตั้งแต่ฝ่ายการตลาดทำการขอออกแคมเปญ จนถึงกระบวนการตรวจสอบใช้เวลาเฉลี่ย ประมาณ 1 วัน ซึ่งผู้วิจัยได้เล็งเห็นปัญหาและนำเสนอเครื่องมือเพื่อช่วยในการตรวจสอบเงื่อนไขของแคมเปญที่ครอบคลุมการทดสอบทุกเงื่อนไข เพื่อช่วยลดข้อผิดพลาดที่อาจเกิดขึ้นและช่วยลดระยะเวลาในการดำเนินการ

1.2 วัตถุประสงค์ของโครงการ

นำเสนอเครื่องมือสำหรับสร้างกรณีทดสอบของเงื่อนไขที่ได้จากการสร้างแคมเปญจากซอฟต์แวร์การจัดการแคมเปญ

1.3 ขอบเขตของโครงการ

- 1) ไฟล์ข้อมูลที่ใช้ในการทดสอบเป็นไฟล์นามสกุลเอสคิวแอล (.sql) โดยรูปแบบของภาษาเอสคิวแอลจัดอยู่ในรูปแบบของพีแอลเอสคิวแอล (PL/SQL) เท่านั้น
- 2) ตรวจสอบเงื่อนไขของแคมเปญในรูปแบบคำสั่งในการค้นหา (Select Statement)

1.4 ขั้นตอนและวิธีการดำเนินโครงการ

ขั้นตอนและวิธีการการดำเนินโครงการมหาบัณฑิตนี้จะดำเนินการดังต่อไปนี้

- 1) ศึกษาทำความเข้าใจเกี่ยวกับการทดสอบ
- 2) ศึกษาการทำการทดสอบของภาษา SQL จากงานวิจัยต่าง ๆ
- 3) พัฒนาเครื่องมือตามที่ได้ออกแบบไว้
- 4) ทดสอบและประเมินผลเครื่องมือที่ได้พัฒนาขึ้น
- 5) จัดทำบทความทางวิชาการและนำเสนอ
- 6) สรุปผลและจัดทำเล่มโครงการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เครื่องมือที่ใช้ในการทดสอบการสร้างเงื่อนไขของแคมเปญที่ถูกจัดใช้โดยฝ่ายสารสนเทศ
- 2) ช่วยลดระยะเวลาในการทำงานของฝ่ายสารสนเทศ

1.6 ผลงานตีพิมพ์

โครงการมหาบัณฑิตนี้ได้รับการตีพิมพ์เป็นบทความวิชาการเรื่อง “ระบบสำหรับการทดสอบเงื่อนไขในแคมเปญการตลาด” ได้รับคัดเลือกและตีพิมพ์ในการประชุมวิชาการ “The 12th National Conference on Computing and Information Technology (NCCIT 2016)” ระหว่างวันที่ 7 – 8 กรกฎาคม 2559 ณ โรงแรมโรงแรมเซ็นทาราและคอนเวนชันเซ็นเตอร์ จังหวัดขอนแก่น ประเทศไทย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ [1] หมายถึง กระบวนการที่ใช้ตรวจสอบว่าซอฟต์แวร์ที่จัดทำขึ้นมาเป็นไปตามความต้องการหรือเพื่อที่จะระบุความแตกต่างระหว่างผลลัพธ์ที่คาดหวังกับผลลัพธ์ที่ได้จริงจากซอฟต์แวร์ โดยสามารถทำได้ทั้งการทำด้วยมือ หรือการทำโดยอัตโนมัติ

เป้าหมายของการทดสอบซอฟต์แวร์ มีดังนี้

1. สามารถป้องกันความผิดพลาด (Bug) ที่เกิดขึ้นกับซอฟต์แวร์ได้
2. สามารถบอกได้ว่าความผิดพลาดที่เกิดขึ้นส่งผลกระทบต่อซอฟต์แวร์อย่างไร ถ้าไม่สามารถป้องกันความผิดพลาดที่เกิดขึ้นได้
3. การทดสอบควรจะบอกแนวทางแก้ไขที่ชัดเจน

การทดสอบซอฟต์แวร์ แบ่งเป็น 4 ระดับ [2] ดังนี้

1. การทดสอบหน่วยโปรแกรม (Unit Testing) คือ การทดสอบโปรแกรมหลังจากที่เขียนโปรแกรมหน่วยนั้นเสร็จ
2. การทดสอบแบบบูรณาการ (Integration Testing) คือ การทดสอบซอฟต์แวร์โดยนำแต่ละหน่วยโปรแกรมที่ผ่านการทดสอบหน่วยโปรแกรมมาแล้วมาทดสอบร่วมกัน
3. การทดสอบระบบ (System Testing) คือ การทดสอบว่าซอฟต์แวร์ที่พัฒนาได้นั้นทำงานได้ตรงกับข้อกำหนดความต้องการของลูกค้า (Customer Requirements) หรือไม่
4. การทดสอบการตรวจรับ (Acceptance Testing) คือ การนำเอาซอฟต์แวร์ไปทดสอบกับสิ่งแวดล้อมที่จะถูกนำไปใช้งานจริง เพื่อยืนยันว่าซอฟต์แวร์นั้นสามารถทำงานได้ตามความต้องการที่กำหนดไว้

2.1.2 การทดสอบแบบชั้นสมมูล

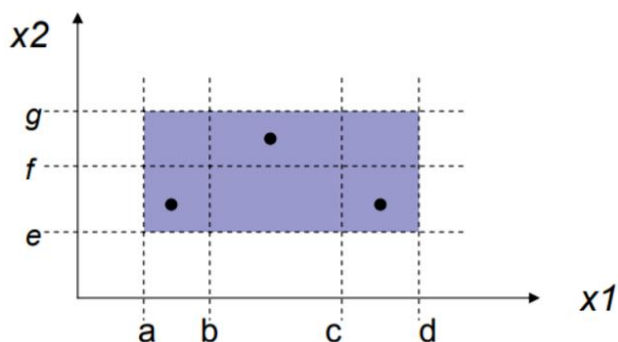
การทดสอบแบบชั้นสมมูล (Equivalence Class Testing) [3,4] เป็นวิธีการออกแบบกรณีทดสอบ โดยแบ่งชุดข้อมูลออกเป็น ส่วน ๆ หรือเป็นชั้น (Class) ซึ่งแต่ละชั้นจะไม่มีข้อมูลที่ซ้ำกัน จึงทำให้แน่ใจว่าการทดสอบมีความสมบูรณ์ไม่มีความซ้ำซ้อนกัน เช่น ตัวแปร $X1$ และ $X2$ ภายใต้เงื่อนไข $a \leq X1 \leq d$ และ $e \leq X2 \leq g$ สามารถแบ่งชั้นสมมูลโดยใช้หลักการการทดสอบโดยใช้ค่าขอบเขต (Boundary Value Testing) ทำให้ได้ค่าแต่ละชั้นตามตารางที่ 2.1 ดังนี้

ตารางที่ 2.1 ชั้นสมมูลของตัวแปร X1 และ X2

Variable	Equivalence Class
X1	$a \leq X1 < b$
	$b \leq X1 < c$
	$c \leq X1 \leq d$
X2	$e \leq X2 < f$
	$f \leq X2 \leq g$

โดยการทดสอบสำหรับชั้นสมมูลสามารถทำได้ 4 วิธีคือ

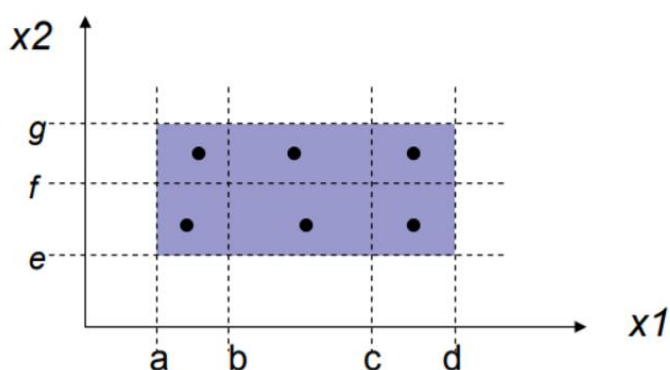
1. การทดสอบโดยใช้ชั้นสมมูลแบบวีคนอร์มอล (Weak Normal Equivalence Class Testing) จากรูปที่ 2.1 การออกแบบกรณีทดสอบขึ้นอยู่กับสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมมักจะไม่ใช่ผลที่เกิดจากข้อผิดพลาดตั้งแต่ 2 อันขึ้นไปพร้อม ๆ กัน หรือจะเกิดข้อผิดพลาดเดียวเท่านั้น (Single Fault Assumption) และชั้นของข้อมูลที่ใช้พิจารณาจะพิจารณาเฉพาะชั้นสมมูลที่ถูกต้อง (Valid Equivalence Class) เท่านั้น ซึ่งกรณีทดสอบที่สร้างออกมาจะต้องครอบคลุมทุก ๆ ชั้น



รูปที่ 2.1 กรณีการทดสอบชั้นสมมูลแบบวีคนอร์มอล

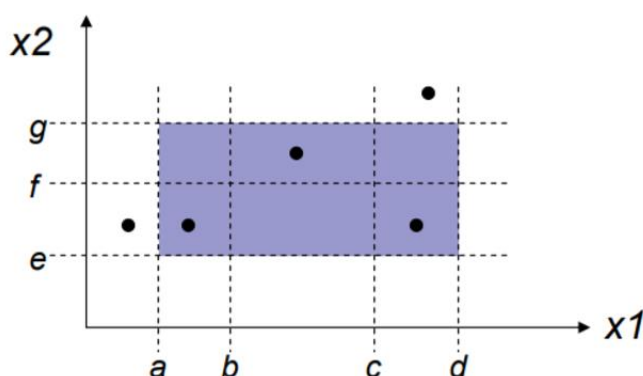
จากรูปที่ 2.1 กำหนดให้พื้นที่ที่แรเงา คือ ชั้นสมมูลที่ถูกต้อง (Valid Equivalence Class) พื้นที่ที่ไม่ได้แรเงา คือ ชั้นสมมูลที่ไม่ถูกต้อง (Invalid Equivalence Class) และจุดสีดำ คือ ข้อผิดพลาด (Fault Assumption)

2. การทดสอบโดยใช้ชั้นสมมูลแบบสตรองนอร์มอล (Strong Normal Equivalence Class Testing) จากรูปที่ 2.2 การออกแบบกรณีทดสอบขึ้นอยู่กับสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมมักจะเป็นผลที่เกิดจากข้อผิดพลาดตั้งแต่ 2 อันขึ้นไปพร้อม ๆ กัน (Multiple Fault Assumption) และชั้นของข้อมูลที่ใช้พิจารณาจะพิจารณาเฉพาะชั้นสมมูลที่ถูกต้องเท่านั้น ซึ่งกรณีทดสอบที่สร้างออกมาจะต้องครอบคลุมทุก ๆ การรวมกันของชั้นที่เป็นไปได้ทั้งหมด หรือทุก ๆ ผลคูณคาร์ทีเซียน (Cartesian Product) ที่เป็นไปได้



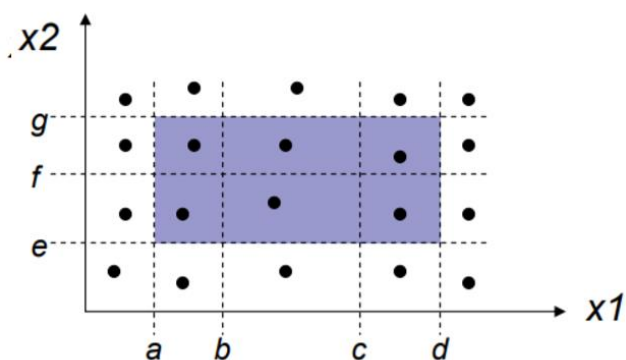
รูปที่ 2.2 กรณีการทดสอบชั้นสมมูลแบบสตรองนอร์มอล

