สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์

MASSIVELY PARALLEL PROCESSING DATABASE PERFORMANCE TOOL USING AGENT TECHNOLOGY

T144224

โดย

ประวิทย์ วงศ์จักรคำ

PRAWIT WONGJAKKHAM

อาจารย์ที่ปรึกษา

ผศ.ดร.ภัทรชัย ลลิตโรจน์วงศ์

ว.ม. ป 37821 2557
เลขหมู่.....
เลขทะเบียน.....144221
วัน,เดือน,ปี.....09 พย. 2559
.b.....12816139
.i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาการศึกษาอิสระ 2
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MASSIVELY PARALLEL PROCESSING DATABASE PERFORMANCE TOOL USING AGENT TECHNOLOGY

PRAWIT WONJAKKHM

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS OF THE COURSE

INDEPENDENT STUDY 2

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2015

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองการศึกษาอิสระ 2 (Independent Study 2)

เรื่อง

โปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์

Massively Parallel Processing Database Performance Tool using Agent Technology

นายประวิทย์ วงศ์จักรคำ

รหัสประจำตัว 53660533

ขอรับรองว่ารายงานฉบับนี้ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ
การศึกษาวิชาการศึกษาอิสระ 2 หลักสูตรวิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศ)
ภาคเรียนที่ 2 ปี การศึกษา 2557

.....**อาจารย์ที่ปรึกษา**
(ผศ.ดร.ภัทรชัย ลลิตโรจน์วงศ์)

.....**กรรมการสอบ**
(รศ.ดร.วรพจน์ กรีสุระเดช)

.....**กรรมการสอบ**
(ดร.กนกวรรณ อัจฉริยะชาญวณิช)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	โปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์
นักศึกษา	นายประวิทย์ วงศ์จักรคำ
รหัสนักศึกษา	53660533
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2557
อาจารย์ที่ปรึกษา	ผศ.ดร.ภัทรชัย ลลิตโรจน์วงศ์

บทคัดย่อ

ในปัจจุบันองค์กรขนาดใหญ่ ได้นำระบบฐานข้อมูลประมวลผลแบบขนานหลายหน่วยมาใช้เป็นระบบฐานข้อมูลของคลังข้อมูล เพื่อรวบรวมฐานข้อมูลจากหลายแหล่งหลายช่วงเวลา มารวม ณ ที่เดียวกัน จึงทำให้ฐานข้อมูลมีขนาดใหญ่ ซึ่งสถาปัตยกรรมของระบบฐานข้อมูลประมวลผลแบบขนานหลายหน่วยนั้นจะมีลักษณะหลายเซิร์ฟเวอร์รวมกันเป็นหนึ่งระบบฐานข้อมูล โดยจะมีหนึ่งเซิร์ฟเวอร์ทำหน้าที่รับชุดคำสั่งจากผู้ใช้งาน แล้วส่งชุดคำสั่งกระจายไปประมวลผลแต่ละเซิร์ฟเวอร์ เมื่อแต่ละเซิร์ฟเวอร์ประมวลผลเสร็จจะส่งผลลัพธ์มายังเซิร์ฟเวอร์ทำหน้าที่รับชุดคำสั่งจากผู้ใช้งาน เนื่องจากระบบฐานข้อมูลประมวลผลแบบขนานหลายหน่วยมีหลายเซิร์ฟเวอร์ ทำให้ผู้ดูแลระบบฐานข้อมูลต้องเฝ้าสังเกตประสิทธิภาพมากขึ้น เพราะว่าถ้าเซิร์ฟเวอร์ไหนมีปัญหาหรือทำงานช้าจะส่งผลต่อประสิทธิภาพการดำเนินการของเอสคิวแอล ในโครงการนี้มีวัตถุประสงค์เพื่อศึกษาและนำเอาเทคโนโลยีเอเจนท์ ซึ่งเป็นซอฟต์แวร์ที่สามารถเคลื่อนย้ายตัวเองไปยังเซิร์ฟเวอร์เครื่องอื่น ๆ ผ่านทางเครือขายคอมพิวเตอร์เพื่อทำงาน มาประยุกต์ใช้ในการเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย โดยทำหน้าที่เฝ้าสังเกตประสิทธิภาพแทนผู้ดูแลระบบฐานข้อมูล และการตั้งค่าเวลาเพื่อบำรุงรักษาระบบฐานข้อมูล รวมถึงการแจ้งเตือนจากการเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Massively Parallel Processing Database Performance Tool Using Agent Technology
Student	Mr. Prawit Wongjakkham
Student ID.	53660533
Degree	Master of Science
Program	Information Technology
Major	Information Science
Academic Year	2014
Advisor	Asst. Prof. Dr. Pattarachai Lalitrojwong

ABSTRACT

Nowadays, large organizations have taken massively parallel processing databases using the data warehouse appliance. Because gathering data from multiple sources and integrating it at the same place the database be comes. The architecture of a massively parallel processing database system requires multiple servers combined into one database system. There is one server receiving commands from users and then sending the command to other servers. When processing is complete, each server sends the results to the server which received commands from the user. Because the massively parallel processing database has multiple servers, the database administrator has to focus on monitoring performance. Because if a problem arises or the database is slow, it will affect the performance of SQL. The objective of this project is study and to employ agent technology to solve the problem processing. Agents are software that can perform autonomous actions on behalf of users and can move spontaneously from one agent server in the network to another until the work is accomplished. The agents monitor a massively parallel processing database to maintain the database inncluding notifying users when any uncommon event occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จได้อย่างดีด้วยคำแนะนำ และคำปรึกษาจาก ผศ.ดร.ภัทรชัย ลลิตโรจน์วงศ์ ซึ่งเป็นอาจารย์ที่ปรึกษาในโครงงานนี้ ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์ภาควิชาวิทยาการสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ประสิทธิประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อน ๆ ในภาควิชาวิทยาการสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกคน ที่ให้คำแนะนำต่าง ๆ และค่อยให้ความช่วยเหลือเสมอมา

ขอขอบคุณบัณฑิตศึกษา และบัณฑิตวิทยาลัย คณะเทคโนโลยีสารสนเทศ ที่ให้ความช่วยเหลือในเรื่องต่าง ๆ

สุดท้ายนี้ ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดา และครอบครัวของข้าพเจ้าที่คอยเป็นกำลังใจ และให้การสนับสนุนในทุกๆเรื่อง ทำให้ข้าพเจ้าสามารถทำโครงงานฉบับนี้ได้สำเร็จลุล่วง คุณค่าและประโยชน์อันพึงได้จากโครงงานฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ประวิทย์ วงศ์จักรคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

ABSTRACT II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 2

1.3 ขอบเขตของการศึกษา 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 ขั้นตอนของการศึกษา 3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4

2.1 เทคโนโลยีซอฟต์แวร์เอเจนท์ 4

2.2 เอเจนท์เคลื่อนที่ 5

2.3 ระบบการจัดการฐานข้อมูลแบบกระจาย 11

2.5 ระบบการจัดการฐานข้อมูลที่ใช้ในการศึกษา 14

บทที่ 3 การวิเคราะห์และการออกแบบระบบ 27

3.1 สถาปัตยกรรมของเอเจนท์ฝ้าสังเกตประสิทธิภาพของฐานข้อมูล 27

3.2 ขั้นตอนการทำงานของระบบ 29

3.3 ยูสเคสไดอะแกรม 33

3.4 คลาสไดอะแกรม 42

3.5 ซีเควนซ์ไดอะแกรม 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 การพัฒนาระบบ 48

4.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ 48

4.2 การจำลองเซิร์ฟเวอร์ 49

4.3 การพัฒนาระบบ 57

4.4 การเชื่อมต่อฐานข้อมูล 61

บทที่ 5 บทสรุป 63

5.1 สรุปโครงการ 63

5.2 ประโยชน์ที่ได้รับ 63

บรรณานุกรม 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 คำอธิบายยูสเคสค้นหาข้อมูลเอสคิวแอล 33

3.2 คำอธิบายยูสเคสจัดการค่าในระบบ 35

3.3 คำอธิบายยูสเคสรวบรวมและจัดเก็บข้อมูล 36

3.4 คำอธิบายยูสเคสบำรุงรักษาฐานข้อมูล 38

3.5 คำอธิบายยูสเคสตรวจสอบข้อมูลประสิทธิภาพ 39

3.6 คำอธิบายยูสเคสแจ้งเตือนปัญหา 41

4.1 รายละเอียดเอนทิตี GP_SERVER_SKEW 53

4.2 รายละเอียดเอนทิตี SERVER_RESOURCE 54

4.3 รายละเอียดเอนทิตี CONFIGURATION 54

4.4 รายละเอียดเอนทิตี RESQUEUE_STATUS 55

4.5 รายละเอียดเอนทิตี LOCK_PROCESS 55

4.6 รายละเอียดเอนทิตี SQL_HISTORY 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 เปรียบเทียบการสื่อสารแบบไคลเอนท์เซิร์ฟเวอร์ และแบบเอเจนท์เคลื่อนที่ 5

2.2 ระบบการจัดการฐานข้อมูลแบบกระจาย 12

2.3 การประมวลผลแบบกระจาย 13

2.4 สถาปัตยกรรมฐานข้อมูลกรีนพลัม 15

2.5 ตัวอย่างฐานข้อมูล star schema 17

2.6 การกระจายข้อมูลในตารางของฐานข้อมูลกรีนพลัม 18

2.7 แผนเอสคิวแอลแสดงการจ่ายงานแบบขนาน 19

2.8 แผนเอสคิวแอลแสดงการจ่ายงานแบบขนานไปยังจุดเป้าหมาย 20

2.9 แผนคำสั่งภาษาเอสคิวแอล 21

3.1 สถาปัตยกรรมโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย โดยใช้เทคโนโลยีเอเจนท์ 27

3.2 แสดงขั้นตอนการทำงานเมื่อผู้ใช้ต้องการค้นหาข้อมูลเอสคิวแอล 29

3.3 แสดงขั้นตอนการทำงานเมื่อผู้ใช้กำหนดให้บำรุงรักษาฐานข้อมูล 30

3.4 แสดงขั้นตอนการทำงานเมื่อผู้ใช้เฝ้าสังเกตประสิทธิภาพฐานของข้อมูล 32

3.5 ยูสเคสไดอะแกรมโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย 33

3.6 ยูสเคสไดอะแกรมเอเจนท์รวบรวมข้อมูล 36

3.7 ยูสเคสไดอะแกรมเอเจนท์บำรุงรักษาฐานข้อมูล 37

3.8 ยูสเคสไดอะแกรมเอเจนท์เฝ้าสังเกต 39

3.9 คลาสไดอะแกรมโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย 42

3.10 คลาสไดอะแกรมโปรแกรมเอเจนท์ 43

3.11 ซีเควนซ์ไดอะแกรมแสดงลำดับการทำงานในการค้นหาเอสคิวแอล 45

3.12 ซีเควนซ์ไดอะแกรมแสดงลำดับการทำงานในการเฝ้าสังเกตค่าของฐานข้อมูลเกินที่กำหนด . 46

4.1 หน้าการค้นหาเอสคิวแอลที่ใช้เวลาดำเนินงานที่ฐานข้อมูลนาน หรือใช้ทรัพยากรเครื่องสูง 57

4.2 หน้าแสดงผลเอสคิวแอล 58

4.3 หน้าแสดงข้อมูลเอสคิวแอล 59

4.4 หน้าจัดการรายการบำรุงรักษาฐานข้อมูล 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.5 หน้าจัดการรายการสำหรับเฝ้าสังเกตประสิทธิภาพของฐานข้อมูล 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้มีการเปลี่ยนแปลงอย่างรวดเร็ว ในด้านของข้อมูลก็เช่นกันก็ต้องการข้อมูลที่ถูกต้องแม่นยำและทันต่อสถานการณ์ การใช้ข้อมูลเป็นเครื่องมือสำคัญในการตัดสินใจขององค์กร ไม่ว่าจะเป็นเพื่อการลงทุนทางธุรกิจ การวางแผนงานในองค์กรภาครัฐ และวางแผนกลยุทธ์ในทุกองค์กร เพื่อให้สามารถตอบโต้กับการเปลี่ยนแปลงอย่างรวดเร็วของสิ่งแวดล้อมและคู่แข่งขัน ฉะนั้นก็อาจกล่าวได้ว่าการมีข้อมูลมากทำให้มีโอกาสและมีชัยเหนือคู่แข่งในระดับหนึ่ง ดังนั้นคลังข้อมูลจึงเกิดขึ้นเพื่อเป็นที่เก็บรวบรวมข้อมูลสำคัญและจำเป็นจากแหล่งต่างๆ ซึ่งเป็นประโยชน์ต่อการตัดสินใจของผู้บริหารและหน่วยงานต่าง ๆ

คลังข้อมูลเป็นฐานข้อมูลขนาดใหญ่ที่รวบรวมฐานข้อมูลจากหลายแหล่งหลายช่วงเวลา ซึ่งอาจมีข้อมูลแตกต่างกันมารวมไว้ที่เดียวกัน จึงมีระบบการจัดการฐานข้อมูลที่เหมาะสมกับคลังข้อมูลที่เรียกว่า ฐานข้อมูลประมวลผลแบบขนานหลายหน่วย (Massively Parallel Processing Database) ซึ่งระบบการจัดการฐานข้อมูลประเภทนี้ จะมีสถาปัตยกรรมที่ออกแบบให้ค้นคืนข้อมูลเดียวกันถูกดำเนินการโดยหน่วยประมวลผลหลายหน่วยซึ่งมีประสิทธิภาพในการประมวลผลคำสั่งภาษาเอสคิวแอลสูง และสามารถขยายขนาดได้ แต่เมื่อระบบฐานข้อมูลประมวลผลแบบขนานหลายหน่วยมีขนาดใหญ่ขึ้นจะมีปัญหาเกี่ยวกับการกระจายข้อมูลไปยังเซกเมนต์ต่าง ๆ และการดำเนินการของคำสั่งภาษาเอสคิวแอลที่ทำให้ฐานข้อมูลมีปัญหา แล้วกระทบกับทรานแซกชันอื่นทำให้เกิดความล่าช้าในการดำเนินการแต่ละเอสคิวแอล

ผู้ทำโครงการจึงมีแนวคิดในการนำเสนอโครงการพัฒนาโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์ เพื่ออำนวยความสะดวกให้กับผู้ดูแลฐานข้อมูล โดยจะทำหน้าที่แทนผู้ดูแลฐานข้อมูลในการเฝ้าระวังการดำเนินการของคำสั่งภาษาเอสคิวแอลในฐานข้อมูล และการใช้ทรัพยากรเครื่อง ตามที่กำหนดเงื่อนไขไว้ได้อย่างอัตโนมัติ

ซึ่งในปัจจุบันยังไม่มีเครื่องมือเฉพาะที่ใช้เฝ้าระวังฐานข้อมูลประมวลผลแบบขนานหลายหน่วย และผู้ดูแลฐานข้อมูลต้องเฝ้าระวังฐานข้อมูลด้วยตนเองตลอดเวลา ทำให้เกิดความล่าช้าในการหาสาเหตุของปัญหาเกี่ยวกับประสิทธิภาพของฐานข้อมูลจากการดำเนินการของคำสั่งภาษาเอสคิวแอลจากผู้ใช้งาน ซึ่งปัญหาที่พบจากการเฝ้าระวังประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยในปัจจุบัน มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ผู้ดูแลฐานข้อมูลไม่สามารถเฝ้าระวังคำสั่งภาษาเอสคิวแอลที่ทำให้ฐานข้อมูลมีปัญหาได้ตลอดเวลา แล้วกระทบกับทรานแซกชันอื่น ทำให้เกิดความล่าช้าในการดำเนินการแต่ละคำสั่งเอสคิวแอล
2. ผู้ดูแลฐานข้อมูลไม่สามารถเฝ้าระวังการใช้ทรัพยากรเครื่องได้ตลอดเวลา เมื่อทรัพยากรเครื่องไม่เพียงพอจะทำให้เกิดปัญหากับฐานข้อมูล
3. ผู้ดูแลฐานข้อมูลต้องรวบรวมข้อมูลการบำรุงรักษาฐานข้อมูลเอง ซึ่งอาจไม่ครอบคลุมเมื่อต้องทำการบำรุงรักษาฐานข้อมูล

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วัตถุประสงค์ของการดำเนินการโครงการพัฒนาโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์ มีดังนี้

1. เพื่อช่วยผู้ดูแลฐานข้อมูลเฝ้าระวังคำสั่งภาษาเอสคิวแอลที่ทำให้ฐานข้อมูลมีปัญหา และมีการแจ้งเตือน
2. เพื่อช่วยผู้ดูแลฐานข้อมูลเฝ้าระวังการใช้ทรัพยากรเครื่อง เมื่อเกินกำหนดกับค่าที่ตั้งไว้จะมีการแจ้งเตือน
3. เพื่อทำหน้าที่บำรุงรักษาฐานข้อมูลแทนผู้ดูแลฐานข้อมูล

1.3 ขอบเขตของการศึกษา

โปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์เป็นโปรแกรมที่สร้างขึ้นเพื่อตอบสนอง และอำนวยความสะดวกแก่ผู้ดูแลฐานข้อมูล โดยขอบเขตของการศึกษาการนำเทคโนโลยีเอเจนท์เฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย มีดังนี้

1. สามารถแจ้งเตือนเมื่อมีคำสั่งภาษาเอสคิวแอลที่ทำให้ฐานข้อมูลมีปัญหา และสามารถค้นหาคำสั่งภาษาเอสคิวแอลที่กำลังดำเนินการอยู่หรือได้ดำเนินการไปแล้ว โดยเลือกช่วงเวลาได้
2. สามารถแจ้งเตือนการใช้ทรัพยากรเครื่องตามค่าที่กำหนด
3. สามารถทำหน้าที่บำรุงรักษาฐานข้อมูลแทนผู้ดูแลระบบฐานข้อมูลได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าผู้ดูแลฐานข้อมูล จะได้รับจากการพัฒนาโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ผู้ดูแลฐานข้อมูลได้รับอีเมลแจ้งเตือนเมื่อพบคำสั่งภาษาเอสคิวแอลที่ทำให้ฐานข้อมูลมีปัญหา และสามารถค้นหาข้อมูลคำสั่งภาษาเอสคิวแอลได้
2. ผู้ดูแลฐานข้อมูลสามารถกำหนดค่าการเฝ้าระวังการใช้ทรัพยากรเครื่อง เมื่อเกินกำหนดจะมีการแจ้งเตือนผ่านทางอีเมล
3. ผู้ดูแลฐานข้อมูลเลือกรายการการบำรุงรักษาฐานข้อมูลที่ต้องการ

1.5 ขั้นตอนของการศึกษา

ขั้นตอนในการดำเนินโครงการประกอบด้วยขั้นตอนการทำงานต่าง ๆ ดังนี้

1. วิเคราะห์ปัญหา และข้อจำกัดของระบบเดิม เพื่อใช้เป็นแนวทางในการนำเสนอระบบงานใหม่
2. ศึกษาทฤษฎี และงานวิจัยที่เกี่ยวข้องกับเทคโนโลยี ที่จะนำมาพัฒนาระบบ
3. กำหนดขอบเขต วิเคราะห์และออกแบบระบบเอเจนท์เฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย นำมาสร้างเป็นแบบจำลองเชิงวัตถุ โดยแสดงด้วยแผนภาพดังต่อไปนี้
   - ยูสเคสไดอะแกรม แสดงให้เห็นถึงฟังก์ชันการทำงานของระบบ และผู้ที่เกี่ยวข้องกับระบบ
   - แอกทิวิตีไดอะแกรม แสดงกระแสลำดับกิจกรรมของงานใด ๆ ที่เกิดขึ้น และผลจากการทำกิจกรรมในขั้นตอนต่าง ๆ
   - คลาสไดอะแกรม แสดงความสัมพันธ์ระหว่างคลาสของวัตถุต่าง ๆ ที่มีในระบบ รวมถึงคุณสมบัติและการกระทำที่วัตถุในคลาสต่าง ๆ สามารถกระทำได้
   - ซีเควนซ์ไดอะแกรม แสดงให้เห็นถึงลำดับขั้นตอนในการติดต่อสื่อสารระหว่างวัตถุในระบบ
4. พัฒนา และทดสอบการทำงานของระบบที่พัฒนา
5. สรุปผลและจัดทำเอกสารคู่มือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีต่าง ๆ และเทคโนโลยีที่เกี่ยวข้องในการพัฒนาระบบ ซึ่งเนื้อหาเกี่ยวข้องกับเทคโนโลยีซอฟต์แวร์เอเจนท์ และเกี่ยวกับฐานข้อมูลประมวลผลแบบขนานหลายหน่วย สำหรับนำมาประยุกต์ใช้ในการพัฒนาเอเจนท์เฝ้าระวังประสิทธิภาพของฐานข้อมูล มีดังต่อไปนี้

2.1 เทคโนโลยีซอฟต์แวร์เอเจนท์

ลักษณะของซอฟต์แวร์เอเจนท์ เป็นโปรแกรมที่สามารถทำงานแทนผู้ใช้ตามที่กำหนดได้ เพื่อช่วยอำนวยความสะดวกแก่ผู้ใช้ และสามารถทำงานเป็นอัตโนมัติ โดยซอฟต์แวร์เอเจนท์คุณสมบัติสำคัญอย่างหนึ่งคือ สามารถสื่อสารกับเอเจนท์อื่น ๆ และทำงานในระบบที่มีความแตกต่างกันได้ (Weigel. 2009)

ซอฟต์แวร์เอเจนท์ต้องอาศัยสภาพแวดล้อมเฉพาะเพื่อให้เอเจนท์ทำการติดตั้ง และทำงานได้ เรียกสิ่งแวดล้อมนี้ว่า เอเจนท์แพลตฟอร์มหรือเอเจนท์เซิร์ฟเวอร์ ซึ่งคอยจัดสรรทรัพยากรที่จำเป็นต่อการทำงานและควบคุมวงจรชีวิตของเอเจนท์ ตัดสินใจว่าเมื่อไรที่เอเจนท์ต้องเริ่มหรือหยุดการทำงาน ต้องส่งข้อมูลหรือไม่ และต้องทำลายเอเจนท์เมื่อใด เป็นต้น นอกจากนั้น ยังจัดเตรียมเครื่องมือที่ช่วยให้ผู้ใช้และผู้พัฒนาระบบ สามารถติดต่อใช้งานเอเจนท์ได้อีกด้วย ดังนั้น เอเจนท์แพลตฟอร์มจึงเปรียบเสมือนส่วนต่อประสานระหว่างผู้ใช้งานกับเอเจนท์ และเอเจนท์กับทรัพยากร

Object Management Group (OMG) ได้แบ่งประเภทของเอเจนท์ไว้เป็น 2 ประเภท ดังนี้

1. เอเจนท์แบบสเตชันนารี่ (Stationary agent) คือ เอเจนท์ที่จะทำการประมวลผลได้เฉพาะบนระบบที่สร้างเอเจนท์นั้น หากเอเจนท์ต้องการติดต่อกับเอเจนท์ที่อยู่ในระบบอื่นก็จะต้องทำการติดต่อผ่านกลไกการติดต่อสื่อสารด้วยการเรียกใช้ส่วนการทำงานระยะไกล (Remote Procedure Call หรือ RPC)
2. เอเจนท์แบบเคลื่อนที่ (Mobile Agent) คือ เอเจน์ที่ไม่ผูกติดกับระบบที่เอเจนท์เริ่มต้นทำการประมวลผล โดยสามารถเคลื่อนที่ตัวมันเองไปยังระบบเอเจนท์ปลายทางที่มีวัตถุที่เอเจนท์ต้องการติดต่อด้วยได้ และเอเจนท์อาจใช้บริการของวัตถุนั้น

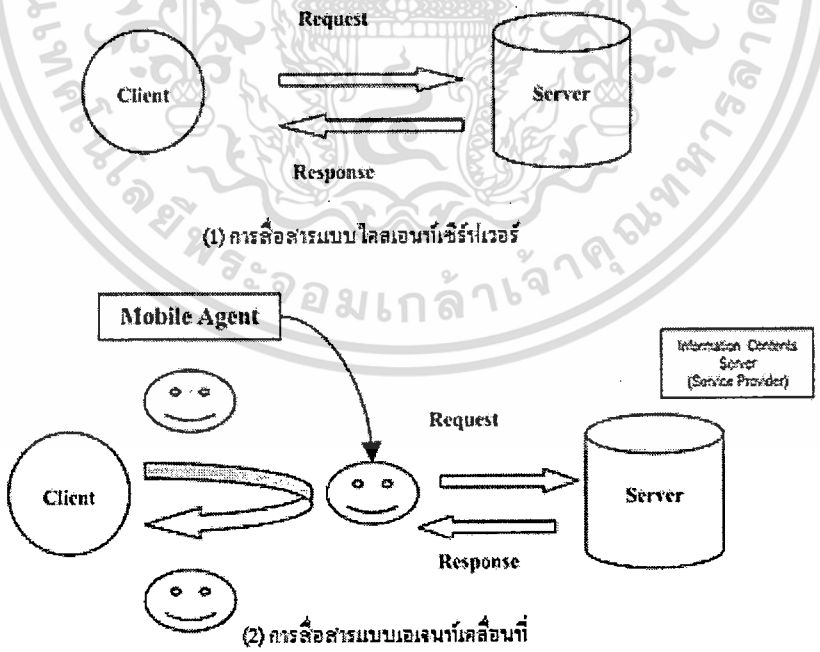
แม้ว่าซอฟต์แวร์เอเจนท์จะสามารถแบ่งออกได้เป็นหลายประเภท แต่มีคุณสมบัติบางอย่างที่ซอฟต์แวร์เอเจนท์แต่ละประเภทมีร่วมกัน เช่น สามารถรับรู้ถึงการเปลี่ยนแปลงของสภาพแวดล้อมที่ทำงานอยู่ และตอบสนองต่อการเปลี่ยนแปลงนั้นได้ สามารถทำงานได้อย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามที่ผู้ใช้มอบหมาย และทำงานแบบไม่ประสานจังหวะ คือระบบสามารถทำงานอย่างอื่น ในขณะที่รอผลลัพธ์จากซอฟต์แวร์เอเจนท์ได้ มีการทำงานแบบเชิงรุก ซึ่งสามารถทำงานได้โดยไม่ต้องรอคำสั่ง และทำงานอย่างต่อเนื่อง เพื่อให้ได้ผลลัพธ์หรือบรรลุงานที่ได้รับมอบหมาย เป็นต้น นอกจากนั้น ยังมีคุณสมบัติอื่น ๆ โดยในซอฟต์แวร์เอเจนท์แต่ละประเภท คือ มีความสามารถในการติดต่อสื่อสารกับเอเจนท์อื่น ๆ ได้ สามารถในการเคลื่อนย้ายตัวเองจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่งผ่านระบบเครือข่าย และความสามารถในการเรียนรู้ และตอบสนองต่อเหตุการณ์ต่าง ๆ ตามที่ได้เรียนรู้มา เป็นต้น (Lange and Oshima. 2009)

2.2 เอเจนท์เคลื่อนที่

เอเจนท์เคลื่อนที่ คือ ซอฟต์แวร์เอเจนท์ประเภทหนึ่งที่มีคุณสมบัติสามารถเคลื่อนย้ายตัวเอง ซึ่งประกอบด้วยชุดคำสั่ง สถานะการทำงานปัจจุบัน และข้อมูลที่เกี่ยวข้อง ไปยังเอเจนท์เซิร์ฟเวอร์อื่น ๆ ผ่านระบบเครือข่ายเพื่อทำงานอย่างใดอย่างหนึ่ง เช่น รวบรวม หรือประมวลผลข้อมูล เป็นต้น ได้อย่างอิสระ (Aglets Development Group. 2009) จากคุณสมบัติในการเคลื่อนย้ายตัวเองนี้ทำให้เอเจนท์สามารถย้ายตัวเองไปหาข้อมูลที่จำเป็นสำหรับการทำงาน ซึ่งเมื่อเปรียบเทียบแล้ว เอเจนท์เคลื่อนที่มักจะมีขนาดเล็กมากกว่าข้อมูล ส่งผลให้ช่วยประหยัดแบนด์วิดธ์ และลดปริมาณการจราจรบนระบบเครือข่ายลงได้ ดังรูปที่ 2.1