3. การทดสอบโดยใช้ชั้นสมมูลแบบวิคโรบัส (Weak Robust Equivalence Class Testing) จากรูปที่ 2.3 การออกแบบกรณีทดสอบขึ้นอยู่กับสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมมักจะเกิดข้อผิดพลาดเดียวเท่านั้น (Single Fault Assumption) และชั้นของข้อมูลที่ใช้พิจารณาจะพิจารณาชั้นสมมูลที่ถูกต้องและชั้นสมมูลที่ไม่ถูกต้อง (Invalid Equivalence Class) ซึ่งกรณีทดสอบที่สร้างออกมาจะต้องครอบคลุมทุก ๆ ชั้น



รูปที่ 2.3 กรณีการทดสอบชั้นสมมูลแบบวิคโรบัส

4. การทดสอบโดยใช้ชั้นสมมูลแบบสตรองโรบัส (Strong Robust Equivalence Class Testing) จากรูปที่ 2.4 การออกแบบกรณีทดสอบขึ้นอยู่กับสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมมักจะเป็นผลที่เกิดจากข้อผิดพลาดตั้งแต่ 2 อันขึ้นไปพร้อม ๆ กัน และชั้นของข้อมูลที่ใช้พิจารณาจะพิจารณาชั้นสมมูลที่ถูกต้องและชั้นสมมูลที่ไม่ถูกต้อง ซึ่งกรณีทดสอบที่สร้างออกมาจะต้องครอบคลุมทุก ๆ การรวมกันของชั้นทั้งหมด หรือทุก ๆ ผลคูณคาร์ทีเซียน (Cartesian Product) ของชั้นทั้งหมด



รูปที่ 2.4 กรณีการทดสอบขั้นสมมูลแบบสตรองโรบัส

2.1.3 การทดสอบแบบไวท์บ็อกซ์

การทดสอบแบบไวท์บ็อกซ์ (White Box) [5] เทคนิคที่ใช้ทดสอบซอฟต์แวร์หรือเรียกว่าการทดสอบเชิงโครงสร้าง ซึ่งวิธีการนี้จะพิจารณาภายในของระบบหรือทางเดินของโปรแกรม โดยจะมุ่งเน้นไปที่โครงสร้างการทำงานภายในของโปรแกรม และผู้ทดสอบต้องออกแบบกรณีทดสอบโดยขึ้นกับการทำงานภายในของโปรแกรม เพื่อให้วัตถุประสงค์เป็นไปตามผลลัพธ์ที่คาดหวังของวัตถุประสงค์เชิงธุรกิจ ซึ่งบ่งชี้ให้เห็นการทำงานที่ถูกต้อง ทั้งนี้เพื่อให้ได้ซอฟต์แวร์ที่มีประสิทธิภาพตรงตามความต้องการของผู้ใช้

2.1.4 ภาษาเอสคิวแอล

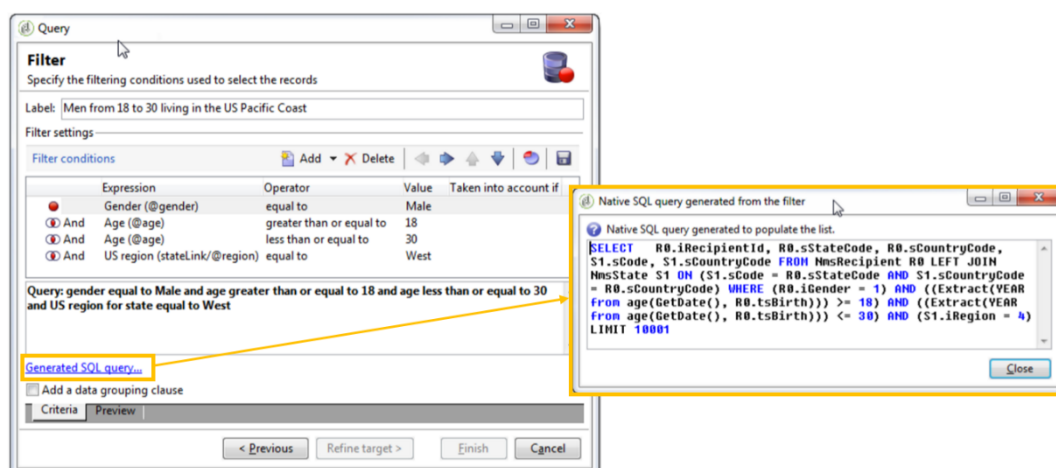
ภาษาเอสคิวแอล (SQL: Structured Query Language) [6] จัดเป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์ โดยผู้ที่เริ่มพัฒนาภาษาเอสคิวแอลครั้งแรก คือ บริษัทไอบีเอ็ม ใช้ชื่อในตอนแรกว่า “ซีเควล” (Sequel) ต่อมาจึงเปลี่ยนมาเป็น “เอสคิวแอล” (SQL) หลังจากนั้นภาษาเอสคิวแอลได้ถูกพัฒนาโดยผู้ผลิตซอฟต์แวร์ด้านระบบจัดการฐานข้อมูลเชิงสัมพันธ์ จนเป็นที่นิยมกันอย่างแพร่หลาย โดยผู้ผลิตแต่ละรายพยายามพัฒนารูปร่างข้อมูลของตัวเองให้มีลักษณะโดดเด่นขึ้นมา ทำให้รูปแบบการใช้คำสั่งมีความแตกต่างกันบ้าง เช่น ออราเคิล (Oracle) และเอสคิวแอลเซิร์ฟเวอร์ของไมโครซอฟต์

2.1.5 ภาษาพีแอลเอสคิวแอล

ภาษาพีแอลเอสคิวแอล (PL/SQL Language: Procedural Language SQL) [7] เป็นภาษาโปรแกรมที่ใช้ในการพัฒนาซอฟต์แวร์ของออราเคิล (Oracle) เป็นภาษาเชิงกระบวนการใช้เพิ่มความสามารถในการใช้เอสคิวแอลของออราเคิล ทำให้สามารถทำงานที่ซับซ้อนขึ้นได้ และสามารถจัดการกับข้อมูลในฐานข้อมูลได้ดี

2.1.6 ซอฟต์แวร์การจัดการแคมเปญ

ซอฟต์แวร์การจัดการแคมเปญ (Campaign Management Software) เป็นซอฟต์แวร์ที่ช่วยในการจัดการแคมเปญให้เป็นไปตามความต้องการฝ่ายการตลาดกับกลุ่มลูกค้าเพื่อให้ได้เป้าหมายที่ต้องการ โดยจะมีการออกสื่อเพื่อชักชวนลูกค้าให้ร่วมแคมเปญนั้น ๆ หากลูกค้าสนใจและตรงตามเงื่อนไขของแคมเปญจะมีการเก็บข้อมูลลูกค้าเพื่อให้ผลตอบแทนตามสื่อที่เสนอออกไป โดยในการสร้างแคมเปญแต่ละครั้งต้องมีการวางแผนสำหรับแคมเปญ เพราะแต่ละแคมเปญที่จะออกไปมักมีเป้าหมายที่แตกต่างกันไป จากนั้นจึงมีการสร้างแคมเปญให้เป็นไปตามความต้องการตามแผนที่ได้วางไว้ ซึ่งการสร้างเงื่อนไขแคมเปญในซอฟต์แวร์การจัดการแคมเปญจะมีลักษณะคล้ายคลึงกับภาษาเอสคิวแอล (SQL) และสามารถสร้างเงื่อนไขของแคมเปญให้อยู่ในรูปแบบของภาษาเอสคิวแอลได้ โดยความสัมพันธ์ของข้อมูลในตารางที่นำมาใช้จะถูกกำหนดตั้งแต่แรกเริ่มแล้ว ผู้ใช้มีหน้าที่เพียงใส่เงื่อนไขตามที่ต้องการลงไปเท่านั้น โดยสามารถเลือกตัวดำเนินการได้เฉพาะ and และ or เท่านั้น จากรูปที่ 2.5 [8] แสดงส่วนการกำหนดเงื่อนไขแคมเปญของโปรแกรมอะโดบีแคมเปญ ที่สนใจกลุ่มเพศชายที่มีอายุตั้งแต่ 18 ปี ถึง 30 ปี และมีภูมิภาคทางทิศตะวันตกของประเทศสหรัฐอเมริกา และส่วนที่สร้างเงื่อนไขตามที่กำหนดให้อยู่ในรูปแบบเอกสารของภาษาเอสคิวแอล เพื่อนำคำสั่งนี้ไปเป็นส่วนนำเข้าข้อมูล (Input Data) ของโปรแกรมอื่น ๆ ต่อไป



รูปที่ 2.5 ตัวอย่างซอฟต์แวร์การจัดการอะโดบีแคมเปญ (Adobe Campaign)

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 การใช้การวัดความครอบคลุมเอสคิวแอลสำหรับแอปพลิเคชันการทดสอบฐานข้อมูล

บทความนี้ (Using an SQL Coverage Measurement for Testing Database Applications) [9] ได้นำเสนอการวัดความครอบคลุมจากการเลือกข้อมูลเอสคิวแอล (SQL Select Statement) เพียงอย่างเดียว ปัญหาที่นำไปสู่การวิจัยชิ้นนี้คือการประเมินกรณีการทดสอบที่

ครอบคลุมและเพียงพอต่อการทดสอบของคำสั่งเอสคิวแอล โดยได้มีการนำโครงสร้างต้นไม้หรือเรียกว่าต้นไม้ครอบคลุม (Coverage Tree) มาใช้ก่อนทำการประเมินผล และนำเสนออัลกอริทึมในการวัดความครอบคลุมของต้นไม้

2.2.2 แนวทางปฏิบัติต่อการทดสอบภาษาเอสคิวแอลแบบไวท์บ็อกซ์

บทความนี้ (A practical guide to SQL white-box testing) [10] ได้นำเสนอแนวทางการทำวิธีการทดสอบแบบไวท์บ็อกซ์ (White Box) โดยเสนอแนะแนวทางการออกแบบการทำการทดสอบสำหรับการทดสอบระดับหน่วย (Unit test) ของภาษาเอสคิวแอลแบบไวท์บ็อกซ์ (White Box) โดยใช้วิธี MCDC (Modified Condition Decision Coverage) และได้แนะนำวิธีการไว้ 5 วิธีการดังนี้

1. ใช้ MCDC สำหรับเงื่อนไขเอสคิวแอล (Adopting MCDC for SQL Conditions)
2. ใช้ MCDC สำหรับแก้ปัญหาค่าว่าง (Adapting MCDC for Tackling with Nulls)
3. การแบ่งหมวดหมู่ในการเลือกข้อมูล (Category Partitioning Selected Data)
4. การตรวจสอบผลลัพธ์ (Checking the Outputs)
5. การตรวจสอบเงื่อนไขบังคับของฐานข้อมูล (Checking the Database Constraints)

2.2.3 การสร้างเอสคิวแอลคิวรีอัตโนมัติ สำหรับระบบการทดสอบของเครื่องประมวลผลฐานข้อมูล

บทความวิจัยนี้ (Automated SQL Query Generation for Systematic Testing of Database Engines) [11] ได้นำเสนอกรอบการทำงานจากเครื่องมือภาษาเอสคิวแอลสำหรับการนำมาสร้างเงื่อนไขแบบอัตโนมัติ และมีการสร้างข้อสอบถาม (Query) เพื่อแก้ปัญหาข้อจำกัดของโครงสร้าง โดยแปลง SQL เป็นปัญหา SAT โดยใช้แบบจำลองของอัลลอย เริ่มจากการรับไวยากรณ์ของ SQL และแปลงสมการจากเครื่องมืออัลลอย เป็นสมการบูลีน จากนั้นแจกแจงทุกกรณีที่เป็นไปได้จากรูปที่ 2.6 คือ ตัวอย่างก่อนการนำข้อมูลไปสร้างข้อสอบถาม (Query) โดยกำหนดรูปแบบของข้อสอบถาม (Query) ตารางที่ต้องการเรียกใช้งาน สดมภ์ที่ต้องการเลือกใช้ในแต่ละตาราง และชุดคำสั่งที่ต้องการนำมาใช้งาน

```

QUERY ::= SELECT FROM
SELECT ::= 'SELECT' selectTerm+
FROM ::= 'FROM' (table | table JOIN table)
selectTerm ::= term | agg(term)
table ::= 'students' | 'grades'
term ::= 'id' | 'name' | 'studentID' | 'courseID' | 'grade'
agg ::= 'MAX' | 'MIN'