รูปที่ 2.1 เปรียบเทียบการสื่อสารแบบไคลเอนท์เซิร์ฟเวอร์(1) และแบบเอเจนท์เคลื่อนที่(2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อสนับสนุนการทำงานของเอเจนท์เคลื่อนที่ เอเจนท์แพลตฟอร์มต้องจัดเตรียมบริการเพื่อสนับสนุนการเคลื่อนย้ายตัวเองของเอเจนท์ ทั้งเข้าและออกจากเอเจนท์แพลตฟอร์ม รวมถึงอำนวยความสะดวกในการส่งข้อความสื่อสารระหว่างเอเจนท์ที่อยู่ต่างเอเจนท์แพลตฟอร์มด้วย โดยเอเจนท์แพลตฟอร์มที่สนับสนุนการทำงานของเอเจนท์เคลื่อนที่นี้เรียกว่า แพลตฟอร์มเอเจนท์เคลื่อนที่

2.2.1 เอเจนท์ และสถานที่

แบบจำลองเอเจนท์เคลื่อนที่ประกอบด้วยส่วนประกอบพื้นฐาน 2 ส่วน ดังนี้ (Lange and Oshima. 1998)

1. **เอเจนท์**

เอเจนท์ (Agent) เป็นเอนทิตีที่ประกอบด้วยลักษณะประจำ 4 ประการ ซึ่งจะถูกเคลื่อนย้ายไปเครื่องต่าง ๆ ในเครือข่ายพร้อมกับเอเจนท์ด้วย ประกอบด้วย

- สถานะ (State) คือ การทำงานต่าง ๆ ที่เกี่ยวข้องกับการทำงานของเอเจนท์ ซึ่งจะช่วยให้เอเจนท์สามารถดำเนินงานต่อไปหลังจากเคลื่อนย้ายไปยังเครื่องอื่น ๆ ได้ ประกอบด้วยสถานการณ์ประมวลผล ซึ่งจะเก็บสถานะที่เกี่ยวข้องกับการทำงาน และสถานะวัตถุ ซึ่งจะเก็บค่าตัวแปรต่าง ๆ ที่ถูกประกาศไว้ในเอเจนท์
- การสร้าง (Implementation) คือ ชุดคำสั่ง หรือโค้ดที่จำเป็นต่อการสร้าง และการประมวลผลของเอเจนท์ ซึ่งการได้มาของโค้ดในเครื่องปลายทางก่อน ถ้าไม่มีจึงค่อยดึงโค้ดจากเครื่องต้นทาง ซึ่งเป็นลักษณะการทำงานแบบโค้ดออนดิมานด์
- ส่วนต่อประสาน (Interface) เป็นสิ่งที่เอเจนท์จัดเตรียมไว้เพื่อให้เอเจน์อื่น ๆ และสภาพแวดล้อม สามารถติดต่อสื่อสารกับเอเจนท์ได้
- ตัวระบุ (Identifier) คือ ตัวเลขที่เป็นเอกลักษณ์ไม่ซ้ำกัน จะถูกกำหนดให้กับเอเจนท์ทุกตัว ใช้ในการะบุถึงเอเจนท์ตัวใดตัวหนึ่งในระบบ ซึ่งเอเจนท์จะใช้ตัวระบุเดิมตลอดช่วงชีวิตของเอเจนท์

2. **สถานที่**

สถานที่ (Place) คือ สภาพแวดล้อมซึ่งคอยให้บริการที่จำเป็นต่อการทำงานเอเจนท์ เปรียบเสมือนระบบปฏิบัติการของเอเจนท์ ประกอบด้วย 4 องค์ประกอบ ดังนี้

- เอ็นจิน (Engine) สถานที่จะต้องอาศัยเอ็นจินในการให้บริการแก่เอเจนท์ ซึ่งจะช่วยให้สถานที่สามารถเชื่อมต่อกับเครือข่าย และเข้าถึงทรัพยากรต่าง ๆ ที่จะจัดสรรให้แก่เอเจนท์ต่อไป ตัวอย่างระบบเอเจนท์เคลื่อนที่ที่พัฒนาด้วยภาษาจาวาจะมีจาวาเวอร์ชวลแมชชีนเป็นเอ็นจิน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทรัพยากร (Resource) เป็นสิ่งที่เอเจนท์ต้องการสำหรับการทำงาน เช่น หน่วยประมวลผล ฐานข้อมูล และหน่วยความจำ เป็นต้น โดยมีเครื่องประมวลผล และสถานที่ เป็นผู้ควบคุมการเข้าถึงทรัพยากรของเอเจนท์
- ที่ตั้ง (Location) ใช้ในการระบุตำแหน่งของเอเจนท์เคลื่อนที่ที่ต้องการอ้างถึง ซึ่งจะอยู่ในรูปของชื่อสถานที่ที่เอเจนท์เคลื่อนที่กำลังทำงานอยู่ เช่น หมายเลขพอร์ต เป็นต้น รวมกับที่อยู่ในเครือข่ายของเครื่องประมวลผล เช่น หมายเลขไอพี เป็นต้น

2.2.2 พฤติกรรมของเอเจนท์เคลื่อนที่

พฤติกรรมหรือลักษณะการทำงานของซอฟต์แวร์เอเจนท์เคลื่อนที่ ประกอบด้วยพฤติกรรมการทำงานต่าง ๆ ดังนี้

1. การสร้าง

เอเจนท์จะถูกสร้างขึ้นภายในสถานที่ ซึ่งการเริ่มต้นการสร้างเอเจนท์นั้น สามารถมาได้ทั้งจากเอเจนท์อื่น ๆ และระบบภายนอกสถานที่ที่ไม่ใช่เอเจนท์ก็ได้ การสร้างเอเจนท์ประกอบด้วย 3 ขั้นตอน ดังนี้ (Lange and Oshima. 1998)

1. การสร้างวัตถุของเอเจนท์ และการกำหนดตัวระบุ คลาสต้นแบบของเอเจนท์จะถูกอ่านเข้ามายังเอเจนท์แพลตฟอร์ม ซึ่งจะถูกใช้ในการสร้างวัตถุของเอเจนท์นั้น หลังจากสร้างวัตถุของเอเจนท์เสร็จสิ้น เอเจนท์แพลตฟอร์มจะกำหนดตัวระบุที่ไม่ซ้ำกันให้แก่เอเจนท์แต่ละตัว เพื่อใช้ในการอ้างถึง
2. การกำหนดค่าเริ่มต้น เมื่อเอเจนท์ถูกสร้างขึ้นเสร็จเรียบร้อย เอเจนท์จะทำการกำหนดค่าเริ่มต้นของตัวแปรต่าง ๆ ที่จำเป็นสำหรับการทำงาน เมื่อกำหนดค่าเริ่มต้นเสร็จสิ้นแล้ว จะถือว่าเอเจนท์ได้ถูกสร้าง และติดตั้งลงในเอเจนท์แพลตฟอร์มเสร็จสมบูรณ์ พร้อมสำหรับการทำงานแล้ว
3. เริ่มการประมวลผลโดยอัตโนมัติ เมื่อเอเจนท์ถูกติดตั้งลงในเอเจนท์แพลตฟอร์มเสร็จสมบูรณ์แล้ว จะเริ่มการประมวลผลตามคำสั่ง ซึ่งเอเจนท์สามารถทำงานได้โดยอิสระจากเอเจนท์แพลตฟอร์ม

2. การกำจัด

เอเจนท์สามารถถูกกำจัดได้ด้วยตัวเอง เอเจนท์อื่น หรือระบบที่ไม่ใช่เอเจนท์ ทั้งภายในและภายนอกเอเจนท์แพลตฟอร์มที่เอเจนท์นั้นทำงานอยู่ก็ได้ โดยการถูกกำจัดมีสาเหตุมาจาก มีการละเมิดกฎความมั่น คง ไม่มีระบบใด ๆ อ้างถึงเอเจนท์เป็ นเวลานาน เอเจนท์หมดอายุ และระบบที่เอเจนท์ทำงานอยู่ถูกปิด เป็นต้น กระบวนการกำจัดประกอบด้วย 2 ขั้นตอน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การเตรียมการสำหรับการกำจัด เอเจนท์ทำการบันทึกค่าตัวแปรต่าง ๆ และสถานการณ์ทำงานที่ทำอยู่เก็บไว้
2. การหยุดการประมวลผล เอเจนท์แพลตฟอร์มหยุดการประมวลผลของเอเจนท์

3. การเคลื่อนย้าย

เอเจนท์เคลื่อนที่มีความสามารถในการเคลื่อนย้ายตัวเองไปยังเซิร์ฟเวอร์ต่าง ๆ ในเครือข่าย เอเจนท์จะถูกส่งออกจากเซิร์ฟเวอร์ต้นทาง ไปยังเซิร์ฟเวอร์ปลายทาง โดยมีสถานที่ต้นทางและปลายทางเป็นผู้ควบคุมดูแลให้การส่งนั้นเสร็จสิ้นสมบูรณ์ ซึ่งการเริ่มต้นกระบวนการในการเคลื่อนย้าย สามารถถูกสั่งโดยเอเจนท์อื่น ระบบที่ไม่ใช่เอเจนท์ หรือตัวเอเจนท์เองก็ได้

เมื่อเอเจนท์ต้องการส่งตัวเองไปยังสถานที่ปลายทาง เอเจนท์จะทำการแจ้งไปยังสถานที่ต้นทางที่เอเจนท์กำลังทำงานอยู่ผ่านทางเอพีไอว่าต้องการเคลื่อนย้ายตัวเองไปยังสถานที่ปลายทางที่อยู่ ณ ตำแหน่งนี้ในเครือข่าย หลังจากนั้น สถานที่ต้นทางจะทำการหยุดการทำงานของเอเจนท์ พร้อมเก็บสถานะที!จำเป็นสำหรับการทำงาน แล้วจึงแปลงเอเจนท์ที่ประกอบด้วยเอเจนท์คลาสและสถานะ ไปเป็นสายอนุกรม เพื่อส่งไปยังสถานที่ปลาย

ทางผ่านช่องทางการสื่อสารที่เครื่องประมวลผลสร้างไว้ เมื่อถึงสถานที่ปลายทางจะทำการถอดสายอนุกรมกลับมาเป็นเอเจนท์ และเริ่มการทำงานต่อไป

4. การสื่อสาร

เอเจนท์มีความสามารถในการสื่อสารกับเอเจนท์อื่น ๆ ทั้งที่อยู่ในเอเจนท์แพลตฟอร์มเดียวกัน และต่างเอเจนท์แพลตฟอร์ม เอเจนท์สามารถในการส่งข้อความ หรือเรียกใช้งานเมธอดของเอเจนท์ตัวอื่นได้ถ้าได้รับอนุญาต นอกจากนั้น ยังสามารถแพร่กระจาย (Broadcast) ข้อความไปยังเอเจนท์หลายตัวพร้อม ๆ กันได้อีกด้วย รูปแบบการสื่อสารของเอเจนท์สามารถแบ่งออกได้เป็น 3 ลักษณะ ดังนี้

1. การส่งข้อความแบบนาวไทป์ เป็นวิธีที่นิยมและใช้มากที่สุด เป็นการส่งในลักษณะประสานจังหวะ คือเมื่อมีการส่งข้อความออกไปแล้ว จะหยุดการทำงาน และรอจนกว่าจะได้รับข้อความตอบกลับ จึงจะเริ่มทำงานต่อ
2. การส่งข้อความแบบฟิวเจอร์ไทป์ เป็นการส่งในลักษณะไม่ประสานจังหวะ คือเมื่อมีการส่งข้อความออกไปแล้ว เอเจนท์ที่ส่งไม่ต้องรอข้อความที่ตอบกลับ สามารถทำงานอย่างอื่นต่อไปได้เลย จึงมีความยืดหยุ่นในการทำงานมากกว่า และเหมาะกับการสื่อสารระหว่างเอเจนท์พร้อมกันหลาย ๆ ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การส่งข้อความแบบวันเวย์ไทป์ เป็นการส่งในลักษณะไม่ประสานจังหวะ เช่นเดียวกันกับแบบฟิวเจอร์ไทป์ แต่เอเจนท์รับจะไม่มีการตอบข้อความกลับมายังเอเจนท์ส่ง ซึ่งเหมาะสำหรับกรณีที่เอเจนท์ต้องการแจ้งข่าวสารบางอย่างโดยไม่ต้องการข้อความตอบกลับ

2.2.3 จุดเด่นของเอเจนท์เคลื่อนที่

เราสามารถนำเอาแนวคิดของเอเจนท์เคลื่อนที่ไปใช้ให้เกิดประโยชน์ เนื่องจากเอเจนท์เคลื่อนที่มีคุณสมบัติที่เป็นข้อดี ดังนี้

1. ช่วยลดปริมาณข้อมูลในเครือข่าย เอเจนท์เคลื่อนที่สามารถส่งตัวเอง ซึ่งประกอบไปด้วยคำสั่งในการทำงาน ไปยังเครื่องเป้าหมายเพื่อทำงานอย่างใดอย่างหนึ่งได้ ดังนั้นการทำงานต่าง ๆ รวมทั้งการแลกเปลี่ยนข้อมูลที่จำเป็นต่อการทำงาน จะเกิดขึ้นภายในเครื่องเป้าหมายเอง จึงช่วยลดปริมาณข้อมูลดิบภายในเครือข่ายที่แลกเปลี่ยนกันระหว่างเครื่องต้นทาง และปลายทางได้
2. ช่วยแก้ปัญหาความล่าช้าที่เกิดจากระบบเครือข่าย เนื่องจากเอเจนท์เคลื่อนที่สามารถเคลื่อนย้ายตัวเองไปทำงานยังเครื่องปลายทาง และสามารถตอบสนองต่อการเปลี่ยนแปลงต่าง ๆ ที่เกิดขึ้นในสภาพแวดล้อมของเครื่องปลายทางได้โดยตรง จึงไม่จำเป็นต้องมีการส่งคำสั่งเพื่อตอบสนองต่อการเปลี่ยนแปลงนั้นจากตัวควบคุมส่วนกลางผ่านระบบเครือข่ายซึ่งอาจทำให้เกิดความล่าช้าอันเนื่องมาจากระบบเครือข่ายได้
3. ทำงานแบบไม่ประสานเวลา คือการทำงานที่ทั้งสองฝ่ายไม่จำเป็นต้องเข้าสู่ระบบหรือเครือข่ายในเวลาเดียวกัน เช่น การรับส่งอีเมลเป็นต้น ซึ่งเหมาะกับการเชื่อมต่อเข้าสู่เครือข่ายที่มีต้นทุนสูง หรือไม่เสถียร เช่น ระบบจีพีอาร์เอส เป็นต้น โดยเมื่อเครื่องเชื่อมต่อเข้ากับเครือข่าย จะส่งเอเจนท์เคลื่อนที่ออกไปทำงาน จากนั้นจึงยกเลิกการเชื่อมต่อกับเครือข่าย เอเจนท์ที่ส่งออกไปนั้นจะเดินทางไปในเครือข่ายเพื่อไปทำงานยังเครื่องปลายทางเมื่อเอเจนท์ทำงานเสร็จ จะส่งผลลัพธ์กลับไปยังเครื่องต้นทางเมื่อเครื่องต้นทางกลับมาเชื่อมต่อกับระบบเครือข่ายอีกครั้ง
4. การทำงานด้วยภาวะอิสระ หมายถึง เอเจนท์เคลื่อนที่สามารถทำงานได้ด้วยตัวเอง โดยไม่ต้องการการสั่งงานจากเครื่องต้นทางตลอดเวลา เอเจนท์เคลื่อนที่จะนำเอาชุดคำสั่งของตัวเองไป และใช้ในการตัดสินใจการทำงานเอง ทำให้เครื่องต้นทางไม่จำเป็นต้องเชื่อมต่อกับเครือข่ายตลอดเวลา
5. สามารถทำงานได้ในระบบเครือข่ายที่มีความหลากหลายทั้งในด้านฮาร์ดแวร์ และซอฟต์แวร์ เนื่องจากการทำงานของเอเจนท์เคลื่อนที่เป็นอิสระจากแพลตฟอร์มของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ และการสื่อสารระหว่างเครือข่าย แต่จะขึ้นอยู่กับเอเจนท์แพลตฟอร์มเพียงอย่างเดียวเท่านั้น

6. มีความทนทานต่อความผิดพลาดที่เกิดขึ้น เนื่องจากเอเจนท์เคลื่อนที่สามารถรับรู้ถึงการเปลี่ยนแปลงต่าง ๆ ที่เกิดขึ้นของเอเจนท์แพลตฟอร์มในเครื่องที่กำลังทำงานอยู่ได้ เอเจนท์จะได้รับการเตือนเมื่อเครื่องที่มันกำลังทำงานอยู่กำลังถูกปิด และจะเคลื่อนย้ายตัวเองไปยังเครื่องอื่นในเครือข่าย และเริ่มทำงานต่อไปได้

2.2.4 งานประยุกต์ที่เหมาะในการนำเอเจนท์เคลื่อนที่ไปประยุกต์ใช้

การพัฒนาโปรแกรมโดยใช้หลักการของเอเจนท์เคลื่อนที่นั้น เหมาะกับงานหลายประเภท เช่น

1. การรวบรวมข้อมูลแบบกระจาย (Distributed Data Collection) โดยทั่วไปแล้ว การค้นหาข้อมูลในอินเทอร์เน็ตจะใช้ขั้นตอนวิธีการค้นหาแบบกระจาย (Distributed Search Algorithm) ซึ่งเราสามารถใช้แนวคิดของเอเจนท์และเอเจนท์ย่อย มาทำงานในลักษณะการประมวลผลแบบขนานได้ โดยที่เอเจนท์หลักจะปล่อยเอเจนท์ย่อยออกมาให้ทำงานแต่ละอย่างเอง แล้วเอเจนท์หลักก็จะทำการประมวลผลลัพธ์ที่เอเจนท์ย่อยส่งกลับมา เพื่อสร้างกลุ่มของผลลัพธ์ตามที่ผู้ใช้เลือก
2. การเฝ้าสังเกต (Monitoring) เราสามารถใช้เอเจนท์ในการตรวจสอบความเปลี่ยนแปลงของทรัพยากรในเครือข่าย หรือทำการรวบรวมข้อมูลต่าง ๆ ที่ทรัพยากรเหล่านั้นสร้างขึ้นมา โดยสร้างโปรแกรมให้เอเจนท์เคลื่อนที่ทำงานเป็นช่วงเวลา แล้วนำข้อมูลที่ตรวจสอบได้ในแต่ละโหนดกลับมายังระบบเอเจนท์ ตัวอย่างเช่น การใช้เอเจนท์ตรวจสอบเซิร์ฟเวอร์ที่ให้บริการข่าว เพื่อดูว่าข่าวใดที่ผู้ใช้สนใจ หรืออาจเป็นการตรวจสอบการใช้งานของเครือข่าย จะเห็นได้ว่า ลักษณะเด่นของการทำงานของเอเจนท์ในงานประเภทนี้ คือการทำงานด้วยภาวะอิสระ
3. การบริการการส่งสารสนเทศในเว็บ (Information Delivery) Push Technology กำลังเติบโตอย่างรวดเร็วในโลกของเวิลด์ไวด์เว็บ ซึ่งเป็นความสามารถในการให้บริการสารสนเทศที่มีการร้องขอสิ่งที่ต้องการออกไป บริการการให้ข้อมูลสารสนเทศในเวิลด์ไวด์เว็บ เป็นงานหนึ่งที่เหมาะกับการนำเอเจนท์มาใช้ ตัวอย่างของเอเจนท์ที่ใช้ในการส่ง (Pushing) สารสนเทศ คือ เอเจนท์ส่งข้อความ (Message Agent) และเอเจนท์สื่อบันเทิง (Entertainment Channel Agent) ซึ่งเป็นลักษณะสื่อประสม โดยจะเผยแพร่หรือกระจายข้อมูลให้แก่สมาชิกในกลุ่ม
4. การเจรจา (Negotiation) เอเจนท์สามารถดึงข้อมูลและตัดสินใจได้โดยไม่ได้อาศัยแต่เพียงข้อมูลทางสถิติอย่างเดียว เช่น ฐานข้อมูล แต่สามารถดึงข้อมูลจากเอเจนท์อื่นเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบการตัดสินใจได้ นั่น คือ เรากำลังประยุกต์ใช้งานกลุ่มเอเจนท์ ซึ่งเอเจนท์แต่ละตัวจะทำหน้าที่แต่ละอย่างแทนคน โดยมีความรู้และตารางการทำงานของตัวเอง จึงสามารถพูดคุยและตกลงเวลาในการพบกันระหว่างเอเจนท์ได้

5. การทำรายการ (Transaction) เอเจนท์สามารถทำให้กระบวนการที่มีการทำงานยาว ๆ ทำงานได้ง่ายขึ้น เช่น ในการทำรายการเตรียมการท่องเที่ยว นอกจากเอเจนท์จะสามารถให้ข้อมูลวัน ราคา และข้อกำหนดอื่น ๆ ในการท่องเที่ยวแก่ผู้ใช้แล้ว ยังสามารถที่จะออกเดินทางไปค้นหาข้อมูลในฐานข้อมูลของสายการบินและโรงแรมเพื่อทำการจองบัตรโดยสารและที่พักให้ งานประยุกต์ในลักษณะนี้เอเจนท์จะใช้ความสามารถในการทำงานเมื่อผู้ใช้ไม่ได้เชื่อมต่ออยู่ในเครือข่าย
6. การประมวลผลแบบขนาน (Parallel Processing) เอเจนท์ยอมให้มีการแบ่งงานของโปรแกรมประยุกต์เป็นส่วนย่อย ๆ แล้วส่งไปให้เอเจนท์ย่อย ๆ ประมวลผลที่เครื่องอื่นที่อยู่ในเครือข่าย เมื่องานเหล่านั้นสำเร็จ เอเจนท์ย่อย ๆ จะมารวมกันเป็น Multi-thread Application

2.3 ระบบการจัดการฐานข้อมูลแบบกระจาย

ระบบการจัดการฐานข้อมูลแบบกระจาย คือ มุมมองเชิงตรรกะระบบการจัดการฐานข้อมูลแบบกระจายจะมีเพียงหนึ่งฐานข้อมูลซึ่งจะแบ่งออกเป็นเซกเมนต์ต่าง ๆ โดยแต่ละเซกเมนต์นั้นจะถูกจัดเก็บลงในคอมพิวเตอร์หนึ่งเครื่องหรือมากกว่าภายใต้การควบคุมของแต่ละระบบการจัดการฐานข้อมูล เครื่องเซิร์พเวอร์แต่ละตัวจะมีการเชื่อมต่อสื่อสารกันในลักษณะของระบบเครือข่ายแต่ละไซต์ หรือแต่ละสาขา สามารถประมวลผลโดยอิสระ และผู้ใช้แต่ละคนสามารถเรียกใช้บริการข้อมูลของสาขาตน รวมทั้งสามารถประมวลผลและจัดเก็บข้อมูลลงในเครื่องเซิร์พเวอร์อื่น ๆ ได้ในเครือข่าย

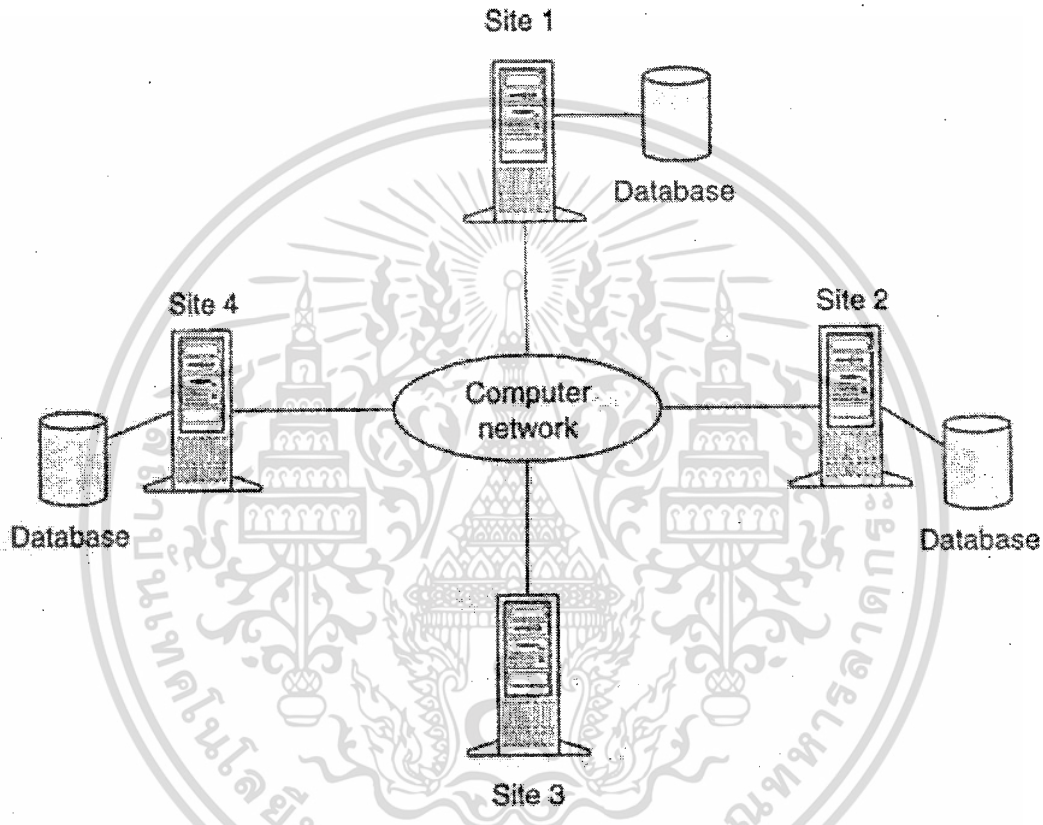
สำหรับในมุมมองทางเชิงกายภาพจะเห็นในลักษณะมีการจัดเก็บฐานข้อมูลที่กระจายกันในเครื่องเซิร์พเวอร์หลาย ๆ เครื่องที่อยู่ต่างพื้นที่และห่างไกลกันแต่สามารถเชื่อมโยงติดต่อกันได้โดยผ่านระบบเครือข่าย

ผู้ใช้สามารถเข้าถึงฐานข้อมูลแบบกระจายผ่านทางโปรแกรมประยุกต์โดยโปรแกรมประยุกต์จะเป็นตัวกำหนดให้เรียกใช้ข้อมูลเฉพาะส่วนสาขาตนหรือเรียกใช้ข้อมูลจากสาขาอื่น ดังนั้นระบบการจัดการฐานข้อมูลแบบกระจายก็คือซอฟต์แวร์ที่ใช้สำหรับจัดการฐานข้อมูลแบบกระจาย เพื่อให้ผู้ใช้งานสามารถมองผ่านสิ่งเหล่านี้ไปได้กล่าวคือผู้ใช้ไม่ต้องสนใจในรายละเอียดว่าฐานข้อมูลที่จัดเก็บนั้นจัดเก็บกระจายลงบนเครื่องเซิร์พเวอร์ต่าง ๆ อย่างไรแต่ผู้ใช้สามารถใช้งานฐานข้อมูลแบบกระจายเสมือนหนึ่งว่าเป็นฐานข้อมูลของตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูลแบบกระจาย (Distributed database) คือระบบที่มีการกระจายฐานข้อมูลไปยังจุดต่าง ๆ โดยข้อมูลที่จัดเก็บไว้ในแต่ละจุดสามารถเชื่อมโยงระหว่างกันได้ด้วยระบบเครือข่าย

ระบบการจัดการฐานข้อมูลแบบกระจาย (Distributed database management system) คือ ซอฟต์แวร์ที่ใช้สำหรับจัดการฐานข้อมูลแบบกระจายเพื่ออำนวยความสะดวกแก่ผู้ใช้งานในการปฏิบัติงานบนฐานข้อมูลแบบกระจาย แต่ทั้งนี้ก็ไม่จำเป็นว่าทุก ๆ จุดหรือทุก ๆ ไซต์จำเป็นต้องมีข้อมูลของสาขาตนเป็นของตนเองซึ่งสามารถพิจารณาจากรูปที่ 2.2



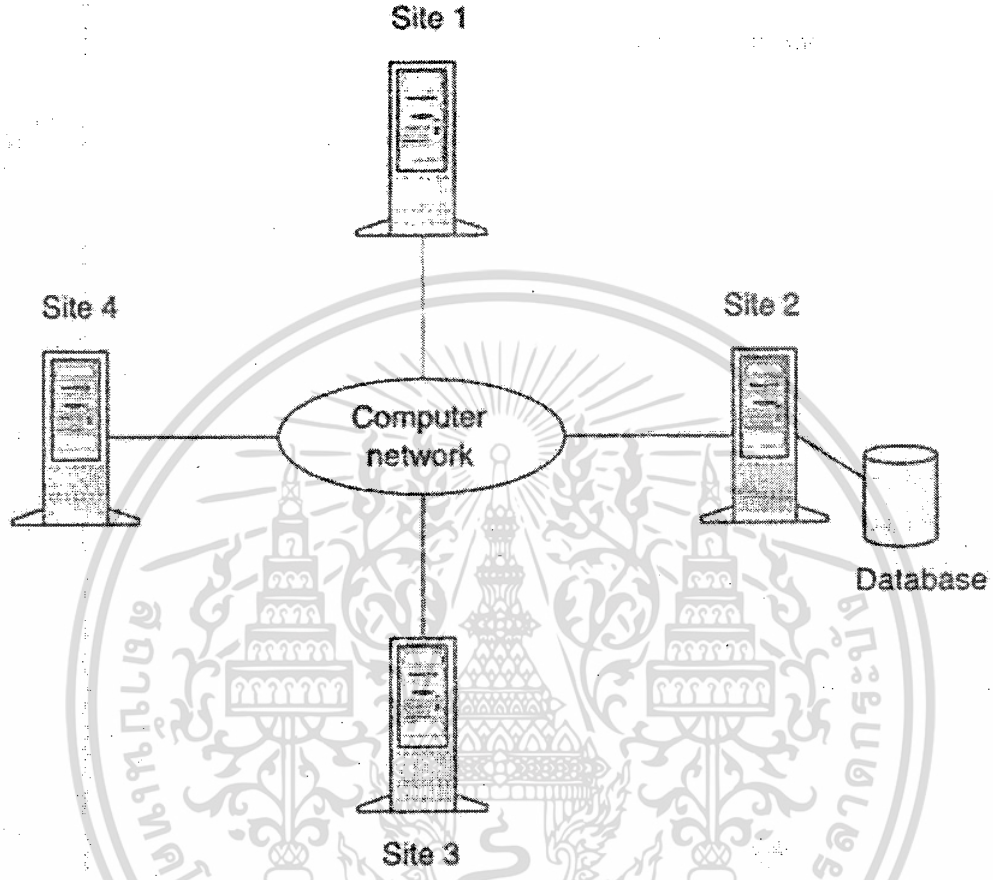
รูปที่ 2.2 ระบบการจัดการฐานข้อมูลแบบกระจาย

2.3.1 การประมวลผลแบบกระจาย (Distributed Processing)

ระบบการจัดการฐานข้อมูลแบบกระจาย กับการประมวลผลแบบกระจายนั้นมีความแตกต่างกันซึ่งถือเป็นสิ่งสำคัญยิ่ง สำหรับความเข้าใจในความแตกต่างของความหมายทั้งสองการประมวลผลแบบกระจายนั้น คือฐานข้อมูลแบบรวมศูนย์ที่สามารถเข้าถึงบนระบบเครือข่ายคอมพิวเตอร์ โดยข้อมูลที่เป็นศูนย์กลางผู้ใช้สามารถเข้าถึงระบบเครือข่าย แต่นั่นไม่ได้หมายถึงว่าเป็นระบบการจัดการฐานข้อมูลแบบกระจาย ซึ่งสามารอธิบายและเปรียบเทียบได้โดยพิจารณาจากรูปที่ 2.3 จะเห็นได้ว่าศูนย์กลางข้อมูลอยู่ที่ไซต์ 2 กับรูปที่ 2.2 เห็นได้ว่านะมีฐานข้อมูลที่จัดเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหลาย ๆ ไซต์ ซึ่งแต่ละไซต์อาจมีข้อมูลของสาขาตน นั่นหมายถึงไซต์งานที่ 3 นี้อาจเป็นไซต์ที่ใช้สำหรับเรียกดูข้อมูลจากไซต์หรือสาขาต่าง ๆ และหากมีความต้องการจัดเก็บข้อมูลก็อาจบันทึกข้อมูลลงบนไซต์อื่น ๆ ได้



รูปที่ 2.3 การประมวลผลแบบกระจาย

2.3.2 ระบบการจัดการฐานข้อมูลแบบขนาน (Parallel DBMS)

ความหมายของระบบการจัดการฐานข้อมูลแบบขนาน คือ ระบบการจัดการฐานข้อมูลที่รันบนหลายประมวลผล และดิสก์จัดเก็บข้อมูลได้ถูกออกแบบให้ดำเนินการในลักษณะคู่ขนาน ส่งผลให้ระบบทำงานมีประสิทธิภาพยิ่งขึ้น ทั้งในด้านของความยืดหยุ่นสามารถขยายได้ ความน่าเชื่อถือและความมีประสิทธิภาพ ซึ่งสามารถนำเครื่องเซิร์ฟเวอร์หลาย ๆ เครื่องมาเชื่อมต่อในลักษณะคู่ขนานเป็นเครือข่ายเสมือนกับเป็นการทำงานบนเครื่องเดียว และมีความสามารถในการประมวลผลด้วยความเร็วเทียบเท่ากับเครื่องระดับใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ระบบการจัดการฐานข้อมูลที่ใช้ในการศึกษา

ระบบการจัดการฐานข้อมูลที่ใช้การศึกษาในการทำโครงการนี้ เป็นฐานข้อมูลประมวลผลแบบขนานหลายหน่วยที่มีใช้ในปัจจุบัน และเป็นฐานข้อมูลที่สามารถนำมาใช้ในการศึกษาโดยไม่มีค่า cost คือ ฐานข้อมูลกรีนพลัม โดยมีเนื้อหาดังต่อไปนี้

ฐานข้อมูลกรีนพลัม (Greenplum Database) เป็นฐานข้อมูลประมวลผลแบบขนานหลายหน่วย (Massively Parallel Processing (MPP)) โดยใช้เทคโนโลยีโอเพนซอร์ซของโพสต์เกรสคิวแอล (PostgreSQL) MPP เป็นสถาปัตยกรรม shared nothing ซึ่งหมายถึงระบบที่มีหน่วยประมวลผลสองตัวขึ้นไปที่ร่วมกันปฏิบัติการ โดยแต่ละหน่วยประมวลผลจะมีหน่วยความจำ ระบบปฏิบัติการและดิสก์เป็นของตนเอง ฐานข้อมูลกรีนพลัมได้เพิ่มสถาปัตยกรรมระบบที่มีประสิทธิภาพสูงเพื่อกระจายโหลดของคลังข้อมูล (Data Warehouse) ในระดับหลาย ๆ เทระไบต์ และยังสามารถใช้ทรัพยากรของระบบในประมวลผลคำสั่งภาษาเอสคิวแอลแบบขนาน

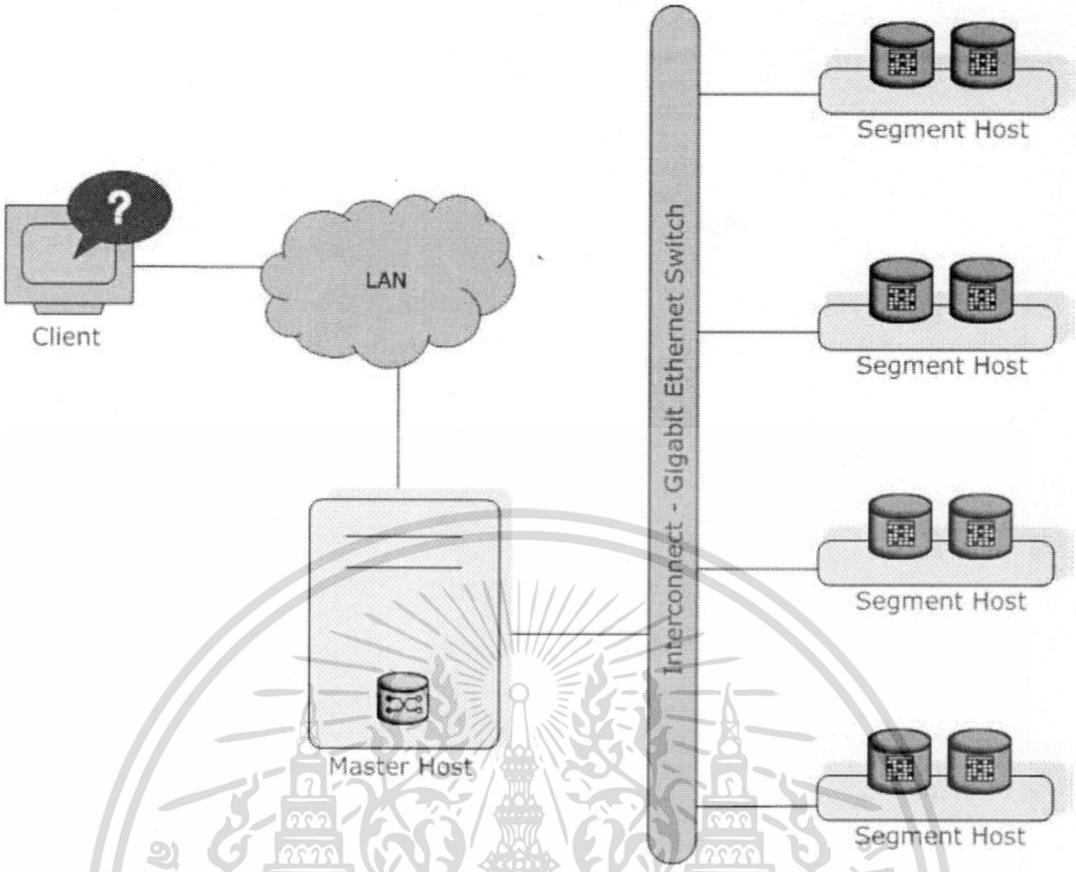
ดังนั้นฐานข้อมูลกรีนพลัมต้องใช้หลาย ๆ อินสแตนส์ของฐานข้อมูลโพสต์เกรสคิวแอล ในการปฏิบัติงานร่วมกันเสมือนเป็นหนึ่งฐานข้อมูลเดียว ผู้ใช้งานฐานข้อมูลกรีนพลัมจะใช้ฟังก์ชันต่าง ๆ เหมือนกับใช้ฐานข้อมูลโพสต์เกรสคิวแอล

ฐานข้อมูลโพสต์เกรสคิวแอลมีการปรับปรุงและเพิ่มเติมเพื่อสนับสนุนการทำงานที่เป็นแบบขนานของฐานข้อมูลกรีนพลัม ตัวอย่างเช่น system catalog, query planner, optimizer, query executor และส่วนประกอบ transaction manager ที่ซึ่งแก้ไขและปรับปรุงเพื่อให้สามารถดำเนินการกับคำสั่งภาษาเอสคิวแอลในแบบขนานในการใช้หลาย ๆ อินสแตนซ์ของฐานข้อมูลโพสต์เกรสคิวแอลในขณะที่มีการดำเนินการครั้งหนึ่ง การเชื่อมต่อของฐานข้อมูลกรีนพลัม (Greenplum interconnect) เป็นระบบเลเยอร์เครือข่ายที่เปิดใช้งานสื่อสารระหว่างอินสแตนซ์ของฐานข้อมูลโพสต์เกรสคิวแอล ทำให้ทำงานเสมือนเป็นหนึ่งฐานข้อมูลเดียว

2.4.1 สถาปัตยกรรมฐานข้อมูลกรีนพลัม

ฐานข้อมูลกรีนพลัมสามารถรองรับการเก็บข้อมูลและการประมวลผลจำนวนมากของข้อมูลที่ในกระจายโหลดในหลายเซิฟเวอร์ ซึ่งฐานข้อมูลกรีนพลัมก็คือกลุ่มของฐานข้อมูลโพสต์เกรสคิวแอลที่ทำงานร่วมกันและแสดงเป็นหนึ่งฐานข้อมูลเดียว มาสเตอร์ (Master) เป็นจุดแรกของฐานข้อมูลกรีนพลัมที่ไคลเอนต์เข้ามาติดต่อ และยืนยันคำสั่งภาษาเอสคิวแอล มาสเตอร์ทำหน้าที่ประสานงานกับอินสแตนซ์ต่าง ๆ ของฐานข้อมูลในระบบ โดยเซกเมนต์ (Segment) ช่วยรองรับการประมวลผลข้อมูลและเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 สถาปัตยกรรมฐานข้อมูลกรีนพลัม

ส่วนประกอบต่าง ๆ ของฐานข้อมูลกรีนพลัม

1. **มาสเตอร์**

มาสเตอร์เป็นจุดเริ่มต้นในการเข้าระบบฐานข้อมูลกรีนพลัม และฐานข้อมูลต้องอนุญาตให้ไคลเอนต์ทำการเชื่อมต่อ และตรวจสอบคำสั่งภาษาเอสคิวแอลจากผู้ใช้ที่ส่งเข้ามา เนื่องจากฐานข้อมูลกรีนพลัมนั้น ใช้โพสต์เกรสคิวแอล ผู้ใช้สามารถเชื่อมต่อฐานข้อมูลกรีนพลัมเช่นเดียวกับเชื่อมต่อกับฐานข้อมูลโพสต์เกรสคิวแอล การเชื่อมต่อฐานข้อมูลโดยใช้โปรแกรมเช่น psql หรือส่วนต่อประสานกับผู้ใช้ผ่านทาง JDBC หรือ ODBC

มาสเตอร์เป็นที่เก็บ global system catalog ซึ่งเป็น system table ที่เก็บ metadata เกี่ยวกับฐานข้อมูลกรีนพลัม และมาสเตอร์ไม่ได้เก็บข้อมูลของผู้ใช้ แต่ข้อมูลจะถูกเก็บในเซกเมนต์ต่าง ๆ มาสเตอร์ทำหน้าที่ตรวจสอบไคลเอนต์ที่เข้าใช้งาน รับคำสั่งภาษาเอสคิวแอลเข้ามา และกระจายงานโหลดไปยังเซกเมนต์ต่าง ๆ ประสานงานรวบรวมผลลัพธ์ที่ได้ในแต่ละเซกเมนต์ และแสดงผลลัพธ์สุดท้ายไปยังโปรแกรมไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เซกเมนต์

ในฐานข้อมูลกรีนพลัม เซกเมนต์เป็นส่งที่เก็บข้อมูลและประมวลผลคำสั่งภาษาเอสคิวแอล โดยตารางและอินเดกซ์ที่ถูกกระจายไปยังเซกเมนต์ที่สามารถใช้งานได้ในระบบฐานข้อมูลกรีนพลัม แต่ละเซกเมนต์เก็บข้อมูลที่แตกต่างกัน อินสแตนท์ของเซกเมนต์เป็นฐานข้อมูลที่ประมวลผลให้เซกเมนต์ ผู้ใช้ไม่สามารถติดต่อโดยตรงกับเซกเมนต์ของฐานข้อมูลกรีนพลัมได้ แต่สามารถเข้าผ่านมาสเตอร์

ส่วนของฮาร์ดแวร์ของฐานข้อมูลกรีนพลัม ให้ 1 เซกเมนต์ที่ใช้งานต่อ 1 หน่วยประมวลผล หรือ 1 CPU core ตัวอย่างเช่น ถ้าเซกเมนต์มี 2 dual-two processor จะสามารถมี 4 primary ของเซกเมนต์ 1 เครื่อง

3. ส่วนเชื่อมต่อ

ส่วนเชื่อมต่อ (interconnect) เป็นระดับ network layer ของฐานข้อมูลกรีนพลัม เมื่อผู้ใช้เชื่อมต่อกับฐานข้อมูลและส่งคำสั่งภาษาเอสคิวแอล โพเซสจะถูกสร้างในแต่ละเซกเมนต์ที่รองรับการทำงาน ส่วนเชื่อมต่อยังหมายถึงการสื่อสารโฟสเซสระหว่างเซกเมนต์ ส่วนเชื่อมต่อนี้ใช้มาตรฐาน Gigabit Ethernet switching fabric

โดยพื้นฐานแล้วส่วนเชื่อมต่อใช้ UDP (User Datagram Protocol) เพื่อสื่อสารในเครือข่าย สำหรับซอฟแวร์ของกรีนพลัมได้เพิ่มชุดยืนยันความถูกต้องและตรวจสอบที่ไม่ได้ทำโดย UDP จะใช้ TCP (Transmission Control Protocol) เพื่อความน่าเชื่อถือและมีประสิทธิภาพ สำหรับ TCP ฐานข้อมูลกรีนพลัมสามารถขยายได้สูงสุด 1,000 อินสแตนซ์ ถ้าเกินกว่านี้ UDP จะเป็นโปรโตคอลพื้นฐานในการเชื่อมต่อ

2.4.2 ระบบฐานข้อมูลแบบกระจาย

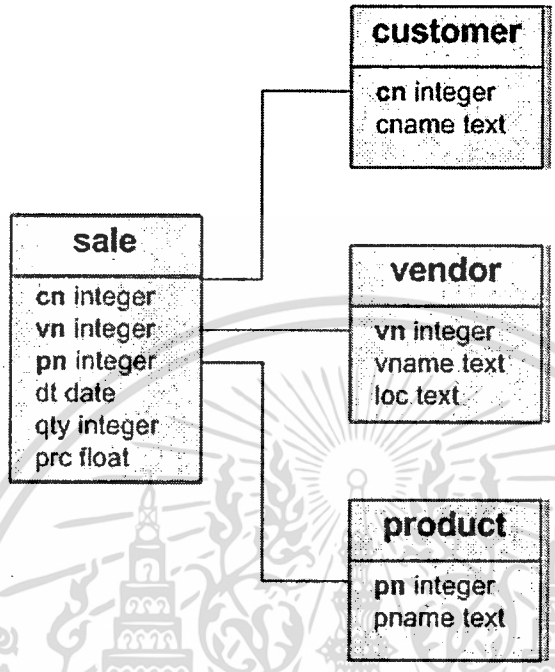
กรีนพลัมเป็นระบบฐานข้อมูลแบบกระจาย (distributed database system) ซึ่งหมายความว่าข้อมูลถูกเก็บไว้ในฐานข้อมูลที่เป็นพื้นที่ดิสก์มากกว่าหนึ่งเซิฟเวอร์ฐานข้อมูล ตามเซกเมนต์ในฐานข้อมูลกรีนพลัม ในแต่ละเซิฟเวอร์ฐานข้อมูลถูกเชื่อมต่อโดยเครือข่ายที่ติดต่อสื่อสาร ส่วนสำคัญในฐานข้อมูลแบบกระจายคือผู้ใช้และโปรแกรมไคลเอนต์สามารถเข้าถึงหนึ่งฐานข้อมูลเดียวสำหรับในกรีนพลัมคือเครื่องมาสเตอร์ ซึ่งในความเป็นจริงแล้วฐานข้อมูลจะถูกกระจายไปหลายเครื่อง

ฐานข้อมูลกรีนพลัมเก็บข้อมูลในหลาย ๆ เครื่องและหลาย ๆ อินสแตนซ์ของเซกเมนต์ เมื่อพิจารณาฐานข้อมูลในระดับตรรกะตามรูปคีย์หลัก (Primary Key) คือตัวอักษรเข้ม และคีย์นอก (Foreign Key) มีความสัมพันธ์ตามเส้นจากคีย์นอกอ้างถึงความสัมพันธ์กับคีย์หลัก ในที่นี้จะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

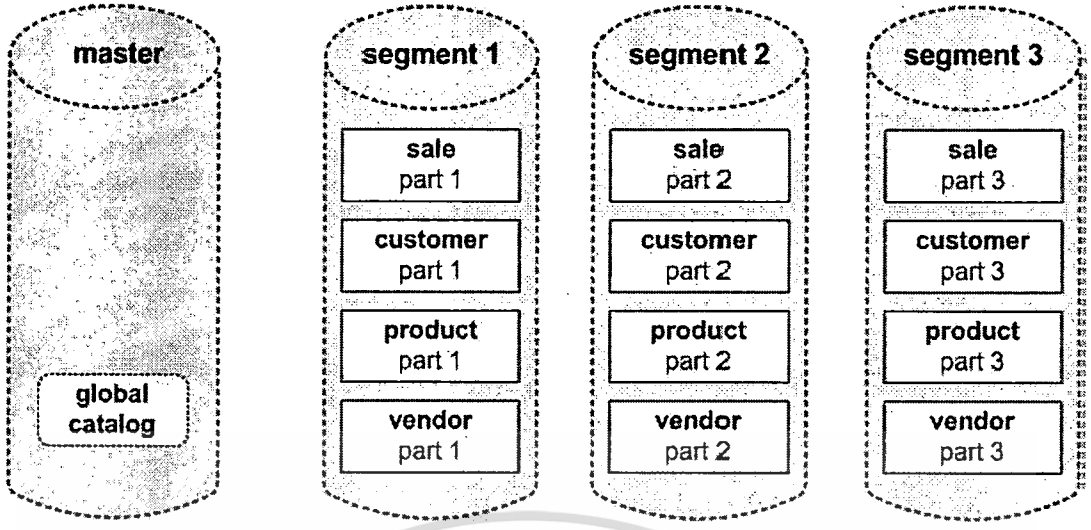
โครงสร้างแบบ star schema ซึ่งตาราง sale เรียนกว่าตาราง fact และตารางอื่น ๆ (customer, vender, product) เรียกว่าตาราง dimension



รูปที่ 2.5 ตัวอย่างฐานข้อมูล star schema

ในฐานข้อมูลกรีนพลัม ตารางทั้งหมดจะถูกกระจาย หมายความว่าตารางหนึ่งจะถูกแบ่งกลุ่มของแถวโดยไม่ซ้ำกันไปยังเซกเมกต์ต่าง ๆ ที่ทำงานอยู่ โดยใช้อัลกอริทึม hashing ในการแบ่ง ผู้ดูแลฐานข้อมูลจะต้องเลือกคีย์ hash (หนึ่งคอลัมภ์ขึ้นไป) เมื่อสร้างตารางฐานข้อมูลกรีนพลัมในระดับกายภาพทำการสร้างฐานข้อมูลในระดับตรรกะในแต่ละอินสแตนซ์ ซึ่งมาสเตอร์ไม่ได้เก็บข้อมูลของผู้ใช้ เก็บแต่ตาราง global catalog เท่านั้น เซกเมนต์เก็บข้อมูลกลุ่มแถวของแต่ละตารางที่กระจายกัน

144224
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ **2.6** การกระจายข้อมูลในตารางของฐานข้อมูลกรีนพลัม

เมื่อสร้างตารางในฐานข้อมูลกรีนพลัม ต้องเพิ่มส่วน DISTRIBUTED เพื่อกำหนด distribution policy ของตาราง โดย distribution policy กำหนดการแบ่งแถวของตารางในแต่ละเซกเมนต์ของฐานข้อมูลกรีนพลัม ซึ่งฐานข้อมูลกรีนพลัมมี distribution policy อยู่ 2 ชนิด

1. Hash Distribution ใช้ 1 คอลัมภ์ขึ้นไปในการเป็นคีย์สำหรับกระจายของตาราง คีย์สำหรับกระจายจะให้อัลกอริทึม hashing เพื่อระบุแต่ละแถวไปยังเซกเมนต์โดยเฉพาะ คีย์ที่มีค่าเหมือนกันจะยังเซกเมนต์เดียวกันตลอด การเลือกคอลัมภ์ที่มีค่าไม่ซ้ำกันเป็นคีย์สำหรับกระจายจะทำให้ข้อมูลกระจายได้มากที่สุด Hash distribution เป็นการกระจายโดยค่าเริ่มต้นของตาราง ถ้าไม่ได้กำหนด DISTRIBUTED จะใช้คีย์หลักหรือคอลัมภ์แรกมาเป็นคีย์สำหรับกระจาย
2. Random Distributed แถวกระจายไปยังเซกเมนต์วนรอบเป็น round-robin แถวที่มีค่าในคอลัมภ์เหมือนกันอาจจะไม่อยู่ที่เดียวกัน

2.4.3 การประมวลผลในฐานข้อมูลกรีพลัม

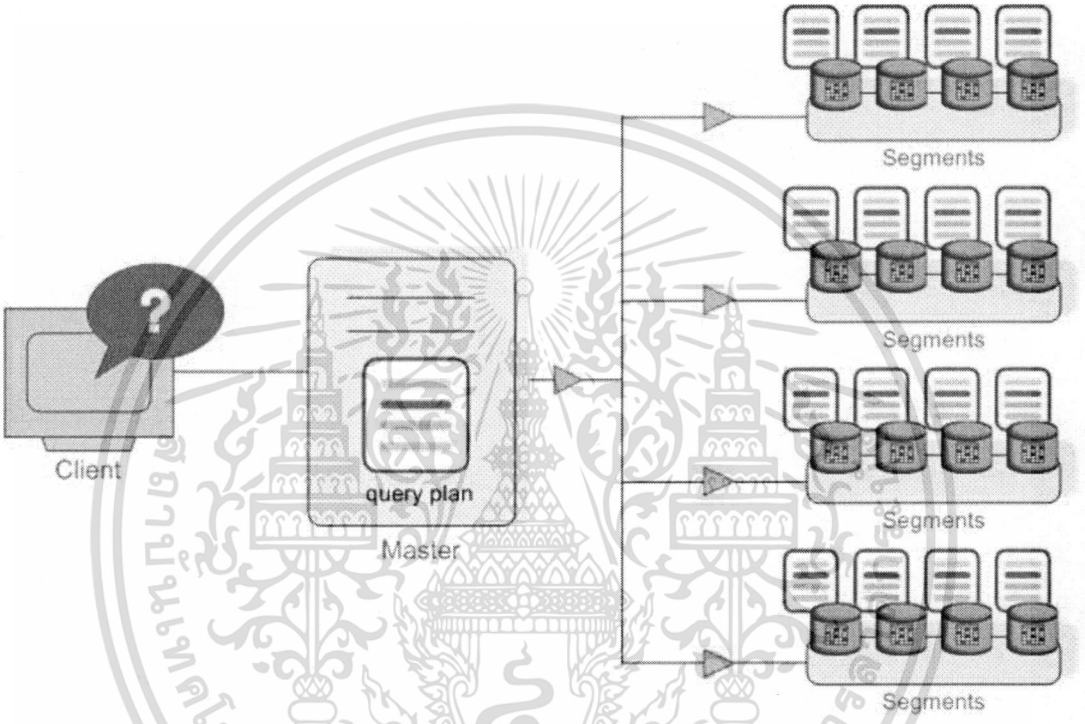
ผู้ใช้งานสามารถส่งคำสั่งภาษาเอสคิวแอล เข้าไปดำเนินงานคล้ายกับระบบฐานข้อมูลอื่น ๆ โดยสามารถเชื่อมต่อกับฐานข้อมูลกรีนพลัมผ่านไคลเอนต์เช่น psql

คำสั่งภาษาเอสคิวแอลถูกทำผ่านเครื่องมาสเตอร์ ที่ซึ่งตรวจสอบคำสั่งภาษาเอสคิวแอลเลือกแผนปฏิบัติที่ทำให้ได้คำตอบที่ดีที่สุด และสร้างทั้งแผนเอสคิวแอลแบบขนานและปลายทางที่เลือกไว้ซึ่งขึ้นอยู่กับคำสั่งภาษาเอสคิวแอลเครื่องมาสเตอร์เลือกจ่ายงานตามแผนไปยังเซกเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับปฏิบัติงาน แต่ละเซกเมนต์มีหน้าที่รับผิดชอบในการปฏิบัติในเครื่องฐานข้อมูลของตนเอง ซึ่งมีกลุ่มของข้อมูลที่ถูกกระจายมา