```

รูปที่ 2.6 ตัวอย่างข้อมูลในการสร้างข้อสอบถาม (Query)

ผลลัพธ์ที่ได้ออกมาเป็นไปตามรูปที่ 2.7 ดังนี้

```

SELECT courseID, studentID FROM GRADES, STUDENT;
SELECT MAX (courseID), MAX (NAME) FROM GRADES, STUDENT;
SELECT MIN (courseID), MIN (NAME) FROM GRADES, STUDENT;
SELECT courseID FROM GRADES, STUDENT;
SELECT MAX (courseID), MIN (NAME) FROM GRADES, STUDENT;
SELECT MAX (NAME), MIN (courseID) FROM GRADES, STUDENT;
SELECT courseID, MIN (NAME) FROM GRADES, STUDENT;
SELECT courseID, MAX (NAME) FROM GRADES, STUDENT;
SELECT NAME, MAX (courseID) FROM GRADES, STUDENT;
SELECT NAME FROM STUDENT;
SELECT MIN (NAME) FROM STUDENT;
SELECT id FROM STUDENT;
SELECT MAX (NAME), MIN (id) FROM STUDENT;
SELECT MAX (id), MIN (NAME) FROM STUDENT;
SELECT id, MAX (NAME) FROM STUDENT;
...

```

รูปที่ 2.7 ผลลัพธ์ที่จากการสร้างแบบอัตโนมัติผ่านชุดเครื่องมือ

2.2.4 การสร้างฐานข้อมูลเซตผ่านการดำเนินการสัญลักษณ์แบบไดนามิกสำหรับเกณฑ์ความครอบคลุม

บทความนี้ (Database State Generation via Dynamic Symbolic Execution for Coverage Criteria) [12] ได้นำเสนอการสร้างฐานข้อมูลโดยอัตโนมัติเพื่อลดการทำงานของมนุษย์ โดยใช้วิธีการขอบเขตข้อมูลที่ครอบคลุม (Boundary Value Coverage) ความครอบคลุมของตรรกะ (Logical Coverage) และใช้ Correlated Active Clause Coverage ซึ่งประยุกต์มาจาก Modified Condition/Decision Coverage (MC/DC) เพื่อให้ทุกเงื่อนไขถูกทดสอบด้วยเครื่องมือ DES สำหรับ .NET โดยมีการทดสอบทุกตัวดำเนินการ (=, <>, >, >=, <, <=, IN, NOT IN) ซึ่งจากการทดสอบพบว่าสามารถทดสอบได้ครอบคลุมในกรณีที่มีเงื่อนไขที่ไม่ซับซ้อน

จากงานวิจัยฉบับนี้พบว่าเครื่องมือที่ใช้วิธีการ Boundary Value Coverage ความครอบคลุมของตรรกะ (Logical Coverage) และใช้ Correlated Active Clause Coverage สามารถทำการทดสอบที่ครอบคลุมได้

2.2.5 เครื่องมือสร้างกรณีทดสอบสำหรับโปรแกรมประยุกต์บนเว็บบนพื้นฐานสคีมาของฐานข้อมูล

บทความนี้ [13] ได้นำเสนอเครื่องมือสร้างกรณีทดสอบโดยคำนึงถึงโครงสร้างของข้อมูล โดยกรณีที่ทดสอบจะสร้างครอบคลุมคำสั่งการใช้งานฐานข้อมูลในการเลือกข้อมูล การเพิ่มข้อมูล การลบข้อมูล และการปรับปรุงข้อมูล ซึ่งเครื่องมือนี้จะสร้างกรณีทดสอบตามโครงสร้างของฐานข้อมูล โดยเครื่องมือนี้ใช้สำหรับทดสอบโปรแกรมประยุกต์บนเว็บจากการรับไฟล์เอชทีเอ็มแอล (HTML Document) และเอกซสาร์พีเอชพี (PHP Document) มาวิเคราะห์ความสัมพันธ์และลำดับการทำงานเพื่อหาชุดคำสั่งเอสคิวแอลภายในเอกซสาร์ ฐานข้อมูลที่ใช้สำหรับเครื่องมือถูกสร้างขึ้นด้วยฐานข้อมูลมายเอสคิวแอล (My SQL Database) ซึ่งผลลัพธ์ที่ได้มาแสดงให้เห็นว่าเครื่องมือมีประสิทธิภาพสามารถใช้ในการสร้างกรณีการทดสอบได้จริง

จากการศึกษางานวิจัยชิ้นนี้พบว่ามีข้อจำกัดในการสร้างกรณีทดสอบโดยจะสามารถสร้างจากคำสั่งเอสควิแอลที่มีเงื่อนไขได้ไม่เกิน 2 ตัวดำเนินการ (Operator)

จากบทความวิจัยในข้างต้นแสดงให้เห็นแนวทางในการสร้างกรณีทดสอบกับภาษาเอสควิแอล จากงานวิจัยแนวทางปฏิบัติต่อการทดสอบภาษาเอสควิแอลแบบไวท์บ็อกซ์ วิธีการพัฒนาเครื่องมือตามงานวิจัยการสร้างเอสควิแอลคิวรีอัตโนมัติ สำหรับการทดสอบของเครื่องประมวลผลฐานข้อมูล และงานวิจัยการสร้างฐานข้อมูลเสถียรผ่านการดำเนินการสัญลักษณ์แบบไดนามิกสำหรับเกณฑ์ความครอบคลุม สำหรับงานวิจัยการสร้างเอสควิแอลคิวรีอัตโนมัติได้เสนอระบบ ซึ่งงานวิจัยแรกได้กำหนดรูปแบบข้อสอบถามสำหรับวิเคราะห์ไวยากรณ์ของภาษาเอสควิแอล โครงการมหาบัณฑิตฉบับนี้ก็ได้พัฒนาโดยการวิเคราะห์ไวยากรณ์ของภาษาเอสควิแอลโดยกำหนดตามรูปแบบที่เกิดขึ้นจากการสร้างเงื่อนไขในแคมเปญไว้คล้ายคลึงการงานวิจัยฉบับนี้ และอีกงานวิจัยเป็นการนำเสนอรูปแบบของตัวดำเนินการทางตรรกศาสตร์เข้ามาวิเคราะห์เพิ่มเติม ซึ่งโครงการฉบับนี้ก็นำส่วนนี้มาเพิ่มเติมในไวยากรณ์ของภาษาที่ต้องการใช้ในการวิเคราะห์เพิ่มเติม

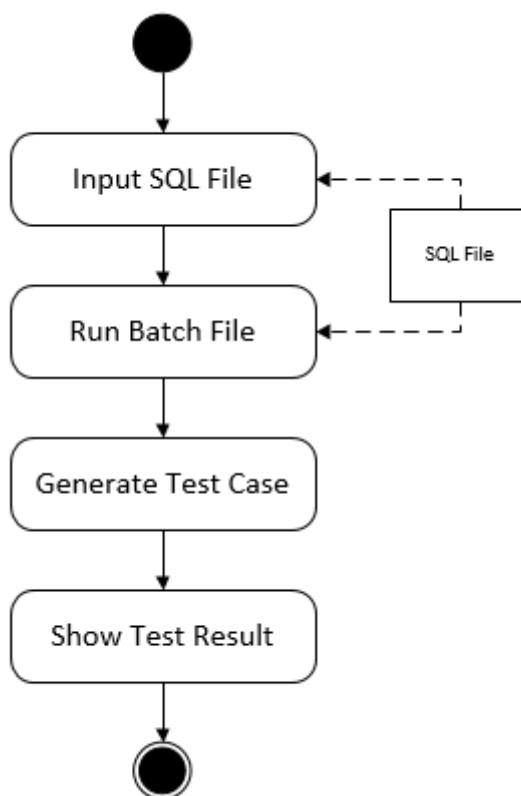
บทที่ 3

แนวคิดและวิธีดำเนินการ

ในบทนี้จะกล่าวถึงรายละเอียดวิธีการที่นำมาใช้ในการพัฒนาระบบ สำหรับการทดสอบเงื่อนไขในแคมเปญการตลาดโดยการทดสอบแบบ white-box ซึ่งเป็นการทดสอบการเดินทางของโปรแกรมไปยังทุก ๆ เส้นทางที่มีโอกาสเกิดขึ้น จึงต้องทำการวิเคราะห์ชุดคำสั่งจากโครงสร้างภาษา SQL ซึ่งเป็นภาษาของชุดคำสั่งที่ใช้ในการสร้างแคมเปญ โดยการใช้ Parser เพื่อหาตัวแปรที่จะนำมาทดสอบชุดคำสั่ง แต่ในการทดสอบเราจะมุ่งเน้นการทดสอบไวท์บ็อกซ์และการทดสอบที่เป็นไปตามความต้องการเงื่อนไขของแคมเปญ

3.1 ภาพรวมการทำงานของระบบการสร้างกรณีทดสอบ

การทำงานของเครื่องมือสำหรับสร้างกรณีการทดสอบนั้น ประกอบด้วยหลายส่วน ดังแสดงในรูปที่ 3.1



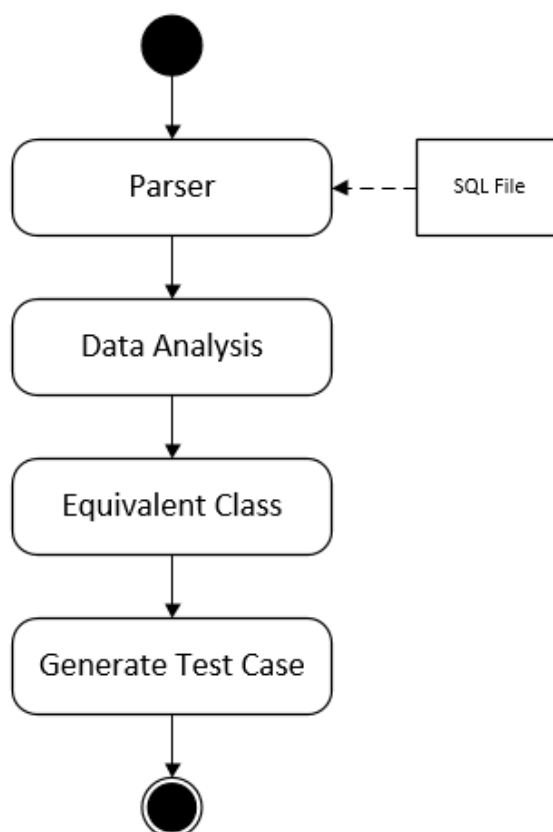
รูปที่ 3.1 แผนภาพแสดงการไหลของระบบงาน

จากรูปที่ 3.1 แสดงภาพรวมในการทำงานของเครื่องมือสร้างกรณีทดสอบแคมเปญสำหรับโครงการฉบับนี้ เริ่มจากการนำไฟล์เอสคิวแอลเข้าไปวิเคราะห์หาโครงสร้างไวยากรณ์ของภาษาเอสคิวแอลเพื่อหาตัวแปรทั้งหมดที่เกิดขึ้นในโปรแกรม และนำตัวแปรที่ได้มาทำการวิเคราะห์หาค่าของตัว

แปรโดยการทำชั้นสมมูลเพื่อให้ได้ค่าตัวแปรที่ตรงตามช่วงของค่า นั้น ๆ ซึ่งจะทำให้ได้การทดสอบที่ครอบคลุมทุกเส้นทางการเดินทางของโปรแกรม เพราะถ้ามีการหาช่วงค่าตัวแปรที่จะนำมาทดสอบทั้งหมดแล้วจะเกิดกรณีการทดสอบจากชั้นสมมูลที่มีการเดินทางไปยังทางเดินของโปรแกรม จากนั้นจึงทำการสร้างกรณีการทดสอบที่ได้จากค่าตัวแปร

3.2 ภาพรวมการทำงาน

การทำงานจะเริ่มจากนำเงื่อนไขมาทำการ Parse จากนั้นทำการวิเคราะห์ข้อมูลให้อยู่ในรูปแบบต้นไม้และทำการสร้างกรณีการทดสอบ ดังรูปที่ 3.2 โดยรายละเอียดของการทำงานมี 4 ขั้นตอนดังนี้



รูปที่ 3.2 แผนภาพกิจกรรมของการทำงาน

3.2.1 ตัวแฉงส่วน

ตัวแฉงส่วน (Parser) คือกระบวนการในการตรวจสอบโครงสร้างของประโยค (Syntax) ของสิ่งที่รับเข้ามา (Input) ว่าถูกต้องตรงตามไวยากรณ์ (Grammar) ที่กำหนด เพื่อทำให้ง่ายและชัดเจนขึ้น ในการพัฒนาระบบมีการวิเคราะห์ตัวแฉงส่วนจากไฟล์ตั้งต้นซึ่งเป็นภาษา SQL นำมาสกัดค่าเพื่อให้ได้บริบทของค่าออกมา และนำข้อมูลมาวิเคราะห์หาค่าสำหรับนำมาใช้ในการทำกรณีทดสอบ

โดยโครงสร้างของภาษา SQL ที่นำมาใช้เป็นไปตามรูปที่ 3.3 เป็นส่วนย่อยของ SQL ที่อ้างมาจากแคมเปญการตลาดที่นำมาใช้ในการวิจัยนี้เท่านั้น โดยประกอบไปด้วยเงื่อนไข (clause) 3 รูปแบบคือการนำเงื่อนไขมาเชื่อมด้วย and, or และเงื่อนไขที่อยู่ภายใต้วงเล็บ ซึ่งในแต่ละเงื่อนไขจะมีรูปแบบคำสั่งที่ประกอบด้วยตัวแปร (name) ตัวดำเนินการ (operator) และค่าของตัวแปร (value) ที่สนใจในแต่ละเงื่อนไข

```

clause := clause and clause
clause := clause or clause
clause := ( clause )

clause := name operator value
clause := name like 'string'
clause := name not like 'string'
clause := name between value and value
clause := name not between value and value
clause := name in list
clause := name not in list
clause := name is null
clause := name is not null

name := iden | fun ( iden )
operator := = | <> | < | <= | =>
list := ( values )
values := value | value , values
value := number | 'string' | fun list

iden identifier
fun function name
    only two functions: upper( ) and to_date('date-string','dd-mm-yyyy')

```

รูปที่ 3.3 ไวยากรณ์ภาษา SQL

3.2.2 การวิเคราะห์หาค่าของตัวแปร

ในการวิเคราะห์หาค่าตัวแปรจะมีการทดสอบผลลัพธ์โดยกำหนดผลลัพธ์จากการวิเคราะห์เป็น 2 รูปแบบคือ ได้ผลลัพธ์ที่ถูกต้องซึ่งจะแทนด้วย "VALID" และ ไม่ถูกต้อง แทนด้วย "INVALID"

AND คือ ตรรกะที่จะให้ผลลัพธ์เป็น VALID ในกรณีเดียวเท่านั้น คือ เมื่อข้อมูลนำเข้าในการทดสอบทั้งสองค่าเป็น VALID โดยมีรูปแบบตรรกะดังตารางที่ 3.1

ตารางที่ 3.1 ผลลัพธ์ที่จากการวิเคราะห์ไวยากรณ์ของภาษาเอสคิวแอลเมื่อใช้ตัวดำเนินการ AND

ข้อมูลนำเข้าในการทดสอบค่าที่ 1	ตัวดำเนินการทางตรรกศาสตร์	ข้อมูลนำเข้าในการทดสอบค่าที่ 2	ผลลัพธ์
VALID	AND	VALID	VALID
VALID	AND	INVALID	INVALID
INVALID	AND	VALID	INVALID
INVALID	AND	INVALID	INVALID

OR คือ ตรรกะที่จะให้ผลลัพธ์เป็น INVALID ในกรณีเดียวกันนั้น คือ เมื่อข้อมูลอินพุตทั้งสองเป็น INVALID โดยมีรูปแบบตรรกะดังตารางที่ 3.2

ตารางที่ 3.2 ผลลัพธ์ที่จากการวิเคราะห์ไวยากรณ์ของภาษาเอสคิวแอลเมื่อใช้ตัวดำเนินการ OR

ข้อมูลนำเข้าในการทดสอบค่าที่ 1	ตัวดำเนินการทางตรรกศาสตร์	ข้อมูลนำเข้าในการทดสอบค่าที่ 2	ผลลัพธ์
VALID	OR	VALID	VALID
VALID	OR	INVALID	VALID
INVALID	OR	VALID	VALID
INVALID	OR	INVALID	INVALID

การวิเคราะห์หาค่าตัวแปรจากการทำชั้นสมมูล (Equivalence Class) เพื่อหาช่วงของข้อมูลที่จะครอบคลุมตัวแปรซึ่งช่วยลดกรณีการทดสอบ โดยจากการวิเคราะห์ข้อมูลของตัวแปรจากค่าถูกต้อง (VALID) จากนั้นจึงหาค่าที่ไม่ถูกต้อง (INVALID) จากข้อมูลของตัวแปรที่มี ตัวอย่างเช่น (table1.amount >= 500 AND table1.currency_cd = '764') OR (table1.amount >= 10000 AND table1.currency_cd <> '764') ตัวแปร table1.amount ได้แบ่งชั้นสมมูลออกเป็น 2 ชั้นคือ table1.amount >= 500 และ table1.amount >= 10000 ซึ่งทั้ง 2 ชั้นถูกกำหนดมาอยู่แล้ว จึงถือว่าทั้งสองชั้นเป็นชั้นที่ถูกต้อง (VALID) เมื่อได้ชั้นที่ถูกต้องแล้วจะทำการหาชั้นที่ไม่ถูกต้องจากทั้งสองชั้น ซึ่งจะได้ชั้น table1.amount < 500 เป็นชั้นที่ไม่ถูกต้อง (INVALID) ของทั้งสองชั้นพอดี และใช้วิธีการนี้หาชั้นที่ถูกต้องและไม่ถูกต้องของ table1.currency_cd ได้เหมือนกันโดยสามารถแบ่งค่าตามชั้นสมมูลแต่ละตัวแปรตามตารางที่ 3.3 ดังนี้

ตารางที่ 3.3 ผลลัพธ์ที่จากการวิเคราะห์ไวยากรณ์ของภาษาเอสคิวแอล

Column	No	Equivalence Class	Value	Type
table1.amount	1	amount >= 500	500	Valid
	2	amount >= 10000	10000	Valid

Column	No	Equivalence Class	Value	Type
table1.currency_cd	1	currency_cd = '764'	'764'	Valid
	2	currency_cd <> '764'	'765'	Valid

3.2.3 การสร้างกรณีทดสอบ

การสร้างกรณีทดสอบแบ่งเป็นกรณีที่ถูกต้อง VALID และกรณีที่ไม่ถูกต้อง INVALID โดยมีตัวอย่างดังตารางที่ 3.4

ตารางที่ 3.4 แสดงตัวอย่างกรณีที่ถูกต้องและไม่ถูกต้อง

ลำดับ	ข้อมูลที่ทดสอบ	กรณีทดสอบที่ถูกต้อง (VALID)	กรณีทดสอบที่ไม่ถูกต้อง (INVALID)
1	A = 500	A = 500	A = 501
2	A = 'abc'	A = 'abc'	A = 'xyz'
3	A <> 500	A <> 500	A = 500
4	A >= 500	A >= 500	A < 500
5	A > 500	A > 500	A <= 500
6	A <= 500	A <= 500	A > 500
7	A < 500	A < 500	A >= 500
8	A in ('10','20')	A in ('10','20')	A not in ('10','20')
9	A not in ('10','20')	A not in ('10','20')	A in ('10','20')
10	A is null	A is null	A is not null
11	A is not null	A is not null	A is null
12	A like 'abc'	A like 'abc'	A not like 'abc'
13	A not like 'abc'	A not like 'abc'	A like 'abc'
14	A between date1 and date2	A between date1 and date2	A not between date1 and date2
15	A not between date1 and date2	A not between date1 and date2	A between date1 and date2

3.2.4 ขั้นตอนการสร้างกรณีทดสอบ

หลักการวิเคราะห์ภาษาเอสคิวแอลโดยใช้ Parser สามารถนำมาสรุปเป็นวิธีการในการวิเคราะห์ได้ดังนี้

1. เมื่อผู้ใช้วางไฟล์ประเภทเอสคิวแอลในตำแหน่งที่จัดเก็บของแฟ้มข้อมูลที่กำหนด ระบบจะส่งเงื่อนไขไปวิเคราะห์ด้วยวิธีการ Parser ตัวอย่างเช่น

```
select *
  from table1
 where table1.amount = 500 and table1.merchant_name like 'THAI%'
```

เงื่อนไขที่สนใจคือ table1.amount = 500 and table1.merchant_name like 'THAI%'

2. จากนั้นตัว Parser จะทำการแยกเงื่อนไขมาจำแนกเป็นโครงสร้างต้นไม้ โดยเริ่มค้นหาจากเครื่องหมายตรรกะได้แก่ and และ or ซึ่งเงื่อนไขที่เหลือจะเรียกว่านิพจน์ (expression) จากตัวอย่างในข้อที่ 1 ตัว Parser จะจำแนก expression เป็น A AND B โดย expression A คือ table1.amount = 500 และ expression B คือ table1.merchant_name like 'THAI%'

3. หลังจากนั้นตัว Parser จะทำการแยก expression มาจำแนกเป็นโครงสร้างต้นไม้ โดยค้นหาจากตัวดำเนินการต่าง ๆ ของภาษาเอสคิวแอล เช่น เครื่องหมายทางเปรียบเทียบได้แก่

เครื่องหมาย < (น้อยกว่า)

เครื่องหมาย > (มากกว่า)

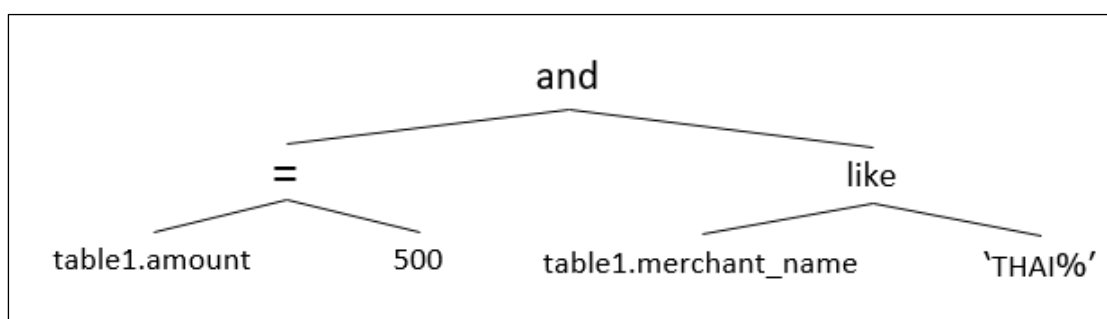
เครื่องหมาย = (เท่ากับ),

เครื่องหมาย <= (น้อยกว่าหรือเท่ากับ)

เครื่องหมาย >= (มากกว่าหรือเท่ากับ)

เครื่องหมาย < > (ไม่เท่ากับ)