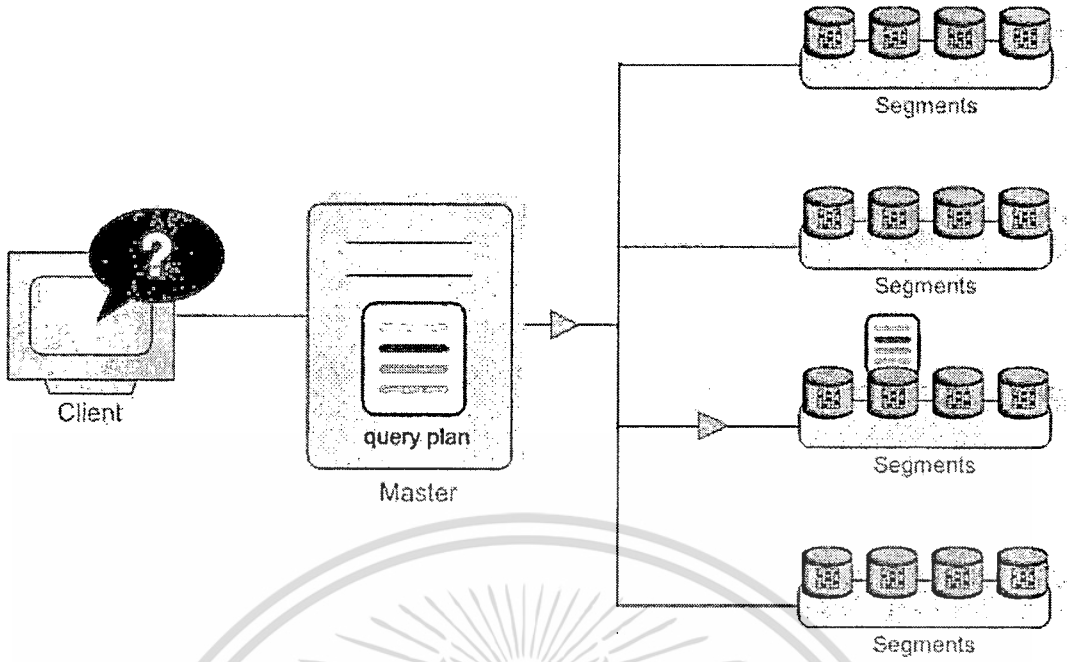
การดำเนินงานของฐานข้อมูลส่วนใหญ่ เช่น การกราดตรวจตาราง(scan table) การเชื่อม (join) การรวมกลุ่ม (aggregation) และการเรียง (sort) จะดำเนินการแบบขนานไปยังเซกเมนต์ต่าง ๆ พร้อม ๆ กัน การดำเนินการแต่ละครั้งจะถูกทำโดยฐานข้อมูลในเซกเมนต์นั้น โดยข้อมูลจะมีความสัมพันธ์กับเซกเมนต์อื่น ๆ



รูปที่ 2.7 แผนเอสคิวแอลแสดงการจ่ายงานแบบขนาน

บางคำสั่งภาษาเอสคิวแอลอาจจะเข้าถึงข้อมูลในเซกเมนต์เดียว เช่น มี 1 แถวที่การดำเนินการ INSERT, UPDATE, DETE หรือ SELECT หรือเอสคิวแอลนั้นคืนผลลัพธ์ที่มีจำนวนแถวไม่มากและถูกกรองโดยคอลัมน์ที่เป็นคีย์กระจาย ในหลาย ๆ เอสคิวแอล แผนเอสคิวแอลจะไม่ได้จ่ายงานไปทุกเซกเมนต์ จะดำเนินการเฉพาะเซกเมนต์ที่มีแถวที่กระทบเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แผนเอสคิวแอลแสดงการจ่ายงานแบบขนานไปยังจุดเป้าหมาย

แผนคำสั่งภาษาเอสคิวแอลเป็นกลุ่มของตัวดำเนินการข้องฐานข้อมูลกรีนพลัมที่จะช่วยค้นหาคำตอบจากเอสคิวแอล แต่ละโหนดหรือขั้นตอนในแผนที่แสดงการดำเนินงานในฐานข้อมูล เช่น การกราดตรวจตาราง (scan table) การเชื่อม (join) การรวมกลุ่ม (aggregation) และการเรียง (sort) แผนจะอ่านและดำเนินการจากข้างล่างไปข้างบน

นอกจากชนิดการดำเนินการของฐานข้อมูลที่กล่าวมาแล้ว ฐานข้อมูลกรีนพลัมได้เพิ่มการดำเนินการที่เรียกว่า "การเคลื่อนที่" (motion) การดำเนินการการเคลื่อนที่เกี่ยวกับการเคลื่อนย้ายทูเพิลระหว่างเซกเมนต์และระหว่างการประมวลผลของเอสคิวแอล แต่ไม่จำเป็นที่ทุกเอสคิวแอลต้องใช้การดำเนินการการเคลื่อนที่ ตัวอย่างเช่น เอสคิวแอลของการใช้ตาราง system catalog ในเครื่องมาสเตอร์ซึ่งไม่ได้เข้าถึงข้อมูลเพื่อย้ายข้อมูลข้ามเครื่อง

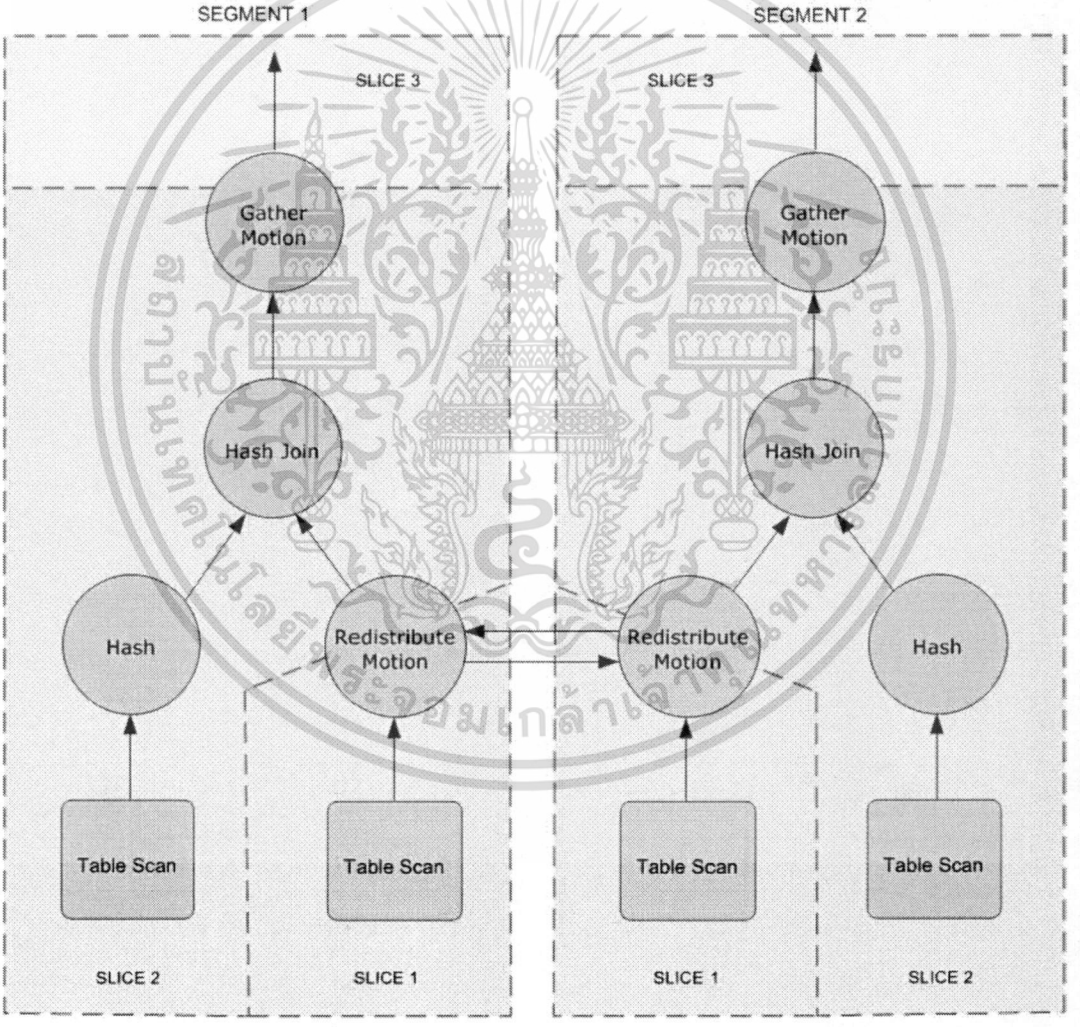
เพื่อให้เป็นแบบขนานสูงสุดในระหว่างเอสคิวแอลดำเนินการ ฐานข้อมูลกรีนพลัมได้แบ่งงานของแผนเอสคิวแอลไปเป็นส่วน ๆ (slice) ส่วนที่กล่าวถึงนี้เป็นส่วนหนึ่งของแผนงานที่ทำงานเป็นอิสระในระบบเซกเมนต์ แผนคำสั่งภาษาเอสคิวแอลที่ถูกแบ่งเป็นส่วน ๆ ที่ใดก็ตามที่มีการดำเนินการแบบการเคลื่อนไหว (motion) ในแผน ส่วนนั้นเป็นแต่ละด้านของการเคลื่อนไหว

ตัวอย่างเช่น เมื่อพิจารณาคำสั่งภาษาเอสคิวแอลที่มีการ join ระหว่างสองตาราง

```
SELECT customer, amount
FROM sales JOIN customer USING (cust_id)
WHERE dateCol = '04-30-2008';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงแผนเอสคิวแอล ที่แต่ละเซกเมนต์ได้คัดลอกแผนเอสคิวแอลและงานเป็นแบบขนาน ส่วนหนึ่งของแผนนี้เป็นการเคลื่อนไหวแบบการกระจายใหม่ (Redistribute Motion) โดยทำการย้ายแถวระหว่างเซกเมนต์เพื่อให้การ join สมบูรณ์ แผนถูกแบ่งเป็นส่วนตามด้านของการเคลื่อนไหวแบบการกระจายใหม่ (slice 1 และ slice 2). แผนเอสคิวแอลยังมีการเคลื่อนไหวแบบอื่นเรียกว่า การเคลื่อนไหวแบบรวบรวม (gather motion) การเคลื่อนไหวแบบรวบรวม คือ เมื่อเซกเมนต์ทั้งหลายส่งผลลัพธ์ไปยังเครื่องมาสเตอร์แล้วแสดงที่ไคลเอนต์ ตั้งแต่แผนเอสคิวแอลถูกแบ่งเป็นส่วนทุกครั้งที่มีการเคลื่อนไหวเกิดขึ้น แผนยังมีส่วนที่เป็นค่าเริ่มต้นที่อยู่บนสุดของแผน (slice 3) แต่ไม่ทุกเอสคิวแอลที่มีส่วนเกี่ยวข้องกับการเคลื่อนไหวแบบรวบรวม เช่น คำสั่ง CREATE TABLE x AS SELECT... จะไม่มีการเคลื่อนไหวแบบรวบรวม



รูปที่ 2.9 แผนคำสั่งภาษาเอสคิวแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 การบำรุงรักษาระบบที่ทำประจำ

ฐานข้อมูลต้องทำการบำรุงรักษาเพื่อช่วยให้มีประสิทธิภาพ โดยสามารถตั้งเวลาให้ทำงานได้ ดังนี้

1. การใช้คำสั่ง Vacuum และ Analyze

เนื่องในฐานข้อมูลมีการใช้งานข้อมูลทั้งการใช้คำสั่ง delete หรือ update ข้อมูลที่เกิดขึ้นกับพื้นที่ในส่วนดิสก์ ซึ่งถ้าฐานข้อมูลมีการใช้คำสั่ง delete หรือ update เป็นจำนวนมาก จะทำให้มีแถวของข้อมูลที่ไม่ใช้แล้ว การใช้คำสั่ง Vacuum ทำให้ตารางได้พื้นที่ว่างจากการกำจัดแถวที่ไม่ใช้งานแล้ว แต่ขนาดของตารางจะยังไม่ลด

การเพิ่มตัวเลือก Full ตามหลังคำสั่ง Vacuum จะทำให้ได้ขนาดของตารางที่มีข้อมูลอยู่จริง แตกต่างกับคำสั่ง Vacuum โดยไม่ใส่ตัวเลือก จะทำให้ตารางจะยังมีขนาดเท่าเดิม แต่การใช้คำสั่ง Vacuum โดยใส่ตัวเลือก Full นี้จะทำให้ตาราง Lock เป็นโหมด Exclusive ซึ่งหมายความว่าจะไม่มีทรานแซกชันอื่นเข้ามาใช้งานตารางนี้ จนกว่าจะทำการใช้คำสั่ง Vacuum เสร็จ

การเพิ่มตัวเลือก Analyze ตามหลังคำสั่ง Vacuum นอกจากกำจัดแถวที่ไม่ใช้งานแล้ว ยังเป็นทำการรวบรวมค่าสถิติของตาราง เช่น จำนวนแถวและเพจที่ใช้งานอยู่

การทำงานในแต่ละวันของฐานข้อมูล จะมีการใช้คำสั่ง Create และการใช้คำสั่ง Drop ตารางจำนวนทำให้ตาราง System Catalog มีแถวที่ไม่ใช้งานเพิ่มขึ้นเรื่อย ๆ ดังนั้นในแต่ละวันควรตั้งเวลาเพื่อทำการใช้คำสั่ง Vacuum และ Analyze ตาราง System Catalog เพื่อช่วยเรื่องประสิทธิภาพของการใช้ฐานข้อมูล แต่ถ้าใส่ตัวเลือก Full ด้วยจะทำให้ช่วยเรื่องประสิทธิภาพของฐานข้อมูลดียิ่งขึ้น แต่ว่าให้เกิดการ Lock แบบโหมด Exclusive ในหนึ่งสัปดาห์ควรตั้งทำงานอย่างน้อย 1 วันในช่วงที่ไม่ใครใช้งานฐานข้อมูล

2. การสร้างดัชนีใหม่

การตั้งเวลาใช้คำสั่ง Reindex สำหรับตาราง System Catalog ควรตั้งในเวลาช่วงที่มีผู้ใช้งานเข้ามาใช้ฐานข้อมูลน้อย เนื่องจากทำให้การ Lock แบบโหมด Exclusive เหมือนกัน

ข้อแนะนำการ Reindex ตาราง ไม่ว่าตารางจะมีดัชนีแบบ B-tree หรือ Bitmap การใช้คำสั่ง Drop index แล้วสร้างดัชนีจะทำได้ไวกว่าการ Reindex

2.4.5 การวิเคราะห์เอสคิวแอล

ฐานข้อมูลกรีนพลัมให้ใช้เส้นทางการทำงานตามคำสั่งสืบค้น (Query Optimization) เป็นแบบแบบ Cost-based Optimization (CBO) เราสามารถใช้คำสั่ง Analyze เพื่อสร้างค่าสถิติต่าง ๆ ที่เกี่ยวกับอ็อบเจกต์ในฐานข้อมูล เช่น จำนวนระเบียนในตารางหรือจำนวนค่าที่แตกต่างกันของคีย์ในดัชนี โดยอาศัยค่าสถิติเหล่านั้น COB จะทำการประเมินค่า cost ของเส้นทางต่าง ๆ การทำงานและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกเส้นทางที่เสียค่า cost น้อยที่สุด ในกรณีที่เราตั้งใจจะใช้ Cost-based Optimizer เราจะต้องไม่ลืมที่จะส่งคำสั่ง Analyze บ่อย ๆ เพื่อให้ในฐานข้อมูลของเรามีค่าสถิติที่ทันสมัยเสมอ

ตัวอย่างการหา cost ของคำสั่งเอสคิวแอล กำหนดให้มีตาราง date_dim และตาราง job_history โดยกำหนดคำสั่งสร้างตารางดังนี้

```
CREATE TABLE hr.date_dim
(
date_key timestamp without time zone NOT NULL,
date_week timestamp without time zone NOT NULL,
date_month timestamp without time zone NOT NULL,
date_quarter timestamp without time zone NOT NULL,
date_year timestamp without time zone NOT NULL
)
DISTRIBUTED BY (date_key);


CREATE TABLE hr.job_history
(
employee_id numeric NOT NULL,
start_date timestamp without time zone NOT NULL,
end_date timestamp without time zone,
job_id character varying(10),
department_id numeric,
CONSTRAINT job_history_pkey PRIMARY KEY (employee_id, start_date)
)
DISTRIBUTED BY (employee_id, start_date);
```

ใช้คำสั่งเอสคิวแอลในการสืบค้นข้อมูลตามด้านล่าง

```
SELECT *
FROM hr.job_history jh
JOIN hr.date_dim d ON d.date_key BETWEEN jh.start_date AND jh.end_date;
```

ขั้นตอนการตรวจสอบคำสั่งเอสคิวแอล

1. ในกรณีคำสั่งเอสคิวแอลเข้าไปดำเนินการในฐานข้อมูลโดยเวลานาน มีสาเหตุสำคัญดังนี้

1.1. ตารางที่ใช้ในคำสั่งเอสคิวแอลที่กำลังดำเนินการในฐานข้อมูลมีการ Lock แบบโหมด Exclusive ทำให้คำสั่งเอสคิวแอลนี้ ต้องรอจนกว่าคำสั่งเอสคิวแอลก่อนหน้าใช้งานตารางเสร็จก่อน จึงสามารถเข้าไปดำเนินการต่อได้ วิธีการตรวจสอบคือ คำสั่งเอสคิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอลมี 2 ตารางที่ใช้งานคือ date_dim และ job_history สามารถตรวจสอบในตาราง pg_lock ซึ่งเป็นตาราง System catalog ว่ามีคำสั่งเอสคิวแอลอื่น ใช้งานตาราง date_dim และ job_history และ Lock แบบโหมด Exclusive หรือไม่ ถ้าตรวจสอบเจอว่ามี แสดงว่าคำสั่งเอสคิวแอลจะยังไม่สามารถดำเนินการได้

1.2. ตารางที่ใช้ในคำสั่งเอสคิวแอลที่กำลังดำเนินการในฐานข้อมูลมีการกระจายในฐานข้อมูล ไม่กระจายทุกเซกเมนต์เกิดจากเลือกคีย์กระจาย (Distribute key) ไม่ถูกต้องหรือไม่ได้กำหนดตอนสร้างตาราง ซึ่งเริ่มต้นฐานข้อมูลจะใช้คอลัมน์แรกของตารางเป็นคีย์กระจาย แต่เมื่อใช้งานคำสั่งเอสคิวแอลแล้วไม่ได้ใช้คีย์กระจายในการ Join หรือ Where จะทำให้ฐานข้อมูลสร้างคีย์กระจายใหม่ชั่วคราวในขณะที่ดำเนินการในฐานข้อมูล ในกรณีที่ตารางมีขนาดใหญ่มาก จะทำให้ฐานข้อมูลมีประสิทธิภาพลดลง รวมถึงกระทบกับคำสั่งเอสคิวแอลอื่น ๆ ที่กำลังดำเนินการในฐานข้อมูลด้วย

2. ในกรณีคำสั่งเอสคิวแอลเข้าไปดำเนินการในฐานข้อมูล แล้วมีค่า cost สูง

คำสั่งเอสคิวแอลที่มีค่า cost สูงทำให้ไม่สามารถเข้าไปรันในฐานข้อมูลได้ เนื่องจากมีการกั้นด้วยคิวการใช้ทรัพยากร หรือถ้าเข้าไปดำเนินการได้ก็จะกระทบกับประสิทธิภาพของฐานข้อมูล การตรวจสอบค่า cost ของคำสั่งเอสคิวแอล โดยเพิ่มคำสั่ง Explain ดังนี้

```
Explain SELECT *
FROM hr.job_history jh
JOIN hr.date_dim d ON d.date_key BETWEEN jh.start_date AND jh.end_date;
```

ผลลัพธ์ของการดำเนินการ ดังนี้

```
Gather Motion 2:1  (slice2; segments: 2)  (cost=2.42..2301.81 rows=2494 width=78)
 ->  Nested Loop  (cost=2.42..2301.81 rows=2494 width=78)
     Join Filter: d.date_key >= jh.start_date AND d.date_key   Seq Scan on date_dim d  (cost=0.00..54.89 rows=2245 width=40)
     ->  Materialize  (cost=2.42..2.62 rows=10 width=38)
        ->  Broadcast Motion 2:2  (slice1; segments: 2)  (cost=0.00..2.40 rows=10 width=38)
           ->  Seq Scan on job_history jh  (cost=0.00..2.10 rows=5 width=38)
```

จากผลลัพธ์แผนของคำสั่งเอสคิวแอลนี้มีค่า cost รวมเท่ากับ 2301.81 ในกรณีที่คำสั่งเอสคิวแอลมีค่า cost สูงสาเหตุมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1. ตารางที่ใช้ในคำสั่งเอสคิวแอลไม่ได้ใช้คำสั่ง Analyze เพื่อสร้างค่าสถิติต่าง ๆ ที่เกี่ยวกับอ็อบเจกต์ในฐานข้อมูลที่เกิดจากการ Delete หรือ Update ข้อมูล ดังนั้นถ้าคำสั่งเอสคิวแอลเดิมที่เคยใช้ประจำแต่มีค่า cost สูงผิดปกติ ให้ Analyze ตารางก่อน

2.2. การเขียนคำสั่งเอสคิวแอลไม่เหมาะสม ตัวอย่างเช่น

```
select distinct job_id from hr.job_history
```

จะทำให้มี cost สูงกว่าคำสั่งเอสคิวแอล

```
select job_id from hr.job_history group by job_id
```

ซึ่งทั้งสองคำสั่งเอสคิวแอลได้ผลลัพธ์เหมือนกัน นอกจากยังมีวิธีการเขียนคำสั่งเอสคิวแอลอื่น ๆ อีกมาก เมื่อตรวจสอบเจอคำสั่งเอสคิวแอลที่มีค่า cost สูง จะตรวจสอบผลจาก Explain ก่อนว่าจุดไหนที่ให้คำสั่งเอสคิวแอลมีค่า cost สูง ตัวอย่างเช่น ผลลัพธ์จากการ Explain คำสั่งเอสคิวแอล

```
Gather Motion 2:1  (slice2; segments: 2)  (cost=2.42..2301.81 rows=2494 width=78)
 ->  Nested Loop  (cost=2.42..2301.81 rows=2494 width=78)
     Join Filter: d.date_key >= jh.start_date AND d.date_key   Seq Scan on date_dim d  (cost=0.00..54.89 rows=2245 width=40)
     ->  Materialize  (cost=2.42..2.62 rows=10 width=38)
        ->  Broadcast Motion 2:2  (slice1; segments: 2)  (cost=0.00..2.40 rows=10 width=38)
           ->  Seq Scan on job_history jh  (cost=0.00..2.10 rows=5 width=38)
```

จุดที่ทำให้ค่า cost สูงคือ Nested Loop โดยฐานข้อมูลกรีนพลัมการเกิด Nested Loop ควรพยายามหลีกเลี่ยง โดยเขียนคำสั่งเอสคิวแอลใหม่ได้เป็น

```
SELECT *
FROM (
SELECT *, start_date + interval '1 day' * (generate_series(0, (EXTRACT('days' FROM end_date - start_date)::int))) AS mydate
FROM hr.job_history
) AS jh
JOIN hr.date_dim d ON jh.mydate = d.date_key;
```

เมื่อตรวจสอบผลลัพธ์จากการ Explain ใหม่จะเห็นว่า cost น้อยลงคือ 68.96