และตัวดำเนินการของภาษาเอสคิวแอลได้แก่ is, in, like, between นำมาจำแนกเพื่อค้นหาพารามิเตอร์และค่าของพารามิเตอร์นั้น จากเงื่อนไขตัวอย่างในข้อ 1 โปรแกรมจะสามารถจำแนกเป็นพารามิเตอร์ได้ จะพบว่า พารามิเตอร์คือ table1.amount และ table1.merchant_name ส่วนค่าของพารามิเตอร์คือ 500 และ 'THAI%' ในรูปแบบโครงสร้างต้นไม้ดังรูปที่ 3.4



รูปที่ 3.4 แผนภาพต้นไม้ที่จากการวิเคราะห์

4. เมื่อโปรแกรมแยกพารามิเตอร์และค่าของพารามิเตอร์ได้แล้วนั้น โปรแกรมจะสร้างกรณีทดสอบย่อยและผลลัพธ์ที่คาดหวังโดยใช้วิธีหาชั้นสมมูล (Equivalent Class) ซึ่งจะเลือกวิธีการสร้างจากค่าของตัวดำเนินการ โดย 1 expression จะสร้างได้เป็น 2 กรณีทดสอบย่อยคือ กรณีทดสอบที่เป็นจริง และกรณีทดสอบที่ไม่เป็นจริง จากตัวอย่างในข้อ 1 สามารถสร้างกรณีทดสอบย่อยได้ดังรูปที่ 3.5 และสามารถอธิบายเพิ่มเติมได้ตารางที่ 3.5 และ ตารางที่ 3.6

พารามิเตอร์: table1.amount
เครื่องหมาย: =
ค่าของพารามิเตอร์: 500
พารามิเตอร์: table1.merchant_name
เครื่องหมาย: like
ค่าของพารามิเตอร์: 'THAI%'

รูปที่ 3.5 พารามิเตอร์และค่าของพารามิเตอร์จากโปรแกรม

ตารางที่ 3.5 ตัวอย่างการสร้างกรณีทดสอบย่อยของพารามิเตอร์ที่ 1

ลำดับ	กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง
1	table1.amount = 500	Valid
2	table1.amount = 501	Invalid

ตารางที่ 3.6 ตัวอย่างการสร้างกรณีทดสอบย่อยของพารามิเตอร์ที่ 2

ลำดับ	กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง
1	table1.merchant_name like 'THAI%'	Valid
2	table1.merchant_name not like 'THAI%'	Invalid

5. เมื่อโปรแกรมได้กรณีทดสอบย่อยแล้ว โปรแกรมจะทำการสร้างกรณีทดสอบหลักของเงื่อนไขทั้งหมดโดยการนำกรณีทดสอบย่อยมาเชื่อมต่อกันทั้งหมดด้วยเครื่องหมายตรรกะของเงื่อนไขนั้น เพื่อให้ได้ออกมาเป็นกรณีทดสอบหลักและผลลัพธ์ที่คาดหวังตามวิธีการทำกรณีการทดสอบชั้นสมมูลแบบสตรองโรบัสต์ จากเงื่อนไข table1.amount = 500 and table1.merchant_name like 'THAI%' โปรแกรมจะสร้างกรณีทดสอบหลักและผลลัพธ์ที่คาดหวังได้ตามตารางที่ 3.7

ตารางที่ 3.7 ตัวอย่างการสร้างกรณีทดสอบชั้นสมมูลแบบสตรองโรบัสต์

ลำดับ	กรณีทดสอบ	ผลลัพธ์ที่คาดหวัง
1	table1.amount = 501 table1.merchant_name not like 'THAI%'	Invalid
2	table1.amount = 501 table1.merchant_name like 'THAI%'	Invalid
3	table1.amount = 500 table1.merchant_name not like 'THAI%'	Invalid
4	table1.amount = 500 table1.merchant_name like 'THAI%'	Valid

3.3 การออกแบบส่วนต่อประสานผู้ใช้งาน

เครื่องมือถูกออกแบบมาให้ใช้งานผ่านการทำงานในรูปแบบของตัวอักษร (Text Mode) บนระบบปฏิบัติการแบบที่ใช้จานบันทึกข้อมูล (Disk Operating System: DOS) ซึ่งในโครงการมหาบัณฑิตฉบับนี้ เครื่องมือถูกเรียกใช้งานผ่านโปรแกรม Command Prompt โดยสามารถอธิบายส่วนการทำงานที่ใช้เงื่อนไขส่วนนำเข้าตามรูปที่ 3.6 และส่วนแสดงผลซึ่งสามารถแบ่งออกเป็น 4 ส่วนได้ตามรูปที่ 3.7 ดังนี้

```
(TRN_HIS.MKT_CD = '3456'
or (TRN_HIS.MERCHANT_TYPE = '5944'
and TRN_HIS.MERCHANT_NAME not like '%GOLD%'))
```

รูปที่ 3.6 เงื่อนไขที่ใช้เป็นส่วนนำเข้าของข้อมูล

```

Select Command Prompt
D:\>cd D:\AON\ProgramTest\sqltest-2\test
D:\AON\ProgramTest\sqltest-2\test>sqltest D:\AON\Test_Files\test.txt
(or (= TRN_HIS.MKT_CD 3456 )(and (= TRN_HIS.MERCHANT_TYPE '5944' )(not like TRN_HIS.MERCHANT_NAME '%GOLD%' )))
asso table 3 entries
key 0 value (TRN_HIS.MKT_CD = 3456 ) (TRN_HIS.MKT_CD = 3457 )
key 1 value (TRN_HIS.MERCHANT_TYPE = '5944' ) (TRN_HIS.MERCHANT_TYPE = '456' )
key 2 value (TRN_HIS.MERCHANT_NAME not like '%GOLD%' ) (TRN_HIS.MERCHANT_NAME like '%GOLD%' )

logic table
0 0 0 0 0
0 0 1 0 0
0 1 0 0 0
0 1 1 1 1
1 0 0 0 1
1 0 1 0 1
1 1 0 0 1
1 1 1 1 1

case 0 expect i
(TRN_HIS.MKT_CD = 3457 ) i
(TRN_HIS.MERCHANT_TYPE = '456' ) i
(TRN_HIS.MERCHANT_NAME like '%GOLD%' ) i
case 1 expect i
(TRN_HIS.MKT_CD = 3457 ) i
(TRN_HIS.MERCHANT_TYPE = '456' ) i
(TRN_HIS.MERCHANT_NAME not like '%GOLD%' ) v
case 2 expect i
(TRN_HIS.MKT_CD = 3457 ) i
(TRN_HIS.MERCHANT_TYPE = '5944' ) v
(TRN_HIS.MERCHANT_NAME like '%GOLD%' ) i
case 3 expect v
(TRN_HIS.MKT_CD = 3457 ) i
(TRN_HIS.MERCHANT_TYPE = '5944' ) v
(TRN_HIS.MERCHANT_NAME not like '%GOLD%' ) v
case 4 expect v
(TRN_HIS.MKT_CD = 3456 ) v
(TRN_HIS.MERCHANT_TYPE = '456' ) i
(TRN_HIS.MERCHANT_NAME like '%GOLD%' ) i
case 5 expect v
(TRN_HIS.MKT_CD = 3456 ) v
(TRN_HIS.MERCHANT_TYPE = '456' ) i
(TRN_HIS.MERCHANT_NAME not like '%GOLD%' ) v
case 6 expect v
(TRN_HIS.MKT_CD = 3456 ) v
(TRN_HIS.MERCHANT_TYPE = '5944' ) v
(TRN_HIS.MERCHANT_NAME like '%GOLD%' ) i
case 7 expect v
(TRN_HIS.MKT_CD = 3456 ) v
(TRN_HIS.MERCHANT_TYPE = '5944' ) v
(TRN_HIS.MERCHANT_NAME not like '%GOLD%' ) v

```

รูปที่ 3.7 หน้าจอแสดงผลลัพธ์ในการสร้างกรณีทดสอบ

1. ส่วนการเรียกใช้งานเครื่องมือผ่านโปรแกรม Command Prompt

เมื่อเปิดโปรแกรม Command Prompt มาแล้ว ให้ใช้คำสั่ง CD (Change Directory) เพื่อให้ระบบชี้ไปยังตำแหน่งที่วางเครื่องมือ จากนั้นจึงระบุชื่อของเครื่องมือ (sqltest) วันวรรคหนึ่งครั้ง และตามด้วยส่วนนำเข้าโดยจะป้อนตำแหน่งที่วางแฟ้มข้อมูลและชื่อของแฟ้มข้อมูลที่ต้องการให้วิเคราะห์ผล เมื่อป้อนข้อมูลเรียบร้อยแล้วจึงจะกดปุ่มเอ็นเทอร์เพื่อสั่งให้เครื่องมือทำงาน ตามตัวอย่างรูปที่ 3.7 ดังนี้

```

รูปแบบ:
D:\>cd [Tool Path]

D:\[Tool Path]>sqltest [File Path]\[File Name]

ตัวอย่าง:
D:\>cd D:\AON\ProgramTest\sqltest-2\test

D:\AON\ProgramTest\sqltest-2\test>sqltest D:\AON\Test_Files\test.txt

```

รูปที่ 3.8 ตัวอย่างการเรียกใช้งานเครื่องมือ

2. ส่วนการแสดงผลการวิเคราะห์ค่าตัวแปรของเครื่องมือ

ในส่วนนี้จะเป็นการแสดงผลที่ได้จากการวิเคราะห์ของเครื่องมือ ซึ่งบรรทัดแรกจะแสดงผลที่ได้จากการวิเคราะห์ส่วนข้อมูลนำเข้าซึ่งจัดอยู่ในรูปแบบโครงสร้างต้นไม้ เพื่อให้ทราบถึงส่วนประกอบที่เป็นตัวดำเนินการ พารามิเตอร์ และค่าของพารามิเตอร์ และในบรรทัดถัดมาจะแสดงผลของเงื่อนไขสำหรับพารามิเตอร์แต่ละตัวที่เป็นแบบเงื่อนไขที่เป็นจริงและเงื่อนไขที่เป็นเท็จ โดยที่ Key 0 หมายถึงพารามิเตอร์ตัวที่ 1 Key 1 หมายถึงพารามิเตอร์ตัวที่ 2 และ Key 2 หมายถึงพารามิเตอร์ตัวที่ 3

3. ส่วนการแสดงตารางตรรกศาสตร์ที่ได้จากการวิเคราะห์ของเครื่องมือ

ตารางจะถูกกำหนดไว้ให้มีขนาดกว้าง 5 แถว 8 บรรทัดโดยสามแถวแรกจะแทนค่าพารามิเตอร์ทั้ง 3 ตัว แถวที่สี่แสดงผลของการดำเนินการทางตรรกศาสตร์ของ Key 1 และ Key 2 แสดงผลของการดำเนินการทางตรรกศาสตร์ของ Key 1 และ Key 2 และแถวสุดท้ายเป็นการแสดงผลของการดำเนินการทางตรรกศาสตร์ของ Key 0 และผลที่ได้จากแถวที่สี่ โดยที่ 0 จะแทนค่าที่เป็นเท็จ และ 1 จะแทนค่าที่เป็นจริง ซึ่งจะมีรูปแบบเป็นไปตามตารางที่ 3.8 โดยจะให้ค่าเริ่มต้นบรรทัดแรกเป็นกรณี 0 (Case 0) และเรียงตามลำดับไปจนถึง 7 ดังนี้

ตารางที่ 3.8 ตารางตรรกศาสตร์ที่ได้จากการวิเคราะห์ของเครื่องมือ

	Key 0	Key 1	Key 2	(Key 1 and Key 2)	(Key 0 or (Key 1 and Key 2))
Case 0	0	0	0	0	0
Case 1	0	0	1	0	0
Case 2	0	1	0	0	0
Case 3	0	1	1	1	1
Case 4	1	0	0	0	1
Case 5	1	0	1	0	1
Case 6	1	1	0	0	1
Case 7	1	1	1	1	1

4. ส่วนการแสดงผลในการแทนค่าพารามิเตอร์ที่ได้จากการวิเคราะห์ของเครื่องมือ

ในส่วนนี้จะเป็นการแสดงกรณีทดสอบที่จะเกิดขึ้น โดยได้ผลที่คาดหวัง (Expected Result) จากแถวสุดท้ายในตารางที่ได้ในส่วนก่อนหน้านี และเงื่อนไขของพารามิเตอร์แต่ละตัวที่จากส่วนที่สองจากการวิเคราะห์หาเงื่อนไขจริงและเท็จ โดยผลที่ได้ออกมาจะใช้อักษรภาษาอังกฤษ v ตัวเล็ก แทนค่าที่เป็นจริง (Valid) และอักษรภาษาอังกฤษ i ตัวเล็ก แทนค่าที่เป็นเท็จ (Invalid) โดยจากรูป 3.6 ส่วนที่ 4 สามารถอธิบายบรรทัดแรกได้ว่า Case 0 ได้ผลที่คาดหวังเป็นเท็จ (i) ซึ่งประกอบไปด้วยเงื่อนไข (TRN_HIS.MKT_CD = 3457) ที่เป็นเท็จ (TRN_HIS.MERCHANT_TYPE = '456') ที่เป็นเท็จ และ (TRN_HIS.MERCHANT_NAME like '%GOLD%') ที่เป็นเท็จ

บทที่ 4

การทดลอง

ในบทนี้จะกล่าวถึงรายละเอียดในการทดลอง เพื่อประเมินประสิทธิภาพระบบสำหรับการทดสอบเงื่อนไขในแคมเปญการตลาด โดยแบ่งออกเป็น 4 ส่วน คือ ส่วนแรกเป็นสภาพแวดล้อมที่ใช้ในการทดลอง ส่วนข้อมูลที่นำมาใช้ในการทดลอง ส่วนผลที่ได้จากการทดลอง และสรุปผลการทดลอง

4.1 สภาพแวดล้อมที่ใช้ทดสอบ

เพื่อให้สภาพแวดล้อมในการทดสอบเงื่อนไขในแคมเปญตลาดมีความใกล้เคียงกับระบบที่ทำงานจริงโครงการนี้จึงได้กำหนด สภาพแวดล้อมทั้ง ฮาร์ดแวร์ที่ใช้ในการทดสอบและซอฟต์แวร์ที่ใช้ในการทดสอบ ดังนี้

1. ฮาร์ดแวร์ที่ใช้ในการทดสอบ

เครื่องคอมพิวเตอร์ที่ใช้พัฒนาระบบ

- 1) หน่วยประมวลผล อินเทล คอร์ไอ 7-5500 ความเร็ว 2.40 กิกะเฮิรท์
- 2) หน่วยความจำ ดีดีย์อาร์ 3 ขนาด 8.00 กิกะไบต์
- 3) ฮาร์ดดิสก์ ความจุ 1 เทราไบต์

2. ซอฟต์แวร์ที่ใช้ในการทดสอบ

- 1) ระบบปฏิบัติการ
 - ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 7 (Microsoft Window 7)
- 2) เครื่องมือเสริมที่ใช้ในการพัฒนาเครื่องมือ
 - วิซวล สตูดิโอ 2015

4.2 ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลองครั้งนี้เป็นเงื่อนไขในการออกแคมเปญทางการตลาดสำหรับจับรายการการใช้จ่ายของกลุ่มลูกค้า โดยสุ่มเงื่อนไขแคมเปญออกมา 10 เงื่อนไขที่เป็นพื้นฐานในการสร้างแคมเปญที่เกิดขึ้น ซึ่งจะมีความยากง่ายที่แตกต่างกันออกไปตามแต่ละเงื่อนไขของแคมเปญที่ต้องการตั้งค่าไว้ ดังนี้ มีตัวแปรไม่เกิน 3 ตัวแปร 2 ตัวดำเนินการ และตัวอย่างชุดนี้ครอบคลุมไวยากรณ์ตามรูปที่ 3.3 ในบทที่ 3

4.3 ผลการทดสอบ

จากการทดสอบการสร้างแคมเปญนั้นได้ผลที่มีรูปแบบที่คล้ายคลึงกันซึ่งจะยกตัวอย่าง ผลการสอบที่แบ่งเป็นกลุ่ม จำนวน 10 ตัวอย่าง โดย กำหนดให้

Campaign ID:	รหัสแคมเปญ
TRN_HIS:	ชื่อของตาราง Transaction_History ที่ถูกย่อเอาไว้เพื่อให้ข้อความสั้นลง
AMOUNT:	สดมภ์ที่เก็บยอดการใช้จ่าย
CURRENCY_CD:	คือ สดมภ์ที่เก็บรหัสเงินตรา (Currency Code) โดย 764 เป็นรหัสของสกุลเงินบาท โดยสกุลเงินอื่นๆ จะอธิบายเพิ่มเติมไว้ในภาคผนวก ก
MERCHANT_TYPE:	สดมภ์ที่เก็บรหัสประเภทร้านค้า (Merchant Category Code) โดยอธิบายเพิ่มเติมไว้ในภาคผนวก ข
MERCHANT_CD:	สดมภ์ที่เก็บรหัสร้านค้า (Merchant Code) ตามที่ภาคธุรกิจกำหนดไว้
MKT_CD:	สดมภ์ที่เก็บรหัสทางการตลาด (Marketing Code) ที่ภาคธุรกิจกำหนดไว้สำหรับให้แต่ละร้านเข้าร่วมโปรโมชั่น
TRN_DATE:	สดมภ์ที่เก็บวันที่ทำรายการ (Transaction Date)

ผลการทดสอบมีดังต่อไปนี้

1. Campaign ID: CM001

Condition: TRN_HIS.AMOUNT = 500 and TRN_HIS.CURRENCY_CD = '764'

(จากรหัสภาคผนวก ข CURRENCY = 764 คือ สกุลเงินบาท)

คำอธิบายเงื่อนไข: รายการใช้จ่ายที่มียอดเงินเท่ากับ 500 และเป็นสกุลเงินบาท

Expression ที่ได้

A	= (TRN_HIS.AMOUNT = 500)	--- (Valid)
A'	= (TRN_HIS.AMOUNT = 501)	--- (Invalid)
B	= (TRN_HIS.CURRENCY_CD = '764')	--- (Valid)
B'	= (TRN_HIS.CURRENCY_CD = '456')	--- (Invalid)

ตารางที่ 4.1 ผลการทดลองของ Campaign ID : CM001

Test ID	Test Case	Expected Result
1	A' and B'	Invalid
2	A' and B	Invalid

Test ID	Test Case	Expected Result
3	A and B'	Invalid
4	A and B	Valid

2. Campaign ID: CM002

Condition: (TRN_HIS.AMOUNT = 1000 or TRN_HIS.MKT_CD = '3456')

คำอธิบายเงื่อนไข : รายการใช้จ่ายที่มียอดเงิน 1,000 หรือ รายการที่มีรหัสส่งเสริมการขายเท่ากับ 3456

Expression ที่ได้

A = (TRN_HIS.AMOUNT = 1000) --- (Valid)
 A' = (TRN_HIS.AMOUNT = 1001) --- (Invalid)
 B = (TRN_HIS.MKT_CD = '3456') --- (Valid)
 B' = (TRN_HIS.MKT_CD = '456') --- (Invalid)

ตารางที่ 4.2 ผลการทดลองของ Campaign ID : CM002

Test ID	Test Case	Expected Result
1	A' or B'	Invalid
2	A' or B	Valid
3	A or B'	Valid
4	A or B	Valid

3. Campaign ID: CM003

Condition: TRN_HIS.MERCHANT_TYPE is not null
 and TRN_HIS.MERCHANT_CD not in ('0022','0000500')

คำอธิบายเงื่อนไข : รายการที่ประเภทร้านค้าไม่มีค่าว่าง และรายการการที่ไม่ใช่รหัสร้านค้า '0022' หรือ '0000500'

Expression ที่ได้

A = (TRN_HIS.MERCHANT_TYPE is not null) --- (Valid)
 A' = (TRN_HIS.MERCHANT_TYPE is null) --- (Invalid)
 B = (TRN_HIS.MERCHANT_CD not in ('0022','0000500')) --- (Valid)

B' = (TRN_HIS.MERCHANT_CD in ('0022','0000500')) --- (Invalid)

ตารางที่ 4.3 ผลการทดลองของ Campaign ID : CM003

Test ID	Test Case	Expected Result
1	A' and B'	Invalid
2	A' and B	Invalid
3	A and B'	Invalid
4	A and B	Valid

4. Campaign ID: CM004

Condition: (TRN_HIS.MERCHANT_TYPE <> '5621'
and TRN_HIS.MERCHANT_CD = '00050019'
and TRN_HIS.MERCHANT_NAME = ""Home"")

คำอธิบายเงื่อนไข: รายการที่ประเภทร้านค้าไม่ใช่ประเภท 5621 โดยมีรหัสร้านค้า '00050019' และมีชื่อร้านค้าเท่ากับ "Home"

Expression ที่ได้

A = (TRN_HIS.MERCHANT_TYPE <> '5621') --- (Valid)
A' = (TRN_HIS.MERCHANT_TYPE = '5621') --- (Invalid)
B = (TRN_HIS.MERCHANT_CD = '00050019') --- (Valid)
B' = (TRN_HIS.MERCHANT_CD = '456') --- (Invalid)
C = (TRN_HIS.MERCHANT_NAME = ""Home"") --- (Valid)
C' = (TRN_HIS.MERCHANT_NAME = 'xyz') --- (Invalid)

ตารางที่ 4.4 ผลการทดลองของ Campaign ID : CM004

Test ID	Test Case	Expected Result
1	A' and B' and C'	Invalid
2	A' and B' and C	Invalid
3	A' and B and C'	Invalid
4	A' and B and C	Invalid
5	A and B' and C'	Invalid
6	A and B' and C	Invalid
7	A and B and C'	Invalid

Test ID	Test Case	Expected Result
8	<i>A and B and C</i>	Valid

5. Campaign ID: CM005

Condition: ((TRN_HIS.AMOUNT < 5001
and TRN_HIS.CURRENCY_CD = '764')
or TRN_HIS.MERCHANT_CD in ('00519','00125'))

คำอธิบายเงื่อนไข: รายการใช้จ่ายที่มียอดเงินน้อยกว่า 5,001 และเป็นสกุลเงินบาท หรือ รายการการใช้จ่ายที่มีรหัสร้านค้า '00050019' และ '00125'

Expression ที่ได้

A = (TRN_HIS.AMOUNT < 5001) --- (Valid)
A' = (TRN_HIS.AMOUNT >= 5001) --- (Invalid)
B = (TRN_HIS.CURRENCY_CD = '764') --- (Valid)
B' = (TRN_HIS.CURRENCY_CD = '456') --- (Invalid)
C = (TRN_HIS.MERCHANT_CD in ('00519','00125')) --- (Valid)
C' = (TRN_HIS.MERCHANT_CD not in ('00519','00125')) --- (Invalid)

ตารางที่ 4.5 ผลการทดลองของ Campaign ID : CM005

Test ID	Test Case	Expected Result
1	<i>(A' and B') or C'</i>	Invalid
2	<i>(A' and B) or C'</i>	Invalid
3	<i>(A and B') or C'</i>	Invalid
4	<i>(A and B) or C'</i>	Valid
5	<i>(A' and B') or C</i>	Valid
6	<i>(A' and B) or C</i>	Valid
7	<i>(A and B') or C</i>	Valid
8	<i>(A and B) or C</i>	Valid

6. Campaign ID: CM006

Condition: (TRN_HIS.AMOUNT >= 10000
 or TRN_HIS.CURRENCY_CD <> '764'
 or TRN_HIS.MERCHANT_TYPE = '5621')

คำอธิบายเงื่อนไข: รายการใช้จ่ายที่มียอดเงินตั้งแต่ 10,000 ขึ้นไป หรือไม่ใช่รายการที่ใช้สกุลเงินบาท หรือ รายการที่ประเภทร้านค้าไม่ใช่ประเภท 5621