```
Gather Motion 2:1  (slice2; segments: 2)  (cost=2.70..68.96 rows=7 width=166)
 ->  Hash Join  (cost=2.70..68.96 rows=7 width=166)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Hash Cond: d.date_key = jh.mydate
-> Seq Scan on date_dim d  (cost=0.00..54.89 rows=2245 width=40)
-> Hash  (cost=2.58..2.58 rows=5 width=126)
   -> Redistribute Motion 2:2  (slice1; segments: 2)  (cost=0.00..2.58 rows=5
width=126)
      Hash Key: jh.mydate
      -> Result  (cost=0.00..2.27 rows=5 width=38)
         -> Seq Scan on job_history  (cost=0.00..2.27 rows=5 width=38)
```

ทั้งนี้ การเฝ้าระวังประสิทธิภาพของฐานข้อมูลเมื่อเจอคำสั่งเอสคิวแอลที่มีค่า cost สูง อาจแจ้งสาเหตุเบื้องตนให้ผู้ดูแลฐานข้อมูลทราบ แต่ยังไม่สามารถปรับคำสั่งเอสคิวแอลให้มีความเหมาะสมที่สุดได้ จึงต้องให้ผู้ดูแลฐานข้อมูลพิจารณา

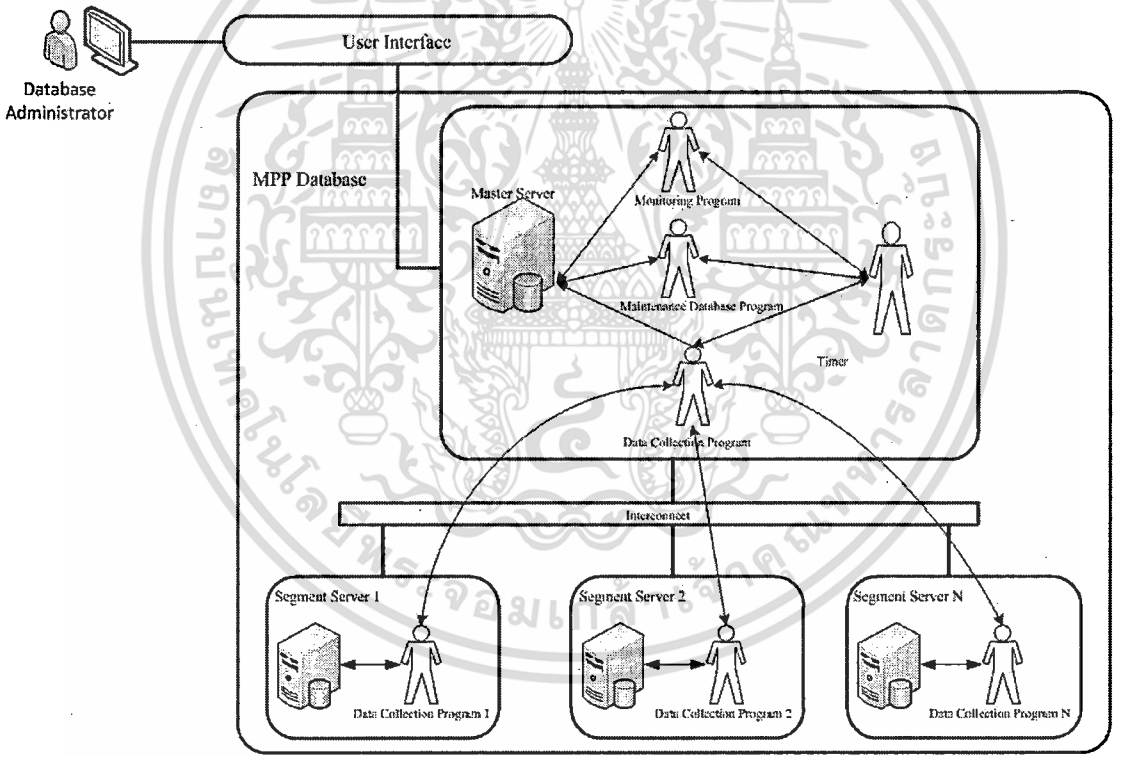
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และการออกแบบระบบ

หลังจากได้ศึกษาลักษณะขั้นตอนการทำงาน ปัญหาที่เกี่ยวข้องแล้ว ในขั้นตอนนี้จะทำการวิเคราะห์ และออกแบบโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์

3.1 สถาปัตยกรรมของเอเจนท์เฝ้าสังเกตประสิทธิภาพของฐานข้อมูล

โปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์ได้ออกแบบไว้ ดังรูป 3.1



รูปที่ 3.1 สถาปัตยกรรมโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์

1. ส่วนต่อประงานกับผู้ใช้ (User Interface) ทำหน้าที่เป็นตัวกลางในการติดต่อสื่อสารระหว่างผู้ใช้และระบบ เพื่อแสดงผลข้อมูลต่าง ๆ และบันทึกค่าต่าง ๆ ที่ต้องการลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมเอเจนท์ (Agent) เป็นโปรแกรมที่ทำงานเป็นเอเจนท์เคลื่อนที่ที่ต้องเดินทางผ่านไปยังเซิร์ฟเวอร์ต่างเพื่อรวบรวมข้อมูล จัดเก็บข้อมูล และเฝ้าสังเกต ซึ่งประกอบด้วย

2.1. ตัวกำหนดเวลา (Timer) จะตรวจสอบเวลาการทำงานที่ถูกบันทึกโดยผู้ดูแลฐานข้อมูล เมื่อถึงเวลาที่กำหนด ตัวกำหนดเวลาจะเรียกโปรแกรมต่าง ๆ ทำงาน โดยขึ้นอยู่กับเงื่อนไขที่ถูกบันทึกไว้ หลังจากนั้นจึงกำจัดตัวเองออกจากระบบ

2.2. โปรแกรมรวบรวมข้อมูล (Data Collection Program) เป็นโปรแกรมที่มีหน้าที่จัดรวบรวมข้อมูลต่าง ๆ ที่เกี่ยวกับประสิทธิภาพของฐานข้อมูล เช่น ข้อมูลเอสคิวแอลที่ได้ดำเนินการแล้ว ข้อมูลการล็อคเทเบิล ข้อมูลการโหลดของเซิร์ฟเวอร์ และข้อมูลทรัพยากรเครื่อง เป็นต้น โดยมีการเรียกใช้งานจากตัวกำหนดเวลา มีการติดตั้งโปรแกรมรวบรวมข้อมูลไว้แต่ละเซิร์ฟเวอร์เพื่อรวบรวมและมาจัดเก็บลงฐานข้อมูล หลังจากนั้นจึงกำจัดตัวเองออกจากระบบ

2.3. โปรแกรมบำรุงรักษาฐานข้อมูล (Maintainance Database Program) เป็นโปรแกรมที่ทำหน้าบำรุงรักษาฐานข้อมูลตามที่ผู้ดูแลฐานข้อมูลกำหนด โดยมีการเรียกใช้งานจากตัวกำหนดเวลา หลังจากนั้นจึงกำจัดตัวเองออกจากระบบ

2.4. โปรแกรมเฝ้าสังเกต (Monitoring Program) เป็นโปรแกรมที่มีหน้าที่ตรวจสอบข้อมูลที่ถูกบันทึกจากโปรแกรมรวบรวมข้อมูล โดยมีการเรียกใช้งานจากตัวกำหนดเวลา ซึ่งโปรแกรมเฝ้าสังเกตจะเปรียบเทียบค่าที่กำหนดโดยผู้ดูแลฐานข้อมูลว่ามีค่าประเภทไหนบ้างที่เกินกำหนด ถ้าพบว่าค่าประเภทไหนเกินกำหนดจะส่งอีเมลแจ้งเตือนผู้ดูแลฐานข้อมูล หลังจากนั้นจึงกำจัดตัวเองออกจากระบบ

3. ฐานข้อมูลประมวลผลแบบขนานหลายหน่วย (MPP Database) เป็นส่วนของฐานข้อมูลที่ต้องเฝ้าระวังประสิทธิภาพ ประกอบด้วย

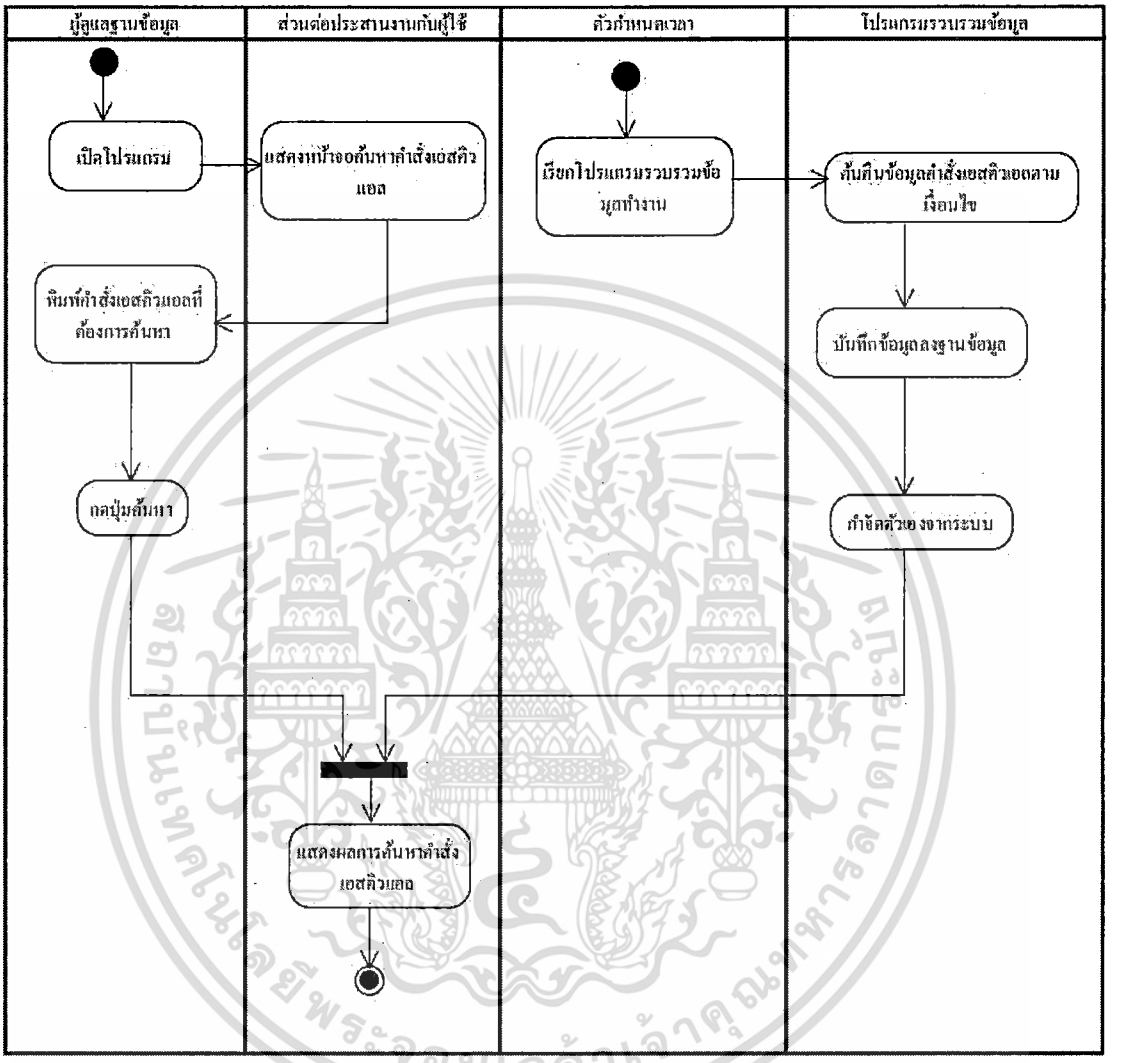
3.1. เซิร์ฟเวอร์มาสเตอร์ (Master server) เป็นเซิร์ฟเวอร์ที่ประสานงานกับผู้ใช้กับฐานข้อมูล ซึ่งเซิร์ฟเวอร์มาสเตอร์ทำหน้าที่รับคำสั่งเอสคิวแอล แล้วส่งไปประมวลผลที่เซิร์ฟเวอร์เซกเมนต์ที่เก็บข้อมูลอยู่ เมื่อประมวลผลเสร็จจะส่งผลลัพธ์มายังเซิร์ฟเวอร์มายังเซิร์ฟเวอร์มาสเตอร์ โดยเซิร์ฟเวอร์มาสเตอร์ยังเก็บการตั้งค่าของฐานข้อมูลอีกด้วย

3.2. เซิร์ฟเวอร์เซกเมนต์ (Segment server) เป็นเซิร์ฟเวอร์ที่เก็บข้อมูล ซึ่งมีอย่างน้อย 2 เซิร์ฟเวอร์ขึ้นไป โดยการเก็บข้อมูลจะมีลักษณะกระจายข้อมูลของตารางไปยังเซิร์ฟเวอร์เซกเมนต์ต่าง ๆ ทำหน้าประมวลผลข้อมูลที่ได้รับจากเซิร์ฟเวอร์มาสเตอร์ จากนั้นจึงส่งผลลัพธ์ไปยังเซิร์ฟเวอร์มาสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการทำงานของระบบ

แอกทิวิตีไดอะแกรมแสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการค้นหาข้อมูลเอสคิวแอล สามารถแสดงได้ดังรูปที่ 3.2



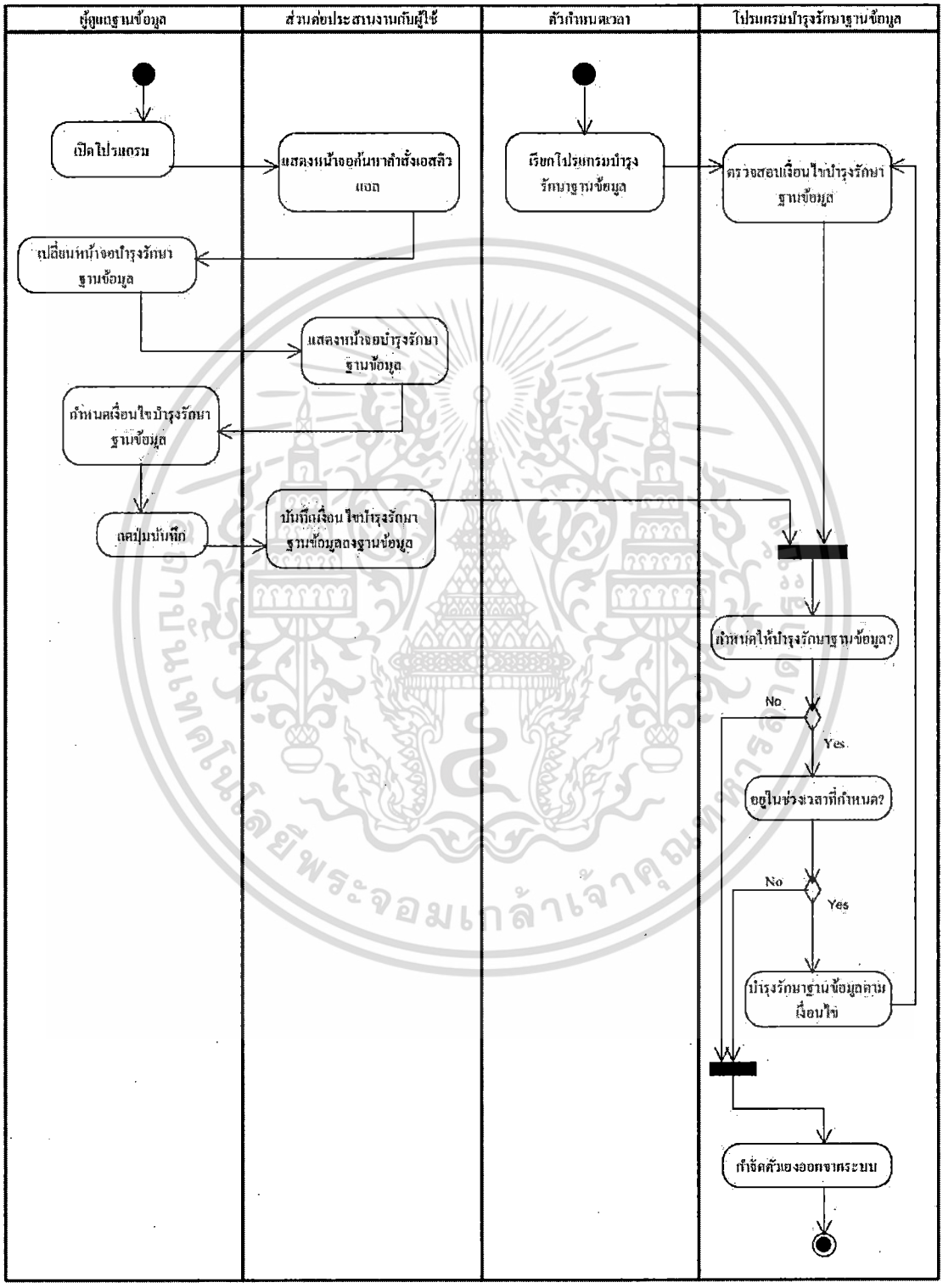
รูปที่ 3.2 แสดงขั้นตอนการทำงานเมื่อผู้ใช้ต้องการค้นหาข้อมูลเอสคิวแอล

จากรูปที่ 3.2 เมื่อผู้ดูแลระบบฐานข้อมูลเปิดโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย จะแสดงหน้าจอค้นหาคำสั่งเอสคิวแอล ผู้ดูแลระบบฐานข้อมูลต้องพิมพ์คำสั่งเอสคิวแอลที่ต้องการค้นหา แล้วจึงกดปุ่มค้นหา

ในส่วนของเอเจนท์รวบรวมข้อมูล จะประกอบด้วย ตัวกำหนดเวลา และโปรแกรมรวบรวมข้อมูล เมื่อถึงเวลาตัวกำหนดเวลาจะเรียกโปรแกรมรวบรวมข้อมูลทำงาน โดยโปรแกรมรวบรวมข้อมูลจะค้นคืนข้อมูลเอสเอลคิวแอลที่ได้ถูกดำเนินการแล้ว มาจัดการลงฐานข้อมูลตามเงื่อนไข และกำจัดตัวเองออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการที่ผู้ดูฐานข้อมูลกดปุ่มค้นหา หน้าจอแสดงผลค้นหาคำสั่งเอสคิวแอลจะแสดงคำสั่งเอสคิวแอลตามเงื่อนไขที่ค้นหา และแสดงรายละเอียดของคำสั่งเอสคิวแอล จากข้อมูลที่ถูกบันทึกโดยเอเจนท์รวบรวมข้อมูล



รูปที่ 3.3 แสดงขั้นตอนการทำงานเมื่อผู้ใช้กำหนดให้บำรุงรักษาฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 เมื่อผู้ดูแลระบบฐานข้อมูลเปิดโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย จะแสดงหน้าจอค้นหาคำสั่งเอสคิวแอล ผู้ดูแลระบบฐานข้อมูลต้องเปลี่ยนหน้าจอเป็นหน้านจอบำรุงรักษาฐานข้อมูล จากนั้นจะแสดงหน้าจอบำรุงรักษาฐานข้อมูล ผู้ดูแลฐานข้อมูลกำหนดเงื่อนไขบำรุงรักษาฐานข้อมูล แล้วกดปุ่มบันทึก

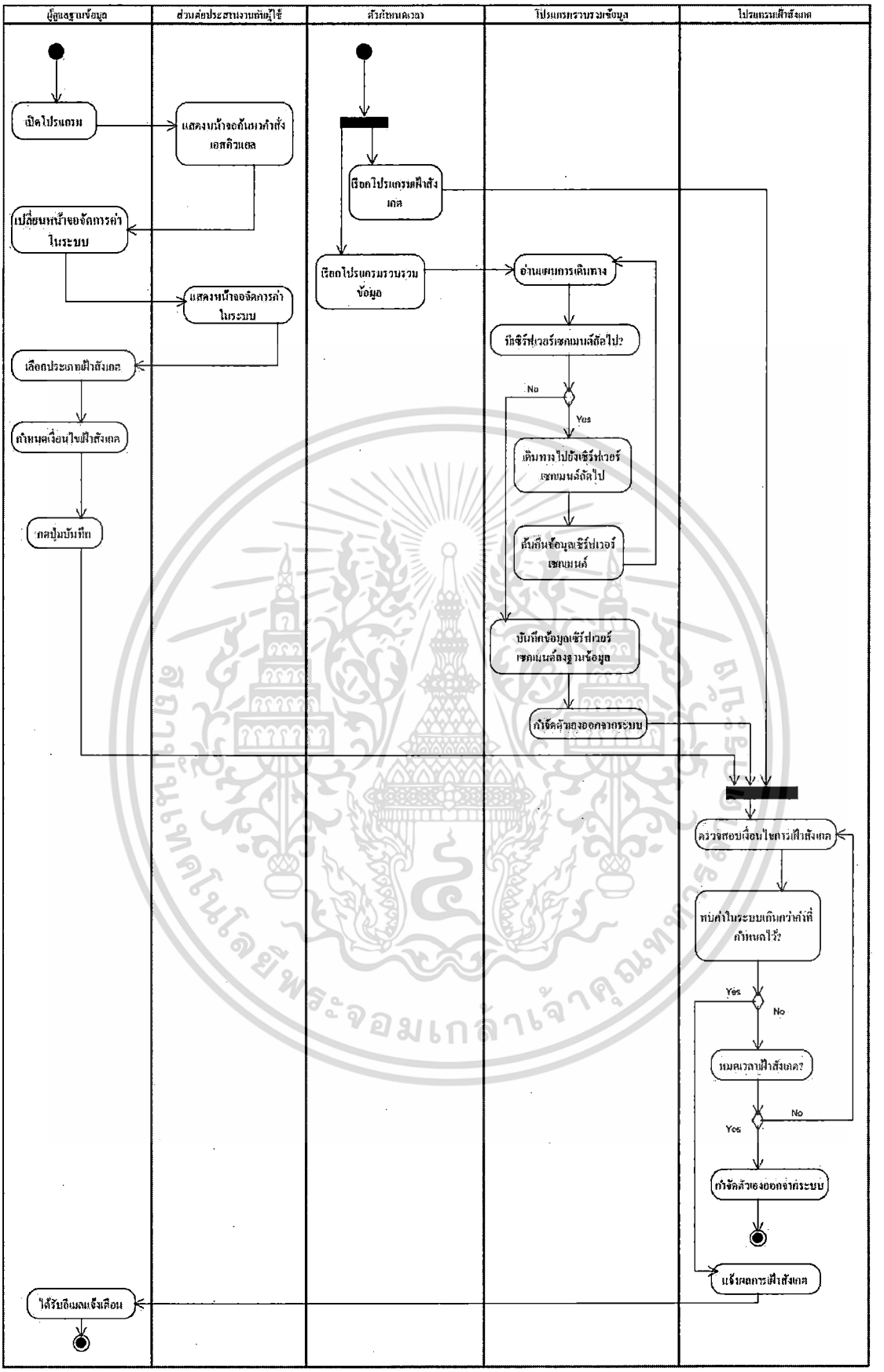
ในส่วนของเอเจนท์บำรุงรักษาฐานข้อมูล จะประกอบด้วย ตัวกำหนดเวลา และโปรแกรมบำรุงรักษาฐานข้อมูล เมื่อถึงเวลาตามที่ผู้ดูแลฐานข้อมูลบันทึก ตัวกำหนดเวลาจะเรียกโปรแกรมบำรุงรักษาฐานข้อมูล โดยบำรุงรักษาฐานข้อมูลจะตรวจสอบเงื่อนไขบำรุงรักษาฐานข้อมูล ถ้าผู้ดูแลฐานข้อมูลกำหนดให้บำรุงรักษาฐานข้อมูล ก็จะทำการตรวสอบช่วงเวลาที่กำหนดต่อ ถ้าตรงตามที่กำหนด โปรแกรมจะดำเนินการบำรุงรักษาฐานข้อมูล ถ้าตรวสอบว่าไม่อยู่ในช่วงเวลาที่กำหนดหรือโปรแกรมบำรุงรักษาฐานข้อมูลเสร็จสิ้น จะกำจัดตัวเองออกจากระบบ

จากรูปที่ 3.4 เมื่อผู้ดูแลระบบฐานข้อมูลเปิดโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย จะแสดงหน้าจอค้นหาคำสั่งเอสคิวแอล ผู้ดูแลระบบฐานข้อมูลต้องเปลี่ยนหน้าจอเป็นหน้านจอจัดการค่าในระบบ จากนั้นจะแสดงหน้าจอจัดการค่าในระบบ ผู้ดูแลฐานข้อมูลเลือกประเภทการเฝ้าสังเกตและกำหนดเงื่อนไขการเฝ้าสังเกต แล้วกดปุ่มบันทึก

ในส่วนของเอเจนท์รวบรวมข้อมูล จะประกอบด้วย ตัวกำหนดเวลา และโปรแกรมรวบรวมข้อมูล เมื่อถึงเวลา ตัวกำหนดเวลาจะเรียกโปรแกรมรวบรวมข้อมูล จากนั้นอ่านแผนที่การเดินทางว่าต้องไปเรียกโปรแกรมรวบรวมข้อมูลที่เซิร์ฟเวอร์ไหนบ้าง ทำการเข้าไปดำเนินการรวบรวมข้อมูลทีละเซิร์ฟเวอร์เซกเมนต์ เมื่อทำการรวบรวมข้อมูลเซกเมนต์หนึ่งเสร็จ จะตรวจว่ามีเซิร์ฟเวอร์เซกเมนต์ที่ต้องรวบรวมข้อมูลอีกหรือไม่ ถ้ามีจะทำการรวบรวมข้อมูลเซิอร์เวอร์เซกเมนต์นั้นต่อ ถ้าไม่มีจะนำผลการรวบรวมข้อมูลทั้งหมดบันทึกลงฐานข้อมูล และกำจัดตัวเองออกจากระบบ

ในส่วนของเอเจนท์เฝ้าสังเกต จะประกอบด้วย ตัวกำหนดเวลา และโปรแกรมเฝ้าสังเกต เมื่อถึงเวลาตามที่ผู้ดูแลฐานบันทึก ตัวกำหนดเวลาจะเรียกโปรแกรมเฝ้าสังเกต ตรวจเงื่อนไขการเฝ้าสังเกตว่ามีค่าในระบบเกินกว่าค่าที่กำหนดไว้หรือไม่ ถ้ามีค่าเกินกำหนดจะผลการเฝ้าสังเกตทางอีเมลให้ผู้ดูแลฐานข้อมูล ต่อมาจะตรวจสอบว่าหมดเวลาเฝ้าสังเกตหรือไม่ ถ้าไม่หมดเวลาให้ตรวจสอบเงื่อนการเฝ้าสั่งเกตต่อ แต่ถ้าหมดเวลาจะกำจัดตังเองออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

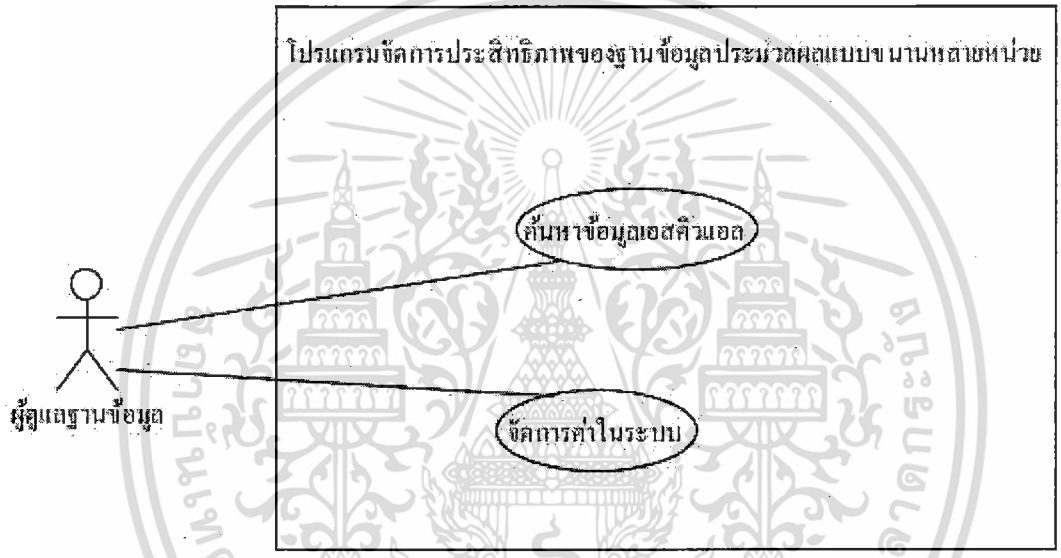


รูปที่ 3.4 แสดงขั้นตอนการทำงานเมื่อผู้ใช้เฝ้าสังเกตประสิทธิภาพฐานของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ยูสเคสไดอะแกรม

ยูสเคสไดอะแกรมเป็นแผนภาพที่แสดงให้เห็นถึงความสัมพันธ์ และปฏิสัมพันธ์ระหว่างระบบสิ่งแวดล้อมภายนอกระบบ ทำให้ทราบถึงความสามารถ หรือฟังก์ชันการทำงานของระบบและผู้ที่เกี่ยวข้องกับระบบที่พัฒนา โดยยูสเคสไดอะแกรมของระบบเอเจนท์เฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยที่ได้ทำการวิเคราะห์ออกแบบ ประกอบด้วย 4 ยูสเคสไดอะแกรม คือ ยูสเคสไดอะแกรมโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย ยูสเคสไดอะแกรมเอเจนท์รวบรวมข้อมูล ยูสเคสไดอะแกรมเอเจนท์บำรุงรักษาฐานข้อมูล และยูสเคสไดอะแกรมเฝ้าสังเกต



รูปที่ 3.5 ยูสเคสไดอะแกรมโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย

ตารางที่ 3.1 คำอธิบายยูสเคสค้นหาข้อมูลเอสคิวแอล

รหัสยูสเคส	UC01
ยูสเคส	ค้นหาข้อมูลเอสคิวแอล
วัตถุประสงค์	เพื่อให้ผู้ดูแลฐานข้อมูลค้นหาข้อมูลเอสคิวแอลที่ได้ถูกดำเนินการในฐานข้อมูลประมวลแบบขนานหลายหน่วยแล้ว เวลาที่ใช้ดำเนินการของคำสั่งเอสคิวแอลและ cost ที่ใช้ เพื่อให้ผู้ดูแลฐานข้อมูลจะได้นำเอสคิวแอลมาปรับปรุงเ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 (ต่อ)

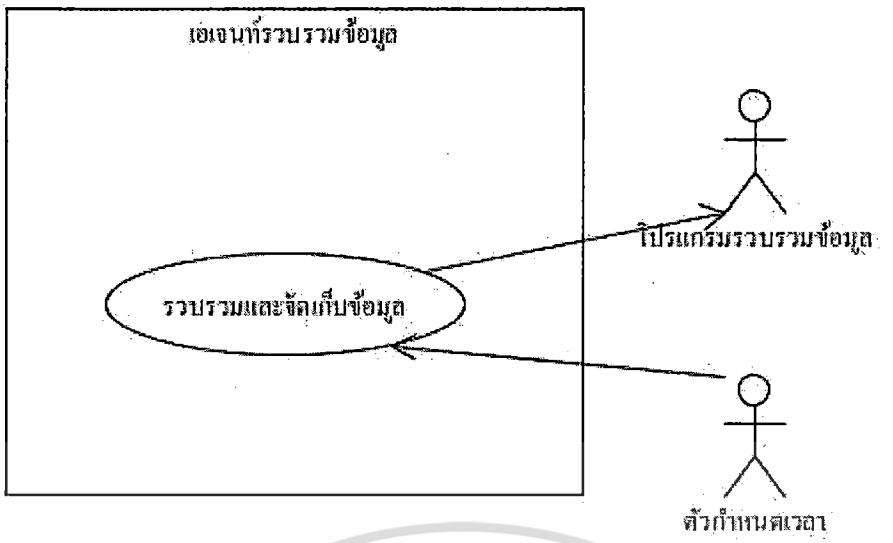
เงื่อนไขเมื่อเริ่มต้น	มีข้อมูลการดำเนินการของเอสคิวแอลในช่วงเวลาต่าง ๆ จนถึงข้อมูลล่าสุด และผู้ดูแลฐานข้อมูลมีคำค้นหา	
เมื่อทำงานสำเร็จ	แสดงผลการค้นหาคำสั่งเอสคิวแอลในช่วงเวลาต่าง ๆ ตามผู้ดูแลฐานข้อมูลกำหนด	
เมื่อทำงานไม่สำเร็จ	-	
แอกเตอร์ที่เกี่ยวข้อง	ผู้ดูแลฐานข้อมูล	
สิ่งกระตุ้นการทำงาน	ผู้ดูแลฐานข้อมูลเข้าสู่หน้าจอค้นหาและพิมพ์คำสั่งเอสคิวแอลที่ต้องการค้นหา และกำหนดช่วงเวลาที่ค้นหา	
อินพุต	คำสั่งเอสคิวแอลที่ต้องการค้นหา และกำหนดช่วงเวลาที่ค้นหา	
เอาท์พุต	ผลของข้อมูลเอสคิวแอลตามเงื่อนไขที่ค้นหา	
สถานการณ์	ผู้ดูแลฐานข้อมูลค้นหาคำสั่งเอสคิวแอลในช่วงเวลาต่าง ๆ	
รายละเอียด	แอกเตอร์	ระบบ
	1. ผู้ดูแลฐานข้อมูลเปิดโปรแกรม เพื่อค้นหาคำสั่งเอสคิวแอล 2. พิมพ์คำสั่งเอสคิวแอลที่ต้องการค้นหา และกำหนดช่วงเวลาที่ค้นหา 3. กดปุ่มค้นหา 4. ผู้ดูแลฐานข้อมูลดับเบิลคลิกคำสั่งเอสเอสคิวแอลเพื่อดูรายละเอียด และแผนของคำสั่งเอสคิวแอล	1.1 แสดงหน้อจอค้นหา 3.1 โปรแกรมส่วนต่อประสานกับผู้ใช้ค้นคืนข้อมูลคำสั่งเอสคิวแอลที่ถูกรวบรวมไว้ 3.2 หน้าจอแสดงผลการค้นหาคำสั่งเอสคิวแอลตามเงื่อนไขที่กำหนด 4.1 โปรแกรมค้นคืนข้อมูลรายละเอียด และแผนของคำสั่งเอสคิวแอล 4.2 โปรแกรมแสดงหน้าจอรายเอียดคำสั่งเอสคิวแอล และแผนของคำสั่งเอสคิวแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 คำอธิบายยูสเคสจัดการค่าในระบบ

รหัสยูสเคส	UC02	
ยูสเคส	จัดการค่าในระบบ	
วัตถุประสงค์	ผู้ดูแลฐานข้อมูลมีการกำหนดให้บำรุงรักษาฐานข้อมูล หรือเลือกประเภทและกำหนดค่าที่ต้องการเฝ้าสังเกตฐานข้อมูล แล้วบันทึก	
เงื่อนไขเมื่อเริ่มต้น	ผู้ดูแลฐานข้อมูลต้องกำหนดให้บำรุงรักษาฐานข้อมูล หรือเลือกประเภทที่ต้องการเฝ้าสังเกต เช่น จำนวนคิวของคำสั่งเอสคิวแอล การล็อคของเทเบิล การโหลดที่ไม่สมดุลของเซิร์ฟเวอร์ และทรัพยากรเครื่อง เป็นต้น	
เมื่อทำงานสำเร็จ	บันทึกค่าที่ได้กำหนด	
เมื่อทำงานไม่สำเร็จ	-	
แอกเตอร์ที่เกี่ยวข้อง	ผู้ดูแลฐานข้อมูล	
สิ่งกระตุ้นการทำงาน	ผู้ดูแลฐานข้อมูลกำหนดให้บำรุงรักษาฐานข้อมูล หรือเลือกประเภทและกำหนดค่าที่ต้องการเฝ้าสังเกตฐานข้อมูล	
อินพุต	ผู้ดูแลฐานข้อมูลกำหนดให้บำรุงรักษาฐานข้อมูล หรือเลือกประเภทและกำหนดค่าที่ต้องการเฝ้าสังเกตฐานข้อมูล	
เอาท์พุต	บันทึกค่าที่ได้กำหนดลงในฐานข้อมูล	
สถานการณ์	ผู้ดูแลฐานข้อมูลกำหนดค่าในระบบ	
รายละเอียด	แอกเตอร์	ระบบ
	1. ผู้ดูแลฐานข้อมูลเปิดโปรแกรม	1.1 แสดงหน้าจอค้นหา
	2. บำรุงรักษาฐานข้อมูล เปลี่ยนหน้าจอ	2.1 แสดงหน้าจอบำรุงรักษาฐานข้อมูล หรือหน้าจอเฝ้าสังเกต
	3. ผู้ดูแลฐานข้อมูลกำหนดห้บำรุงรักษาฐานข้อมูล หรือเลือกประเภทของการเฝ้าสังเกต	
	4. ผู้ดูแลฐานข้อมูลกำหนดค่าที่ต้องการ เลือกช่วงเวลาบันทึกค่าลงในระบบ	4.1 โปรแกรมบันทึกค่าที่กำหนดลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ยูสเคสไดอะแกรมเอเจนท์รวบรวมข้อมูล

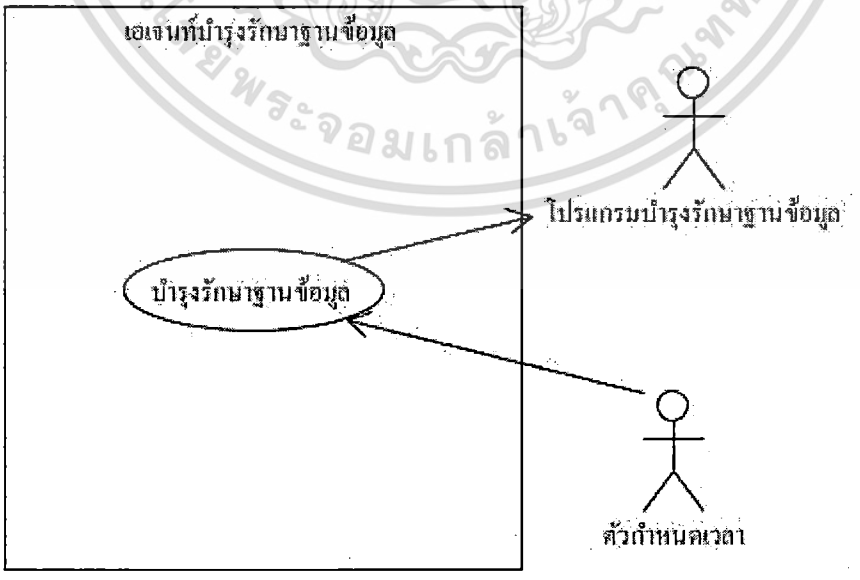
ตารางที่ 3.3 คำอธิบายยูสเคสรวบรวมและจัดเก็บข้อมูล

รหัสยูสเคส	UC03
ยูสเคส	รวบรวมและจัดเก็บข้อมูล
วัตถุประสงค์	เพื่อรวบรวมข้อมูลต่าง ๆ เกี่ยวกับประสิทธิของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยมาจัดเก็บไว้
เงื่อนไขเมื่อเริ่มต้น	เมื่อถึงเวลาตัวกำหนดเวลาเรียกโปรแกรมรวบรวมข้อมูลทำงาน
เมื่อทำงานสำเร็จ	โปรแกรมรวบรวมข้อมูลรวบรวมข้อมูลต่าง ๆ เกี่ยวกับประสิทธิของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยมาจัดเก็บไว้
เมื่อทำงานไม่สำเร็จ	-
แอกเตอร์ที่เกี่ยวข้อง	ตัวกำหนดเวลา และโปรแกรมรวบรวมข้อมูล
สิ่งกระตุ้นการทำงาน	เมื่อถึงเวลาตัวกำหนดเวลาทำงาน
อินพุต	เวลาสำหรับเริ่มทำงาน และแผนการเดินทางที่ประกอบด้วยคำสั่งค้นคืนข้อมูล ที่อยู่ของเซิร์ฟเวอร์แต่ละแห่ง
เอาท์พุต	รวบรวมข้อมูลต่าง ๆ เกี่ยวกับประสิทธิของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยมาจัดเก็บไว้
สถานการณ์	ตัวกำหนดเวลาเรียกโปรแกรมรวบรวมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 (ต่อ)

รายละเอียด	แอกเตอร์	ระบบ
	1. เมื่อถึงเวลาตัวกำหนดเวลาทำงาน 2. โปรแกรมรวบรวมข้อมูลอ่านแผนที่เดินทาง 3. ตรวจว่ามีเซิร์ฟเวอร์เซกเมนต์หรือไม่ 4. ถ้ามีเดินทางไปยังเซิร์ฟเวอร์เซกเมนต์นั้นและโปรแกรมรวบรวมข้อมูลย่อยทำงาน 5. ค้นคืนข้อมูลเซิร์ฟเวอร์เซกเมนต์ 6. ตรวจสอบว่ามีเซิร์ฟเวอร์เซกเมนต์อีกหรือไม่ 7. ถ้าไม่มี ให้บันทึกข้อมูลเซิร์ฟเวอร์เซกต์เมนต์ลงฐานข้อมูล 8. กำจัดตัวเองออกจากระบบ	1.1 เรียกโปรแกรมรวบรวมข้อมูลเพื่อรวบรวมและจัดเก็บข้อมูล 7.1 บันทึกข้อมูลเซิร์ฟเวอร์เซกเมนต์ลงฐานข้อมูล



รูปที่ 3.7 ยูสเคสไดอะแกรมเอเจนท์บำรุงรักษาฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

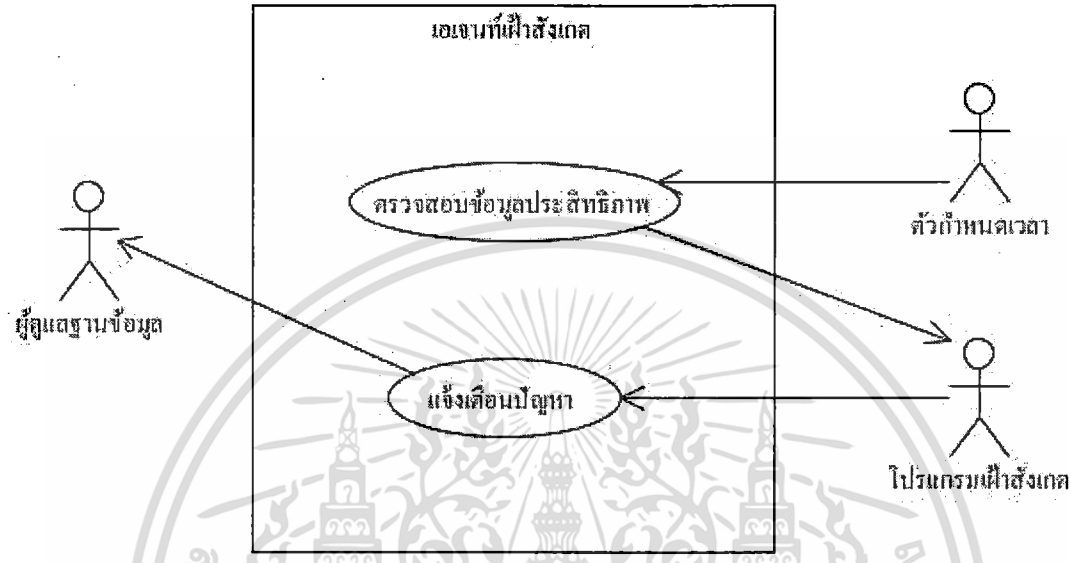
ตารางที่ 3.4 คำอธิบายยูสเคสบำรุงรักษาฐานข้อมูล

รหัสยูสเคส	UC04	
ยูสเคส	บำรุงรักษาฐานข้อมูล	
วัตถุประสงค์	เพื่อบำรุงรักษาฐานข้อมูล	
เงื่อนไขเมื่อเริ่มต้น	ผู้ดูแลฐานข้อมูลกำหนดให้มีการบำรุงรักษาฐานข้อมูล	
เมื่อทำงานสำเร็จ	บำรุงรักษาฐานข้อมูล	
เมื่อทำงานไม่สำเร็จ	-	
แอกเตอร์ที่เกี่ยวข้อง	ตัวกำหนดเวลา และโปรแกรมบำรุงรักษาฐานข้อมูล	
สิ่งกระตุ้นการทำงาน	เมื่อถึงเวลาที่ผู้ดูแลฐานข้อมูลกำหนด	
อินพุต	เวลาที่ผู้ดูแลฐานข้อมูลกำหนด และเงื่อนไขการบำรุงรักษา	
เอาท์พุต	บำรุงรักษาฐานข้อมูล	
สถานการณ์	ตัวกำหนดเวลาเรียกโปรแกรมบำรุงรักษาฐานข้อมูล	
รายละเอียด	แอกเตอร์	ระบบ