Expression ที่ได้

A = (TRN_HIS.AMOUNT >= 10000) --- (Valid)
 A' = (TRN_HIS.AMOUNT < 10000) --- (Invalid)
 B = (TRN_HIS.CURRENCY_CD <> '764') --- (Valid)
 B' = (TRN_HIS.CURRENCY_CD = '764') --- (Invalid)
 C = (TRN_HIS.MERCHANT_TYPE = '5621') --- (Valid)
 C' = (TRN_HIS.MERCHANT_TYPE = '456') --- (Invalid)

ตารางที่ 4.6 ผลการทดลองของ Campaign ID : CM006

Test ID	Test Case	Expected Result
1	A' or B' or C'	Invalid
2	A' or B' or C	Valid
3	A' or B or C'	Valid
4	A' or B or C	Valid
5	A or B' or C'	Valid
6	A or B' or C	Valid
7	A or B or C'	Valid
8	A or B or C	Valid

7. Campaign ID: CM007

Condition: TRN_HIS.MERCHANT_CD in ('00519','00125')
 and (TRN_HIS.TRN_DATE between to_date('01-12-2015','DD-MM-YYYY')
 and to_date('10-12-2015','DD-MM-YYYY')
 or TRN_HIS.TRN_DATE between to_date('20-12-2015','DD-MM-YYYY')
 and to_date('31-12-2015','DD-MM-YYYY'))

คำอธิบายเงื่อนไข: รายการการใช้จ่ายที่มีรหัสร้านค้า '00519'หรือ '00125' และเป็นรายการที่ทำรายการตั้งแต่วันที่ 1 ถึง 10 ธันวาคม 2015 หรือ ตั้งแต่วันที่ 20 ถึง 31 ธันวาคม 2015

Expression ที่ได้

A = (TRN_HIS.MERCHANT_CD in ('00519','00125')) --- (Valid)
 A' = (TRN_HIS.MERCHANT_CD not in ('00519','00125')) --- (Invalid)
 B = (TRN_HIS.TRN_DATE between (to_date('01-12-2015','DD-MM-YYYY')) and (to_date('10-12-2015','DD-MM-YYYY'))) --- (Valid)
 B' = (TRN_HIS.TRN_DATE not between (to_date('01-12-2015','DD-MM-YYYY')) and (to_date('10-12-2015','DD-MM-YYYY'))) --- (Invalid)
 C = (TRN_HIS.TRN_DATE between (to_date('20-12-2015','DD-MM-YYYY')) and (to_date('31-12-2015' 'DD-MM-YYYY'))) --- (Valid)
 C' = (TRN_HIS.TRN_DATE between (to_date('20-12-2015','DD-MM-YYYY')) and (to_date('31-12-2015' 'DD-MM-YYYY'))) --- (Invalid)

ตารางที่ 4.7 ผลการทดลองของ Campaign ID : CM007

Test ID	Test Case	Expected Result
1	A' and (B' or C')	Invalid
2	A' and (B' or C)	Invalid
3	A' and (B or C')	Invalid
4	A' and (B or C)	Invalid
5	A and (B' or C')	Invalid
6	A and (B' or C)	Valid
7	A and (B or C')	Valid
8	A and (B or C)	Valid

8. Campaign ID: CM008

Condition: (TRN_HIS.MERCHANT_TYPE not in ('5621','6300','6211')
 or (TRN_HIS.MERCHANT_TYPE <> '5621'
 or TRN_HIS.MERCHANT_NAME not like 'TOP%'))

คำอธิบายเงื่อนไข: รายการที่ประเภทร้านค้าไม่ใช่ 5621, '6300' และ '6211' โดยที่ประเภทร้านค้า '5621' จะไม่ใช่ชื่อที่ขึ้นต้นด้วย TOP

Expression ที่ได้

A = (TRN_HIS.MERCHANT_TYPE not in ('5621','6300','6211')) --- (Valid)
 A' = (TRN_HIS.MERCHANT_TYPE in ('5621','6300','6211')) --- (Invalid)
 B = (TRN_HIS.MERCHANT_TYPE <> '5621') --- (Valid)
 B' = (TRN_HIS.MERCHANT_TYPE = '5621') --- (Invalid)
 C = (TRN_HIS.MERCHANT_NAME not like 'TOP%') --- (Valid)
 C' = (TRN_HIS.MERCHANT_NAME like 'TOP%') --- (Invalid)

ตารางที่ 4.8 ผลการทดลองของ Campaign ID: CM008

Test ID	Test Case	Expected Result
1	A' or (B' or C')	Invalid
2	A' or (B' or C)	Valid
3	A' or (B or C')	Valid
4	A' or (B or C)	Valid
5	A or (B' or C')	Valid
6	A or (B' or C)	Valid
7	A or (B or C')	Valid
8	A or (B or C)	Valid

9. Campaign ID: CM009

Condition: ((TRN_HIS.MERCHANT_TYPE is not null
 or upper(TRN_HIS.MERCHANT_NAME) not like '%AIRLINE%')
 and TRN_HIS.AMOUNT between 10000 and 30000)

คำอธิบายเงื่อนไข : รายการที่ประเภทร้านค้าไม่ใช่ค่าว่าง หรือมีชื่อร้านค้าที่ไม่ได้ประกอบด้วยคำว่า AIRLINE โดยที่ต้องมียอดใช้จ่ายตั้งแต่ 10,000 ถึง 30,000

Expression ที่ได้

A = (TRN_HIS.MERCHANT_TYPE is not null) --- (Valid)
 A' = (TRN_HIS.MERCHANT_TYPE is null) --- (Invalid)
 B = ((upper(TRN_HIS.MERCHANT_NAME)) not like '%AIRLINE%') --- (Valid)
 B' = ((upper(TRN_HIS.MERCHANT_NAME)) like '%AIRLINE%') --- (Invalid)
 C = (TRN_HIS.AMOUNT between 10000 and 30000) --- (Valid)
 C' = (TRN_HIS.AMOUNT not between 10000 and 30000) --- (Invalid)

ตารางที่ 4.9 ผลการทดลองของ Campaign ID : CM009

Test ID	Test Case	Expected Result
1	(A' or B') and C'	Invalid
2	(A' or B) and C'	Invalid
3	(A or B') and C'	Invalid
4	(A or B) and C'	Valid
5	(A' or B') and C	Valid
6	(A' or B) and C	Valid
7	(A or B') and C	Valid
8	(A or B) and C	Valid

10. Campaign ID: CM010

Condition: (TRN_HIS.MKT_CD = '3456'
 or (TRN_HIS.MERCHANT_TYPE = '5944'
 and TRN_HIS.MERCHANT_NAME not like '%GOLD%'))

คำอธิบายเงื่อนไข: รายการใช้จ่ายที่มีรหัสส่งเสริมการขายเท่ากับ 3456 หรือ รายการที่ประเภทร้านค้าเป็นประเภท 5944 และมีชื่อร้านค้าที่ไม่ได้ประกอบด้วยคำว่า GOLD

Expression ที่ได้

A = (TRN_HIS.MKT_CD = '3456') --- (Valid)
 A' = (TRN_HIS.MKT_CD = '456') --- (Invalid)
 B = (TRN_HIS.MERCHANT_TYPE = '5944') --- (Valid)
 B' = (TRN_HIS.MERCHANT_TYPE = '456') --- (Invalid)
 C = (TRN_HIS.MERCHANT_NAME not like '%GOLD%') --- (Valid)
 C' = (TRN_HIS.MERCHANT_NAME like '%GOLD%') --- (Invalid)

ตารางที่ 4.10 ผลการทดลองของ Campaign ID: CM010

Test ID	Test Case	Expected Result
1	A' or (B' and C')	Invalid
2	A' or (B' and C)	Invalid
3	A' or (B and C')	Invalid
4	A' or (B and C)	Valid

Test ID	Test Case	Expected Result
5	<i>A or (B' and C')</i>	Valid
6	<i>A or (B' and C)</i>	Valid
7	<i>A or (B and C')</i>	Valid
8	<i>A or (B and C)</i>	Valid

4.4 ผลการทดสอบเครื่องมือ

ผลการทดสอบเครื่องมือที่ช่วยในการตรวจสอบแคมเปญ และการประเมินผลในส่วนของผลลัพธ์ที่ได้ ซึ่งทำการประเมินผลโดยการนำค่าผลลัพธ์ที่ได้จากเครื่องมือมาทำการเปรียบเทียบกับผลลัพธ์ที่ได้จากผู้เชี่ยวชาญภาษาแอสคิวิแอลโดยดูจากผลลัพธ์ที่คาดหวังในกรณีทดสอบที่สร้างจากเครื่องมือ พบว่าได้ผลลัพธ์ที่ถูกต้องทุกกรณีตามที่นำมาทดสอบ

บทที่ 5

บทสรุปโครงการและข้อเสนอแนะ

5.1 สรุปผลโครงการมหาบัณฑิต

โครงการมหาบัณฑิตฉบับนี้มีวัตถุประสงค์เพื่อนำเสนอแนวคิด และเครื่องมือในการสร้างกรณีการทดสอบจากเงื่อนไขแคมเปญทางการตลาด ผลลัพธ์ที่ได้จากโครงการมหาบัณฑิตนี้ทำให้เห็นกรณีการทดสอบที่สามารถเกิดขึ้นได้ของเงื่อนไขแคมเปญทางการตลาด และผลลัพธ์ที่คาดหวังว่าจะเกิดขึ้นในแต่ละกรณี ซึ่งสามารถช่วยให้ผู้ใช้งานเห็นถึงกรณีต่าง ๆ ที่อาจจะเกิดขึ้นกับเงื่อนไขแคมเปญ ต่างจากเดิมที่ทำการทดสอบเพียงแคเงื่อนไขนี้สามารถผ่านตัวประมวลผลได้หรือไม่ และดูเพียงแค่จำนวนรายการที่เกิดขึ้นจากเงื่อนไขของแคมเปญทางการตลาดเพียงเท่านั้น

5.2 ข้อเสนอแนะและข้อปรับปรุงงานวิจัยในอนาคต

1. พัฒนาเครื่องมือที่สามารถรองรับฐานข้อมูลประเภทอื่น ๆ ได้เพื่อให้สามารถใช้งานได้หลากหลายประเภท
2. พัฒนาเครื่องมือให้สามารถรองรับคำสั่งสร้าง ลบ เพิ่มและปรับปรุงในภาษาเอสคิวแอลได้
3. พัฒนาการทดสอบให้สามารถทดสอบการทำงานของฟังก์ชันพิเศษของภาษาได้ เช่น NVL และ AVG เป็นต้น
4. พัฒนาเครื่องมือให้สามารถทดสอบคำสั่ง and or และตัวดำเนินการ in, not in, is null ฯลฯ ในรูปแบบตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ได้

รายการอ้างอิง

- [1] IEEE Computer Society. IEEE Standard Glossary of Software Engineering Terminology, 1983. ANSI/IEEE Std 729-1983.
- [2] Boris Beizer. Software Testing Techniques Second Edition, New York: Van Nostrand Reinhold (1990).
- [3] สุมนตรา ปัญจรัตน์. วิธีการสร้างกรณีทดสอบจากยูสเคสบนพื้นฐานของตารางการตัดสินใจ, วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ สาขาวิศวกรรมซอฟต์แวร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2549.
- [4] P. C. Jorgensen, “Software Testing: A Craftsman’s Approach”, third edition. United States of America: CRC Press LLC, 2007.
- [5] ดร. ภาณุชาติ บุญยเกียรติ และ ศศิธร มงคลศรีพัฒนา. เกณฑ์การประเมินความครอบคลุมของการทดสอบซอฟต์แวร์แบบโครงสร้าง. กรุงเทพมหานคร: คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยหอการค้าไทย.
- [6] โอภาส เอี่ยมสิริวงศ์. ระบบฐานข้อมูล, กรุงเทพมหานคร: ซีเอ็ดดูเคชั่น, 2551.
- [7] S. Feuerstein and B. Pribyl, Oracle PL/SQL programming, 6th ed. Sebastopol, CA: O’Reilly Media, 2014.
- [8] Docs.campaign.adobe.com, Targeting activities, 2015. [Online]. Available: https://docs.campaign.adobe.com/doc/AC6.1/en/WKF_Repository_of_activities_Targeting_activities.html.
- [9] M. Suárez-Cabal and J. Tuya, “Using an SQL coverage measurement for testing database applications”, SIGSOFT Softw. Eng. Notes, vol. 29, no. 6, p. 253, 2004.
- [10] J. Tuya, M. Suárez-Cabal and C. de la Riva, “A practical guide to SQL white-box testing”, ACM SIGPLAN Notices, vol. 41, no. 4, p. 36, 2006.
- [11] Abdul Khalek and S. Khurshid, “Automated SQL query generation for systematic testing of database engines”, Proceedings of the IEEE/ACM international conference on Automated software engineering - ASE '10, 2010.