	1. เมื่อถึงเวลาที่ผู้ดูแลฐานข้อมูลกำหนดตัวกำหนดเวลาทำงาน 2. โปรแกรมบำรุงรักษาฐานข้อมูลตรวจสอบเงื่อนไขการบำรุงรักษา 3. ตรวจสอบว่ามีการกำหนดให้บำรุงรักษาฐานข้อมูลหรือไม่ 4. ถ้ามี ให้ตรวจสอบว่าอยู่ในช่วงเวลาที่กำหนดหรือไม่ 5. ถ้าอยู่ในช่วงเวลาที่กำหนดให้บำรุงรักษาฐานข้อมูลตามเงื่อนไขที่กำหนด 6. ตรวจสอบว่ามีการกำหนดบำรักษาฐานข้อมูลหรือไม่	1.1 เรียกโปรแกรมบำรุงรักษาฐานข้อมูล 5.1 บำรุงรักษาฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 (ต่อ)

รายละเอียด	แอกเตอร์	ระบบ
	7. ถ้าไม่มี กำจัดตัวเองออกจากระบบ	



รูปที่ 3.8 ยูสเคสไดอะแกรมเอเจนท์เฝ้าสังเกต

ตารางที่ 3.5 คำอธิบายยูสเคสตรวจสอบข้อมูลประสิทธิภาพ

รหัสยูสเคส	UC05
ยูสเคส	ตรวจสอบข้อมูลประสิทธิภาพ
วัตถุประสงค์	เพื่อเฝ้าสังเกตและตรวจสอบค่าในฐานข้อมูลมีค่าเกินกำหนดตามได้ที่บันทึกไว้
เงื่อนไขเมื่อเริ่มต้น	ผู้ดูแลฐานข้อมูลกำหนดเวลาการเฝ้าสังเกต
เมื่อทำงานสำเร็จ	ตรวจสอบค่าในฐานข้อมูลมีค่าเกินกำหนดตามได้ที่บันทึกไว้หรือไม่
เมื่อทำงานไม่สำเร็จ	-
แอกเตอร์ที่เกี่ยวข้อง	ตัวกำหนดเวลา และโปรแกรมเฝ้าสังเกต
สิ่งกระตุ้นการทำงาน	เมื่อถึงเวลาที่ผู้ดูแลฐานข้อมูลกำหนด
อินพุต	เวลาที่ผู้ดูแลฐานข้อมูลกำหนด และเงื่อนไขการเฝ้าสังเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 (ต่อ)

เอาท์พุต	ผลการตรวจสอบค่าในฐานข้อมูลมีค่าเกินกำหนดตามได้ที่บันทึกไว้หรือไม่	
สถานการณ์	ตัวกำหนดเวลาเรียกโปรแกรมเฝ้าสังเกต	
รายละเอียด	แอกเตอร์	ระบบ
	1. เมื่อถึงเวลาที่ผู้ดูแลฐานข้อมูลกำหนดตัวกำหนดเวลาทำงาน 2. โปรแกรมเฝ้าสังเกตตรวจสอบเงื่อนไขการเฝ้าสังเกต 3. ตรวจสอบว่ามีค่าในระบบเกินกำหนดหรือไม่ 4. ถ้าพบว่าเกินกำหนด ให้แสดงผลลัพธ์ 5. ตรวจสอบว่าอยู่ในช่วงเวลาที่กำหนดให้เฝ้าสังเกตหรือไม่ 6. ถ้าอยู่ในช่วงเวลาให้ตรวจเงื่อนไขเฝ้าสังเกตต่อ 7. ตรวจสอบว่าอยู่ในช่วงเวลาที่กำหนดให้เฝ้าสังเกตหรือไม่ 8. ถ้าไม่อยู่ในช่วงเวลาที่กำหนด ให้กำจัดตัวเองออกจากระบบ	1.1 เรียกโปรแกรมเฝ้าสังเกต 4.1 แจ้งผลการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

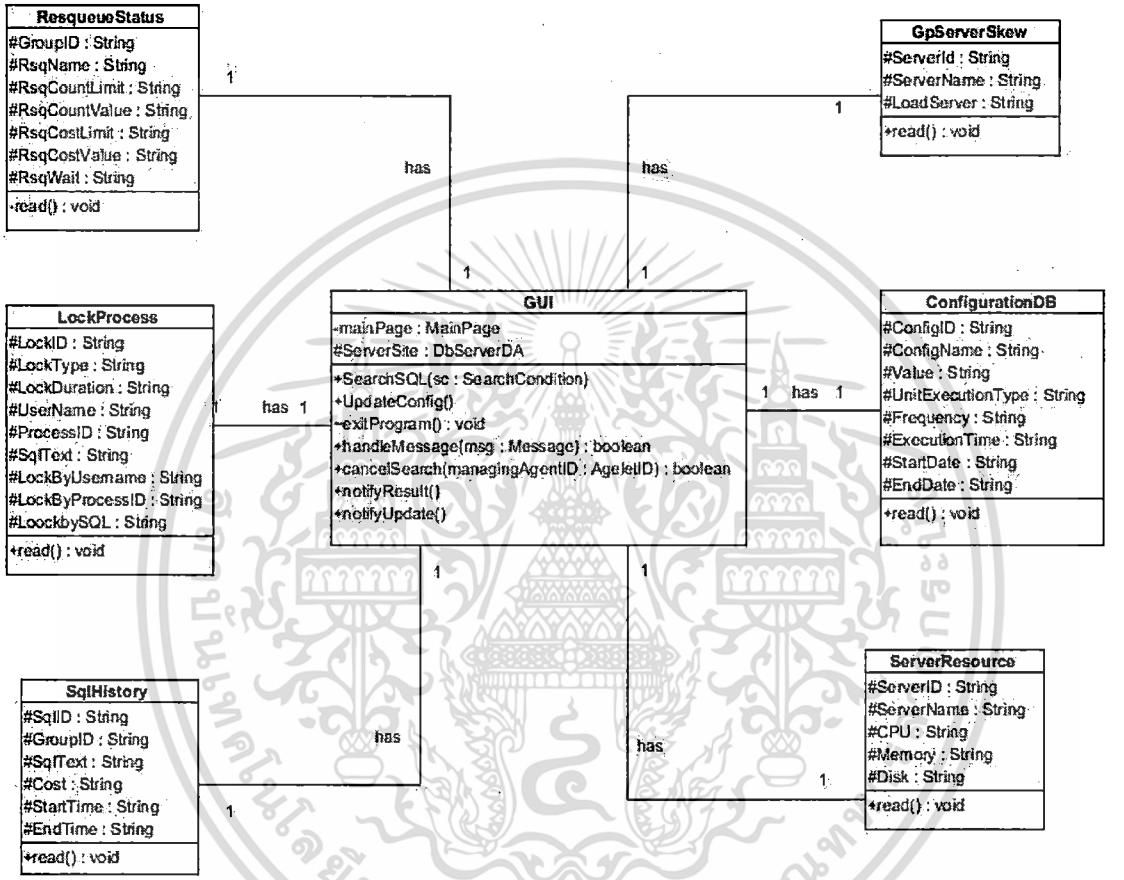
ตารางที่ 3.6 คำอธิบายยูสเคสแจ้งเตือนปัญหา

รหัสยูสเคส	UC06	
ยูสเคส	**แจ้งเตือนปัญหา**	
วัตถุประสงค์	เพื่อแจ้งเตือนเมื่อตรวจพบค่าในฐานข้อมูลมีค่าเกินกำหนดตามได้ที่บันทึกไว้ และแจ้งเตือนผู้ดูแลฐานข้อมูล	
เงื่อนไขเมื่อเริ่มต้น	โปรแกรมเฝ้าสังเกตตรวจพบค่าในฐานข้อมูลมีค่าเกินกำหนดตามได้ที่บันทึกไว้	
เมื่อทำงานสำเร็จ	แจ้งเตือนผู้ดูแลฐานข้อมูลเมื่อพบค่าในฐานข้อมูลมีค่าเกินกำหนด	
เมื่อทำงานไม่สำเร็จ	-	
แอกเตอร์ที่เกี่ยวข้อง	ผู้ดูแลฐานข้อมูล และโปรแกรมเฝ้าสังเกต	
สิ่งกระตุ้นการทำงาน	ผลการตรวจสอบพบว่ามีค่าในระบบฐานข้อมูลมีค่าเกินกำหนด	
อินพุต	ผลการตรวจสอบจากเอเจนท์เฝ้าสังเกต	
เอาท์พุต	แจ้งผลการเฝ้าสังเกตให้ผู้ดูแลฐานข้อมูล	
สถานการณ์	โปรแกรมเฝ้าสังเกตตรวจสอบพบว่ามีค่าในระบบฐานข้อมูลมีค่าเกินกำหนด แล้วแจ้งผลการเฝ้าสังเกตให้ผู้ดูแลฐานข้อมูล	
รายละเอียด	แอกเตอร์	ระบบ
	1. โปรแกรมเฝ้าสังเกตได้ตรวจสอบพบว่ามีค่าในระบบฐานข้อมูลมีค่าเกินกำหนด 2. ผู้ดูแลฐานได้รับอีเมลผลการเฝ้าสังเกต	2.1 แจ้งเตือนผลการเฝ้าสังเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 คลาสไดอะแกรม

คลาสไดอะแกรมเป็นแผนภาพที่ใช้แสดงโครงสร้างความสัมพันธ์ระหว่างคลาสของวัตถุต่าง ๆ ที่มีในระบบ รวมถึงคุณสมบัติ และการกระทำที่วัตถุในคลาสต่าง ๆ สามารถกระทำได้โดยระบบที่ได้ทำการออกแบบประกอบด้วยคลาสต่าง ๆ ดังรูปที่ 3.9 จากรูประบบที่ออกแบบประกอบด้วยคลาสต่าง ๆ ดังนี้



รูปที่ 3.9 คลาสไดอะแกรมโปรแกรมจัดการประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย

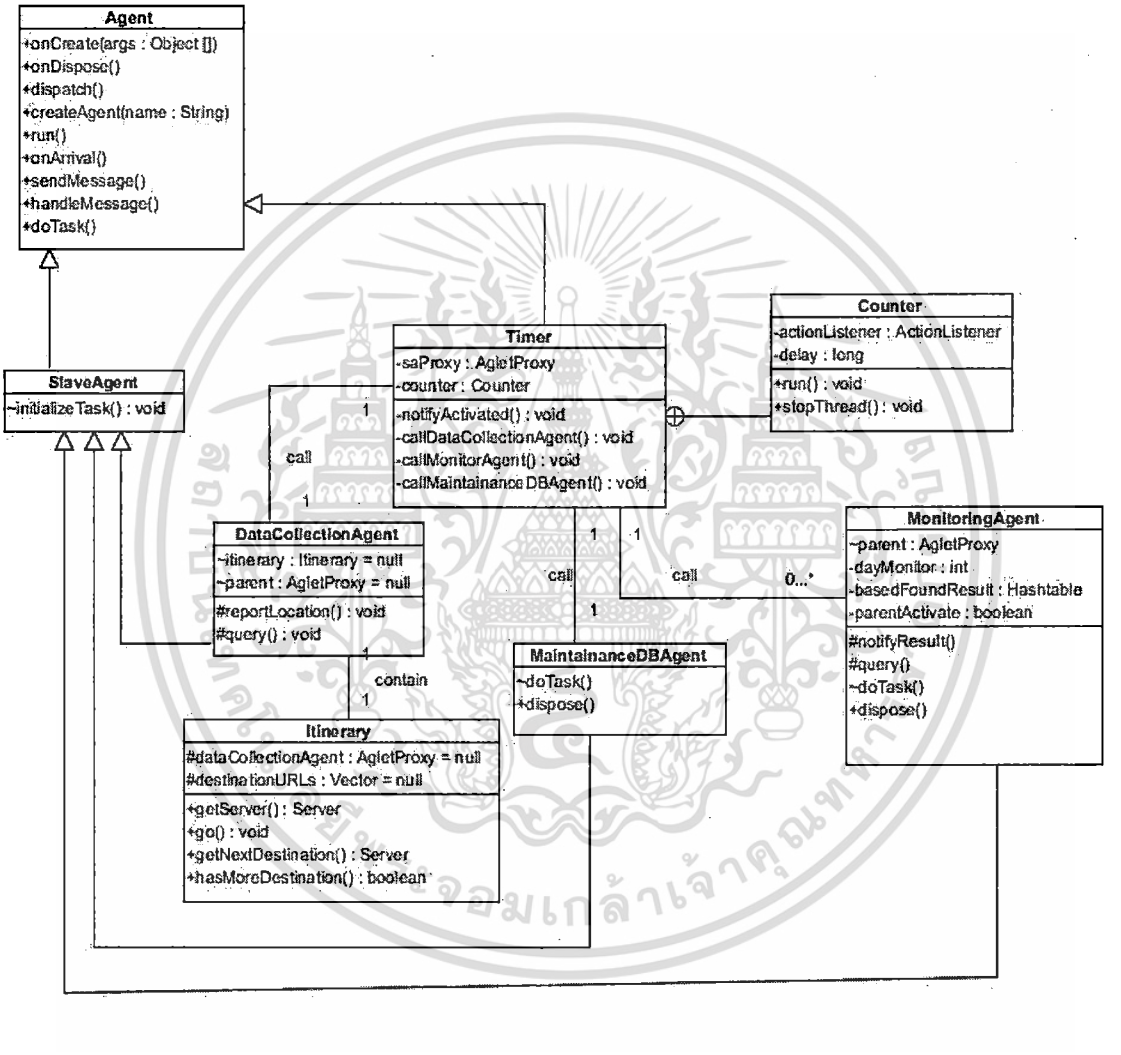
- GUI ทำหน้าที่สร้างส่วนติดต่อกับผู้ใช้ รวมถึงเป็นคลาสเริ่มต้นการทำงานของโปรแกรม
- ResquenceStatus คือคลาสที่เป็นตัวแทนของข้อมูลการจัดสรรทรัพยากรฐานข้อมูล
- LockProcess คือคลาสที่เป็นตัวแทนของข้อมูลล็อคเทเบิล
- GpServerSkew คือคลาสที่เป็นตัวแทนของข้อมูลการโหลดสมุลของเซิร์ฟเวอร์
- Configuration คือคลาสที่เป็นตัวแทนของข้อมูลของค่าที่ถูกกำหนดจากผู้ดูแลฐานข้อมูล
- ServerResource คือคลาสที่เป็นตัวแทนของข้อมูลทรัพยากรเครื่องเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SqlHistory คือคลาสที่เป็นตัวแทนของข้อมูลประวัติการดำเนินการของเอสคิวแอล

โดยความสัมพันธ์ระหวางคลาส มีดังนี้

คลาส GUIAgent สัมพันธ์กับคลาส ResquenceStatus LockProcess GpServerSkew ServerResource และ SqlHistory โดยคลาส GUIAgent ทำหน้าที่สร้างอินสเตนซ์ของคลาสขึ้นมาเพื่อใช้สำหรับการอ่านข้อมูล



รูปที่ 3.10 คลาสไดอะแกรมโปรแกรมเอเจนท์

- Aglet คือคลาสหลักของเอเจนท์ ประกอบด้วยคำสั่ง และคุณสมบัติพื้นฐานในการทำงานของเอเจนท์เคลื่อนที่ สำหรับให้คลาสเอเจนท์เคลื่อนที่อื่นๆ มารับสืบทอดคุณสมบัติเพื่อให้สามารถทำงานในลักษณะของเอเจนท์เคลื่อนที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Timer ทำหน้าที่เป็นตัวควบคุมจัดการกลางของระบบคอยควบคุมประสานการทำงานกับระบบเอเจนท์เคลื่อนที่
- Itinerary คือคลาสที่ใช้ในการจัดการเส้นทางการเดินทางของเอเจนท์เคลื่อนที่ เพื่อบอกให้เอเจนท์เคลื่อนที่รู้ว่าจะต้องเดินทางไปยังเซิร์ฟเวอร์ใดบ้าง
- SlaveAgent คือคลาสนามธรรม ทำหน้าที่รับงานจากคลาส Timer เพื่อไปทำงานยังเซิร์ฟเวอร์ต่างๆ และส่งผลลัพธ์การทำงานกลับมาเมื่อทำงานเสร็จสิ้น
- DataCollectionAgent คือคลาสเอเจนท์เคลื่อนที่ที่รับทอดคุณสมบัติมาจากคลาสนามธรรม SlaveAgent ทำหน้าที่เดินทางไปยังเซิร์ฟเวอร์ต่าง ๆ โดยใช้เส้นทางที่กำหนดโดยคลาส Itinerary เพื่อไปรวบรวมและจัดเก็บข้อมูล ตามที่ผู้ดูแลฐานข้อมูลกำหนด
- MonitoringAgent คือคลาสเอเจนท์เคลื่อนที่รับทอดคุณสมบัติมาจากคลาสนามธรรม SlaveAgent ทำหน้าที่เฝ้าสังเกตค่าของระบบที่เกินกำหนด โดยจะถูกส่งออกไปตามจำนวนของแหล่งข้อมูล
- Counter คือคลาสภายใน (Inner Class) ที่อยู่ภายในคลาส Timer ทำหน้าที่ในการนับเวลาในการทำงานของเอเจนท์ โดยเมื่อครบเวลาตามที่กำหนดคลาส Counter จะกระตุ้นให้เอเจนท์เฝ้าสังเกตทำงานตามที่ได้กำหนดไว้

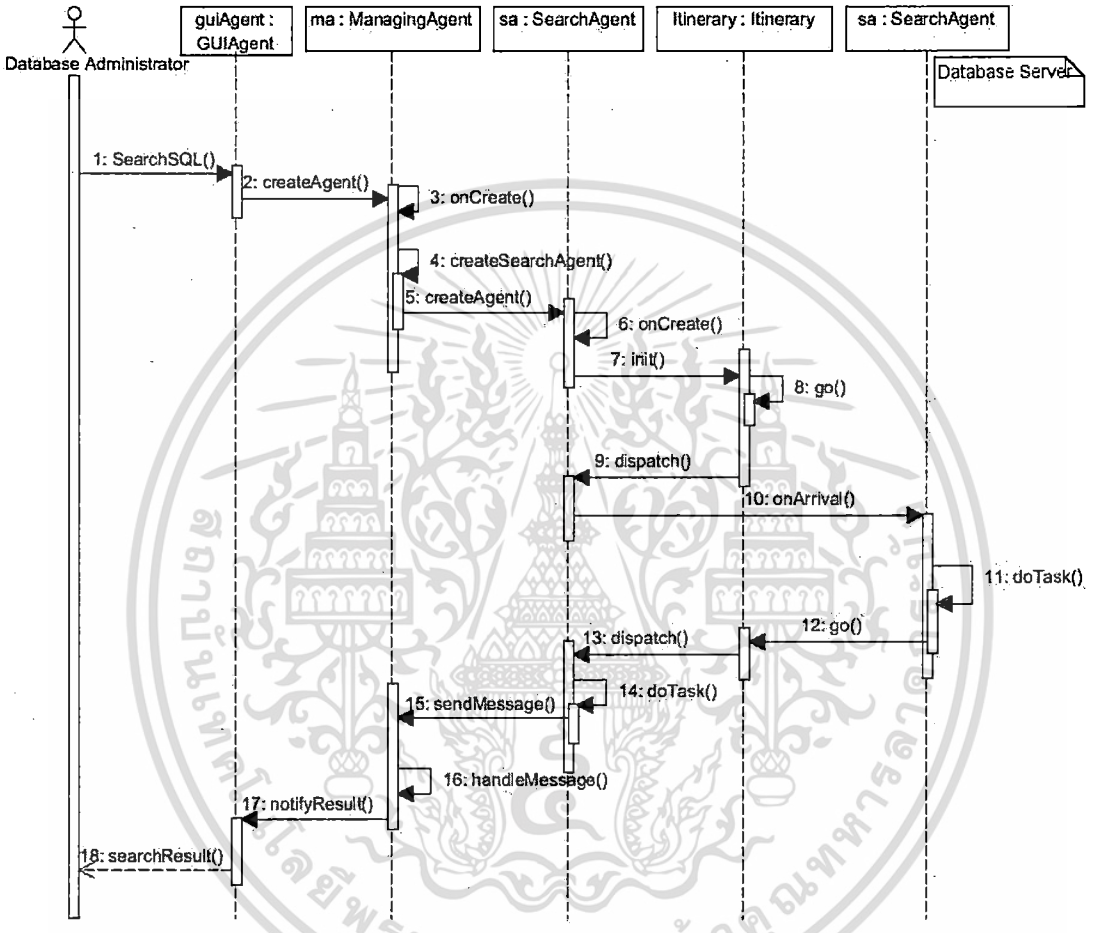
โดยความสัมพันธ์ระหวางคลาส มีดังนี้

1. คลาส Timer สัมพันธ์กับคลาส DataCollectionAgent โดยคลาส Time จะทำหน้าที่เรียกคลาส DataCollectionAgent ขึ้นมาหนึ่งอินเสตนซ์ต่อการค้นหาแต่ละครั้งเมื่อผู้ดูแลฐานข้อมูลค้นหาข้อมูลคำสั่งเอสคิวแอล ซึ่ง Timer จะทำหน้าที่ประสานการทำงานกับคลาส DataCollectionAgent ในการมอบหมายงานการค้นหา รายงานสถานะของระบบและรับผลลัพธ์การค้นหาที่ได้จากคลาส DataCollectionAgent เป็นต้น
2. คลาส Timer สัมพันธ์กับคลาส MonitoringAgent โดยคลาส Timer จะทำหน้าที่สร้างอินสเตนซ์ของคลาส MonitoringAgent ขึ้นมาตามจำนวนของเซิร์ฟเวอร์ปลายทางที่ผู้หางานระบุ ซึ่งคลาส Timer จะทำหน้าที่ประสานการทำงานกับ คลาส MonitoringAgent ในการมอบหมายงานการค้นหาเช่นเดียวกับคลาส DataCollectionAgent
3. คลาส DataCollectionAgent สัมพันธ์กับคลาส Itinerary ภายในคลาส DataCollectionAgent จะต้องประกอบด้วยอินเสเตนซ์ของคลาส Itinerary ซึ่งเก็บแผนการเดินทางของเอเจนท์ค้นหาหรือคลาส DataCollectionAgent เอาไว้ และจะถูกเคลื่อนย้ายไปยังเซิร์ฟเวอร์ต่างๆ พร้อมกับคลาส DataCollectionAgent เพื่อใช้ตรวจสอบเซิร์ฟเวอร์ที่จะต้องเดินทางถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ซีเควนซ์ไดอะแกรม

ซีเควนซ์ไดอะแกรม หรือแผนภาพลำดับเหตุการณ์เป็นแผนภาพแสดงลำดับเวลาของการทำรายการที่เกิดขึ้นระหว่างวัตถุหนึ่งกับอีกวัตถุหนึ่ง โดยลำดับเหตุการณ์การทำงานที่สำคัญของการค้นหาเอสคิวแอลที่ใช้เวลาดำเนินงานที่ฐานข้อมูลนาน หรือใช้ทรัพยากรเครื่องสูง สามารถแสดงลำดับเหตุการณ์ด้วยซีเควนซ์ไดอะแกรมได้ดังรูปที่ 3.11 ดังนี้

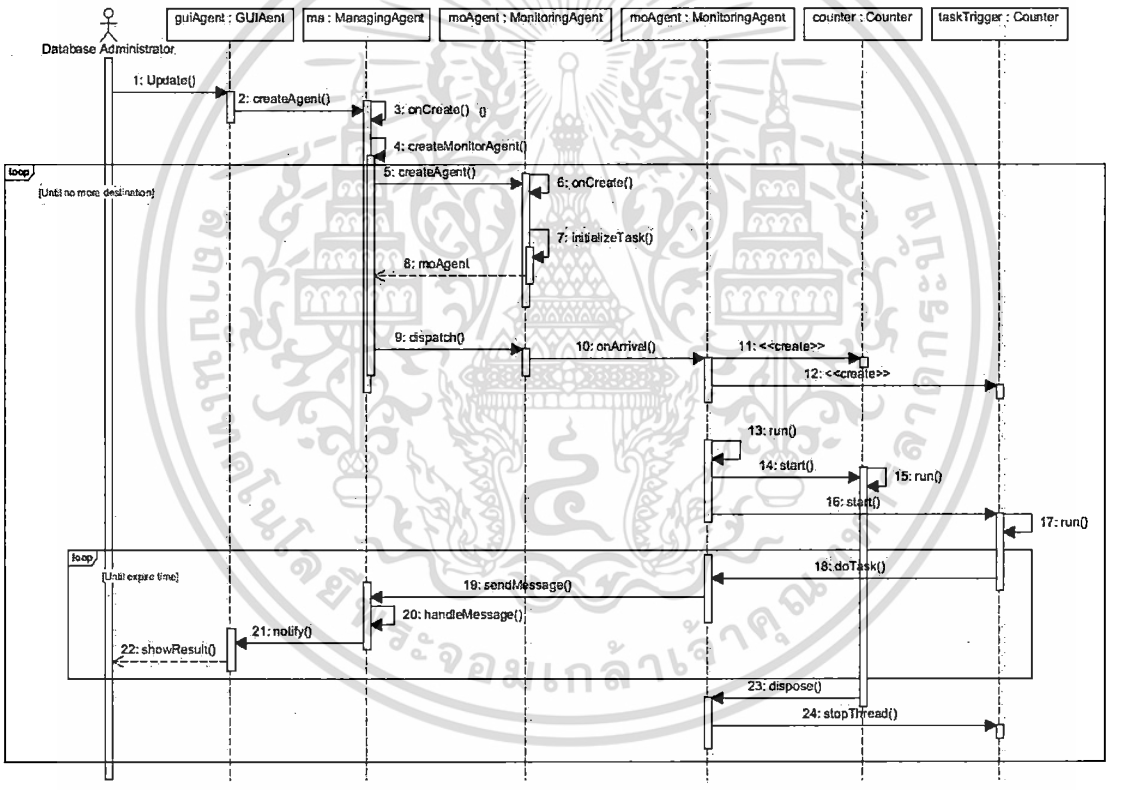


รูปที่ 3.11 ซีเควนซ์ไดอะแกรมแสดงลำดับการทำงานในการค้นหาเอสคิวแอล

จากรูปที่ 3.11 แสดงลำดับการทำงานในการค้นหาข้อมูลคำสั่งเอสคิวแอล โดยเริ่มจากผู้ดูแลฐาข้อมูลสั่งค้นหาคำสั่งเอสคิวแอลไปยังเอเจนท์ส่วนต่อประสานกับผู้ใช้งาน เอเจนท์ส่วนต่อประสานกับผู้ใช้งานจะสร้างเอเจนท์การจัดการเพื่อประสานงานการค้นหากับเอเจนท์เคลื่อนที่ที่ถูกสร้างขึ้นสำหรับการค้นหาคำสั่งเอสคิวแอลครั้งนั้น เมื่อเอเจนท์การจัดการถูกสร้างขึ้น เอเจนท์การจัดการจะสร้างแผนการเดินทางสำหรับเอเจนท์ค้นหาตามเซิร์ฟเวอร์ที่ผู้ดูแลฐานข้อมูลเลือกในหน้าจอค้นหา หลังจากสร้างแผนการเดินทางแล้วเสร็จ เอเจนท์การจัดการจะสร้างเอเจนท์ค้นหาขึ้นมาในเซิร์ฟเวอร์ต้นทางหนึ่งตัว พร้อมส่งแผนการเดินทางเป็นพารามิเตอร์หนึ่งให้กับเอเจนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหา หลังจากเอเจนท์ค้นหาถูกสร้างขึ้นแล้ว จึงเริ่มทำงาน โดยส่งข้อความ init() ไปยังแผนการเดินทางเพื่อเริ่มต้นการอ่านตำแหน่งที่อยู่ของเซิร์ฟเวอร์ที่ต้องเดินทางไปทำงาน แผนการเดินทางจะตรวจสอบรายการเซิร์ฟเวอร์ ถ้ามีรายการเซิร์ฟเวอร์ที่ต้องไปค้นหาข้อมูล จึงจะส่งข้อความ dispatch() เพื่อสั่งให้เอเจนท์ค้นหาเดินทางไปยังเซิร์ฟเวอร์นั้นเพื่อค้นหาข้อมูลเอสคิวแอลในฐานข้อมูลตามเงื่อนไขการค้นหาที่ผู้ดูแลฐานข้อมูลกำหนด เอเจนท์ค้นหาจะส่งข้อความ go() ไปยังแผนการเดินทางเพื่ออ่านแผนการเดินทางและเดินทางไปยังเซิร์ฟเวอร์ถัดไปจนครบ จึงจะเดินทางกลับไปยังเซิร์ฟเวอร์ต้นทางเพื่อรายงานผลลัพธ์ให้แก่เอเจนท์การจัดการโดยส่ งข้อความ sendMessage() ไปยังเอเจนท์การจัดการ เมื่อการรายงานผลลัพธ์เสร็จสิ้น เอเจนท์ค้นหาจึงกำจัดตัวเองออกจากระบบ หลังจากที่เอเจนท์การจัดการได้รับผลลัพธ์การค้นหา จะรายงานผลต่อไปยังเอเจนท์ส่วนต่อประสานกับผู้ใช้ผู้ผ่านการส่งข้อความ notifyResult() เพื่อแสดงผลให้ผู้ดูแลฐานข้อมูล



รูปที่ 3.12 ซีเควนซ์ไดอะแกรมแสดงลำดับการทำงานในการเฝ้าสังเกตค่าของฐานข้อมูลเกินที่กำหนด

จากรูปที่ 3.12 แสดงลำดับการทำงานของการเฝ้าสังเกตการค่าของฐานข้อมูลเกินที่กำหนด โดยหลักจากผู้ดูแลฐานข้อมูลทำการกำหนดค่าการเฝ้าสังเกตอผ่านหน้าจอและบันทึก เอเจนท์การจัดการจะสร้างเอเจนท์เฝ้ าสังเกตขึ้นมาในเซิร์ฟเวอร์ต้นทาง แล้วจึงส่งข้อความ dispatch() ไปยังเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจนท์เฝ้าสังเกตเพื่อส่งเอเจนท์เฝ้าสังเกตออกไปทำงานยังเซิร์ฟเวอร์ฐานข้อมูลแต่ละแห่ง เมื่อเอเจนท์เฝ้าสังเกตเดินทางไปถึงจะเรียกคำสั่ง onArrival() เพื่อกระตุ้นให้ตัวเองเริ่มทำงาน เริ่มจากสร้างตัวนับขึ้นมาสองตัวสำหรับนับเวลาการเฝ้าสังเกต ชื่อ Counter และนับเวลาในการตรวจสอบค่าที่กำหนด ชื่อ taskTrigger แล้วจึงส่งข้อความ start() ไปยังตัวนับทั้งสองเพื่อสั่งให้เริ่มการนับ เมื่อถึงเวลาตรวจสอบการเปลี่ยนแปลง ตัวนับจะส่งข้อความ doTask() เพื่อกระตุ้นให้เอเจนท์เฝ้าสังเกตเริ่มค้นคืนข้อมูลในฐานข้อมูลแล้วนำมาเปรียบเทียบกับข้อมูลตั้งต้นที่มีอยู่เพื่อตรวจสอบค่าที่กำหนด เมื่อพบค่าที่มากกว่า เอเจนท์เฝ้าสังเกตจะส่งข้อความ sendMessage() ไปยังเอเจนท์การจัดการเพื่อส่งผลลัพธ์ และเมื่อเอเจนท์การจัดการได้รับผลลัพธ์แล้ว จะส่งข้อความ notifyUpdate() ไปยังเอเจนท์ส่วนต่อประสานกับผู้ใช้ เพื่อแจ้งให้ผู้ดูแลฐานข้อมูลทราบต่อไป เมื่อครบกำหนดเวลาการเฝ้าสังเกต ตัวนับจะส่งข้อความ dispose() เพื่อสั่งให้เอเจนท์เฝ้าสังเกตกำจัดตัวเองออกจากเซิร์ฟเวอร์ฐานข้อมูล โดยเอเจนท์เฝ้าสังเกตจะส่งข้อความ stopThread() ไปยังตัวนับเพื่อหยุดการนับก่อนกำจัดตัวเองจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การพัฒนาระบบ

4.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ

การออกแบบและพัฒนาระบบเอเจนท์เฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยในโครงการนี้ใช้เครื่องมือ และภาษาโปรแกรมพัฒนาระบบ ดังนี้

4.1.1 ฮาร์ดแวร์

เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบ จำลอง และทดสอบระบบ มีคุณสมบัติดังนี้

- คอมพิวเตอร์โน้ตบุ๊ก Dell รุ่น N4120
- หน่วยประมวลผล Intel(R) Core(TM) i7-2620M ความเร็ว 2.70 GHz
- หน่วยความจำหลัก 8 GB
- ฮาร์ดดิสก์ ความจุ 600 GB
- หน่วยแสดงผล AMD Radeon HD 6630M

4.1.2 ซอฟต์แวร์

ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ ประกอบด้วยซอฟแวร์ต่าง ๆ ดังนี้

- โปรแกรมสร้างคอมพิวเตอร์เสมือน VMWare Workstation 7.1.6
- ภาษาโปรแกรม Java J2SE 5.0
- เฟรมเวิร์คสำหรับเอเจนท์เคลื่อนที่ IBM Aglets Framework 2.0.2
- ระบบปฏิบัติการ Linux CentOS 5 64 bit
- ระบบปฏิบัติการ Microsoft Windows 7 64bit
- ระบบจัดการฐานข้อมูล Greenplum 4.0.1 for Linux

4.1.3 เครื่องมือ

เครื่องที่ใช้พัฒนาระบบ และออกแบบระบบมีดังนี้

- Eclipse IDE Kepler เป็นซอฟต์แวร์ที่ช่วยในการเขียนโปรแกรมได้สะดวกและรวดเร็วมากขึ้น ประกอบด้วย คอมไพเลอร์ และ ดีบักเกอร์ เป็นต้น ในโครงงานนี้ได้นำ Eclipse IDE มาช่วยในการเขียนโปรแกรมในส่วนของระบบ
- Visual Paradigm for UML Enterprise Edition เป็นซอฟต์แวร์ที่ช่วยในการจัดทำแบบจำลองระบบตามมาตรฐาน UML ในโครงการนี้ได้นำมาใช้ในการจัดทำแบบจำลองในขั้นตอนกาวิเคราะห์และออกแบบระบบ เช่น ยูสเคสไดอะแกรม คลาสไดอะแกรม และแอกทิวิตีไดอะแกรม เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PgAdmin III เป็นซอฟต์แวร์ที่ช่วยให้สามารถจัดการข้อมูลที่จัดเก็บในระบบจัดการฐานข้อมูล Greenplum ได้ง่ายขึ้นซึ่งนำมาช่วยในขั้นตอนการจำลองฐานข้อมูลประมวลผลแบบขนานหลายหน่วย

4.2 การจำลองเซิร์ฟเวอร์

การจำลองเซิร์ฟเวอร์แบ่งออกเป็นสองส่วน คือ การจำลองเอเจนท์เคลื่อนที่ และการจำลองฐานข้อมูลกรีนพลัมซึ่งเป็นฐานข้อมูลประมวลผลแบบขนานหลายหน่วย ดังนี้

4.2.1 การจำลองเอเจนท์เคลื่อน

การจำลองเซิร์ฟเวอร์ที่จะนำใช้กับระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์เคลื่อนที่ในโครงการนี้นั้น สามารถแบ่งการจำลองออกเป็น 2 ส่วน คือ การจำลองเอเจนท์เคลื่อนที่ ซึ่งคอยจัดสรรทรัพยากรที่จำเป็นต่อการทำงาน และควบคุมวัฏจักรชีวิตของเจนท์เคลื่อนที่ สำหรับแอกเลตส์เฟรมเวิร์คนั้น มีตาฮิติเซิร์ฟเวอร์ (Tahiti Server) เป็นเอเจนท์เซิร์ฟเวอร์ ทำหน้าที่ควยจัดสรรทรัพยากรให้แก่เอเจนท์เคลื่อนที่ที่พัฒนาด้วยแอกเลตส์เฟรมเวิร์ค ให้สามารถติดตั้งและเริ่มการทำงานได้ โดยในเซิร์ฟเวอร์ของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยแต่ละแห่งที่ต้องการให้เอเจนท์เคลื่อนที่ของระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย สามารถเข้าไปค้นหาข้อมูลในฐานข้อมูลได้ รวมถึงเครื่องที่ติดตั้งระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยเอง จะต้องติดตั้งตาฮิติเซิร์ฟเวอร์ไว้ภายใน แอกเลตส์เฟรมเวิร์คที่นำมาใช้ในการพัฒนาระบบจะอยู่ในรูปของแฟ้มเอกสารจาวา (Java Archive) ซึ่งภายในประกอบไปด้วยชุดเครื่องมือสำหรับการพัฒนาแอกเลตส์ และตาฮิติเซิร์ฟเวอร์ โดยการติดตั้งตาฮิตินั้น ต้องคลายการบีบอัดแฟ้มเอกสารจาวาก่อนเป็นขั้นตอนแรก ด้วยการพิมพ์คำสั่งผ่านคำสั่ง ดังนั้น

```
jarxvf aglets-2.0.2.jar
```

หลังจากคลายการบีบอัด จะได้โฟลเดอร์ข้อมูลที่อยู่ภายในแฟ้มเอกสารจาวาออก แล้วจึงเริ่มการติดตั้งตาฮิติเซิร์ฟเวอร์โดยพิมพ์คำสั่ง

```
bin\ant
```

หลังจากนั้นทำงากรติดตั้งแฟ้มนโยบายจาวา (Java Policy) ซึ่งจะกำหนดสิทธิการใช้ทรัพยากรของตาฮิติเซิร์ฟเวอร์เช่น การเปิดซ็อกเก็ต การประมวลผลเอเจนท์ และการเข้าถึงแฟ้มข้อมูล เป็นต้น โดยพิมพ์คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bin\ant install-home
```

จะได้แฟ้ม .java.policy ในโฟลเดอร์ผู้ใช้ปริยาย ซึ่งขึ้นอยู่กับระบบปฏิบัติการที่ใช้ เมื่อการติดตั้งเสร็จสิ้น ขั้นตอนต่อมาเป็นการกำหนดการตั้งค่าตาฮิติเซิร์ฟเวอร์ของแต่ละเว็บไซต์ที่ต้องการจำลองโดยใช้แฟ้มข้อความ (Text File) แม่แบบที่แอกเลตส์เฟรมเวิร์คได้จัดเตรียมไว้ให้ มาแก้ไขหมายเลขพอร์ต ชื่อโดเมน และจัดเก็บเป็นแฟ้มข้อความใหม่ 4 แฟ้มสำหรับแต่ละเซิร์ฟเวอร์ที่จำลองขึ้น โดยมีรายละเอียดตั้งค่า ดังนี้

1. แฟ้ม origin_aglets.props เป็นแฟ้มการตั้งค่าตาฮิติเซิร์ฟเวอร์สำหรับเครื่องที่ติดตั้งระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยที่พัฒนาขึ้น โดยมีรายละเอียดการตั้งค่าดังนี้

```
...
# (optional) Port number used by agents server.
# default: 4434
maf.port=4434
...
# (optional) TCP/IP domain name of the host
atp.domain=home
...
```

2. แฟ้ม greenplum01_aglets.props เป็นแฟ้มการตั้งค่าตาฮิติเซิร์ฟเวอร์สำหรับเซิร์ฟเวอร์ของฐานข้อมูล Greenplum เครื่องที่หนึ่ง โดยมีรายละเอียดการตั้งค่าดังนี้

```
...
# (optional) Port number used by agents server.
# default: 4434
maf.port=5000
...
# (optional) TCP/IP domain name of the host
atp.domain=server01
...
```

3. แฟ้ม greenplum02_aglets.props เป็นแฟ้มการตั้งค่าตาฮิติเซิร์ฟเวอร์สำหรับเซิร์ฟเวอร์ของฐานข้อมูล Greenplum เครื่องที่สอง โดยมีรายละเอียดการตั้งค่าดังนี้

```
...
# (optional) Port number used by agents server.
# default: 4434
maf.port=5001
...
# (optional) TCP/IP domain name of the host
atp.domain=server02
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดแฟ้มข้อความกาตั้งค่าสำหรับแต่ละเซิร์ฟเวอร์เสร็จเรียนร้อย การเริ่มการทำงานตาฮิติเซิร์ฟเวอร์สามารถทำได้โดยพิมพ์คำสั่ง

```
agletsd –f c:\java\aglets\cnf\origin_aglets.props –port 4434
```

โดยเปลี่ยนตำแหน่งและชื่อของแฟ้มข้อความตั้งค่าไปตามที่จัดเก็บและชื่อของแต่ละแฟ้มตาฮิติเซิร์ฟเวอร์จะเริ่มการทำงาน โดยแสดงหน้าจอดังรูปที่

1.1.1 การจำลองฐานข้อมูลกรีนพลัม

ฐานข้อมูลกรีนพลัม มี 3 ส่วน คือ มาสเตอร์ เซกเมนต์ และส่วนเชื่อมต่อ ในโครงการนี้จะจำลองฐานข้อมูลบนระบบปฏิบัติการลีนุกซ์ โดยมีขั้นตอนการจำลองดังนี้

1.1.1.1 การติดตั้งฐานข้อมูลกรีนพลัม

โดยมีขั้นตอนและรายละเอียดดังนี้

1. ดาวน์โหลดโปรแกรมติดตั้งฐานข้อมูลระบบปฏิบัติการลีนุกซ์จากเว็บไซต์ของฐานข้อมูลกรีนพลัม หลังจากนั้นอัพโหลดโปรแกรมเข้าเครื่องเซิร์ฟเวอร์ลีนุกซ์ที่ได้เตรียม โดยกำหนดให้เครื่องนี้เป็นเครื่องมาสเตอร์
2. ทำการคลายการบีบอัดของไฟล์

```
unzip greenplum-db-4.0.1.0-SuSE10-x86_64.zip
```

3. รันคำสั่งติดตั้งโปรแกรมฐานฐานข้อมูล

```
/bin/bash greenplum-db-4.0.1.0-SuSE10-x86_64.bin
```

4. ต่อมาจะแสดงสารบบที่ติดตั้งโปรแกรมโดยปริยาย คือ /usr/local/greenplum-db-4.0.1.0 ถ้าเลือกสารบบนี้ได้ให้กด enter
5. เมื่อติดตั้งเสร็จจะกำหนดพารามิเตอร์ $GPHOME คือ /usr/local/greenplum-db-4.0.1.0
6. ลงบันทึกเข้าเครื่องเซิร์ฟเวอร์มาสเตอร์ด้วยชื่อผู้ใช้ root

```
su –
```

7. รันคำสั่งเพื่อกำหนดพารามิเตอร์ที่ฐานข้อมูลกรีนพลัมต้องใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
source /usr/local/greenplum-db-4.0.1.0/greenplum_path.sh
```

8. สร้างไฟล์ hostfile_exkeys ที่เก็บชื่อเครื่องเซิร์ฟเวอร์มาสเตอร์และเซกเมนต์ ตามตัวอย่างด้านล่างจะมีเซิร์ฟเวอร์มาสเตอร์ 1 เครื่อง คือ mdw และเซกเมนต์ 2 เครื่อง คือ sdw1 และ sdw2 ซึ่งต่ละเครื่องมี 2 เซกเมนต์ย่อย

```
mdw
mdw-1
mdw-2
sdw1
sdw1-1
sdw1-2
sdw2
sdw2-1
sdw2-2
```

ในไฟล์ /etc/hosts ของแต่เซิร์ฟเวอร์ต้องกำหนดไอพีและชื่อเครื่องที่ต้องสามารถติดต่อกันได้

9. รันคำสั่ง gpseginstall เพื่อติดตั้งโปรแกรมฐานข้อมูลที่เซกเมนต์ โดยอ้างอิงไฟล์ hostfile_exkeys ที่ได้สร้างไว้ พร้อมกับสร้างชื่อผู้ใช้และรหัสที่เป็นของผู้ดูแลฐานข้อมูล

```
gpseginstall -f hostfile_exkeys -u gpadmin -p password
```

10. ลงบันทึกเข้าเครื่องเซิร์ฟเวอร์มาสเตอร์ด้วยชื่อผู้ใช้ gpadmin

```
su – gpadmin
```

11. สร้างไฟล์ hostfile_gpinitsystem เพื่อเก็บชื่อเครื่องเซกเมนต์ที่เก็บสารบบของข้อมูล โดยเก็บชื่อเครื่องเซกเมนต์ตามด้านล่าง

```
sdw1-1
sdw1-2
sdw2-1
sdw2-2
```

12. คัดลอกไฟล์ gpinitsystem_config จากสารบบที่ได้ติดตั้งโปรแกรมฐานข้อมูลกรีนพลัมมาที่สารบบของผู้ใช้ gpadmin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cp $GPHOME/docs/cli_help/gpconfigs/gpinitsystem_config /home/gpadmin/gpconfigs/gpinitsystem_config
```

13. แก้ไขไฟล์ gpinitsystem_config โดยมี 4 เซกเมนต์

```
SEG_PREFIX=gpseg
PORT_BASE=40000
declare -a DATA_DIRECTORY=(/data1/primary /data1/primary
/data2/primary /data2/primary)
MASTER_HOSTNAME=mdw
MASTER_DIRECTORY=/data/master
MASTER_PORT=5432
TRUSTED SHELL=ssh
CHECK_POINT_SEGMENT=4
ENCODING=UNICODE
```

14. สร้างฐานข้อมูลกรีนพลัม โดยใช้คำสั่ง gpinitsystem โดยอ้างอิงไฟล์ gpinitsystem_config และ hostfile_gpinitsystem ที่ได้สร้างไว้

```
gpinitsystem -c gpconfigs/gpinitsystem_config -h gpconfigs/hostfile_gpinitsystem
```

15. เมื่อสร้างฐานข้อมูลกรีนพลัมสำเร็จแล้วจะแสดงข้อความ

```
=> Greenplum Database instance successfully created.
```

1.1.1.2 แอตทริบิวต์ภายในฐานข้อมูลกรีนพลัมสำหรับระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูล

แอตทริบิวต์ภายในฐานข้อมูลกรีนพลัม ในแต่ละเซิร์ฟเวอร์จะใช้รูปที่เหมือนกัน และได้ออกแบบความสัมพันธ์ของเอนทิตีภายในฐานข้อมูลดังนี้

1. **การโหลดของเซิร์ฟเวอร์ของฐานข้อมูล**

ตารางที่ 4.1 รายละเอียดเอนทิตี GP_SERVER_SKEW

เอนทิตี SERVER_SKEW ตารางหน่วยการโหลดของเซิร์ฟเวอร์				
ชื่อแอตทริบิวต์	ชนิดข้อมูล	คำอธิบาย	คีย์	ตารางอ้างอิง
SERVER_ID	integer	รหัสเซิร์ฟเวอร์	PK	
SERVER_NAME	varchar(20)	ชื่อเซิร์ฟเวอร์		
LOAD_SERVER	integer	จำนวนหน่วยของการโหลด		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **ทรัพยากรของเครื่องเซิร์ฟเวอร์**

ตารางที่ 4.2 รายละเอียดเอนทิตี SERVER_RESOURCE

เอนทิตี SERVER_RESOURCE ตารางการใช้ทรัพยากรของเซิร์ฟเวอร์				
ชื่อแอตทริบิวต์	ชนิดข้อมูล	คำอธิบาย	คีย์	ตารางอ้างอิง
SERVER_ID	integer	รหัสเซิร์ฟเวอร์	PK	
SERVER_NAME	varchar(20)	ชื่อเซิร์ฟเวอร์		
CPU	integer	เปอร์เซ็นต์ของซีพียูที่ใช้ปัจจุบัน		
MEMORY	integer	เปอร์เซ็นต์ของหน่วยความจำที่ใช้ปัจจุบัน		
Disk	integer	เปอร์เซ็นต์ของพื้นที่ดิสก์ที่ใช้		

3. **การกำหนดค่าที่ต้องการเฝ้าสังเกต**

ตารางที่ 4.3 รายละเอียดเอนทิตี CONFIGURATION

เอนทิตี CONFIGURATION ตาราง โครงแบบสำหรับตั้งค่าในระบบ				
ชื่อแอตทริบิวต์	ชนิดข้อมูล	คำอธิบาย	คีย์	ตารางอ้างอิง
CONFIG_ID	integer	รหัสรายการ	PK	
CONFIG_NAME	varchar(20)	ชื่อรายการตั้งค่า		
VALUE	Decimal(18,2)	ค่าที่ถูกตั้งไว้		
UNIT	varchar(20)	หน่วยของรายการที่ตั้งค่า		
EXECUTION_TYPE	varchar(20)	ประเภทการดำเนินการ		
FREQUENCY	integer	ความถี่ในการดำเนินการ		
EXECUTION_TIME	time	เวลาที่ดำเนินการ		
START_DATE	date	วันที่เริ่ม		
END_DATE	date	วันที่สิ้นสุด		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การจัดสรรทรัพยากรฐานข้อมูล

ตารางที่ 4.4 รายละเอียดเอนทิตี RESQUEUE_STATUS

เอนทิตี RESQUEUE_STATUS ตารางรายละเอียดกลุ่ม Resource queue				
ชื่อแอตทริบิวต์	ชนิดข้อมูล	คำอธิบาย	คีย์	ตารางอ้างอิง
GROUP_ID	integer	รหัสกลุ่มของ Resource queue	PK	
RSQ_NAME	varchar(20)	ชื่อของกลุ่ม Resource queue		
RSQ_COUNT_LIMIT	integer	จำนวนเอสคิวแอลที่ดำเนินการได้มากที่สุด		
RSQ_COUNT_VALUE	integer	จำนวนเอสคิวแอลที่กำลังดำเนินการ		
RSQ_COST_LIMIT	decimal(18,2)	ผลรวมของ cost ของเอสคิวแอลที่สามารถดำเนินการได้มากที่สุด		
RSQ_COST_VALUE	decimal(18,2)	ผลรวมของ cost ของเอสคิวแอลที่สามารถดำเนินการอยู่		
RSQ_WAIT	integer	จำนวนเอสคิวแอลที่กำลังรอดำเนินการ		

5. การล็อคเทเบิล

ตารางที่ 4.5 รายละเอียดเอนทิตี LOCK_PROCESS

เอนทิตี LOCK_PROCESS ตารางเอสคิวแอลที่กำลังถูกล็อกจากเอสคิวแอลอื่น				
ชื่อแอตทริบิวต์	ชนิดข้อมูล	คำอธิบาย	คีย์	ตารางอ้างอิง
LOCK_ID	integer	รหัสการล็อก	PK	
LOCK_TYPE	varchar(20)	ชนิดของการล็อก		
LOCK_DURATION	Integer	ระยะเวลาที่ล็อก(นาที)		
USERNAME	varchar(50)	ชื่อเจ้าของเอสคิวแอลที่ถูกล็อก		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 (ต่อ)

ชื่อแอตทริบิวต์	ชนิดข้อมูล	คำอธิบาย	คีย์	ตารางอ้างอิง
PROCESS_ID	integer	รหัสการดำเนินการของเอสคิวแอลที่ถูกล็อก		
SQL_TEXT	text	เอสคิวแอลที่ถูกล็อก		
LOCK_BY_USERNAME	varchar(50)	ชื่อเจ้าของเอสคิวแอลที่ล็อกอยู่		
LOCK_BY_PROCESS_ID	integer	รหัสการดำเนินการของเอสคิวแอลที่ล็อกอยู่		
LOCK_BY_SQL	text	เอสคิวแอลที่ล็อก		

6. ประวัติการดำเนินการของคำสั่งเอสคิวแอล

ตารางที่ 4.6 รายละเอียดเอนทิตี SQL_HISTORY

เอนทิตี SQL_HISTORY ตารางประวัติ SQL ที่ดำเนินการในฐานข้อมูล				
ชื่อแอตทริบิวต์	ชนิดข้อมูล	คำอธิบาย	คีย์	ตารางอ้างอิง
SQL_ID	integer	รหัสเอสคิวแอล	PK	
GROUP_ID	integer	รหัสกลุ่มของ Resource queue	FK	RESQUEUE_STATUS
SQL_TEXT	varchar(20)	ชื่อของกลุ่ม Resource queue		
COST	decimal(18,2)	cost ที่ใช้ในการดำเนินการ		
START_TIME	timestamp	เวลาที่เริ่มดำเนินการ		
END_TIME	timestamp	เวลาที่สิ้นสุดดำเนินการ		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การพัฒนาระบบ

ในส่วนการพัฒนาระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วยโดยใช้เทคโนโลยีเอเจนท์เคลื่อนที่ จะกล่าวถึงรายละเอียดขั้นตอนการทำงานของเอเจนท์เคลื่อนที่แต่ละตัวที่ทำงานภายในระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย ซึ่งประกอบไปด้วยเอเจนท์เคลื่อนที่ 4 ตัวดังต่อไปนี้

4.3.1 เอเจนท์ส่วนต่อประสานกับผู้ใช้

เอเจนท์ส่วนต่อประสานกับผู้ใช้ หรือคลาส GUIAgent เป็นเอเจนท์แบบอยู่ประจำที่ ทำหน้าที่เป็นคลาสเริ่มต้นการทำงานของระบบ โดยเมื่อเริ่มการทำงานของระบบ เอเจนท์ส่วนต่อประสานกับผู้ใช้จะถูกสร้างขึ้นภายในตาฮิติเซิร์ฟเวอร์ ซึ่งเป็นสภาพแวดล้อมการทำงานของระบบเอเจนท์เคลื่อนที่ที่พัฒนาด้วยแอกเลตส์เฟรมเวิร์ค

นอกจากเป็นคลาสเริ่มต้นของระบบแล้ว เอเจนท์ส่วนต่อประสานกับผู้ใช้ ยังทำหน้าที่สร้างส่วนติดต่อกับผู้ใช้ ทั้งการจัดการข้อมูลตั้งต้นของระบบ การรับเงื่อนไขการค้นหา รวมถึงการแสดงผลการค้นหา ซึ่งประกอบไปด้วย 4 หน้าจอหลัก ดังต่อไปนี้

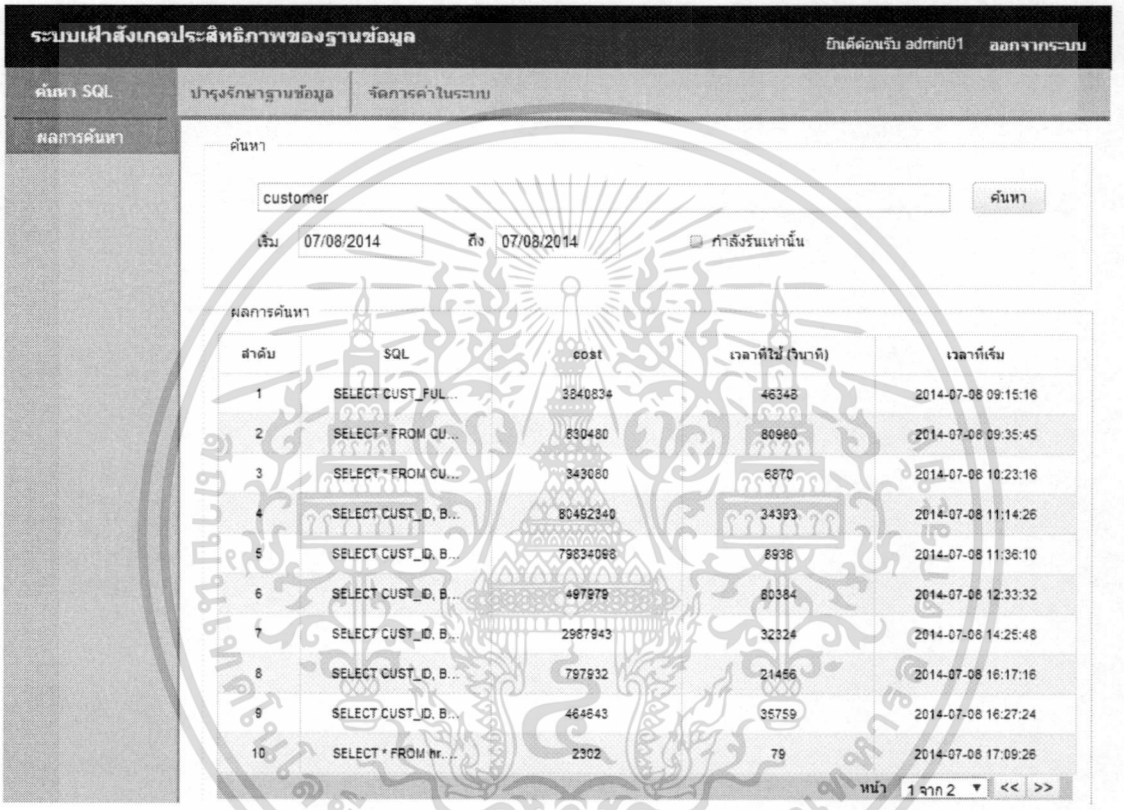
1. **หน้าการค้นหา และแสดงผลเอสคิวแอลที่ใช้เวลาดำเนินงานที่ฐานข้อมูลนาน หรือใช้ทรัพยากรเครื่องสูง**

รูปที่ 4.1 หน้าการค้นหาเอสคิวแอลที่ใช้เวลาดำเนินงานที่ฐานข้อมูลนาน หรือใช้ทรัพยากรเครื่องสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 แสดงหน้าการค้นหาและแสดงผลลัพธ์การค้นหาซึ่งเป็นหน้าจอหลัก จะถูกแสดงขึ้นมาหลังจากที่ผู้ใช้งานเข้าสู่ระบบ และเอเจนท์ส่วนต่อประสานกับผู้ใช้ถูกสร้างขึ้นบนตาฮิติเซิร์ฟเวอร์เสร็จสิ้น โดยจะประกอบด้วยส่วนระบุคำที่ต้องการค้นหา ตามช่วงเวลาที่ผู้ใช้งานกำหนด

เมื่อกดปุ่มค้นหา ส่วนแสดงผลลัพธ์การค้นหาจะแสดงขึ้นมาทางด้านล่างของหน้าจอ และเมื่อเอเจนท์ค้นหารายงานผลลัพธ์การค้นหากลับมายังต้นทาง ผลลัพธ์การค้นหาจะถูกแสดงในตารางของส่วนแสดงผลลัพธ์ตามรูปที่ 4.2

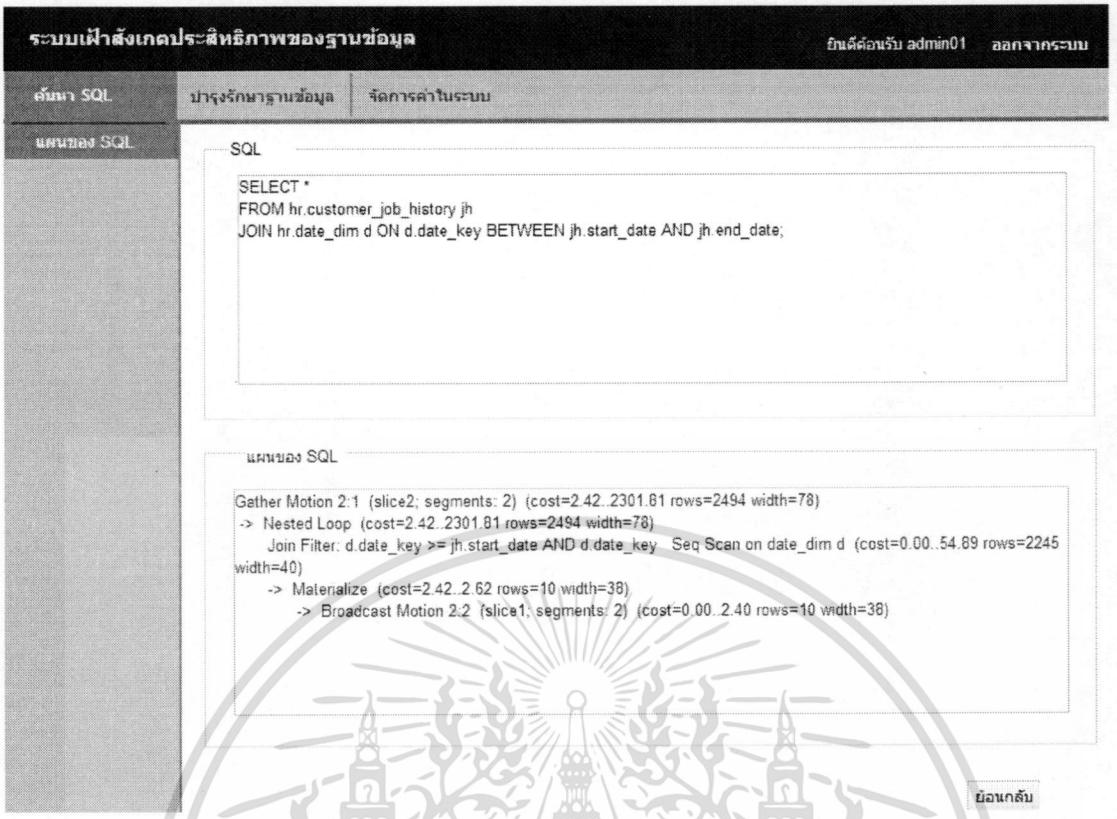


รูปที่ 4.2 หน้าแสดงผลเอสคิวแอล

2. หน้าแสดงข้อมูลเอสคิวแอล

หลังจากที่ข้อมูลเอสคิวแอลที่ค้นพบ ได้ถูกนำมาแสดงในส่วนแสดงผลลัพธ์การค้นหาแล้ว ผู้ใช้งานสามารถดูรายละเอียดแผนการดำเนินการของเอสคิวแอลได้โดยดับเบิลคลิกที่รายการเอสคิวแอลนั้น จะปรากฏหน้าแสดงแผนการดำเนินการของเอสคิวแอลขึ้นมาตามที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าแสดงข้อมูลเอสคิวแอล

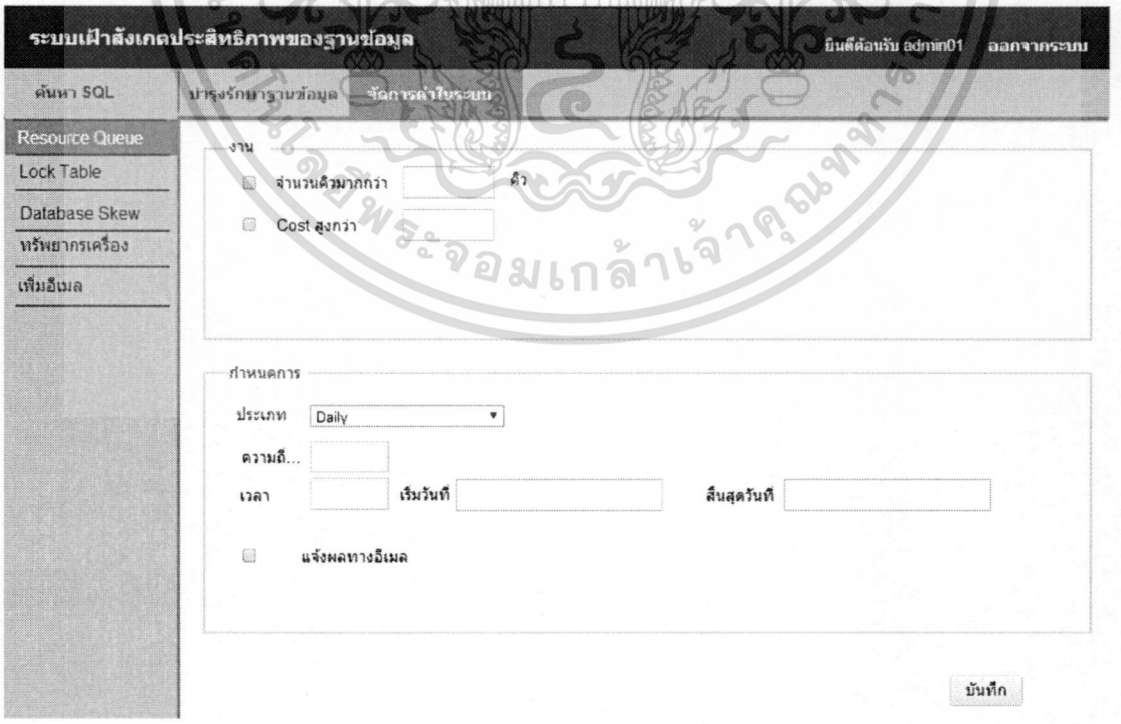
3. หน้าจัดการรายการบำรุงรักษาฐานข้อมูล

ผู้ใช้งานสามารถจัดการรายการบำรุงรักษาฐานข้อมูลเพื่อป้องกันเอสคิวแอลดำเนินการที่ฐานข้อมูลนาน โดยการตั้งเวลาที่ต้องการทำจัดการรายการบำรุงรักษาฐานข้อมูล โดยผู้ใช้งานควรเลือกเวลาดำเนินการในช่วงที่มีผู้ใช้งานน้อย เพราะบางรายการจะทำให้ table lock ดังนั้นผู้ดูและฐานข้อมูลจึงพิจารณาช่วงที่เหมาะสมในการตั้งเวลา โดยเลือกช่วงเวลาที่กระทบกับผู้ใช้งานฐานข้อมูลให้น้อยที่สุด โดยตัวอย่างหน้าจอตามรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 หน้าจัดการรายการบำรุงรักษาฐานข้อมูล

4. หน้าจัดการรายการสำหรับเฝ้าสังเกตประสิทธิภาพของฐานข้อมูล



รูปที่ 4.5 หน้าจัดการรายการสำหรับเฝ้าสังเกตประสิทธิภาพของฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.5 ผู้ใช้งานสามารถจัดการรายการสำหรับเฝ้าสังเกตประสิทธิภาพของฐานข้อมูล โดยเลือกรายการที่ต้องกการเฝ้าสังเกต สามารถกำหนดช่วงเวลาที่ต้องการเฝ้าสังเกตและตรวจสอบประสิทธิภาพของฐานข้อมูลในทุก ๆ ช่วงเท่าไร ผู้ใช้งานสามารถปรับเปลี่ยนเกณฑ์สำหรับแจ้งเตือนผ่านทางอีเมล

4.4 การเชื่อมต่อฐานข้อมูล

ระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูลในโครงการนี้มีรายละเอียดการเชื่อมต่อฐานข้อมูล และค้นคืนข้อมูลในฐานข้อมูล ดังต่อไปนี้

4.3.2 การเชื่อมต่อฐานข้อมูล

เนื่องจากระบบเฝ้าสังเกตประสิทธิภาพของฐานข้อมูล พัฒนาขึ้นด้วยภาษาจาวา ดังนั้นการเชื่อมต่อฐานข้อมูลจึงกระทำผ่านทางการเชื่อมต่อฐานข้อมูลจาวา (Java Database Connection) หรือ JDBC ซึ่งเปรียบเสมือนตัวกลางในการเชื่อมโยงระหว่างภาษาจาวากับฐานข้อมูลที่ต้องการติดต่อ โดยระบบบริหารจัดการฐานข้อมูลที่รองรับการทำงานร่วมกับ JDBC จะมีชุดไดรฟ์เวอร์สำหรับฐานข้อมูลนั้น ซึ่งจะทำหน้าที่แปลคำสั่งจาก JDBC ไปเป็นคำสั่งสำหรับเข้าใช้งานระบบฐานข้อมูลอีกครั้งหนึ่ง

ดังนั้น เพื่อให้เอเจนท์เคลื่อนที่สามารถเชื่อมต่อกับระบบบริหารจัดการฐานข้อมูลได้ จึงติดตั้งไดรฟ์เวอร์สำหรับฐานข้อมูลในเซิร์ฟเวอร์ โดยนำไดรฟ์เวอร์สำหรับระบบบริหารจัดการฐานข้อมูลแต่ละตัวไปติดตั้งยังคลาสพาธของตาฮิติเซิร์ฟเวอร์ของฐานข้อมูลแบบขนานหลายหน่วยที่จำลองขึ้น โดยปกติคลาสพาธของตาฮิติเซิร์ฟเวอร์ คือ

```
%AGLETS-INSTALL-HOME%\aglets\lib
```

โดยที่ %AGLETS-INSTALL-HOME% คือตำแหน่งไดเรคทอรีที่ติดตั้งแอกเลตส์เฟรมเวิร์ค สำหรับการสร้างการเชื่อมต่อกับฐานข้อมูลประกอบด้วย 2 ขั้นตอน คือ โหลดไดรฟ์เวอร์สำหรับฐานข้อมูลที่ต้องการเชื่อมต่อขึ้นมาทำงาน โดยใช้คำสั่ง

```
Class.forName( <Driver Name> );
```

และหลังจากนั้นจึงเชื่อมต่อฐานข้อมูลโดยใช้คำสั่ง

```
DriverManager.getConnection( <Database URL> );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั้งชื่อไดรฟ์เวอร์ (Driver Name) และ URL ของฐานข้อมูล (Database URL) จะถูกกำหนดอยู่ที่แฟ้มนิยามทรัพยากร ซึ่งทั้งเอเจนท์ค้นหาและเอเจนท์เฝ้าสังเกต จะนำข้อมูลเหล่านี้มาใช้ในการสร้างการเชื่อมต่อกับฐานข้อมูลแบบขนานหลายหน่วยที่จำลองขึ้น หลังจากที่เชื่อต่อกับฐานข้อมูลเสร็จสิ้น เอเจนท์เคลื่อนที่จะสามารถค้นคืนข้อมูลในฐานข้อมูลได้ตามคำสั่งค้นคืนที่ถูกสร้างขึ้นตามเงื่อนไขการค้นหาที่ผู้ใช้งานระบุ

4.4.2 การค้นคืนข้อมูล

หลักการทำงานในการสร้างคำสั่ง SQL สำหรับค้นคืนข้อมูลนั้น เริ่มจากเมื่อผู้ใช้งานเริ่มค้นหาข้อมูล เอเจนท์ส่วนต่อประสานกับผู้ใช้ จะอ่านข้อมูลภายในแฟ้มนิยามทรัพยากร หลังจากนั้นจึงส่งข้อมูลดังกล่าวให้เอเจนท์การจัดการพร้อมกับเงื่อนไขการค้นหาที่ผู้ใช้งานกำหนด เอเจนท์การจัดการจะนำคำสั่งสำหรับการค้นคืนข้อมูล สามารถแสดงได้ เมื่อได้คำสั่ง SQL สำหรับค้นคืนข้อมูลแล้ว จึงส่งให้เอเจนท์เคลื่อนที่ ทั้งเอเจนท์ค้นหาและเอเจนท์เฝ้าสังเกตเพื่อนนำไปใช้ค้นคืนข้อมูลในฐานข้อมูลที่ได้รับมอบหมาย

วิธีการสร้างคำสั่ง SQL สำหรับค้นคืนข้อมูลจากฐานข้อมูลประมวลผลแบบขนานหลายหน่วย สามารถแสดงได้ดังรูปที่ คือ ตัวอย่างข้อมูลที่จัดกับอยู่ในแฟ้นนิยามทรัพยากร ซึ่งเป็นข้อมูลที่อธิบายรายละเอียดของเอสคิวแอล เช่น ชื่อแอตทริบิวต์ และชื่อตารางที่จัดเก็บข้อมูล เป็นต้น ซึ่งถูกนำมาประกอบเข้ากับคำสั่งเอสคิวแอล ทั้งในส่วนแอตทริบิวต์ เพื่อระบุว่าต้องการข้อมูลจากแอตริบิวต์ใดบ้าง และส่วนตารางข้อมูลเพื่อระบุว่าข้อมูลที่ต้องการอยู่ในตารางใด นอกจากนั้น เงื่อนไขการค้นหาที่ผู้ใช้งานกำหนดในหน้าจอหลัก เช่น คำสำคัญ ช่วงเวลาที่เอสคิวแอลดำเนินการ เป็นต้น จะถูกนำมาเป็นเงื่อนไขของคำสั่งเอสคิวแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุปโครงการ

โครงการพัฒนาระบบเอเจนท์เฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย เพื่ออำนวยความสะดวกให้กับผู้ดูแลฐานข้อมูล โดยจะทำหน้าที่แทนผู้ดูแลฐานข้อมูลในการเฝ้าระวังการดำเนินการของคำสั่งภาษาเอสคิวแอลในฐานข้อมูล และการใช้ทรัพยากรเครื่องตามที่กำหนดเงื่อนไขไว้ได้อย่างอัตโนมัติ ผู้ดูแลฐานข้อมูลไม่ต้องเฝ้าระวังฐานข้อมูลด้วยตนเองตลอดเวลา และช่วยลดความล่าช้าในการหาสาเหตุของปัญหาเกี่ยวกับประสิทธิภาพของฐานข้อมูลในการดำเนินการของคำสั่งภาษาเอสคิวแอลจากผู้ใช้

5.2 ประโยชน์ที่ได้รับ

ระบบเอเจนท์เฝ้าสังเกตประสิทธิภาพของฐานข้อมูลประมวลผลแบบขนานหลายหน่วย เป็นระบบที่สร้างขึ้นเพื่อตอบสนอง และอำนวยความสะดวกแก่ผู้ดูแลฐานข้อมูล โดยประโยชน์ที่ได้รับ มีดังนี้

1. สามารถแจ้งเตือนเมื่อมีคำสั่งภาษาเอสคิวแอลที่ทำให้ฐานข้อมูลมีปัญหา และสามารถค้นหาคำสั่งภาษาเอสคิวแอลที่กำลังดำเนินการอยู่หรือได้ดำเนินการไปแล้ว โดยเลือกช่วงเวลาได้
2. สามารถแจ้งเตือนการใช้ทรัพยากรเครื่องตามค่าที่กำหนด
3. สามารถทำหน้าที่บำรุงรักษาฐานข้อมูลแทนผู้ดูแลระบบฐานข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Aglets Development Group. 2009. **The Aglets 2.0.2 User's Manual.** [Online]. Available: http://biznetnetworks.dl.sourceforge.net/project/aglets/User_s%20Manual/March%202009/manual031209.pdf.

Arkady, Z. 2009. **Mobile and Distributed Computing Systems**. [Online]. Available : http://www.csse.monash.edu.au/~azaslavs/cse5501/

EMC Corporation. 2011. **Greenplum Database 4.1 Administrator Guide.** [Online]. Available : https://community.emc.com/community/support/greenplum?view=documents.

Lange, D. and Oshima M. 2009. **Introduction to Mobile Agents.** [Online]. Available : www.springerlink.com.

Nikhil, K. 1997. **Agentos – A Java Based Mobile Agent System.** [Online]. Available : http://netresearch.ics.uci.edu/agentos/nkothari/.

Weigel, C. 2009. IBM Agents. [Online]. Available : http://ai.ijs.si/Mezi/pedagosko/AgentsWeigel.pdf.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้