[12] K. Pan, X. Wu and T. Xie, “Database state generation via dynamic symbolic execution for coverage criteria”, Proceedings of the Fourth International Workshop on Testing Database Systems - DBTest '11, 2011.

[13] อธิพิล เจียมอุทิศศักดิ์. เครื่องมือสร้างกรณีทดสอบสำหรับโปรแกรมประยุกต์บนเว็บ พื้นฐานสคีมาของฐานข้อมูล, วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาวิศวกรรมซอฟต์แวร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2555.

ภาคผนวก

ภาคผนวก ก

รหัสเงินตรา

รหัสเงินตรา (Currency Code) เป็นรหัสที่ถูกกำหนดขึ้นโดย องค์การระหว่างประเทศว่าด้วยการมาตรฐาน (The International Organization for Standardization: ISO) เพื่อใช้เป็นสื่อกลางในการทำธุรกรรมทางการเงินระหว่างประเทศ :ซึ่งได้กำหนดด้วยย่อของสกุลเงินไว้ -3 ตัวอักษร และรหัสตัวเลข 3 หลัก ไว้ด้วยกัน โดยกำหนดไว้ในมาตรฐาน ISO 4217 Currency Code ดังตารางที่ ก-1

ตารางที่ ก-1 : สกุลเงินปัจจุบัน

ENTITY	Currency	Alphabetic Code	Numeric Code
AFGHANISTAN	Afghani	AFN	971
ÅLAND ISLANDS	Euro	EUR	978
AUSTRIA	Euro	EUR	978
AZERBAIJAN	Azerbaijan Manat	AZN	944
BAHAMAS (THE)	Bahamian Dollar	BSD	044
BAHRAIN	Bahraini Dinar	BHD	048
BANGLADESH	Taka	BDT	050
BARBADOS	Barbados Dollar	BBD	052
CANADA	Canadian Dollar	CAD	124
CAYMAN ISLANDS (THE)	Cayman Islands Dollar	KYD	136
CENTRAL AFRICAN REPUBLIC (THE)	CFA Franc BEAC	XAF	950
GERMANY	Euro	EUR	978
GHANA	Ghana Cedi	GHS	936
GIBRALTAR	Gibraltar Pound	GIP	292
HAITI	Gourde	HTG	332
HONG KONG	Hong Kong Dollar	HKD	344
ITALY	Euro	EUR	978
JAMAICA	Jamaican Dollar	JMD	388
JAPAN	Yen	JPY	392
KOREA (THE DEMOCRATIC	North Korean	KPW	408

ENTITY	Currency	Alphabetic Code	Numeric Code
PEOPLE'S REPUBLIC OF)	Won		
KOREA (THE REPUBLIC OF)	Won	KRW	410
KUWAIT	Kuwaiti Dinar	KWD	414
KYRGYZSTAN	Som	KGS	417
LAO PEOPLE'S DEMOCRATIC REPUBLIC (THE)	Lao Kip	LAK	418
LUXEMBOURG	Euro	EUR	978
MACAO	Pataca	MOP	446
MACEDONIA (THE FORMER YUGOSLAV REPUBLIC OF)	Denar	MKD	807
MADAGASCAR	Malagasy Ariary	MGA	969
MALAWI	Malawi Kwacha	MWK	454
MALAYSIA	Malaysian Ringgit	MYR	458
MARSHALL ISLANDS (THE)	US Dollar	USD	840
MARTINIQUE	Euro	EUR	978
MAURITANIA	Ouguiya	MRO	478
MAURITIUS	Mauritius Rupee	MUR	480
NICARAGUA	Cordoba Oro	NIO	558
NIGER (THE)	CFA Franc BCEAO	XOF	952
NIGERIA	Naira	NGN	566
NIUE	New Zealand Dollar	NZD	554
NORTHERN MARIANA ISLANDS (THE)	US Dollar	USD	840
NORWAY	Norwegian Krone	NOK	578
PHILIPPINES (THE)	Philippine Peso	PHP	608
QATAR	Qatari Rial	QAR	634
SINGAPORE	Singapore Dollar	SGD	702
SWITZERLAND	Swiss Franc	CHF	756
SWITZERLAND	WIR Euro	CHE	947
SWITZERLAND	WIR Franc	CHW	948
TAIWAN (PROVINCE OF CHINA)	New Taiwan Dollar	TWD	901
TAJIKISTAN	Somoni	TJS	972

ENTITY	Currency	Alphabetic Code	Numeric Code
TANZANIA, UNITED REPUBLIC OF	Tanzanian Shilling	TZS	834
THAILAND	Baht	THB	764
UNITED KINGDOM OF GREAT BRITAIN AND NORTHERN IRELAND (THE)	Pound Sterling	GBP	826
UNITED STATES MINOR OUTLYING ISLANDS (THE)	US Dollar	USD	840
UNITED STATES OF AMERICA (THE)	US Dollar	USD	840
UNITED STATES OF AMERICA (THE)	US Dollar (Next day)	USN	997
VIET NAM	Dong	VND	704
VIRGIN ISLANDS (BRITISH)	US Dollar	USD	840
VIRGIN ISLANDS (U.S.)	US Dollar	USD	840
WALLIS AND FUTUNA	CFP Franc	XPF	953
WESTERN SAHARA	Moroccan Dirham	MAD	504
YEMEN	Yemeni Rial	YER	886
ZAMBIA	Zambian Kwacha	ZMW	967
ZIMBABWE	Zimbabwe Dollar	ZWL	932

ภาคผนวก ข

รหัสประเภทร้านค้า

รหัสประเภทร้านค้า (Merchant Category Code) คือตัวเลข 4 หลัก ที่ใช้สำหรับแบ่งประเภทของร้านค้าแต่ละประเภทตามธุรกิจ สินค้า และการให้บริการ ซึ่งถูกกำหนดไว้ตามมาตรฐานสากล ISO 18245 Retail financial services — Merchant category codes ดังตารางที่ ข-1

ตารางที่ ข-1 : ตัวอย่างรหัสของประเภทร้านค้า

MCC	Merchant Category
0742	Veterinary Services
0763	Agricultural Cooperative
0780	Landscaping Services
2791	Typesetting, Plate Making, and Related Services
2842	Specialty Cleaning
3000-3299	Airlines
3351-3441	Car Rental
3501-3790	Hotels/Motels/Inns/Resorts
4011	Railroads
4111	Commuter Transport, Ferries
5039	Construction Materials (Not Elsewhere Classified)
5111	Stationary, Office Supplies, Printing and Writing Paper
5122	Drugs, Drug Proprieties, and Druggist Sundries
5131	Piece Goods, Notions, and Other Dry Goods
5137	Uniforms, Commercial Clothing
5139	Commercial Footwear
5169	Chemicals and Allied Products (Not Elsewhere Classified)
5172	Petroleum and Petroleum Products
5192	Books, Periodicals, and Newspapers

MCC	Merchant Category
5193	Florists Supplies, Nursery Stock, and Flowers
5198	Paints, Varnishes, and Supplies
5199	Nondurable Goods (Not Elsewhere Classified)
5200	Home Supply Warehouse Stores
5211	Lumber, Building Materials Stores
5231	Glass, Paint, and Wallpaper Stores
5499	Miscellaneous Food Stores - Convenience Stores and Specialty Markets
5511	Car and Truck Dealers (New & Used) Sales, Service, Repairs Parts and Leasing
5521	Car and Truck Dealers (Used Only) Sales, Service, Repairs Parts and Leasing
5531	Auto and Home Supply Stores
5532	Automotive Tire Stores
5533	Automotive Parts and Accessories Stores
5541	Service Stations
5542	Automated Fuel Dispensers
5551	Boat Dealers
5561	Motorcycle Shops, Dealers
5571	Motorcycle Shops and Dealers
5592	Motor Homes Dealers
5598	Snowmobile Dealers
5599	Miscellaneous Auto Dealers
5611	Men's and Boy's Clothing and Accessories Stores
5621	Women's Ready-To-Wear Stores
5631	Women's Accessory and Specialty Shops
5641	Children's and Infant's Wear Stores
5651	Family Clothing Stores
5655	Sports and Riding Apparel Stores
5661	Shoe Stores

MCC	Merchant Category
5681	Furriers and Fur Shops
5691	Men's, Women's Clothing Stores
5697	Tailors, Alterations
5698	Wig and Toupee Stores
5699	Miscellaneous Apparel and Accessory Shops
5712	Furniture, Home Furnishings, and Equipment Stores, Except Appliances
5713	Floor Covering Stores
5714	Drapery, Window Covering, and Upholstery Stores
5718	Fireplace, Fireplace Screens, and Accessories Stores
5719	Miscellaneous Home Furnishing Specialty Stores
5722	Household Appliance Stores
5732	Electronics Stores
5733	Music Stores-Musical Instruments, Pianos, and Sheet Music
5734	Computer Software Stores
5735	Record Stores
5811	Caterers
5812	Eating Places, Restaurants
5813	Drinking Places
5814	Fast Food Restaurants
5912	Drug Stores and Pharmacies
5921	Package Stores-Beer, Wine, and Liquor
5931	Used Merchandise and Secondhand Stores
5932	Antique Shops
5933	Pawn Shops
5935	Wrecking and Salvage Yards
5937	Antique Reproductions
5940	Bicycle Shops
5941	Sporting Goods Stores
5942	Book Stores

MCC	Merchant Category
5943	Stationery Stores, Office, and School Supply Stores
5944	Jewelry Stores, Watches, Clocks, and Silverware Stores
5945	Hobby, Toy, and Game Shops
5946	Camera and Photographic Supply Stores
5947	Gift, Card, Novelty, and Souvenir Shops
5948	Luggage and Leather Goods Stores
5949	Sewing, Needlework, Fabric, and Piece Goods Stores
5950	Glassware, Crystal Stores
5960	Direct Marketing - Insurance Services
5962	Direct Marketing - Travel
5963	Door-To-Door Sales
5964	Direct Marketing - Catalog Merchant
5997	Electric Razor Stores
5998	Tent and Awning Shops
5999	Miscellaneous Specialty Retail
6010	Manual Cash Disburse
6011	Automated Cash Disburse
6012	Financial Institutions
6051	Non-FI, Money Orders
6211	Security Brokers/Dealers
6300	Insurance Underwriting, Premiums
6399	Insurance - Default

ประวัติผู้เขียนโครงการมหาบัณฑิต

นางสาวฤกษ์ระวี พันธุ์ลุกา เกิดเมื่อวันที่ 4 มิถุนายน พ.ศ. 2530 สำเร็จการศึกษาระดับปริญญาตรีหลักสูตรวิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ศรีราชา มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา ปีการศึกษา 2553

เข้าศึกษาต่อระดับปริญญาโทมหาบัณฑิต ปีการศึกษา 2557 หลักสูตรวิทยาศาสตร์มหาบัณฑิต (วท.ม) สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย