

ศึกษาและพัฒนาระบบกระจายโหลดฐานข้อมูล

STUDY AND DEVELOP DATABASE SERVER LOAD BALANCING SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STUDY AND DEVELOP DATABASE SERVER LOAD BALANCING SYSTEM



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบเสนอปริญญาบัตร

หัวข้อปริญญาบัตร ศึกษาและพัฒนาระบบกระจายโหนดฐานข้อมูล
ชื่อนักศึกษา นางสาวชาติณี ชูคินันท์ รหัสประจำตัว 43010104
นายณัฐ ถิမ် โสภารธรรม รหัสประจำตัว 43010123
อาจารย์ที่ปรึกษา อาจารย์สุธีรา พันธุ์ธีรานุรักษ์
อาจารย์ชงค์ หงษ์สุวรรณ
ระดับการศึกษา ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ
ภาควิชา วิศวกรรมสารสนเทศ
ปีการศึกษา 2546

ปริญญาบัตรฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....
(อาจารย์สุธีรา พันธุ์ธีรานุรักษ์)

.....
(อาจารย์ชงค์ หงษ์สุวรรณ)

.....
อาจารย์ผู้ควบคุมวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|--------------------|---|-----------------------|
| หัวข้อปริญญานิพนธ์ | ศึกษาและพัฒนาระบบกระจายโพลฐานข้อมูล | |
| ชื่อนักศึกษา | นางสาวชาลิณี ชุตินันท์ | รหัสประจำตัว 43010104 |
| | นายณัฐ ลิม โสภาทรรณ | รหัสประจำตัว 43010123 |
| อาจารย์ที่ปรึกษา | อาจารย์สุธีรา พันธุ์ธีรานุกัณฑ์ อาจารย์ภูษงค์ หงษ์สุวรรณ | |
| ระดับการศึกษา | ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ | |
| ภาควิชา | วิศวกรรมสารสนเทศ | |
| ปีการศึกษา | 2546 | |

บทคัดย่อ

ระบบกระจายโพลฐานข้อมูลนี้ถูกพัฒนาขึ้นมา เพื่อจัดการการกระจายโพล ไปยังเซิร์ฟเวอร์ฐานข้อมูลหลาย ๆ ตัว เพื่อให้เกิดประสิทธิภาพในการทำงานที่สูงขึ้น โดยที่เมื่อ โปรแกรมประยุกต์ต้องการใช้งานเซิร์ฟเวอร์ฐานข้อมูล ค่าส่งที่ใช้ในการคิดคํ่ากับฐานข้อมูลก็จะถูกส่งไปยังเครื่องคํ่าเบสเซิร์ฟเวอร์ โพลคบาลานซ์ซึ่ง จากนั้นเครื่องคํ่าเบสเซิร์ฟเวอร์โพลคบาลานซ์ซึ่งจะเป็นตัวกำหนดว่าจะคํ่าต่อไปยังเซิร์ฟเวอร์ฐานข้อมูลตัวใด โดยพิจารณาจากเซิร์ฟเวอร์ฐานข้อมูลที่มีภาระงานน้อยที่สุด จากนั้นก็จะส่งคํ่าส่งไปยังเซิร์ฟเวอร์ฐานข้อมูลตัวนั้น เพื่อให้เซิร์ฟเวอร์ฐานข้อมูลตัวนั้นนำไปประมวลผล และส่งผลที่ได้กลับมายังเครื่องคํ่าเบสเซิร์ฟเวอร์โพลคบาลานซ์ซึ่ง จะทำการส่งผลลัพธ์กลับมายัง โปรแกรมประยุกต์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Study and Develop Database Server Load Balancing System

Student Ms. Chalinee Chutinun ID. 43010104
Mr. Nut Limsopatham ID. 43010123

Advisor Ms. Sutheera Puntheeranurak
Mr. Puchong Hongsuwan

Graduate Level Bachelor Degree of Information Engineering

Department Information Engineering

Academic Year 2003



ABSTRACT

Database server load balancing system is developed for manage the load of database server to increase efficiency of system. When the application want to connect to database it will connect to Database load balancing that define the database server is connected and connect it by select from database server that has the least tasks. After SQL is sent it to execute. Result is sent Database load balancing that send result to applications.

สารบัญ

| เรื่อง | หน้า |
|--|------|
| สารบัญ | จ |
| สารบัญรูปภาพ | ช |
| สารบัญตาราง | ฉ |
| บทที่ 1 บทนำ | 1 |
| 1.1 แนวความคิดและที่มา | 1 |
| 1.2 จุดประสงค์ | 3 |
| 1.3 ขอบเขตโครงการ | 3 |
| 1.4 ขั้นตอนการดำเนินโครงการ | 4 |
| 1.5 ผลที่คาดว่าจะได้รับ | 4 |
| บทที่ 2 ทฤษฎีและหลักการที่ใช้ในโครงการ | 5 |
| 2.1 ความรู้พื้นฐานเกี่ยวกับเจดีบีซี | 5 |
| 2.2 ชนิดของไดร์เวอร์เจดีบีซี | 5 |
| 2.2.1 ประเภท 1 (Type1: JDBC-ODBC Bridge) | 6 |
| 2.2.2 ประเภท 2 (Type2: Native-API/partly Java driver) | 7 |
| 2.2.3 ประเภท 3 (Type 3: Net-protocol/all-Java driver) | 8 |
| 2.2.4 ประเภท 4 (Type 4: Native-protocol/all-Java driver) | 9 |
| 2.3 วิธีการสร้างไดร์เวอร์เจดีบีซีประเภท 3 | 13 |
| 2.3.1 สถาปัตยกรรมของไดร์เวอร์เจดีบีซี | 13 |
| 2.3.2 คลาสไลบรารีของเจดีบีซีไดร์ฟเวอร์ | 16 |
| 2.3.2.1 คลาสต่างๆ ของไคลเอนต์เทียร์ | 16 |
| 2.3.2.1.1 JWDriver class | 17 |
| 2.3.2.1.2 JWConnection class | 19 |
| 2.3.2.1.3 JWStatement class | 20 |
| 2.3.2.1.4 JWResultSet class | 22 |
| 2.3.2.2 คลาสต่างๆ ของมิดเดิลเทียร์ | 24 |
| 2.3.2.2.1 RemoteDriverImpl | 24 |
| 2.3.2.2.2 RemoteConnectionImpl | 26 |
| 2.3.2.2.3 RemoteStatementImpl | 27 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

| เรื่อง | หน้า |
|---|------|
| 2.3.2.2.4 RemoteResultSetImpl | 27 |
| 2.4 การเพิ่มความเร็วด้วยการใช้งานคอนเน็คชันพูล | 29 |
| 2.4.1 คอนเน็คชันพูล | 29 |
| 2.4.1.1 การสร้างคอนเน็คชันพูล | 30 |
| 2.4.1.2 การดึงคอนเน็คชันจากพูลมาใช้งาน | 31 |
| 2.4.1.3 การคืนคอนเน็คชันสู่พูล | 32 |
| บทที่ 3 การออกแบบ | 33 |
| 3.1 ข้อมูลเบื้องต้น | 33 |
| 3.1.1 ส่วนผู้ดูแลระบบ | 33 |
| 3.1.2 ส่วนทดสอบระบบ | 33 |
| 3.2 การออกแบบระบบ | 33 |
| 3.2.1 สถาปัตยกรรมของระบบ (System Architecture) | 33 |
| 3.2.2 ไคลเอนต์อินเทอร์วิว (Client Interview) | 35 |
| 3.2.3 ไฮเลเวลยูสเคส (High Level Use Case) | 36 |
| 3.2.4 ยูสเคสไดอะแกรม (Use Case Diagram) | 36 |
| 3.2.5 แอกทิวิตีไดอะแกรม (Activity Diagram) | 43 |
| 3.2.6 การออกแบบส่วนติดต่อผู้ใช้ (User Interface) | 54 |
| บทที่ 4 ผลการทดลอง | 59 |
| 4.1 ฟังก์ชันสำหรับผู้ดูแลระบบ (Administrator and Moderator) | 59 |
| 4.1.1 ฟังก์ชันสำหรับเพิ่มผู้ใช้ใหม่เข้าไปในระบบ | 59 |
| 4.1.2 ฟังก์ชันสำหรับเปลี่ยนแปลงรหัสผ่านของผู้ใช้ภายในระบบ | 61 |
| 4.1.3 ฟังก์ชันสำหรับลบชื่อผู้ใช้งานในระบบ | 63 |
| 4.2 ฟังก์ชันสำหรับการทำงานของระบบ | 64 |
| 4.2.1 ฟังก์ชันในการลงทะเบียนเพื่อรอรับการติดต่อระยะไกล จากเครื่องไคลเอนต์ (RMI Registry) | 64 |
| 4.2.2 ฟังก์ชันในการกำหนดค่าเซิร์ฟเวอร์ฐานข้อมูลที่ใช้งานให้ กับเซิร์ฟเวอร์ | 65 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

| เรื่อง | หน้า |
|---|------|
| 4.2.3 ฟังก์ชันในการเริ่มต้นการใช้งานเซิร์ฟเวอร์ (Start Load Balance Server) และการแสดงค่าสถานะของเซิร์ฟเวอร์ฐานข้อมูล | 66 |
| 4.2.4 ฟังก์ชันในการรายงานค่าสถานะมายังเซิร์ฟเวอร์ | 69 |
| 4.2.5 ฟังก์ชันในการทดสอบประสิทธิภาพของระบบ | 70 |
| บทที่ 5 สรุปผลการทดลอง | 72 |
| 5.1 สรุปผลการทดลอง | 72 |
| 5.2 ปัญหาที่เกิดขึ้นระหว่างการทดลอง | 75 |
| 5.3 แนวทางการพัฒนาโครงการ | 76 |
| บรรณานุกรม | 77 |



สารบัญรูปภาพ

| ภาพ | หน้า |
|--|------|
| รูปที่ 1.1 สถาปัตยกรรมของระบบกระจาย โหลดฐานข้อมูล | 2 |
| รูปที่ 2.1 ประเภท 1 : เจดีบีซี - โอดีบีซีบริดจ์ | 6 |
| รูปที่ 2.2 ประเภท 2 : เนทีฟ เอพีไอ | 7 |
| รูปที่ 2.3 ประเภท 3 : เน็ตเวิร์ค โปรโตคอล | 8 |
| รูปที่ 2.4 ประเภท 4 : เนทีฟโปรโตคอล | 10 |
| รูปที่ 2.5 สถาปัตยกรรมไดรฟ์เวอร์เจดีบีซีประเภท 3 | 14 |
| รูปที่ 2.6 คลาสไดอะแกรมที่แสดงความสัมพันธ์ระหว่างคลาส Driver และคลาส Connection บน โคลเอนต์ทีเซอร์และมิดเคิลทีเซอร์ | 16 |
| รูปที่ 2.7 คลาสไดอะแกรมที่แสดงความสัมพันธ์ระหว่างคลาส Connection และ Statement บนฝั่งโคลเอนต์และมิดเคิลทีเซอร์ | 20 |
| รูปที่ 2.8 คลาสไดอะแกรมที่แสดงความสัมพันธ์ระหว่างคลาส Statement และ ResultSet บนฝั่งบนฝั่ง โคลเอนต์และมิดเคิลทีเซอร์ | 22 |
| รูปที่ 2.9 แสดงซีควเอนซ์ไดอะแกรมของการรับการเชื่อมต่อกลับมาจากพูล | 31 |
| รูปที่ 3.1 สถาปัตยกรรมของระบบ | 33 |
| รูปที่ 3.2 โดเมนโมเดล | 35 |
| รูปที่ 3.3 ไฮเลเวลยูสเคส | 36 |
| รูปที่ 3.4 ยูสเคสไดอะแกรมสำหรับการเชื่อมต่อกับเซิร์ฟเวอร์ (Connect Server) | 36 |
| รูปที่ 3.5 ยูสเคสไดอะแกรมสำหรับการสร้างพูล (Create Pool) | 37 |
| รูปที่ 3.6 ยูสเคสไดอะแกรมสำหรับการกรับคอนเน็คชั่น (Get Connection) | 37 |
| รูปที่ 3.7 ยูสเคสไดอะแกรมสำหรับการสร้างคอนเน็คชั่น (New Connection) | 38 |
| รูปที่ 3.8 ยูสเคสไดอะแกรมสำหรับการฟรีคอนเน็คชั่น (Free Connection) | 38 |
| รูปที่ 3.9 ยูสเคสไดอะแกรมสำหรับการอ่านค่าข้อมูลผู้ใช้ (Get Account) | 38 |
| รูปที่ 3.10 ยูสเคสไดอะแกรมสำหรับการจัดการเกี่ยวกับข้อมูลผู้ใช้ (Manage Account) | 39 |
| รูปที่ 3.11 ยูสเคสไดอะแกรมสำหรับการรายงานสถานะ (Send Status) | 39 |
| รูปที่ 3.12 ยูสเคสไดอะแกรมสำหรับการยกเลิกการเชื่อมต่อกับเซิร์ฟเวอร์ (Disconnect Server) | 39 |
| รูปที่ 3.13 ยูสเคสไดอะแกรมสำหรับการส่งคำสั่งคิวรีไปยังเซิร์ฟเวอร์ (send Query to Server) | 40 |
| รูปที่ 3.14 ยูสเคสไดอะแกรมสำหรับการส่งผลลัพธ์ไปยังโคลเอนต์ (send resultset to Client) | 40 |

สารบัญรูปภาพ (ต่อ)

| ภาพ | หน้า |
|--|------|
| รูปที่ 3.15 ยูสเคสโค้ดแอมแกรมสำหรับการร้องขอว่าเซิร์ฟเวอร์ฐานข้อมูล ขณะนี้ตัวใดถูกใช้งานน้อยสุด (request DB) | 40 |
| รูปที่ 3.16 ยูสเคสโค้ดแอมแกรมสำหรับการส่งผลไปแจ้งว่า ในขณะที่เซิร์ฟเวอร์ฐานข้อมูลตัวใดถูกใช้งานน้อยสุด (send Result DB) | 41 |
| รูปที่ 3.17 ยูสเคสโค้ดแอมแกรมสำหรับการส่งคิวรีผ่านไปในคอนเน็คชัน | 41 |
| รูปที่ 3.18 ยูสเคสโค้ดแอมแกรมสำหรับการส่งผลลัพธ์ไปยังเซิร์ฟเวอร์ (send Resultset) | 41 |
| รูปที่ 3.19 ยูสเคสโค้ดแอมแกรมสำหรับการ โค้ดแอมแกรมสำหรับการส่งคิวรีไปยังเซิร์ฟเวอร์ฐานข้อมูล (send Query to DB) | 42 |
| รูปที่ 3.20 ยูสเคสโค้ดแอมแกรมสำหรับการส่งผลลัพธ์ผ่านไปในคอนเน็คชัน | 42 |
| รูปที่ 3.21 ยูสเคสโค้ดแอมแกรมสำหรับการประมวลผลคำสั่ง (execute Command) | 42 |
| รูปที่ 3.22 แอคทิวิตีโค้ดแอมแกรมสำหรับการเชื่อมต่อกับเซิร์ฟเวอร์ (Connect Server) | 43 |
| รูปที่ 3.23 แอคทิวิตีโค้ดแอมแกรมสำหรับการสร้างพูล (Create Pool) | 44 |
| รูปที่ 3.24 แอคทิวิตีโค้ดแอมแกรมสำหรับการรับคอนเน็คชัน (Get Connection) | 45 |
| รูปที่ 3.25 แอคทิวิตีโค้ดแอมแกรมสำหรับการสร้างคอนเน็คชัน (New Connection) | 46 |
| รูปที่ 3.26 แอคทิวิตีโค้ดแอมแกรมสำหรับการฟรีคอนเน็คชัน (Free Connection) | 46 |
| รูปที่ 3.27 แอคทิวิตีโค้ดแอมแกรมสำหรับการอ่านค่าข้อมูลผู้ใช้ (Get Account) | 47 |
| รูปที่ 3.28 แอคทิวิตีโค้ดแอมแกรมสำหรับการจัดการเกี่ยวกับข้อมูลผู้ใช้ (Manage Account) | 47 |
| รูปที่ 3.29 แอคทิวิตีโค้ดแอมแกรมสำหรับการรายงานสถานะ (Send Status) | 48 |
| รูปที่ 3.30 แอคทิวิตีโค้ดแอมแกรมสำหรับการยกเลิกการเชื่อมต่อกับเซิร์ฟเวอร์ | 48 |
| รูปที่ 3.31 แอคทิวิตีโค้ดแอมแกรมสำหรับการส่งคำสั่ง Query ไปยังเซิร์ฟเวอร์ | 49 |
| รูปที่ 3.32 แอคทิวิตีโค้ดแอมแกรมสำหรับการส่งผลลัพธ์ไปยังไคลเอนต์ (send resultset to Client) | 49 |
| รูปที่ 3.33 แอคทิวิตีโค้ดแอมแกรมสำหรับการร้องขอว่าเซิร์ฟเวอร์ฐานข้อมูลขณะนี้ตัวใดถูกใช้งานน้อยสุด (request DB) | 50 |
| รูปที่ 3.34 แอคทิวิตีโค้ดแอมแกรมสำหรับการส่งผลไปแจ้งว่า ในขณะที่เซิร์ฟเวอร์ฐานข้อมูลตัวใดถูกใช้งานน้อยสุด (send Result DB) | 50 |
| รูปที่ 3.35 แอคทิวิตีโค้ดแอมแกรมสำหรับการการส่งคิวรีผ่านไปในคอนเน็คชัน | 51 |
| รูปที่ 3.36 แอคทิวิตีโค้ดแอมแกรมสำหรับการส่งผลลัพธ์ไปยังเซิร์ฟเวอร์ (send Resultset) | 51 |
| รูปที่ 3.37 แอคทิวิตีโค้ดแอมแกรมสำหรับการส่งคิวรีไปยังเซิร์ฟเวอร์ฐานข้อมูล | 52 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

| ภาพ | หน้า |
|--|------|
| รูปที่ 3.38 แอคทีวิตีไดอะแกรมสำหรับการส่งผลลัพธ์ผ่านไปในคอนเน็คชัน | 53 |
| รูปที่ 3.39 แอคทีวิตีไดอะแกรมสำหรับการประมวลผลคำสั่ง (execute Command) | 53 |
| รูปที่ 3.40 หน้าที่ใช้ในการเพิ่มชื่อผู้ใช้ใหม่เข้าสู่ระบบ | 54 |
| รูปที่ 3.41 หน้าที่ใช้ในการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ | 54 |
| รูปที่ 3.42 หน้าที่ใช้ในการลบชื่อผู้ใช้ออกจากกระบวน | 55 |
| รูปที่ 3.43 หน้าที่ใช้ในการกำหนดค่าให้กับเซิร์ฟเวอร์ | 55 |
| รูปที่ 3.44 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด กรณีที่ใช้เซิร์ฟเวอร์ ฐานข้อมูลทั้งหมด 3 เครื่อง | 56 |
| รูปที่ 3.45 หน้าที่ใช้สำหรับแสดงข้อมูลการทำงาน (log file) | 56 |
| รูปที่ 3.46 หน้าที่ใช้ในการทำการรายงานค่าสถานะมายังเซิร์ฟเวอร์ | 57 |
| รูปที่ 3.47 หน้าที่ใช้ในการทดสอบประสิทธิภาพของระบบ | 57 |
| รูปที่ 3.48 หน้าที่ใช้ในการแสดงเวลาที่ใช้ในการประมวลผลของแต่ละ ไคลเอนต์ | 58 |
| รูปที่ 4.1 หน้าที่ใช้ในการเพิ่มชื่อผู้ใช้ใหม่เข้าสู่ระบบ | 59 |
| รูปที่ 4.2 หน้าเพิ่มชื่อผู้ใช้ใหม่เข้าสู่ระบบเมื่อผู้ดูแลระบบป้อนข้อมูลอย่างถูกต้อง | 60 |
| รูปที่ 4.3 หน้าเพิ่มชื่อผู้ใช้ใหม่เข้าสู่ระบบเมื่อผู้ดูแลระบบป้อนข้อมูลชื่อผู้ใช้ที่มีอยู่แล้วในระบบ | 60 |
| รูปที่ 4.4 หน้าเพิ่มชื่อผู้ใช้ใหม่เมื่อผู้ดูแลระบบป้อนรหัสผ่านและยืนยันรหัสผ่านไม่ตรงกัน | 61 |
| รูปที่ 4.5 หน้าที่ใช้ในการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ | 61 |
| รูปที่ 4.6 หน้าเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในกรณีที่รหัสผ่านและยืนยันรหัสผ่านตรงกัน | 62 |
| รูปที่ 4.7 หน้าเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในกรณีที่รหัสผ่านและยืนยันรหัสผ่านไม่ตรงกัน | 63 |
| รูปที่ 4.8 หน้าที่ใช้ในการลบชื่อผู้ใช้ออกจากกระบวน | 63 |
| รูปที่ 4.9 หน้าที่ใช้ในการลงทะเบียนเพื่อรอรับการติดต่อระยะไกลจากเครื่องไคลเอนต์ | 65 |
| รูปที่ 4.10 หน้าที่ใช้ในการกำหนดค่าให้กับเซิร์ฟเวอร์ | 65 |
| รูปที่ 4.11 หน้าที่ใช้ในการเริ่มต้นการใช้งานเซิร์ฟเวอร์ | 66 |
| รูปที่ 4.12 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด กรณีที่ใช้เซิร์ฟเวอร์ ฐานข้อมูลเพียงเครื่องเดียว | 67 |
| รูปที่ 4.13 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด กรณีที่ใช้เซิร์ฟเวอร์ ฐานข้อมูลทั้งหมด 3 เครื่อง | 67 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

| ภาพ | หน้า |
|--|------|
| รูปที่ 4.14 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด กรณีที่ใช้เซิร์ฟเวอร์ฐานข้อมูล 6 เครื่อง | 68 |
| รูปที่ 4.15 หน้าที่ใช้สำหรับแสดงข้อมูลการทำงาน (log file) | 69 |
| รูปที่ 4.16 หน้าที่ใช้ในการทำกรายงานค่าสถานะมายังเซิร์ฟเวอร์ | 69 |
| รูปที่ 4.17 หน้าที่ใช้ในการทดสอบประสิทธิภาพของระบบ | 70 |
| รูปที่ 4.18 หน้าที่ใช้ในการแสดงเวลาที่ใช้ในการประมวลผลของแต่ละไคลเอนต์ | 71 |
| รูปที่ 5.1 หน้าที่ใช้แสดงสคริปไฟล์ที่มีการทำงานในรูปแบบคำสั่งซีเลค | 72 |
| รูปที่ 5.2 หน้าที่ใช้แสดงประสิทธิภาพการใช้งานระบบกระจายโหนดฐานข้อมูลในกรณีที่มีการทำงานในรูปแบบคำสั่งซีเลค | 73 |
| รูปที่ 5.3 หน้าที่ใช้แสดงสคริปไฟล์ที่มีการทำงานในรูปแบบคำสั่งอินเชิร์ท | 73 |
| รูปที่ 5.4 หน้าที่ใช้แสดงประสิทธิภาพการใช้งานระบบกระจายโหนดฐานข้อมูลในกรณีที่มีการทำงานในรูปแบบคำสั่งอินเชิร์ท | 74 |
| รูปที่ 5.5 หน้าที่ใช้แสดงสคริปไฟล์ที่มีการทำงานในรูปแบบคำสั่งจอย | 74 |
| รูปที่ 5.6 หน้าที่ใช้แสดงประสิทธิภาพการใช้งานระบบกระจายโหนดฐานข้อมูลในกรณีที่มีการทำงานในรูปแบบคำสั่งจอย | 75 |

สารบัญตาราง

ตาราง
ตารางที่ 2.1 การเปรียบเทียบความแตกต่างของไคร์ฟเวอร์ทั้ง 4 ชนิด

หน้า
11



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 แนวคิดและที่มาของปัญหา

เนื่องจากปัจจุบันมีการประยุกต์ใช้ระบบฐานข้อมูลในกิจกรรมต่าง ๆ เป็นจำนวนมากไม่ว่าจะเป็นธุรกิจ อุตสาหกรรม สถานศึกษา เป็นต้น ทำให้ระบบฐานข้อมูลจะต้องรองรับการใช้งานในหลายรูปแบบ ไม่ว่าจะเป็นการใช้งานจากผู้ใช้คนเดียว การใช้งานจากผู้ใช้หลายคนผ่านเครือข่าย การใช้งานข้อมูลเก็บข้อมูลทั่วไป หรือการใช้งานข้อมูลเก็บข้อมูลทางอุตสาหกรรม ซึ่งทำให้บางครั้งระบบฐานข้อมูลที่มีอยู่ในท้องตลาดปัจจุบันไม่สามารถจัดการกับปัญหาที่เกิดขึ้นจากความต้องการใช้งานที่ผู้ใช้ต้องการ ได้อย่างเพียงพอ หรือผู้ใช้ อาจจะต้องเสียเงินจำนวนมากเพื่อซื้อบริการที่ต้องการเพิ่มขึ้นเป็นพิเศษ ซึ่งจากการศึกษาได้พบว่า การใช้งานระบบฐานข้อมูลในปัจจุบันได้มีแนวโน้มที่จะใช้ในอุตสาหกรรมขนาดใหญ่เป็นจำนวนมาก และมีแนวโน้มการใช้งานที่ต้องรองรับผู้ใช้ได้หลายคนในเวลาเดียวกัน ซึ่งเป็นคุณสมบัติพื้นฐานของระบบฐานข้อมูลในท้องตลาดปัจจุบัน แต่เมื่อเข้าสู่ภาวะไม่ปกติคือ มีเรียกใช้งานเซิร์ฟเวอร์ฐานข้อมูลเครื่องหนึ่ง ๆ จากผู้ใช้จำนวนมากในเวลาเดียวกัน หรือการใช้งานฐานข้อมูลอย่างหนักเป็นเวลานาน จะส่งผลให้เซิร์ฟเวอร์ฐานข้อมูลต้องทำการประมวลผลอย่างหนัก ซึ่งจะทำให้ประสิทธิภาพในการประมวลผลลดลงซึ่งทำให้เกิดความล่าช้าในการทำงาน ซึ่งหลักการฐานข้อมูลแบบกระจาย (Distributed database) ก็เป็นวิธีการหนึ่งที่จะช่วยเพิ่มประสิทธิภาพในการประมวลผลได้ โดยจะแบ่งคำสั่งที่ใช้ติดต่อกับฐานข้อมูลออกเป็นส่วน ๆ และกระจายไปยังเซิร์ฟเวอร์ฐานข้อมูลเพื่อช่วยกันทำการประมวลผล แต่ในทางปฏิบัติฐานข้อมูลแบบกระจายสามารถพัฒนาขึ้นมาได้ยาก เราจึงได้ทำการศึกษาและพัฒนากระบวนการกระจายโหลดฐานข้อมูล (Database server load balancing system) ขึ้นมา เพื่อช่วยเพิ่มประสิทธิภาพในการทำงานให้ดีขึ้น ซึ่งในทางปฏิบัติ การพัฒนาระบบนี้ขึ้นมาใช้งานจะมีความเป็นไปได้มากกว่า

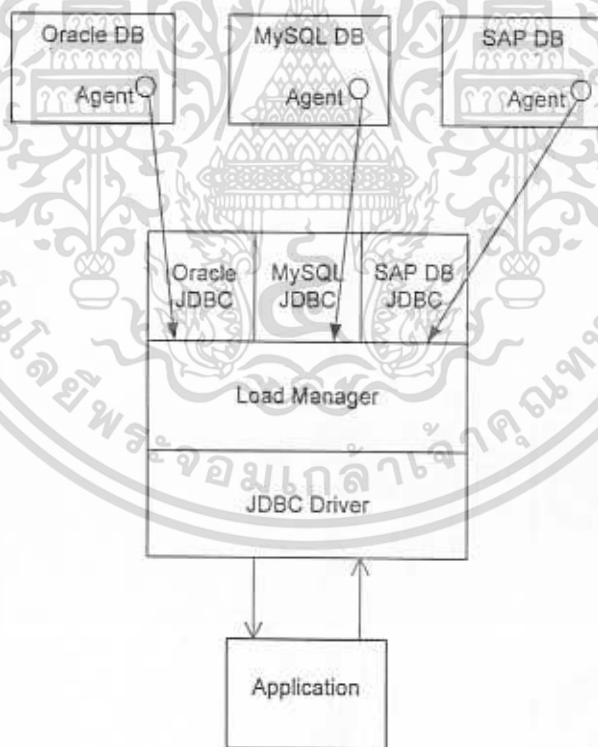
ขั้นตอนการทำงานของระบบกระจายโหลดฐานข้อมูล ก็คือขั้นแรกเราจะต้องทำการเก็บข้อมูลชุดเดียวกันนี้ไว้ที่เซิร์ฟเวอร์ฐานข้อมูลทุก ๆ ตัว และมีเครื่องค้ำเบสเซิร์ฟเวอร์โหลดบาลานซ์ซึ่ง (Database server load balancing) ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่าง โปรแกรมประยุกต์และเซิร์ฟเวอร์ฐานข้อมูลทุก ๆ ตัว โดยที่ผู้ใช้ทุกคนที่ใช้งานผ่าน โปรแกรมประยุกต์จะรู้สึกว่าเหมือนการใช้งานผ่านระบบฐานข้อมูลปกติโดยทั่วไปโดยที่ไม่รู้เลยว่ากำลังใช้งานผ่านระบบ

กระจายโหลคฐานข้อมูลอยู่ โดยที่โปรแกรมประยุกต์จะทำการติดต่อกับฐานข้อมูล โดยผ่านเครื่อง
 ค้าแบบเซิร์ฟเวอร์โหลคบาลานซ์ซึ่ง ซึ่งประกอบด้วย

- ส่วน ไดรฟ์เวอร์เจตบีซี (JDBC Driver) ซึ่งทำหน้าที่ในการติดต่อกับโปรแกรมประยุกต์
 โดยทำหน้าที่รับคำสั่ง (SQL) ที่ส่งมาจากโปรแกรมประยุกต์ และทำหน้าที่ส่งผลลัพธ์ที่อยู่ใน
 รูปแบบของรีซัลเซต (ResultSet) กลับไปยังฝั่งโปรแกรมประยุกต์

- ส่วน โหลคเมนเจอร์ (Load Manager) ทำหน้าที่รับข้อมูลภาระงานของเครื่องเซิร์ฟเวอร์
 ฐานข้อมูลต่าง ๆ ที่ถูกส่งเข้ามาโดยเอเจนต์ (Agent) จากนั้นก็ทำการเลือกเครื่องเซิร์ฟเวอร์
 ฐานข้อมูลที่เหมาะสมที่สุดเพื่อที่จะส่งคำสั่งไปให้ประมวลผล

- ส่วน เอเจนต์ (Agent) ทำหน้าที่ตรวจสอบภาระงานของเครื่องเซิร์ฟเวอร์ฐานข้อมูล โดย
 ดูจาก เปอร์เซ็นต์การถูกใช้งานของซีพียู (CPU) และหน่วยความจำ (RAM) และแจ้งกลับไปที่
 ส่วนโหลคเมนเจอร์ เพื่อทำการตัดสินใจต่อไป



รูปที่ 1.1 สถาปัตยกรรมของระบบกระจายโหลคฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 จุดประสงค์

1. เพื่อพัฒนาระบบให้สามารถกระจาย โหลดให้แก่เซิร์ฟเวอร์ฐานข้อมูลได้อย่างมีประสิทธิภาพ
2. เพื่อศึกษาการเขียนโปรแกรมด้วยภาษาจาวา รวมถึงการใช้งานจาวาเอพีไอ (JAVA APIs) และคลาสต่างๆ ในภาษาจาวา
3. เพื่อศึกษาชนิดของเจดีบีซี (JDBC) และการติดต่อกับฐานข้อมูลโดยผ่านเจดีบีซี (JDBC)
4. เพื่อศึกษาวิธีการออกแบบฐานข้อมูล
5. เพื่อศึกษารายละเอียดโครงสร้าง และวิธีการใช้งานฐานข้อมูล Oracle , MySQL , SAP DB
6. เพื่อศึกษาและพัฒนาทักษะในการเขียนโปรแกรมประยุกต์

1.3 ขอบเขตโครงการ

1. พัฒนาไคร์ฟเวอร์เจดีบีซี ซึ่งมีหน้าที่ดังนี้
 - ติดต่อกับโปรแกรมประยุกต์ (ฟิงโกลอนต์) ในการรับคำสั่ง (SQL) ต่าง ๆ จากโปรแกรมประยุกต์ และส่งผ่านคำสั่งนั้นต่อไปให้เซิร์ฟเวอร์ฐานข้อมูล ตามการตัดสินใจของโหลดเมนเจอร์
 - ส่งผลลัพธ์ที่ได้จากการประมวลผล (ในรูปแบบของรีซัลเซต) กลับไปยังโปรแกรมประยุกต์ เพื่อทำการแสดงผลต่อไป
2. พัฒนาโหลดเมนเจอร์ เพื่อทำหน้าที่รับข้อมูลที่ได้มาจากเอเจนต์ และทำการจัดการในการตัดสินใจเกี่ยวกับการเลือกส่งคำสั่ง (query) ไปยังเครื่องเซิร์ฟเวอร์ฐานข้อมูลที่เหมาะสม โดยจะเลือกส่งคำสั่งไปที่เครื่องเซิร์ฟเวอร์ฐานข้อมูลตัวที่มีภาระงานน้อยที่สุด ซึ่งภาระงานดังกล่าวพิจารณาจากค่าเปอร์เซ็นต์ที่ถูกใช้งานของซีพียู และค่าจำนวนพื้นที่ว่างในหน่วยความจำ
3. พัฒนาเอเจนต์ ซึ่งเป็นคอมโพเนนท์ (component) ที่ฝังอยู่บนเครื่องเซิร์ฟเวอร์ฐานข้อมูลทุกตัว เพื่อทำการรายงานค่าเปอร์เซ็นต์ที่ถูกใช้งานของซีพียูและหน่วยความจำ ไปยังโหลดเมนเจอร์เพื่อทำการตัดสินใจต่อไป

4. พัฒนาส่วนที่แสดงผลที่ทำหน้าที่รายงานสถิติการถูกใช้งานของเครื่องเซิร์ฟเวอร์ฐานข้อมูล โดยทำหน้าที่รายงานค่าเปอร์เซ็นต์ที่ถูกใช้งานของซีพียูและหน่วยความจำ และเวลาที่ใช้ในการประมวลผลออกทางจอภาพเพื่อไว้สำหรับตรวจสอบความถูกต้อง และเพื่อวัดประสิทธิภาพของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ โดยจะนำค่านี้ไปเปรียบเทียบกับการใช้ฐานข้อมูลโดยไม่ผ่านระบบนี้ด้วย และเราจะทำการเปิด-ปิด ปรับค่าเริ่มต้น และซ่อมบำรุงจากส่วนนี้

5. พัฒนาส่วนโปรแกรมประยุกต์ที่ใช้ในการทดสอบระบบ

1.4 ขั้นตอนการดำเนินโครงการ

1. ศึกษาการเขียน โปรแกรมด้วยภาษาจาวา
2. ศึกษาหลักการของเทคโนโลยีเจดีบีซี
3. ศึกษาระบบฐานข้อมูล
4. ทำการออกแบบ และพัฒนาส่วนต่าง ๆ
5. ทำการติดตั้ง และทดสอบ

1.5 ผลที่คาดว่าจะได้รับ

1. ระบบที่พัฒนาขึ้นสามารถนำไปใช้ได้มีประสิทธิภาพ และสามารถนำไปประยุกต์ใช้ได้จริงในอุตสาหกรรม
2. มีความรู้ความเข้าใจในการเขียน โปรแกรมด้วยภาษาจาวา รวมถึงการใช้งานจาวาเอพีไอ และคลาสต่าง ๆ ในภาษาจาวา
3. มีความรู้ความเข้าใจในการประยุกต์การติดต่อกับฐานข้อมูลโดยผ่านเจดีบีซี
4. มีความรู้ความเข้าใจในหลักการออกแบบฐานข้อมูล และสามารถประยุกต์ใช้ได้อย่างเหมาะสม
5. มีความรู้ความเข้าใจในรายละเอียด และสามารถประยุกต์ใช้งานฐานข้อมูล Oracle , MySQL และ SAP DB ได้
6. มีความรู้และทักษะในการเขียน โปรแกรมประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่ใช้ในโครงการ

2.1 ความรู้พื้นฐานเกี่ยวกับเจดีบีซี

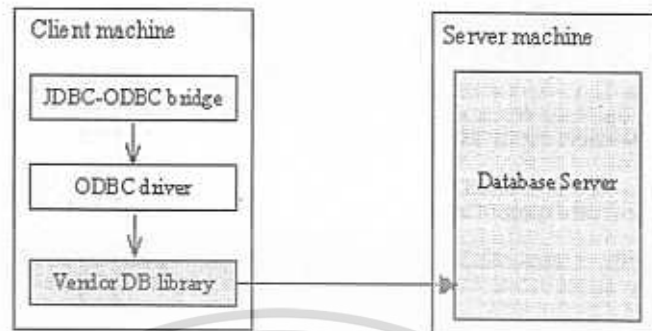
JDBC (Java Database Connectivity) คือชุดคำสั่งที่ทำให้โปรแกรมภาษาจาวาติดต่อกับฐานข้อมูลอะไรก็ได้ จุดเด่นของชุดคำสั่งเจดีบีซี ก็คือ รูปแบบของคำสั่งเป็นอย่างเดียวกันหมดไม่ว่าจะใช้ติดต่อกับฐานข้อมูลอะไร นั่นก็คือ เจดีบีซีจะไม่บังคับให้เขียนโปรแกรมตัวหนึ่งเพื่อใช้กับฐานข้อมูลของ Sybase และอีกโปรแกรมหนึ่งสำหรับฐานข้อมูลของ Oracle และอีกหนึ่งโปรแกรมสำหรับฐานข้อมูลของ Informix และอีกหลายโปรแกรมสำหรับฐานข้อมูลชนิดต่าง ๆ เราสามารถเขียนโปรแกรมโดยใช้จาวาเอพีไอเพียงโปรแกรมเดียว และโปรแกรมนั้นจะสามารถส่งคำสั่งเอสคิวแอล (SQL) ไปยังฐานข้อมูลชนิดต่าง ๆ นั้นได้ทำให้ประหยัดเวลาในการเรียนรู้ และไม่ต้องแก้ไขคำสั่งถ้ามีการเปลี่ยนฐานข้อมูลที่ใช้

นอกจากนี้การเขียนโปรแกรมโดยใช้ภาษาจาวานี้ เราก็ไม่ต้องกังวลว่าจะต้องเขียนโปรแกรมอีกหลายตัวเพื่อนำไปใช้กับคอมพิวเตอร์ที่อยู่บนแพลตฟอร์ม (platform) ชนิดต่าง ๆ เลยในการรวมตัวกันระหว่างจาวาและเจดีบีซีนี้ ทำให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมเพียงหนึ่งโปรแกรมแล้วสามารถนำไปใช้ได้ในทุก ๆ แห่งที่ต้องการ

2.2 ชนิดของไดร์ฟเวอร์เจดีบีซี

เจดีบีซีอาศัย ไดร์ฟเวอร์เจดีบีซี เป็นสะพานที่เชื่อมระหว่างชุดคำสั่งเจดีบีซี ที่อยู่ในโปรแกรมภาษาจาวา กับตัวฐานข้อมูล ไดร์ฟเวอร์เจดีบีซี ก็เหมือนกับซอร์ฟแวร์ตัวกลางที่รับคำสั่งเจดีบีซีมา แล้วส่งผ่านไปยังฐานข้อมูลโดยเปลี่ยนเป็นคำสั่งแบบที่ฐานข้อมูลนั้นเข้าใจ ซึ่งไดร์ฟเวอร์เจดีบีซี ถูกแบ่งออกเป็น 4 ประเภทด้วยกันคือ ประเภท 1, ประเภท 2, ประเภท 3 และประเภท 4 โดยเป็นข้อกำหนดที่ถูกกำหนดมาโดยบริษัทซัน ไมโครซิสเต็มส์ ซึ่งเป็นผู้กำหนดรายละเอียดของแต่ละประเภท ดังนี้

2.2.1 ประเภท 1 (Type1: JDBC-ODBC Bridge)



รูปที่ 2.1 ประเภท 1 : เจดีบีซี - โอดีบีซีบริดจ์

เจดีบีซี - โอดีบีซีบริดจ์ (JDBC-ODBC bridge) ได้ถูกพัฒนาขึ้น โดยการร่วมมือกันระหว่าง JavaSoft และ Intersolv เพื่อที่จะเพิ่มความสามารถให้กับฐานข้อมูลจำนวนมากที่ใช้โอดีบีซี (ODBC) ในส่วนของไคลเอนต์, จาวาแอปเพล็ต (Java applet) หรือโปรแกรมอื่น จะถูกเขียนขึ้นด้วยเจดีบีซีเอพีไอ (JDBC API) โดยบริดจ์ตัวนี้จะทำการแปลงคำสั่งจากเจดีบีซีไปเป็นคำสั่งของโอดีบีซี แล้วส่งคำสั่งนั้น ไปยังตัวโอดีบีซีไดรฟ์เวอร์ (ODBC driver) เพื่อจัดการกับฐานข้อมูล ซึ่งโอดีบีซีไดรฟ์เวอร์จะต้องมีอยู่บนเครื่องไคลเอนต์ ข้อได้เปรียบหลักของบริดจ์ตัวนี้คือ โปรแกรมที่เขียนขึ้นนั้นจะง่ายต่อการเข้าถึงข้อมูลจากระบบฐานข้อมูลของผู้ผลิตต่าง ๆ โดยการเลือกโอดีบีซีไดรฟ์เวอร์ที่เหมาะสม อย่างไรก็ตามข้อติดต่อกับฐานข้อมูลชนิดนี้ ทำให้ต้องนึกถึงผลข้างเคียงที่อาจเกิดขึ้นและความยุ่งยากเพราะว่าคำสั่งต้องส่งจากเจดีบีซี ไปยังบริดจ์ที่เชื่อมไปยังโอดีบีซีไดรฟ์เวอร์ และสุดท้ายจาก โอดีบีซีก็ส่งไปยังเนทีฟไคลเอนต์เอพีไอ (native client-API) เพื่อไปยังฐานข้อมูล

ข้อดี

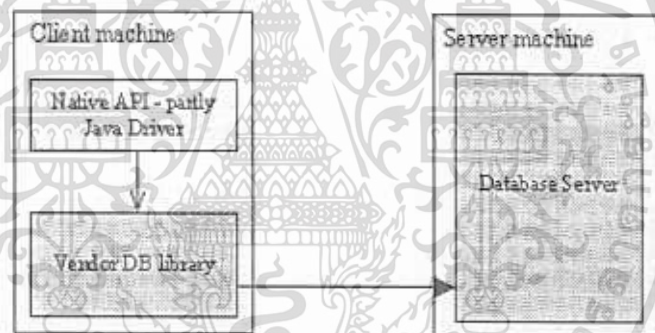
เจดีบีซี - โอดีบีซีบริดจ์ จะอนุญาตให้เข้าถึงฐานข้อมูลได้เกือบทุกชนิด เพราะว่าไดรฟ์เวอร์โอดีบีซีของฐานข้อมูลมันมีอยู่แล้ว ทำให้ไม่ต้องหาไดรฟ์เวอร์มาลงอีก ไดรฟ์เวอร์ประเภทนี้ จะเป็นประโยชน์กับองค์กรซึ่งมีไดรฟ์เวอร์โอดีบีซีติดตั้งอยู่บนเครื่องไคลเอนต์เรียบร้อยแล้ว

ข้อเสีย

ประสิทธิภาพมันจะต่ำ อันเนื่องมาจากเจดีย์จะถูกส่งผ่านไปยังบริษัทซึ่งจะส่งต่อไปยังไคร์ฟเวอร์โอดีบีซี ต่อจากนั้นก็ส่งไปให้เนทีฟดาต้าเบสคอนเนคทีวิตีอินเทอร์เฟซ (native database connectivity interface) ส่วนผลลัพธ์ที่ได้จะถูกส่งกลับมาโดยกระบวนการย้อนกลับ เมื่อพิจารณาในแง่ของประสิทธิภาพไคร์ฟเวอร์ประเภทนี้อาจจะไม่เหมาะสมกับแอปพลิเคชันที่มีขนาดใหญ่

ไคร์ฟเวอร์โอดีบีซี และเนทีฟดาต้าเบสคอนเนคทีวิตีอินเทอร์เฟซ ต้องถูกติดตั้งบนเครื่องโฮลเอนต์ด้วยเหตุนี้ ข้อดีของการใช้ภาษาแอฟเพลต ในระบบอินทราเน็ตก็จะสูญเสียไป เนื่องจากปัญหาของการติดตั้งของแอปพลิเคชันเดิมมันยังคงอยู่

2.2.2 ประเภท 2 (Type2: Native-API/partly Java driver)



รูปที่ 2.2 ประเภท 2 : เนทีฟ เอพีโอ

ไคร์ฟเวอร์เป็นแบบทูเทียร์ (two-tier) นั่นคือ ไคร์ฟเวอร์เจดีย์ต้องการไลบรารี (library) เพื่อแปลงฟังก์ชันของเจดีย์ไปเป็นควิรี่ (query language) ต่าง ๆ ของ DBMS (เช่น ไลบรารีสำหรับ Sybase คือ dlib , สำหรับ Oracle คือ ocilib และอื่น ๆ) ไคร์ฟเวอร์เหล่านี้โดยปกติจะเขียนขึ้นโดยภาษาจาวา และ C/C++ เนื่องจากไคร์ฟเวอร์ต้องใช้เลเยอร์ (layer) ของ C ในการเรียกไปยังไลบรารี สำหรับไคร์ฟเวอร์ประเภท 2 นี้จะทำการติดต่อสื่อสารโดยตรงกับตัวฐานข้อมูล ดังนั้นมันเลยต้องการไบนารีโค้ด (binary code) บางส่วน (ซึ่งก็คือไลบรารีของแต่ละผู้ผลิต) อยู่บนเครื่องโฮลเอนต์

ข้อดี

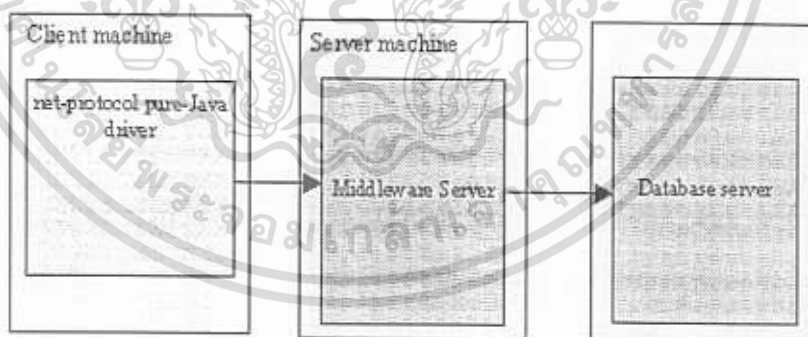
ไดร์ฟเวอร์ประเภทนี้จะให้ประสิทธิภาพที่ดีกว่าประเภท 1 เพราะเลขอร์พิเศษของการแปลงเป็น โอดีบีซีจะถูกเอาออกไป

ข้อเสีย

เวนเคอร์ค้ำเบสไลบรารี (vendor database library) จำเป็นจะต้องถูกโหลดไปไว้บนแต่ละเครื่องของไคลเอนต์ เพราะฉะนั้นไดร์ฟเวอร์ประเภท 2 นี้จะไม่สามารถใช้งานบนอินเทอร์เน็ตได้ ไดร์ฟเวอร์ประเภท 2 นี้จะมีประสิทธิภาพต่ำกว่าไดร์ฟเวอร์ประเภท 3 และประเภท 4

2.2.3 ประเภท 3 (Type 3: Net-protocol/all-Java driver)

ไดร์ฟเวอร์ประเภท 3 มันจะทำงานในรูปแบบที่เรียกว่าทรีเทียร์ (three-tiered) โดยมีขั้นตอนการทำงานคือ โอดีบีซีจะถูกส่งผ่านไปบนเครือข่าย เพื่อส่งไปยังเครื่องตัวกลาง (middle-tier server) ซึ่งเครื่องตัวกลางจะทำการแปลงไปเป็น database-specific native-connectivity interface ซึ่งมันก็จะทำการส่งต่อไปยังเซิร์ฟเวอร์ฐานข้อมูลอีกต่อหนึ่ง เครื่องตัวกลางอาจใช้ไดร์ฟเวอร์ ประเภทที่ 1, 2 หรือ ประเภทที่ 4 ก็ได้ถ้าเขียนโดยจาวา



รูปที่ 2.3 ประเภท 3 : เน็ตเวิร์ค โปร โดคอล

ไดร์ฟเวอร์ชนิดนี้เหมาะสำหรับแอปพลิเคชันที่มีหลายผู้ใช้งานอินเทอร์เน็ต/ อินทราเน็ตมากที่สุด ที่ซึ่งการกระทำของข้อมูลต่อเนื่องจำนวนมาก เช่น คิวรี (queries) , เซิร์ช (searches) และอื่น ๆ ถูกคาดหวังประสิทธิภาพเป็นสิ่งสำคัญเซิร์ฟเวอร์ตัวกลางสามารถจัดการฐานข้อมูลจำนวนมากรวมกันได้, สามารถให้การตรวจสอบและดูแลข้อมูล, สามารถทำโหลดบาลานซ์ (loadbalancing) และสนับสนุนแคตตาล็อก (catalog) และคิวรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แคช (query caches) และอย่างที่ได้อธิบายไปแล้วแอปพลิเคชันฐานข้อมูลบนเว็บ แบบทรีเทียร์เกี่ยวข้องกับความปลอดภัย , ไฟร์วอลล์ (firewalls) และพร็อกซี (proxies) ซึ่งไดร์ฟเวอร์ประเภทที่ 3 สนับสนุนสิ่งเหล่านี้

ข้อดี

ไดร์ฟเวอร์ชนิดนี้มันจะอยู่ที่ฝั่งเซิร์ฟเวอร์ตัวกลาง ดังนั้นจึงไม่มีความจำเป็นที่จะต้องมีความเร็วของเครือข่ายบนเครื่องของไคลเอนต์ ดังนั้นจึงมีความสามารถที่คิดหลายอย่าง เช่น สามารถใช้ได้อย่างสะดวกและเข้ากับระบบอื่นได้ง่าย ประสิทธิภาพดี สามารถขยายปรับปรุงได้ นอกจากนี้ยังสามารถที่จะถูกออกแบบไดร์ฟเวอร์เจดีบีซีให้มีขนาดเล็กมาก ซึ่งทำให้สามารถโหลดได้เร็ว นอกจากนี้ยังสนับสนุนคุณสมบัติต่าง ๆ เช่น แลชชิ่ง (การเชื่อมต่อ, การคิวรี) , การกระจายโหลด และการจัดการระบบขั้นสูงอย่างเช่น การลงบันทึก และการการตรวจสอบ

ข้อเสีย

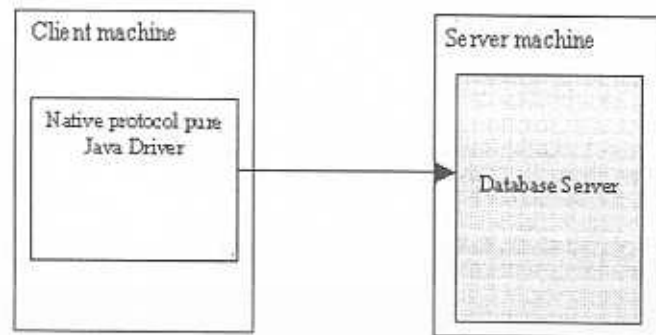
ไดร์ฟเวอร์ชนิดนี้ต้องการโค้ดของฐานข้อมูลเฉพาะไว้ที่เซิร์ฟเวอร์ตัวกลาง (ยังต้องการไดร์ฟเวอร์สำหรับแต่ละผลิตภัณฑ์เพื่อติดตั้งไว้ที่เซิร์ฟเวอร์ตัวกลาง) นอกจากนี้การเข้าไปตรวจสอบและใช้งานเรคคอร์ดเซต (recordset) ใช้เวลานานกว่า เพราะข้อมูลจะมาจากส่วนของเซิร์ฟเวอร์ที่อยู่หลังสุด (เซิร์ฟเวอร์ฐานข้อมูล)

2.2.4 ประเภท 4 (Type 4: Native-protocol/all-Java driver)

ไดร์ฟเวอร์จะทำการแปลงเจดีบีซี ไปเป็นโปรโตคอลเครือข่ายโดยตรงโดยใช้ไดร์ฟเวอร์ซึ่งเขียนขึ้นมาโดยเฉพาะเพื่อว่าแอปพลิเคชัน จะสามารถติดต่อสื่อสารกับเซิร์ฟเวอร์ฐานข้อมูลได้โดยตรง ไดร์ฟเวอร์เหล่านี้จะถูกสร้างขึ้นมาจากภาษาจาวาล้วนๆ เพื่อที่จะไม่ต้องขึ้นอยู่กับแพลตฟอร์ม และกำจัดปัญหาในเรื่องของการติดตั้ง

การส่งข้อมูลแบบจิสต์ไทม์ (just-in-time) ของแอปพลิเคชัน (เหมือนประเภทที่ 3) เนื่องจาก ไดร์ฟเวอร์เหล่านี้แปลงเจดีบีซี ไปเป็นโปรโตคอลดั้งเดิมโดยตรง โดยไม่มีการใช้โอทีบีซี หรือ เอทีไอคั้งเดิมจึงสามารถให้การเข้าถึงฐานข้อมูลที่มีประสิทธิภาพสูง ไดร์ฟเวอร์เหล่านี้ทำขึ้นจากผู้ผลิตดีบีเอ็มเอส (DBMS) เท่านั้น จากความจริงที่ว่าความรู้ในเรื่องโปรโตคอลเป็นของผู้ผลิต ในปัจจุบันยังมีไดร์ฟเวอร์ประเภท 4 ใช้อยู่ น้อย แต่จำนวนน่าจะมากขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ประเภท 4 : เนทีฟโปรโตคอล

ข้อดี

เนื่องจากไดรฟ์เวอร์เจดีย์ซี ประเภท 4 ไม่ต้องทำการแปลงคำสั่งที่ติดต่อกับฐานข้อมูล แล้วส่งไปให้โอดีบีซี หรือเนทีฟคอนเนกทิวตี้อินเทอร์เฟซ (native connectivity interface) หรือว่าส่งผ่านไปให้เซิร์ฟเวอร์ตัวอื่น ดังนั้นประสิทธิภาพของมันจึงค่อนข้างดี นอกจากนั้นแล้วยังถือว่าไดรฟ์เวอร์ประเภทนี้มีประสิทธิภาพดีกว่าประเภท 1 และ 2 อีกด้วย – และยังไม่มีความจำเป็นที่จะต้องติดตั้งซอร์ฟแวร์เฉพาะ บนเครื่องไคลเอนต์ หรือเครื่องเซิร์ฟเวอร์อีกด้วย เพราะ ไดรฟ์เวอร์เหล่านี้มันจะถูกดาวน์โหลดแบบไดนามิก

ข้อเสีย

ผู้ใช้งานจำเป็นต้องมีไดรฟ์เวอร์ที่แตกต่างกันสำหรับฐานข้อมูลแต่ละชนิด

ตารางที่ 2.1 การเปรียบเทียบความแตกต่างของไดรฟ์เวอร์ทั้ง 4 ชนิด

| ชนิดของ JDBC | ข้อดี | ข้อเสีย |
|--------------------------------------|--|--|
| JDBC- ODBC Bridge driver | <ul style="list-style-type: none"> - สามารถใช้ในการเข้าถึงฐานข้อมูลได้หลายรูปแบบเนื่องจากไดรฟ์เวอร์หาได้ง่ายโดยเลือกใช้ไดรฟ์เวอร์ที่เหมาะสมกับฐานข้อมูลนั้นๆ - เหมาะกับระบบงานที่มีไดรฟ์เวอร์ในการใช้งานอยู่แล้ว | <ul style="list-style-type: none"> - ชนิดที่ 1 นี้ไม่สามารถรองรับการทำงานที่ส่งข้อมูลแบบจัสท์อินไทม์ได้ เนื่องจากเนทีฟโค้ด (native code) ที่ใช้จะต้องมีการติดตั้งบนเครื่องไคลเอนต์ก่อนแล้วจึงจะสามารถใช้เจดีบีซี - ไดรฟ์เวอร์บริดจ์ ในการเรียกใช้เอพีไอได้ ในการที่จะต้องทำการติดตั้งโปรแกรมที่ส่วนไคลเอนต์ก่อนนี้ ก็สามารถเปรียบเทียบได้กับระบบ ไคลเอนต์ - เซิร์ฟเวอร์แบบเก่า ๆ ที่ไม่สามารถแก้ไขปัญหากการติดตั้งโปรแกรมในส่วนของ ไคลเอนต์ได้ - มีผลข้างเคียงที่อาจเกิดขึ้นและยุ่งยาก เพราะคำสั่งต้องส่งจากเจดีบีซีไปยังบริดจ์ที่เชื่อมไปยังไดรฟ์เวอร์ และสุดท้ายจากไดรฟ์เวอร์ก็ส่งไปยังเนทีฟไคลเอนต์เอพีไอเพื่อไปยัง ฐานข้อมูล ทำให้การส่งข้อมูลทำได้ช้าหรือมีความเร็วในการส่งข้อมูลต่ำนั่นเอง |
| Native-API, partly Java driver | <ul style="list-style-type: none"> - มีความเร็วในการใช้งานมากกว่า ชนิดที่ 1 เพราะ ส่วนที่ใช้ในการเปลี่ยนเจดีบีซีไปเป็น ไดรฟ์เวอร์นั้น ได้ถูกนำออกไปทำให้การทำงานมีความเร็วมากขึ้น | <ul style="list-style-type: none"> - ต้องการโค้ด ติดตั้งที่ส่วนไคลเอนต์ก่อน ดังนั้น ทั้งสอง ชนิดจึงมีปัญหาเกี่ยวกับการบำรุงรักษาซอร์ฟแวร์อยู่เสมอๆ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ชนิดของ JDBC | ข้อดี | ข้อเสีย |
|---|--|--|
| net-protocol, all-Java driver | <ul style="list-style-type: none"> - สามารถรองรับองค์กรที่มีเซิร์ฟเวอร์ทางด้านฐานข้อมูลที่หลากหลายรูปแบบ - มีความเหมาะสมกับระบบเครือข่ายอินเทอร์เน็ต/อินทราเน็ต, แอปพลิเคชันที่มีการใช้งานจากผู้ใช้งานจำนวนมาก ซึ่งมีการทำโอเปอเรชัน (operation) หลาย ๆ อย่างพร้อม ๆ กันเช่นการควรี, การค้นหาข้อมูล และอื่น ๆ ซึ่งจะแบ่งงานกัน ได้อย่างมีประสิทธิภาพ - การทำงานของไคลเอนต์ ไม่จำเป็นต้องติดตั้งเซิร์ฟเวอร์ไว้ในทุกตัว | <ul style="list-style-type: none"> - ยังต้องการไครฟ์เวอร์ สำหรับแต่ละผลิตภัณฑ์ เพื่อติดตั้งไว้ที่เครื่องตัวกลาง - ในกรณีที่เครื่องตัวกลาง เป็นแบบที่ใช้ภาษาซี หรือซีพลัสพลัส เขียนขึ้นถ้าเกิดมีการเปลี่ยนชนิดของฐานข้อมูลจำเป็นที่จะต้องเปลี่ยนโค้ดภายในเครื่องตัวกลาง |
| Native- Protocol, All Java driver | <ul style="list-style-type: none"> - สามารถทำการส่งข้อมูลแบบจัสท์อินไทม์ได้ - มีประสิทธิภาพ ในการเข้าถึงฐานข้อมูลสูงมากเพราะเข้าถึงได้โดยตรง | <ul style="list-style-type: none"> - เขียนยากเพราะต้องเรียนรู้การทำงานของเซิร์ฟเวอร์ฐานข้อมูลแต่ละตัวก่อน - ผู้ใช้งานมีความจำเป็น ต้องมีไครฟ์เวอร์ที่แตกต่างกันสำหรับฐานข้อมูลแต่ละชนิด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วิธีการสร้างไคร์ฟเวอร์เจดีบีซีประเภท 3

เราจะสร้างไคร์ฟเวอร์เจดีบีซีประเภท 3 ขึ้นมาได้อย่างไร หรือจะทำการแปลงไคร์ฟเวอร์เจดีบีซีที่มีอยู่ให้เป็นประเภท 3 ได้อย่างไร ไคร์ฟเวอร์ในรูปแบบประเภท 3 นี้จะเป็นประโยชน์กับแอปพลิเคชันที่ใช้งานบน อินเทอร์เน็ต หรืออินทราเน็ต ที่ไม่ต้องการการเซตค่าทางฝั่งไคลเอนต์ ซึ่งจะช่วยให้การจัดการกับระบบมีความยืดหยุ่นมากขึ้น ในส่วนนี้เราจะอธิบายถึงโครงสร้างของไคร์ฟเวอร์เจดีบีซี โครงร่างโมเดลของไคร์ฟเวอร์ประเภท 3 และแสดงถึงวิธีการที่จะสร้างไคร์ฟเวอร์เจดีบีซีประเภท 3 ขึ้นมา

ในการที่จะเข้าถึงระบบการจัดการฐานข้อมูลหรือคิมเอ็มเอสด้วยภาษาจาวา จำเป็นจะต้องมีไคร์ฟเวอร์เจดีบีซี ซึ่งเราอาจจะเขียนไคร์ฟเวอร์ (ซึ่งมีด้วยกัน 4 ประเภท) ด้วยวิธีการแบบใช้ภาษาจาวาทั้งหมด หรืออาจจะเขียนด้วยจาวา ร่วมกับจาวาเนทีฟอินเทอร์เฟซ (JNI) ก็ได้ แนวโน้มทางธุรกิจมันจะมุ่งเน้นไปที่ ไคร์ฟเวอร์ประเภท 3 และ 4 แบบเพียวจาวา ซึ่งมีประสิทธิภาพมากกว่า ดังนั้นการพัฒนาประสิทธิภาพให้มีประสิทธิภาพที่สูงสำหรับแอปพลิเคชันที่ใช้งานบนอินเทอร์เน็ต เราจะพบอยู่บ่อยครั้งที่เราจะทำการแปลงไคร์ฟเวอร์ประเภท 1 หรือ 2 ไปเป็นประเภท 3

2.3.1 สถาปัตยกรรมของไคร์ฟเวอร์เจดีบีซี

เจดีบีซีเป็นอินเทอร์เฟซ สำหรับติดต่อกับฐานข้อมูลในลักษณะที่เป็นมาตรฐานการใช้เจดีบีซี เอพีไอ กับคิมเอ็มเอส แต่ละชนิด คุณจำเป็นต้องใช้ไคร์ฟเวอร์เจดีบีซี เพื่อเป็นตัวกลางในการติดต่อระหว่าง เทคโนโลยีเจดีบีซีกับฐานข้อมูล โดยเจดีบีซี แบ่งเป็น 4 ประเภท หรือ 4 ระดับ แต่ละชนิดจะเป็นตัวกำหนดวิธีการสร้างไคร์ฟเวอร์เจดีบีซี ขึ้นมา

- ประเภท 1: เจดีบีซี - โอดีบีซีบริดจ์ (JDBC-ODBC Bridge)
- ประเภท 2: เนทีฟ เอพีไอ (Native-API, partly Java driver)
- ประเภท 3: เน็ตเวิร์ค โปรโตคอล (Network-protocol, all-Java driver)
- ประเภท 4: เนทีฟ โปรโตคอล (Native-protocol, all-Java driver)

ไคร์ฟเวอร์เจดีบีซี ทุกชนิดจะต้องสร้างคลาสเจดีบีซี ที่สำคัญขึ้นมา 4 คลาสด้วยกัน คือคลาส Driver, Connection, Statement และ ResultSet โปรแกรมประยุกต์จะเชื่อมต่อกับฐานข้อมูลที่ต้องการผ่านคลาส DriverManager ที่อยู่ในแพ็คเกจ java.sql. คลาส Driver จะถูกโหลดเมื่อมีการเรียกคำสั่ง

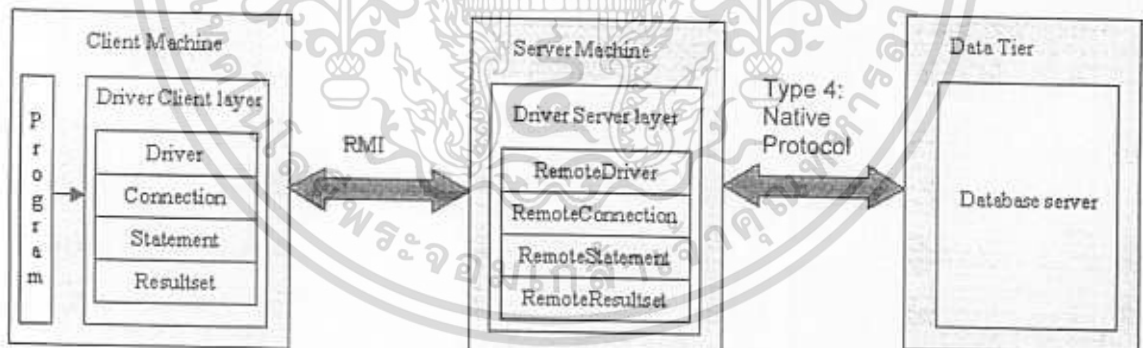
```
Class.forName("ชื่อคลาส");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มของโค้ดที่เป็นสแตติก (static) จะถูกรันในระหว่างที่คลาสของเจดีบีซี ไคร์ฟเวอร์กำลังโหลด ซึ่งจะทำภารกิจสแตติกไคร์ฟเวอร์โดย DriverManager จากนั้น โปรแกรมในฝั่งไคลเอนต์จะเชื่อมต่อไปที่ฐานข้อมูลด้วยเมธอด DriverManager.getConnection() ซึ่ง DriverManager จะเรียกไปที่เมธอด Driver.connect ()

ไคร์ฟเวอร์เจดีบีซี ทุกชนิดจะต้องอิมพลีเมนต์อินเทอร์เฟซ java.sql.Driver ขึ้นมา เมธอด connect () จะติดต่อไปยังรีโมตไคร์ฟเวอร์เพื่อให้ทำการสร้างการเชื่อมต่อกับฐานข้อมูลให้ โดยจะรีเทิร์นคอนเนคชัน (Connection) กลับมา

ในขั้นตอนนี้จะอธิบายการทำงานภายในของเจดีบีซีไคร์ฟเวอร์ประเภท 3 หรือที่เรียกว่า เน็ตเวิร์ค โปรโตคอล / ออลจาวาไคร์ฟเวอร์ (network-protocol/all-Java driver) ซึ่งทำงานแบบทรีเทียร์ โดยที่คำสั่งที่ใช้ติดต่อกับฐานข้อมูลในรูปแบบของเจดีบีซี (“คำสั่งที่ใช้ร้องขอติดต่อกับฐานข้อมูล” ในที่นี้ ขอเรียกว่า “รีเควส” (“request”)) จะถูกส่งผ่านเน็ตเวิร์ค ไปยังเซิร์ฟเวอร์ตัวกลาง และจะทำการส่งคอรี้เควส ไปยังเซิร์ฟเวอร์ฐานข้อมูลต่อไป ในที่นี้เซิร์ฟเวอร์ตัวกลางเขียนด้วยจาวา และเข้าถึงเซิร์ฟเวอร์ฐานข้อมูลด้วยไคร์ฟเวอร์ประเภท 4 (Native-protocol, all-Java driver)



รูปที่ 2.5 สถาปัตยกรรมไคร์ฟเวอร์เจดีบีซีประเภท 3

สำหรับแอปพลิเคชัน ไฟล์ของไคลเอนต์เทียร์ (client-tier) จะอยู่ในเซิร์ฟเวอร์ตัวกลาง และจะดาวน์โหลดมาพร้อมกับแอปพลิเคชัน ในไคร์ฟเวอร์เจดีบีซีจะมีอาร์เอ็มไอ (RMI) ทำหน้าที่เป็นเน็ตโปรโตคอล (net protocol) เพื่อเชื่อมการติดต่อสื่อสารระหว่างไคลเอนต์เทียร์ และเซิร์ฟเวอร์เทียร์ของไคร์ฟเวอร์ ส่วนไคร์ฟเวอร์เน็ตโปรโตคอลเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้โดยไม่ได้รับอนุญาตไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Native-protocol) จะช่วยเซิร์ฟเวอร์ตัวกลางในการแปลงเจดีบีซีรีเคส แล้วส่งไปยังเซิร์ฟเวอร์ฐานข้อมูล

ไคลเอนต์เทียร์ จะมีอินเทอร์เฟซเจดีบีซี ที่เป็นมาตรฐานเพื่อติดต่อกับโปรแกรมของไคลเอนต์ ซึ่งจะประกอบด้วยคลาส Driver ที่อิมพลิเมนต์มาจากอินเทอร์เฟซ java.sql.driver และยังประกอบไปด้วย การอิมพลิเมนต์ของ Connection อินเทอร์เฟซ, Statement อินเทอร์เฟซ และ Resultset อินเทอร์เฟซของเจดีบีซี อีกด้วย

โปรแกรม อย่างเช่น โปรแกรมแอปพลิเคชัน ที่ถูกพัฒนาขึ้นมาโดยใช้คลาสไคลเอนต์เทียร์ของไคร์ฟเวอร์ประเภท 3 เนื่องจาก คลาส Driver ของไคลเอนต์นั้นอิมพลิเมนต์มาจากอินเทอร์เฟซของเจดีบีซี ดังนั้น โปรแกรมของไคลเอนต์เลยได้รับฟังก์ชันเจดีบีซีที่เป็นมาตรฐานมาด้วย โดยคลาส Driver ในฝั่งไคลเอนต์จะมีการอ้างอิงถึงรีโมตอินเทอร์เฟซที่สัมพันธ์กัน ซึ่งจะอยู่ที่ฝั่งมิดเดิลเทียร์หรือเครื่องตัวกลาง ซึ่งรีโมตอินเทอร์เฟซก็จะมีเมธอดพื้นฐานที่ให้คลาสในฝั่งไคลเอนต์ ใช้ในการจัดการกับเจดีบีซีรีเคสของโปรแกรม โดยโปรแกรมแอปพลิเคชันจะเรียกเมธอดเจดีบีซี ผ่านออบเจกต์ในฝั่งไคลเอนต์ ที่อิมพลิเมนต์มาจากคลาส Driver ซึ่งการเรียกนั้นจะถูกส่งต่อไปยังฝั่งมิดเดิลเทียร์ โดยใช้เมธอดของรีโมตอินเทอร์เฟซ ผ่านทางอาร์เอ็มไอ ดังนั้นคลาส Driver ในไคลเอนต์จะจัดการการติดต่อสื่อสารภายในอาร์เอ็มไอกับมิดเดิลเทียร์

เซิร์ฟเวอร์เทียร์ หรือเซิร์ฟเวอร์อาร์เอ็มไอ จะใช้เน็ตเวิร์กโพรโตคอล ออลจาวาไคร์ฟเวอร์ เพื่อที่จะติดต่อไปยังฐานข้อมูล ซึ่งเซิร์ฟเวอร์เทียร์จะประกอบไปด้วย 4 รีโมตอินเทอร์เฟซ และอิมพลิเมนต์ของอินเทอร์เฟซนั้น ซึ่งรีโมตอินเทอร์เฟซจะประกอบด้วยอินเทอร์เฟซ Driver, Connection, Statement และ ResultSet ตามลำดับ ภายในคลาสที่อิมพลิเมนต์ รีโมตอินเทอร์เฟซก็จะมีการเรียกใช้ออบเจกต์ Connection, Statement, และ ResultSet ของไคร์ฟเวอร์ประเภท 4 และเมื่อคลาสของไคลเอนต์เทียร์ ส่งผ่านรีเคสมายังรีโมตอินเทอร์เฟซ อิมพลิเมนต์ของรีโมตอินเทอร์เฟซจะใช้ออบเจกต์เจดีบีซี ที่มีอยู่ทำการติดต่อกับฐานข้อมูล

ตอนนี้เราได้พอรู้จักกับสถาปัตยกรรมของเจดีบีซีไคร์ฟเวอร์ ประเภท 3 ไปบ้างแล้ว ต่อไปเราจะมาศึกษาคลาสต่าง ๆ ของไคลเอนต์เทียร์ และคลาสต่าง ๆ ของเซิร์ฟเวอร์เทียร์ ของเจดีบีซี

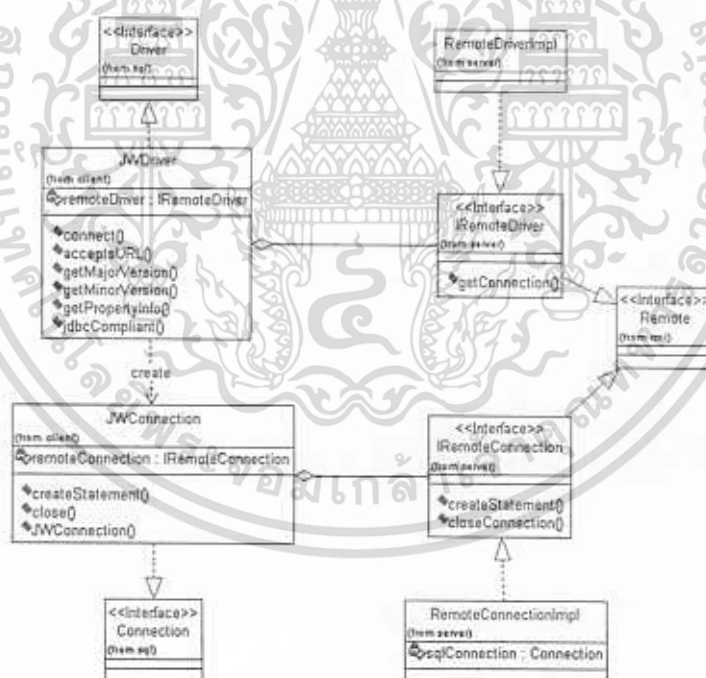
2.3.2 คลาสไดอะแกรมของเจดีบีซีไดรฟ์เวอร์ (JDBC driver class diagrams)

การสร้างเจดีบีซีไดรฟ์เวอร์ประเภท 3 จะต้องสร้างทั้งในส่วนของมิดเดิล-tier และไคลเอนต์-tier ของไดรฟ์เวอร์ ซึ่งไดรฟ์เวอร์ของไคลเอนต์-tier จะอยู่ในแพ็คเกจ com.jw.client ขณะที่ไดรฟ์เวอร์ของมิดเดิล-tier จะต้องอยู่ในแพ็คเกจ com.jw.server

2.3.2.1 คลาสต่าง ๆ ของไคลเอนต์-tier (Client-tier classes)

แพ็คเกจของไคลเอนต์-tier หรือ com.jw.client ก็จะมีคลาสพวกนี้คือ

- com.jw.client.JWDriver
- com.jw.client.JWConnection
- com.jw.client.JWStatement
- com.jw.client.JWResultSet



รูปที่ 2.6 คลาสไดอะแกรมที่แสดงความสัมพันธ์ระหว่างคลาส Driver และคลาส Connection บน ไคลเอนต์-tier และมิดเดิล-tier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปจะเป็นการอธิบายแต่ละคลาสโดยละเอียด

2.3.2.1.1 JWDriver class

คลาส `com.jw.client.JWDriver` มันจะอิมพลิเมนต์อินเทอร์เฟซ `java.sql.Driver` ซึ่งประกอบด้วยเมธอดที่ทำหน้าที่ลงทะเบียนตัวมันเองไว้กับ `DriverManager` และสร้างการเชื่อมต่อกับฐานข้อมูล คลาสนี้จะทำตัวเหมือนกับท่อหุ้มอยู่บนคลาส `Driver` ตัวที่เป็นรีโมต เพื่อทำหน้าที่เป็นอินเทอร์เฟซและ คลาส `JWDriver` จะถูกโหลดเมื่อโปรแกรมมีการเรียก :

```
Class.forName("com.jw.client.JWDriver");
```

เมื่อเมธอด `forName ()` ถูกเรียกจะทำให้มีการกระตุ้นส่วนที่เป็นสแตติก (`static clause`) ใน `JWDriver` ซึ่งมันจะทำการลงทะเบียนกับ `DriverManager` และนี่ ก็คือตัวอย่างของส่วนที่เป็นสแตติก :

```
static
{
    try
    {
        // Register the JWDriver with DriverManager
        JWDriver driverInst = new JWDriver();
        DriverManager.registerDriver(driverInst);
    }
    ...
}
```

คลาส `JWDriver` ยังเป็นตัวอ้างถึงรีโมตไดร์ฟเวอร์ (`com.jw.server.IremoteDriver`) ที่อยู่บนมิดเดิลเทียร์เซิร์ฟเวอร์ ซึ่งตัวอ้างถึงรีโมตไดร์ฟเวอร์นี้จะทำการสร้างการเชื่อมต่อกับฐานข้อมูลให้กับคลาส `JWDriver` ของไคลเอนต์ ดังนั้นเมื่อต้องการติดต่อกับฐานข้อมูล มันก็จะมีการเรียกใช้เมธอด `DriverManager.getConnection ()` ซึ่งจะทำ

ให้เมธอด JWDriver.connect ถูกเรียกใช้ ซึ่งมันก็จะใช้ตัวอ้างอิงของรีโมต ไคล์ฟเวอร์ในการเชื่อมต่อกับฐานข้อมูล

อธิบายเมธอด JWDriver.connect () :

- ถ้ายังไม่มีตัวอ้างอิงของรีโมต ไคล์ฟเวอร์ มันก็จะสร้าง ตัวอ้างอิงขึ้นเองโดยใช้ เมธอด Naming.lookup () และเมธอด JWDriver.connect () จะมีตัวอ้างอิงของรีโมต ไคล์ฟเวอร์เพื่อที่จะสร้างการเชื่อมต่อกับฐานข้อมูล

รายละเอียด :

```
if(remoteDriver == null)
{
    remoteDriver=
    RemoteDriver)Naming.lookup("rmi://" +serverName
    +":1099"+" /RemoteDriver");
}
```

- สร้างการเชื่อมต่อกับฐานข้อมูล โดยการใช้รีโมต ไคล์ฟเวอร์ที่สร้างขึ้นข้างล่างนี้ และส่งกลับไปยังโปรแกรมที่เรียกใช้ ฟังก์ชันที่ชื่อว่า remoteDriver.getConnection () เป็นการเริ่มต้นเรียกใช้ฝั่งมิดเดิลเทียร์ ของไคล์ฟเวอร์เจดีบีซี รีโมต Connection ที่ได้รับกลับมาจากมิดเดิลเทียร์จะถูกเก็บเป็นตัวอ้างอิงในออบเจกต์ของคลาส JWConnection จากนั้นออบเจกต์ของคลาส JWConnection จะส่งกลับไปยังโปรแกรมที่เรียก :

```
{
    IRemoteConnection remoteConInstance =
    (IRemoteConnection)remoteDriver.getConnection();
    localConInstance = new
    JWConnection(remoteConInstance);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
...
return (Connection)localConInstance; }
```

2.3.2.1.2 JWConnection class

คลาส `com.jw.client.JWConnection` นี้มันจะทำการอิมพลิเมนต์คอนเน็คชั่นอินเทอร์เน็ตขึ้นมา และเป็นตัวอ้างอิงไปยังอินเทอร์เน็ต `IRemoteConnection` ของรีโมตเซิร์ฟเวอร์ด้วย

เมธอด `JWDriver.connect()` จะส่งค่าออบเจกต์ `JWConnection` กลับไปยังโปรแกรมไคลเอนต์ที่เรียก นั่นก็คือการสร้างออบเจกต์ `JWConnection` นั้นเอง จากนั้นไคลเอนต์จะเรียกใช้เมธอดต่าง ๆ ของอินเทอร์เน็ต `Connection` โดยการเรียกใช้ผ่านออบเจกต์ `JWConnection` ที่ถูกรีเทิร์นค่ากลับมา ออบเจกต์ `JWConnection` จะเป็นตัวแทนในการเรียกใช้ `Connection` ในรีโมตเซิร์ฟเวอร์ให้ทำงานต่อไป ตัวอย่างเช่น เมื่อไคลเอนต์เรียก `conn.createStatement()` (`conn` คือ ออบเจกต์ของ `JWConnection`) มันก็จะไปเรียกใช้ `RemoteConnection.createStatement()` (โดยอัตโนมัติ) ซึ่งมันจะรีเทิร์นออบเจกต์ของรีโมต `Statement` กลับมา ต่อมาเมธอด `createStatement()` ของ `JWConnection` จะทำการสร้างออบเจกต์ `JWStatement` ขึ้นมา สุดท้ายออบเจกต์ `JWStatement` จะถูกส่งกลับไปยังโปรแกรมของไคลเอนต์ ซึ่งมีเรเฟอเรนซ์ (reference) ของรีโมต `Statement` อยู่ :

```
public Statement createStatement() throws SQLException
```

```
{
    try
    {
        IRemoteStatement remStmnt =
        (IRemoteStatement) remoteConnection.createStatement();
        JWStatement localStmntInstance = new JWStatement(remStmnt);
        return (Statement)localStmntInstance;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

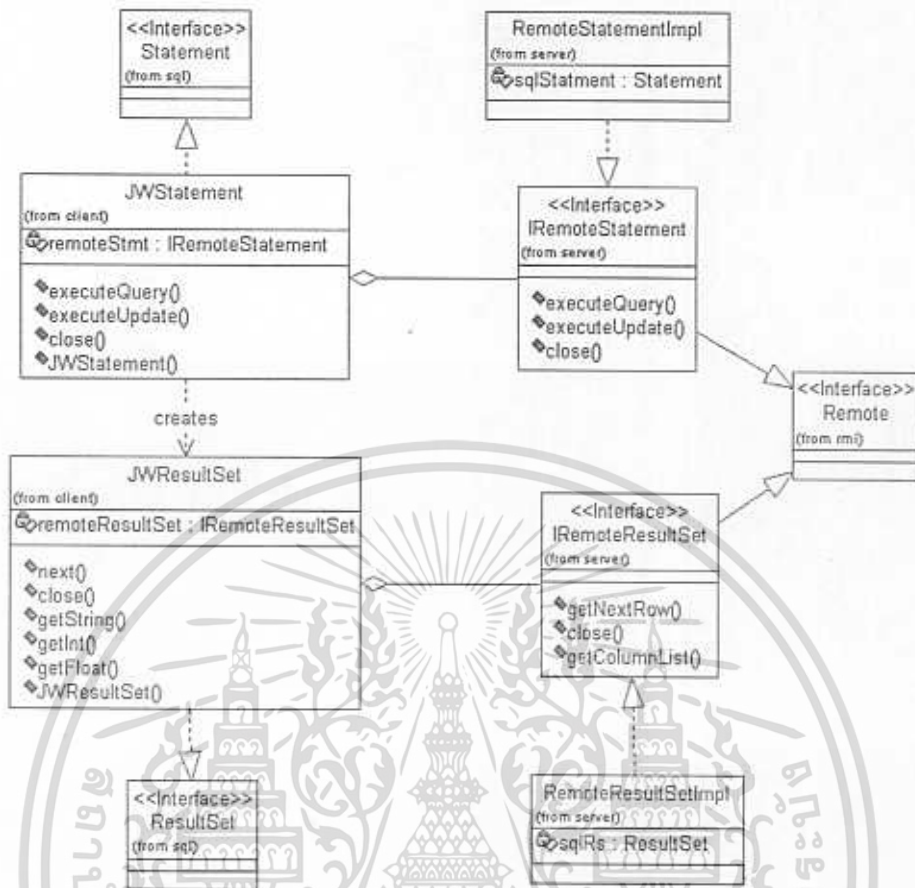
เจตของ JWStatement) เมธอด executeQuery () จะทำการเรียกไปยัง RemoteStatement.executeQuery () โดยอัตโนมัติ ซึ่งคืนค่าเป็นออบเจกต์ของรีโมต ResultSet สุดท้ายออบเจกต์ JWResultSet จะถูกส่งกลับไปยังโปรแกรมในฝั่งไคลเอนต์ ซึ่งมีเรฟเฟอร์เรนซ์ ของรีโมต ResultSet อยู่ :

```

public ResultSet executeQuery(String sqlQuery)
    throws SQLException
{
    try
    {
        IRemoteResultSet remoteRsInstance =
        (IRemoteResultSet) remoteStmt.executeQuery(sqlQuery);
        JWResultSet localRsInstance = new
        JWResultSet(remoteRsInstance);
        return (ResultSet)localRsInstance;
    }
    catch (RemoteException ex)
    {
        throw (new SQLException(ex.getMessage()));
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 คลาส โค้ดแกรมที่แสดงความสัมพันธ์ระหว่างคลาส Statement และ ResultSet บนฝั่งบนฝั่ง ไคลเอนต์และมิดเคิลทีयर

2.3.2.1.4 JWResultSet class

คลาส `com.jw.client.JWResultSet` อิมพลีเมนต์มาจาก อินเทอร์เฟส `ResultSet` ซึ่งจะอ้างถึงอินเทอร์เฟส `ResultSet` ของรีโมต เซิร์ฟเวอร์ และทำงานเหมือนห่ออยู่บนรีโมต `ResultSet` stub ซึ่ง เมธอด `JWStatement.executeQuery ()` ทำหน้าที่สร้าง `JWResultSet` ซึ่งจะได้ค่า ออบเจกต์ `JWResultSet` กลับมายังโปรแกรมฝั่งไคลเอนต์จากนั้น ไคลเอนต์จะเรียกเมธอดต่าง ๆ ของอินเทอร์เฟส `ResultSet` ผ่านทางออบเจกต์ที่ `JWResultSet` รีเทิร์นค่ากลับมา ออบเจกต์ `JWResultSet` จะส่งต่อการร้องขอไปที่ `ResultSet` ของรีโมตเซิร์ฟเวอร์เพื่อที่จะทำงานต่อไปโดยอัตโนมัติ ตัวอย่างเช่น เมื่อไคลเอนต์เรียก `rs.next ()` (ซึ่ง `rs` เป็นของออบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่นิพนธ์ได้เห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจกต์ `JWResultSet`) เมธอด `next ()` จะเรียกไปยัง `RemoteResultSet.getNextRow ()` โดยอัตโนมัติ ซึ่งจะได้ผลลัพธ์ส่งกลับมาเป็นอาร์เรย์ของข้อมูลในลักษณะที่เป็นแถว ซึ่งแถวนี้จะเก็บอยู่ใน `JWResultSet` เป็นแถวปัจจุบันเท่านั้น เมื่อไคลเอนต์เรียกเมธอด `rs.getString (1)` ข้อมูลก็จะถูกเลือกจากแถวข้อมูลและส่งค่ากลับคืนมา

```
public class JWResultSet implements java.sql.ResultSet
{
    // The current ResultSet data row
    private Object[] row;
    ...
    public boolean next() throws SQLException
    {
        try
        {
            // Get the current data row from remote ResultSet
            // All the getXXX methods will get data from local 'row'
            row = remoteResultSet.getNextRow();
        }
        catch(Exception ex)
        {
            return false;
        }

        if(row == null)
        {
            return false;
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    return true;
}
}

```

2.3.2.2 คลาสต่าง ๆ ของมิดเดิลเทียร์ (The middle-tier classes)

แพ็คเกจของมิดเดิลเทียร์ (com.jw.server) ประกอบด้วยคลาสต่อไปนี้

- com.jw.server.RemoteDriverImpl : คลาส Driver นี้ช่วยในการรับการติดต่อกับฐานข้อมูล และทำหน้าที่เหมือนกับเป็นเซิร์ฟเวอร์อาร์เอ็มไอ
- com.jw.server.RemoteConnectionImpl : คลาส Connection นี้ทำหน้าที่รับเจดีย์ซี สเตทเมนต์ (statement)
- com.jw.server.RemoteStatementImpl : คลาส Statement ทำหน้าที่รับ ResultSet
- com.jw.server.RemoteResultSetImpl : เป็นคลาส ResultSet

ต่อไปจะเป็นการอธิบายแต่ละคลาสโดยละเอียด

2.3.2.2.1 RemoteDriverImpl

คลาส RemoteDriverImpl ทำหน้าที่เหมือนเป็นเซิร์ฟเวอร์อาร์เอ็มไอ (โดยทำการขยายมาจากคลาส UnicastRemoteObject) และมีเมธอด getConnection() ให้กับ ไคลเอนต์ในฝั่งไคลเอนต์เทียร์ ในการเรียกใช้ (รูปที่ 2 แสดงความสัมพันธ์ของคลาส และอินเทอร์เฟซอื่น ๆ ของ RemoteDriverImpl) RemoteDriverImpl อิมพลีเมนต์มาจาก com.jw.server.IRemoteDriver ซึ่งเป็นอินเทอร์เฟซที่ขยาย Remote และเตรียมเมธอด getConnection() ซึ่งเมธอด getConnection() นี้จะทำการสร้างเจดีย์ซี คอนเนคชัน (JDBC Connection) และส่งกลับไปเป็น เรฟเฟอร์เรนซ์ ของ IRemoteConnection

เพราะว่า RemoteDriverImpl เป็นอาร์เอ็มไอเซิร์ฟเวอร์ ดังนั้นการลงทะเบียนด้วยตัวเองของ RemoteDriverImpl โดยใช้การเนมมิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Naming) เหมือนกับ RemoteDriver ทำให้ส่วนที่อยู่ในฝั่งไคลเอนต์ของ
ไดร์ฟเวอร์เจดีบีซีจะสามารถติดต่อไปยังฝั่งเซิร์ฟเวอร์ได้โดยใช้เมธอด
Naming.lookup () ต่อจากนั้นมันก็จะโหลดไดร์ฟเวอร์ประเภท 4
เพื่อที่จะสร้างการติดต่อกับฐานข้อมูล

```
public static void main(String args[])
{
    try
    {
        // Create a RemoteDriverImpl instance to register with naming service
        RemoteDriverImpl serverInstance = new RemoteDriverImpl();
        Naming.rebind("RemoteDriver",serverInstance);
        // Load the JDBC-ODBC Bridge driver
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    }
}
```

เมธอด getConnection () จะสร้างการเชื่อมต่อแบบเจดีบีซี
ไดร์ฟเวอร์ประเภท 4 จากนั้นจะส่งค่าออบเจกต์ กลับไปยังรีโมท
อินเทอร์เฟซของออบเจกต์ RemoteConnectionImpl เพื่อถือครองการ
เชื่อมต่อ (holding connection)

```
public IRemoteConnection getConnection()
throws RemoteException,SQLException
{
    String URL="jdbc:odbc:"+DSN;
    Connection sqlCon =
    DriverManager.getConnection(URL,dsUser,dsPassword);
    RemoteConnectionImpl ConnectionInstance =
    new RemoteConnectionImpl(sqlCon);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return (IRemoteConnection)ConnectionInstance;
}

```

2.3.2.2.2 RemoteConnectionImpl

คลาสนี้ขยายคลาส UnicastRemoteObject และอิมพลิเมนต์มาจากอินเทอร์เฟซ IRemoteConnection (ในรูปที่ 2 และ 3 แสดงความสัมพันธ์ของคลาสและอินเทอร์เฟซของ RemoteConnectionImpl) อินเทอร์เฟซ IRemoteConnection จะมีเมธอดที่สร้างเจดีบีซี Statement และปิดการเชื่อมต่อกับฐานข้อมูล คลาส RemoteConnectionImpl จะรวมออบเจกต์ Connection และส่งผ่าน ไปยังตัวที่สร้างมันขึ้นมา

เมธอด createStatement () สร้าง statement และส่งกลับค่าของออบเจกต์ RemoteStatement ที่จัดการ Statement

```

public IRemoteStatement createStatement() throws
RemoteException,SQLException
{
RemoteStatementImpl StmtImplInstance = new
RemoteStatementImpl(sqlConnection.createStatement());
return (IRemoteStatement)StmtImplInstance;
}

```

เมธอด closeConnection () เป็นเมธอดที่ทำการปิดออบเจกต์ Connection ซึ่งมันจะทำงาน โดยการเรียกเมทอด close () :

```

public void closeConnection() throws RemoteException,SQLException
{
sqlConnection.close();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2.3 RemoteStatementImpl

คลาส RemoteStatementImpl ขยายมาจาก คลาส UnicastRemoteObject และอิมพลีเมนต์อินเทอร์เฟซ IRemoteStatement ซึ่งอินเทอร์เฟซ IRemoteStatement จะมีเมธอดสำหรับสร้างเจดีย์ ResultSet และปิดอ็อบเจกต์ Statement คลาส RemoteStatementImpl จะหุ้มอ็อบเจกต์เจดีย์ Statement และถูกส่งผ่านไปในตัวที่สร้างมันขึ้นมา

เมธอด executeQuery() สร้าง JDBC ResultSet จากนั้นจะรีเทิร์นไปยังรีโมตอินเทอร์เฟซของอ็อบเจกต์ RemoteResultSetImpl ที่เกี่ยวข้องกับเจดีย์ ResultSet :

```
public IRemoteResultSet executeQuery(String Query) throws
RemoteException,SQLException
{
    ResultSet rs = sqlStatment.executeQuery(Query);
    RemoteResultSetImpl remoteRs = new RemoteResultSetImpl(rs);
    return (IRemoteResultSet)remoteRs;
}
```

เมธอด close() ทำการปิดอ็อบเจกต์ JDBC Statement

```
public void close() throws RemoteException, SQLException
{
    sqlStatment.close();
}
```

2.3.2.2.4 RemoteResultSetImpl

คลาส RemoteResultSetImpl ขยายมาจากคลาส UnicastRemoteObject และอิมพลีเมนต์อินเทอร์เฟซ IRemoteResultSet (รูป 4 แสดงความสัมพันธ์ของคลาสและอินเทอร์เฟซ) ซึ่งเมธอดของอินเทอร์เฟซจะช่วยรับ ResultSet row และปิดอ็อบเจกต์ของ ResultSet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส RemoteResultSetImpl จะถูกหุ้มด้วยอ็อบเจกต์เจดีย์ ResultSet และถูกส่งผ่านไปในตัวที่สร้างมันขึ้นมา

เมธอด getNextRow () จะส่งผลลัพธ์ 1 แถวของ ResultSet ในรูปแบบของอาร์เรย์ของอ็อบเจกต์ ไปยัง JWResultSet ของไคลเอนต์ซึ่งมันจะส่งค่า null ถ้า ResultSet ไม่มีแถวอื่น ๆ อีกแล้ว

```
public Object[] getNextRow() throws RemoteException,SQLException
{
    // Return null if all data has already been iterated
    if(sqlRs.next() == false)
        return null;
    // Prepare the data row in an array of Objects
    Object []row = new Object[colNum];
    for(int i = 1; i <= colNum; i++)
        row[i-1] = sqlRs.getString(i);
    return row;
}
```

The close() method closes the JDBC ResultSet object:

```
public void close() throws RemoteException,SQLException
{
    sqlRs.close();
}
```

ในเนื้อหาส่วนข้างบนนี้เราได้แสดงให้เห็นถึงสถาปัตยกรรมของเจดีย์ไคร์ฟเวอร์ประเภท 3 ซึ่งประกอบด้วยส่วนประกอบ 2 ส่วนคือ ไคลเอนต์และเจียร์ฟเวอร์ และใช้อาร์เอ็มไอในการติดต่อระหว่างไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเซิร์ฟเวอร์ นอกจากนี้เรายังได้ทำการแสดงให้เห็นว่าออบเจกต์ต่าง ๆ ใน ไคลเอนต์มีการอ้างอิงรีโมคออบเจกต์ในฝั่งเซิร์ฟเวอร์ที่เหมาะสมกันได้อย่างไร

2.4 การเพิ่มความเร็วด้วยการใช้งานคอนเน็คชันพูล (Database connection pooling for faster connection retrieval)

การเริ่มต้นการเชื่อมต่อฐานข้อมูลเพื่อที่จะทำการใช้งานสแตทเมนเอสคิวเอล (SQL) จะต้องเสียเวลาเสมอ เพื่อเพิ่มประสิทธิภาพของแอปพลิเคชัน เราสามารถซ่อนการเชื่อมต่อฐานข้อมูล ซึ่งเราสามารถอิมพลิเมนต์ได้ง่ายใน JWDriver โดยเมื่อไหร่ก็ตามที่มีการร้องขอการเชื่อมต่อมายังเซิร์ฟเวอร์ฐานข้อมูลจากเครื่องตัวกลางเพื่อที่จะบริการการร้องขอของแอปพลิเคชันที่มาจากไคลเอนต์ การเชื่อมต่อนั้นจะได้รับกลับมาจากพูล (pool) เมื่อแอปพลิเคชันของไคลเอนต์ต้องการปิดการเชื่อมต่อ การเชื่อมต่อดังกล่าวในเครื่องตัวกลางจะคืนให้กับพูล ดังนั้นมันจึงสามารถให้บริการการร้องขอของไคลเอนต์อื่นได้ ซึ่งเหตุการณ์ทั้งหมดไคลเอนต์จะไม่ทราบเลยว่าคอนเน็คชันพูลถึง (connection pooling) ได้ถูกสร้างขึ้นในเครื่องตัวกลาง

2.4.1 คอนเน็คชันพูล (The connection pool)

คลาส `ConnectionPool` ถูกสร้างในส่วนไดร์ฟเวอร์ของเครื่องตัวกลาง โดย `ConnectionPool` จะมีเมธอด `newConnection ()` สำหรับเพิ่มการเชื่อมต่อให้กับพูล ในระหว่างการทำงานของเซิร์ฟเวอร์ เมธอด `freeConnection ()` สำหรับคืนการเชื่อมต่อให้กับพูลเมื่อใช้งานเสร็จ รวมถึงเมธอด `getConnection ()` รับการเชื่อมต่อมาจากพูล คอนสตรัคเตอร์ (Constructor) จะสร้างเวกเตอร์ว่างเพื่อเก็บการเชื่อมต่อทั้งหมดไว้ ดังนี้

```
ConnectionPool()
{
    connectionPool = new Vector ( );
}
```

คลาส `ConnectionPool` จะปกปิดการเชื่อมต่อฐานข้อมูลที่ใช้พูลไว้ในเวกเตอร์ที่ถูกอิมพลิเมนต์โดยออบเจกต์ `Vector` คลาส `ConnectionPool` มี 2 เมธอด คือ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

freeConnection () และ getConnection () เพื่อจัดการกับเวกเตอร์ของคอนเน็คชันพูล (connection pool) ซึ่งเมธอดนี้จะคืนและรับ การเชื่อมต่อออกมาจากเวกเตอร์ ตามลำดับ และมีเมธอด newConnection () ในการสร้างการเชื่อมต่ออันใหม่กับฐานข้อมูล

```
public synchronized void freeConnection(Connection con){
    // Add the JDBC-ODBC bridge connection to the pool
    connectionPool.addElement(con);
}

public synchronized Connection getConnection(){
    Connection con = null;
    if(connectionPool.size() > 0){
        con = (Connection)connectionPool.lastElement();
        connectionPool.removeElementAt(connectionPool.size() -1);
    }
    return con;
}
```

2.4.1.1 การสร้างคอนเน็คชันพูล (Initialize the connect pool)

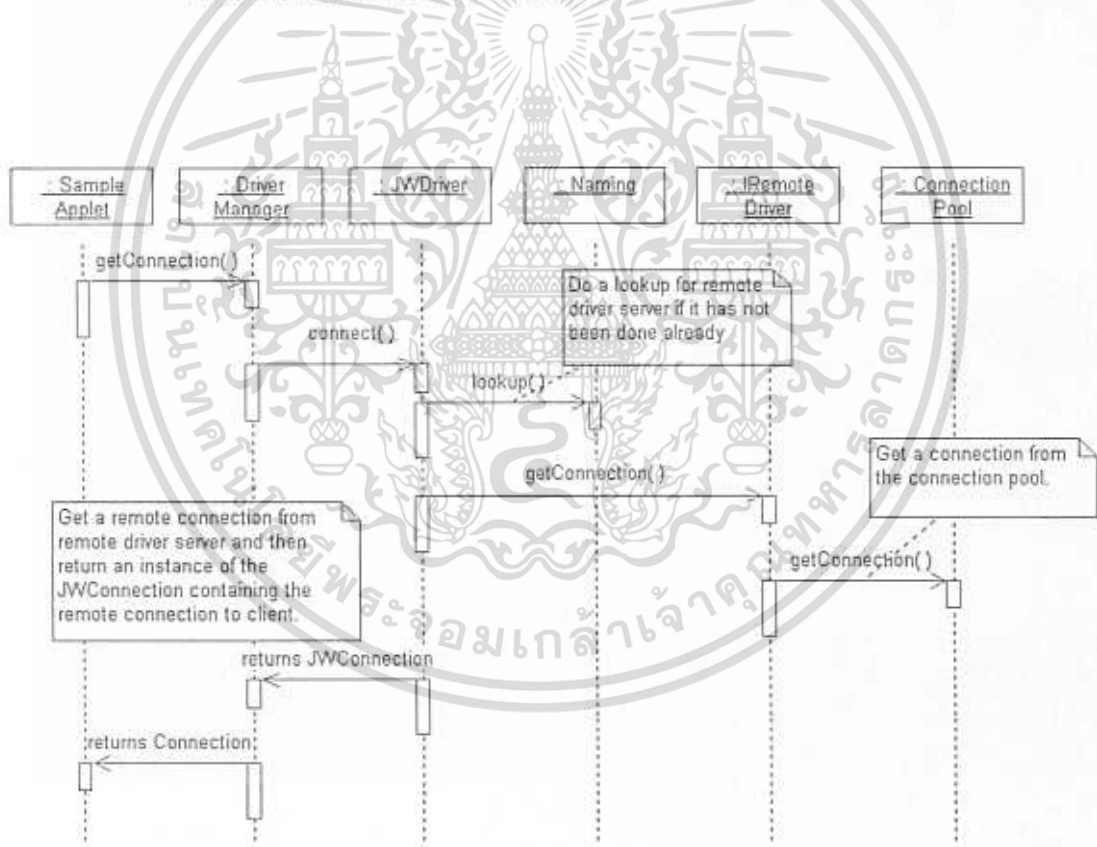
คอนเน็คชันพูลจะเริ่มต้นเมื่ออาร์เอ็มไอเอชอีวีเวอร์ของ JWDriver ถูก สตาร์ท (start) คลาส RemoteDriverImpl สร้างโหนดไดร์ฟเวอร์ที่ใช้ในการติดต่อกับฐานข้อมูล และสร้างเวกเตอร์ว่างขึ้นมาเพื่อเอาไว้คอยเก็บการเชื่อมต่อที่จะถูกสร้างขึ้น

```
try {
    Class.forName(source);
    log(new Date() + " "+"INITIALIZED POOL :"+DBName+" ");
} catch(Exception e){ }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.2 การดึงคอนเน็คชันจากพูลมาใช้งาน (Retrieve the connection from the pool)

เมื่อไรก็ตามที่โปรแกรมของไคลเอนต์ร้องขอการเชื่อมต่อ ฟังก์ชันของ JWDriver จะส่งต่อการร้องขอไปยังเครื่องกลางเพื่อให้ทำหน้าที่แทน จากนั้นเมธอด getConnection () ของ RemoteDriverImpl จะได้รับการเชื่อมต่อจากคอนเน็คชันพูล แต่ถ้าคอนเน็คชันพูลไม่มีการเชื่อมต่อว่างก็จะทำการตรวจสอบว่าจำนวนการเชื่อมต่อที่มีอยู่กับฐานข้อมูลมีจำนวนมากกว่าค่ามากที่สุดที่กำหนดไว้หรือไม่ ถ้าพบว่างไม่มากกว่าก็จะทำการเรียกเมธอด newConnection () เพื่อสร้างการเชื่อมต่ออันใหม่ แต่ถ้าพบว่ามีมากกว่าค่าที่กำหนดไว้แล้วก็จะเข้าสู่สถานะ wait เพื่อรอการเชื่อมต่อที่จะว่างต่อไป



รูปที่ 2.9 แสดงซีควเอนซ์โคอะแกรมของการรับการเชื่อมต่อกลับมาจากพูล

2.4.1.3 การคืนคอนเน็คชันสู่พูล (Return the connection to the pool)

เมื่อไรก็ตามที่โปรแกรมของไคลเอนต์ต้องการปิดการเชื่อมต่อฐานข้อมูล การเชื่อมต่อจะถูกส่งกลับคืนพูล โดยที่ฝั่งไคลเอนต์ของ JWDriver จะส่งต่อการร้องขอเพื่อปิดการเชื่อมต่อไปยังเครื่องตัวกลางผ่านคลาส IRemoteConnection จากนั้น RemoteConnectionImpl จะคืนการเชื่อมต่อฐานข้อมูลไปยังพูลดังนี้ :

```
public synchronized void freeConnection(Connection conn) {
    try{
        connectionPool.addElement( conn );
        usingConnection--;
        notifyAll();
    }catch(Exception e) {System.out.println(e);}
}
```



บทที่ 3

การออกแบบ

3.1 ข้อมูลเบื้องต้น

ทำการเก็บข้อมูลการใช้งานระบบฐานข้อมูลจากแหล่งต่าง ๆ เช่น จากผู้ใช้ทั่วไป จากผู้ดูแลระบบ และจากการศึกษาทำให้สามารถออกแบบฟังก์ชันการทำงานต่าง ๆ ได้ดังนี้

3.1.1 ส่วนผู้ดูแลระบบ

1. หน้าจัดการข้อมูลผู้ใช้ จะใช้ในการเก็บข้อมูลผู้ใช้นี้

1.1 ชื่อในการล็อกอิน (Username)

- จะต้องไม่ซ้ำกับชื่อในการล็อกอินที่มีอยู่

1.2 รหัสเวิร์ด (Password)

2. หน้าควบคุมระบบ มีหน้าที่ในการ

2.1 เปิด-ปิด การใช้งานระบบ

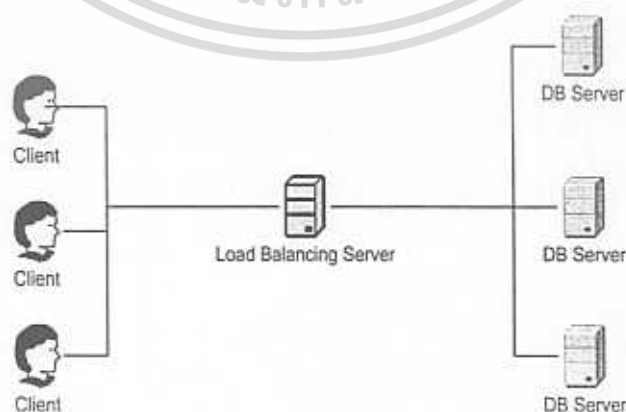
2.2 แสดงสถานะของระบบ

3. หน้าเก็บสถิติการใช้งาน (Log file)

3.1.2 ส่วนทดสอบระบบ

3.2 การออกแบบระบบ

3.2.1 สถาปัตยกรรมของระบบ (System Architecture)



รูปที่ 3.1 สถาปัตยกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบของระบบ

1. โพลิบาลานซ์เซิร์ฟเวอร์ ทำหน้าที่ในการรองรับการเชื่อมต่อจากผู้ใช้และเป็นเซิร์ฟเวอร์ ให้เครื่องผู้ใช้ติดต่อเพื่อเลือกติดต่อเซิร์ฟเวอร์ฐานข้อมูลที่เหมาะสม

2. เซิร์ฟเวอร์ฐานข้อมูลทำหน้าที่เก็บข้อมูลฐานข้อมูล

ในการเชื่อมต่อระหว่างโพลิบาลานซ์เซิร์ฟเวอร์ กับเซิร์ฟเวอร์ฐานข้อมูล จะใช้การเชื่อมต่อ โดยสถาปัตยกรรมเจคบีซีบีบนอาร์เอ็มไอ

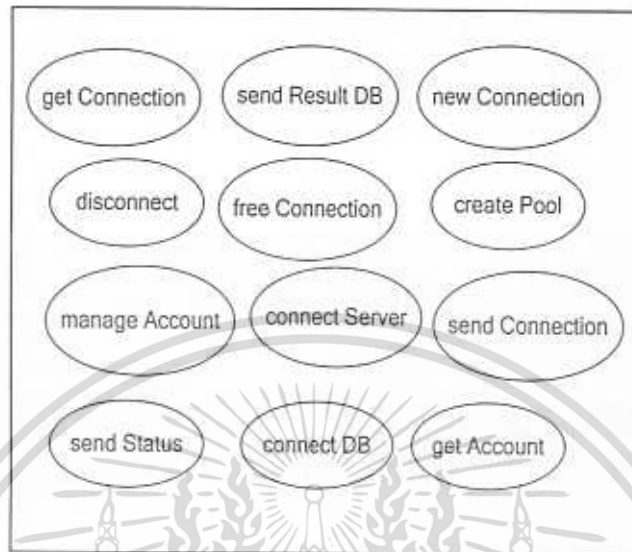
เครื่องมือที่จำเป็น

- เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น โพลิบาลานซ์เซิร์ฟเวอร์ จำนวน 1 เครื่อง
- เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ฐานข้อมูลจำนวน 3 เครื่อง
- เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น โคลเอนต์จำนวน 1 เครื่อง



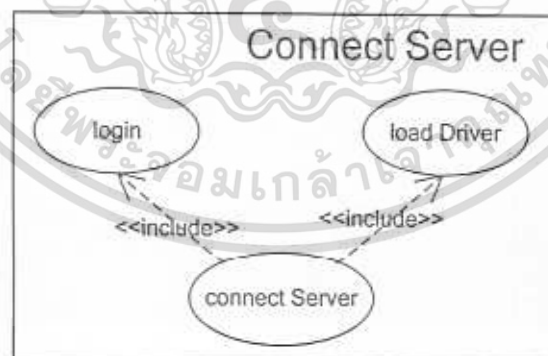
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ไฮเลเวลยูสเคส (High Level Use Case)



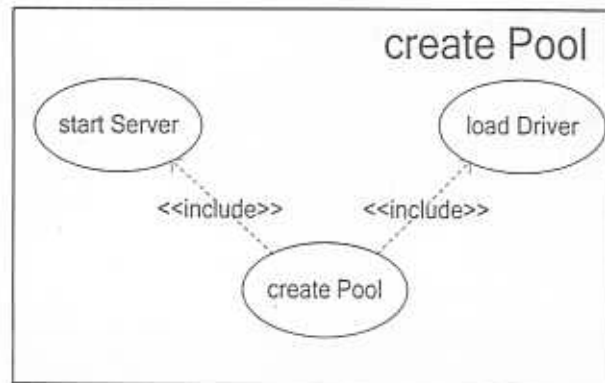
รูปที่ 3.3 ไฮเลเวลยูสเคส

3.2.4 ยูสเคสไดอะแกรม (Use Case Diagram)

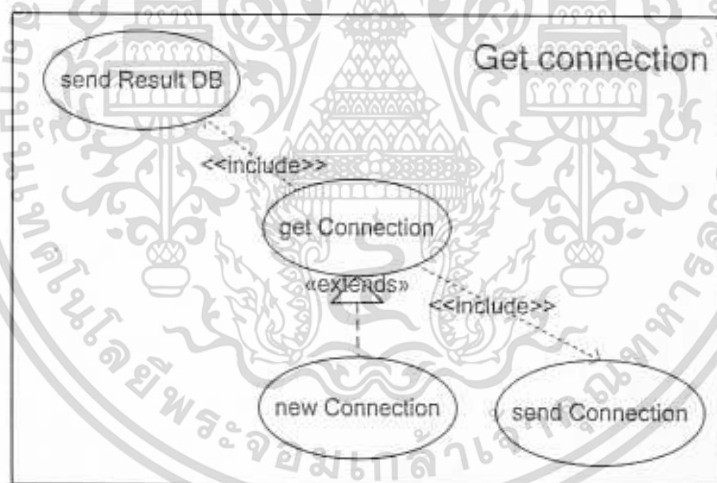


รูปที่ 3.4 ยูสเคสไดอะแกรมสำหรับการเชื่อมต่อกับเซิร์ฟเวอร์ (Connect Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

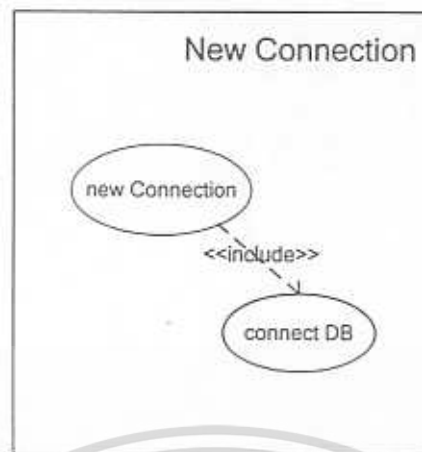


รูปที่ 3.5 ยูสเคสไดอะแกรมสำหรับการสร้างพูล (Create Pool)



รูปที่ 3.6 ยูสเคสไดอะแกรมสำหรับการรับคอนเน็คชั่น (Get Connection)

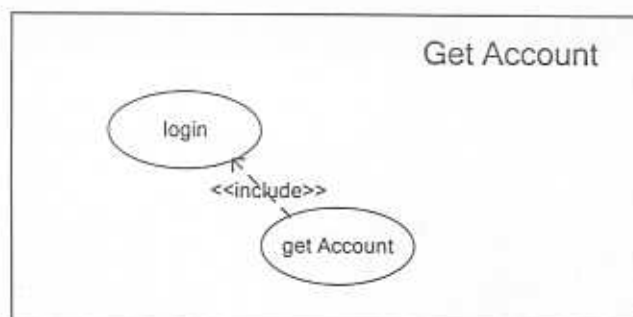
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ยูสเคสไดอะแกรมสำหรับการสร้างคอนเนคชั่น (New Connection)

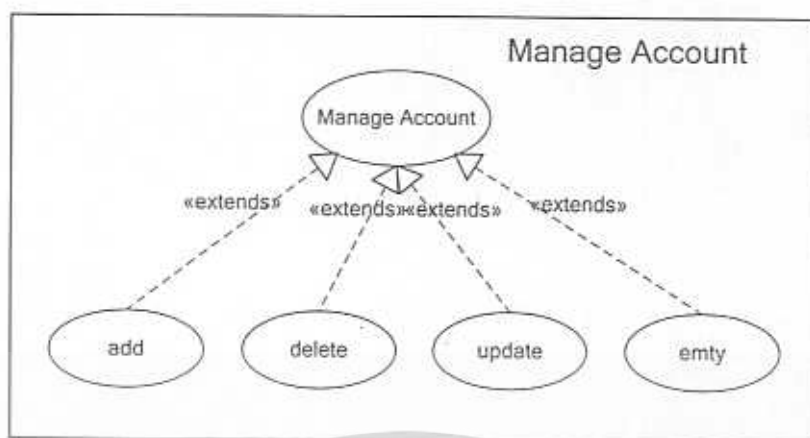


รูปที่ 3.8 ยูสเคสไดอะแกรมสำหรับการฟรีคอนเนคชั่น (Free Connection)

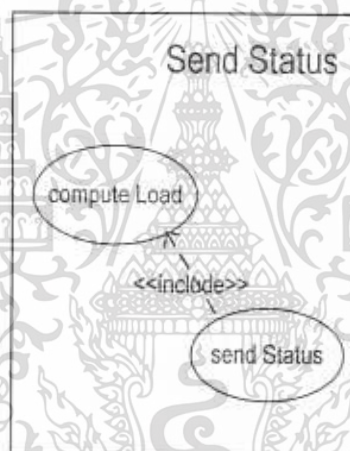


รูปที่ 3.9 ยูสเคสไดอะแกรมสำหรับการอ่านค่าข้อมูลผู้ใช้ (Get Account)

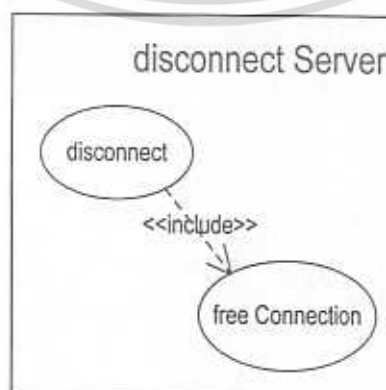
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ยูสเคสไดอะแกรมสำหรับการจัดการเกี่ยวกับข้อมูลผู้ใช้ (Manage Account)



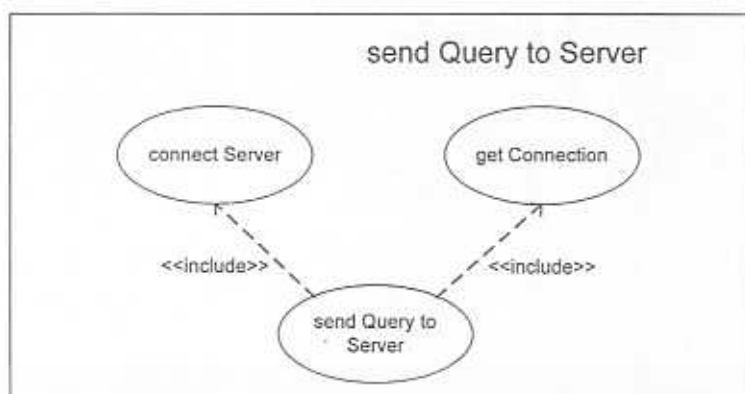
รูปที่ 3.11 ยูสเคสไดอะแกรมสำหรับการรายงานสถานะ (Send Status)



รูปที่ 3.12 ยูสเคสไดอะแกรมสำหรับการยกเลิกการเชื่อมต่อกับเซิร์ฟเวอร์ (Disconnect Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

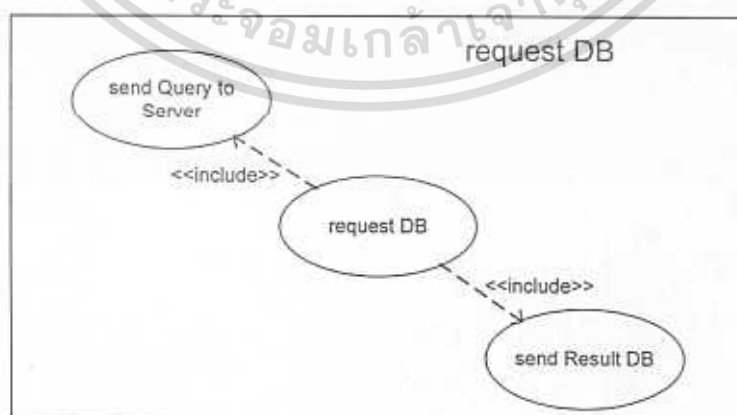
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 ยูสเคสไดอะแกรมสำหรับการส่งคำสั่งคิวรีไปยังเซิร์ฟเวอร์ (send Query to Server)

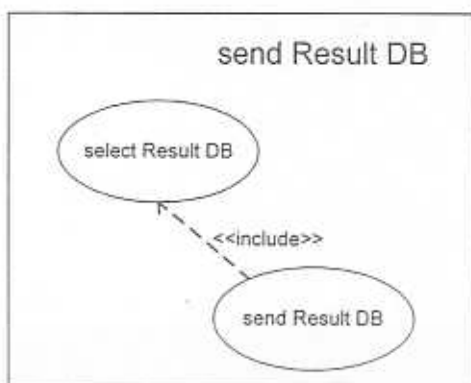


รูปที่ 3.14 ยูสเคสไดอะแกรมสำหรับการส่งผลลัพธ์ไปยังไคลเอนต์ (send resultset to Client)

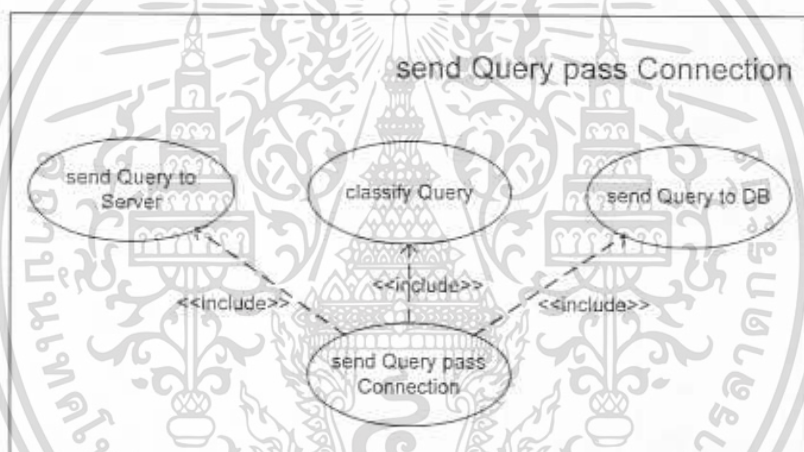


รูปที่ 3.15 ยูสเคสไดอะแกรมสำหรับการร้องขอว่าเซิร์ฟเวอร์ฐานข้อมูล
ขณะนี้ตัวใดถูกใช้งานน้อยสุด (request DB)

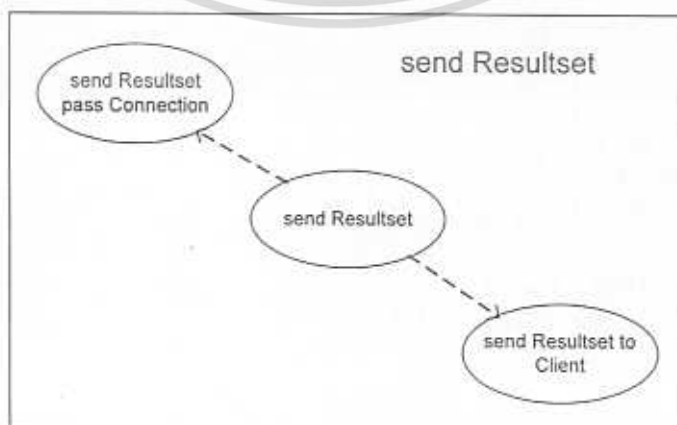
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ยูสเคสโคอะแกรมสำหรับการส่งผลไปแจ้งว่า
ในขณะนี้เซิร์ฟเวอร์ฐานข้อมูลตัวใดถูกใช้งานน้อยสุด (send Result DB)

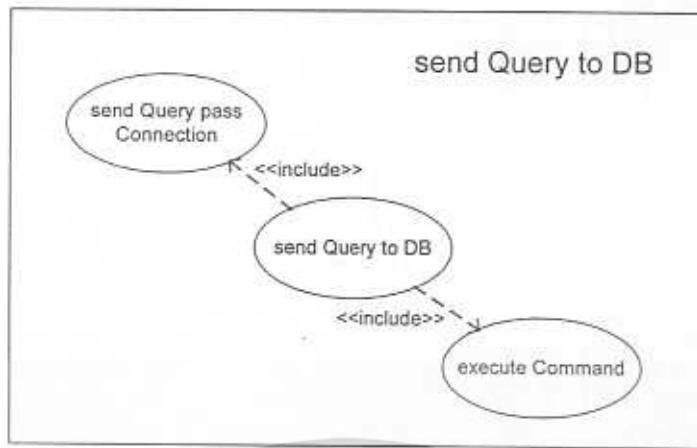


รูปที่ 3.17 ยูสเคสโคอะแกรมสำหรับการส่งคิวรีผ่าน ไปในคอนเน็กชัน
(send Query pass Connection)

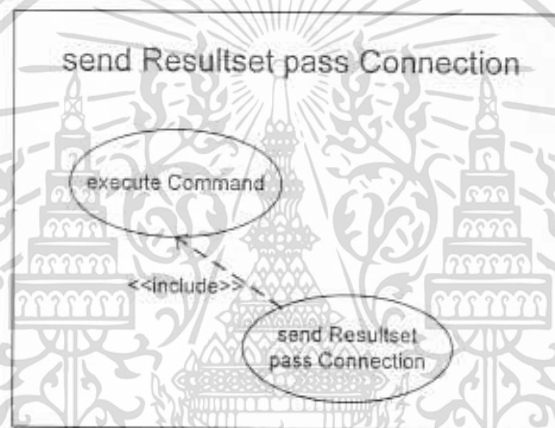


รูปที่ 3.18 ยูสเคสโคอะแกรมสำหรับการส่งผลลัพธ์ไปยังเซิร์ฟเวอร์ (send Resultset)

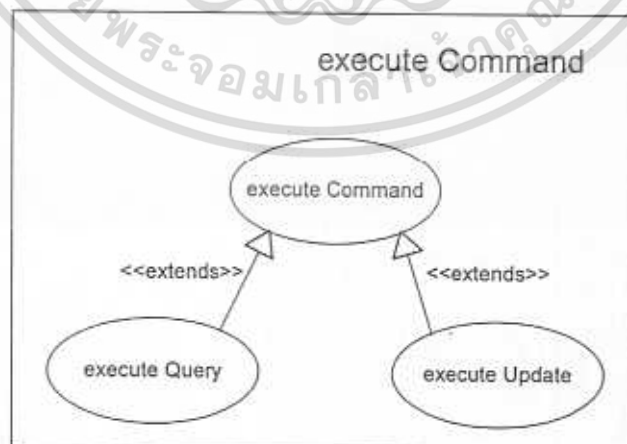
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 ยูสเคสไดอะแกรมสำหรับการส่งคิวรีไปยังเซิร์ฟเวอร์ฐานข้อมูล (send Query to DB)



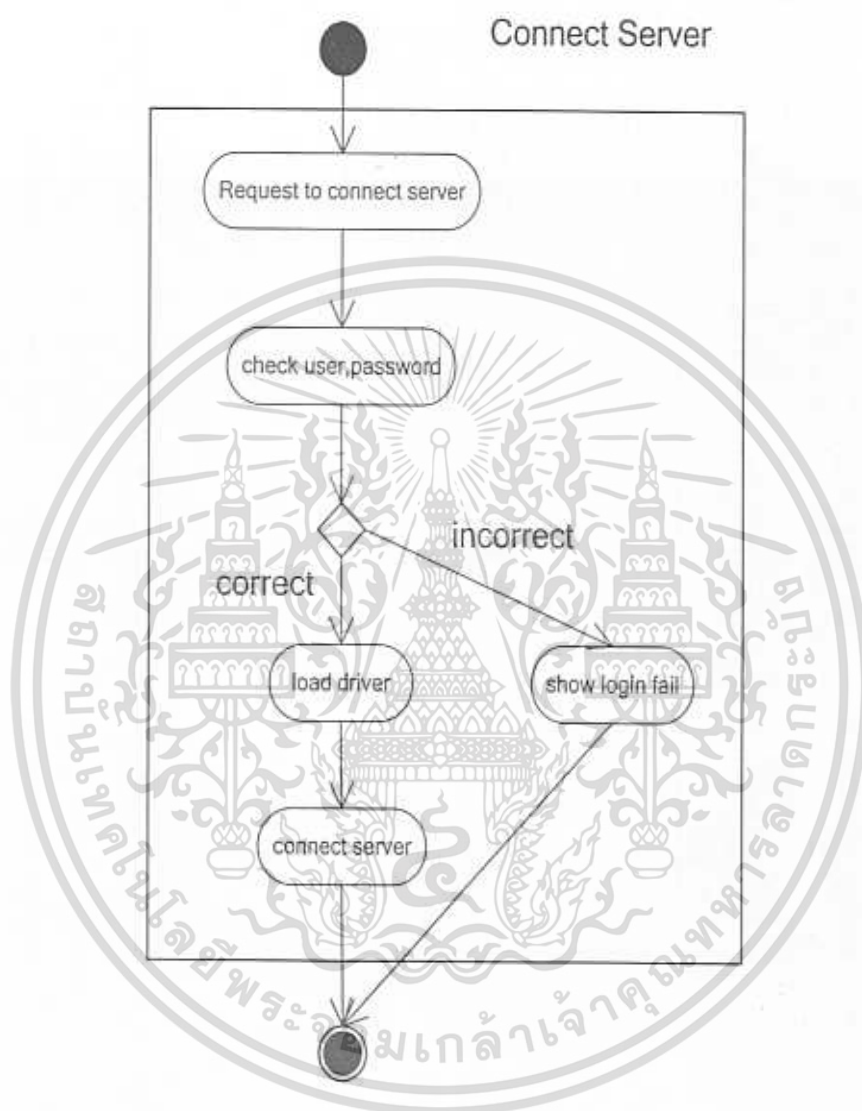
รูปที่ 3.20 ยูสเคสไดอะแกรมสำหรับการส่งผลลัพธ์ผ่านไปในคอนเน็คชั่น
(send Resultset pass Connection)



รูปที่ 3.21 ยูสเคสไดอะแกรมสำหรับการประมวลผลคำสั่ง (execute Command)

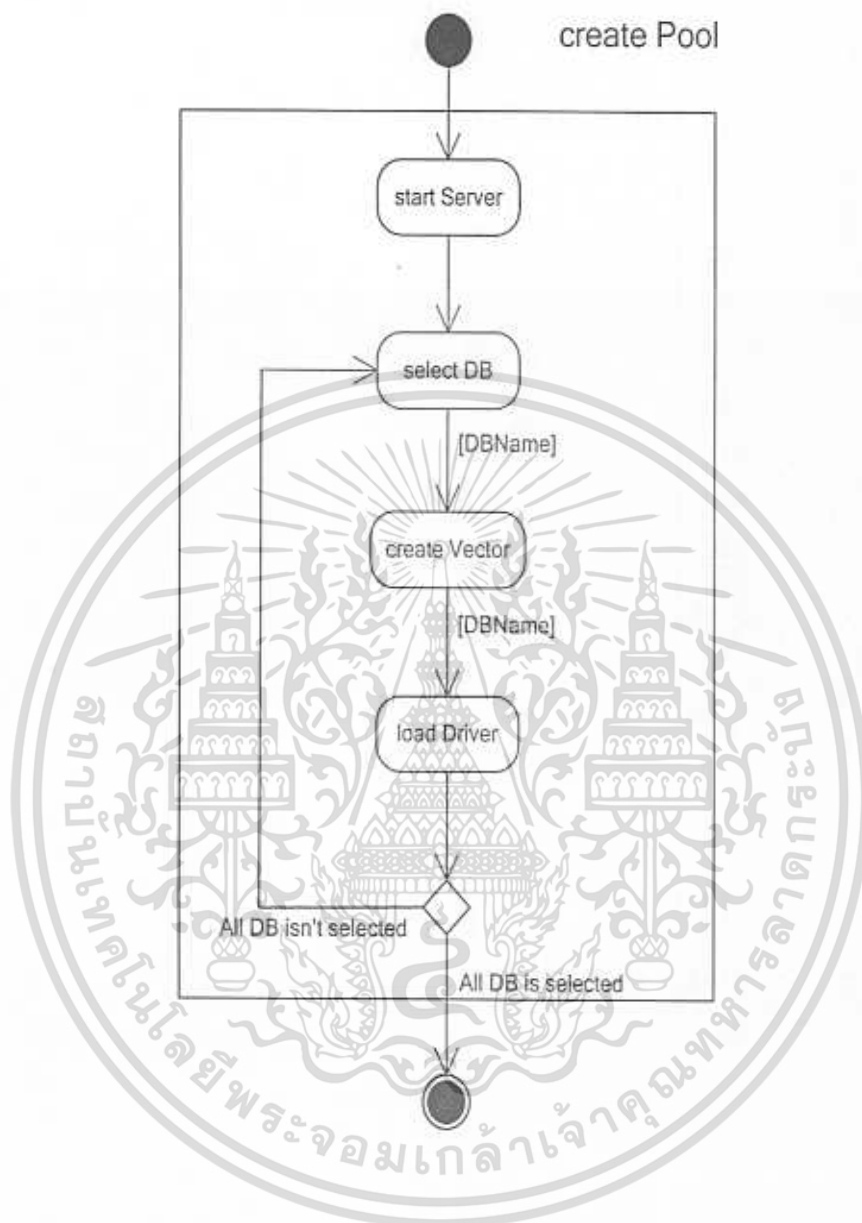
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 แอคทิวิตี้ไดอะแกรม (Activity Diagram)



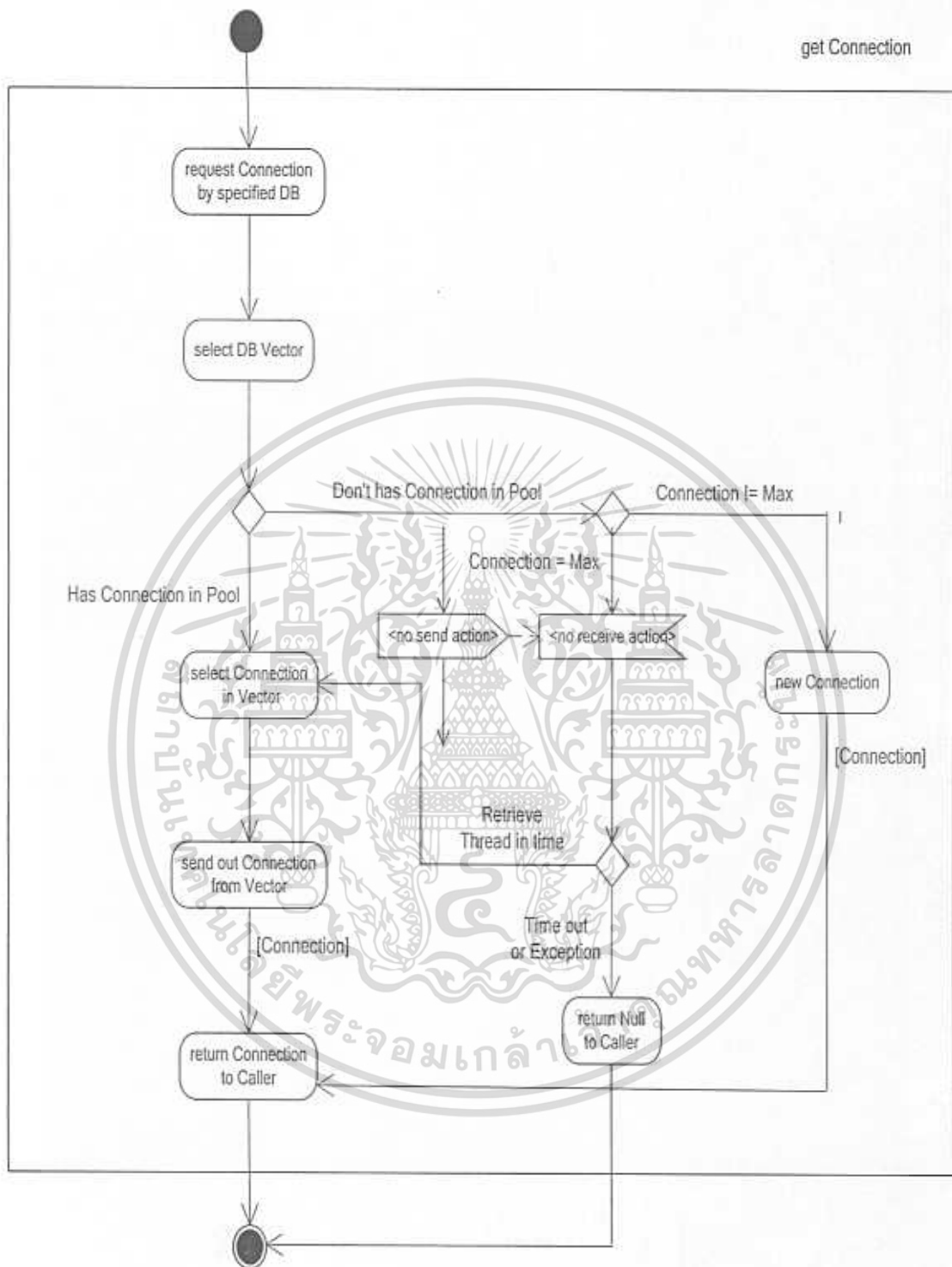
รูปที่ 3.22 แอคทิวิตี้ไดอะแกรมสำหรับการเชื่อมต่อกับเซิร์ฟเวอร์ (Connect Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



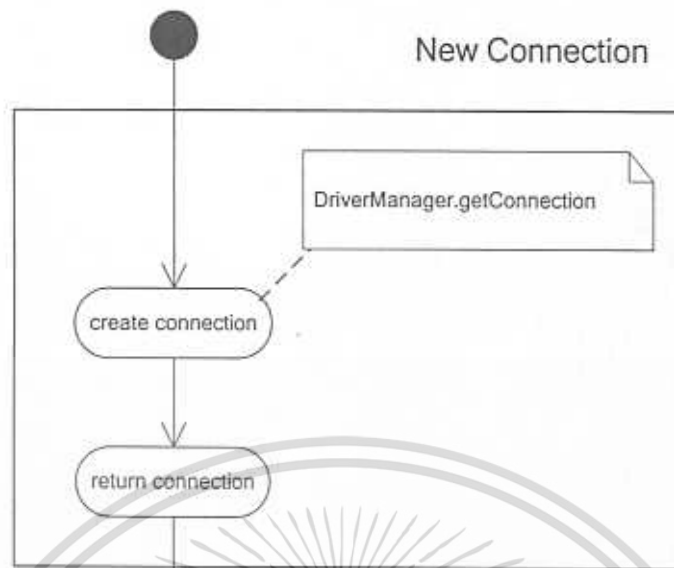
รูปที่ 3.23 แอคทิวิตี้ไดอะแกรมสำหรับการสร้างพูล (Create Pool)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

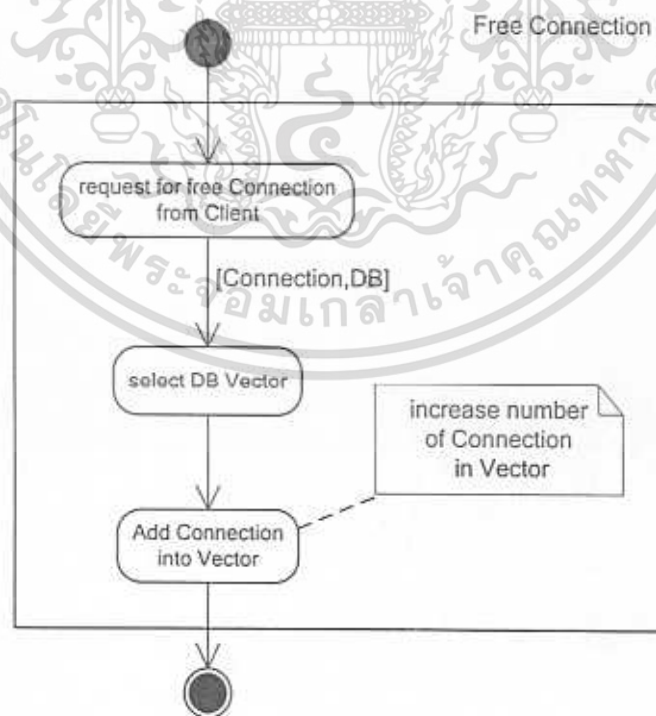


รูปที่ 3.24 แอกทิวิตี้ไดอะแกรมสำหรับการรับคอนเน็คชัน (Get Connection)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

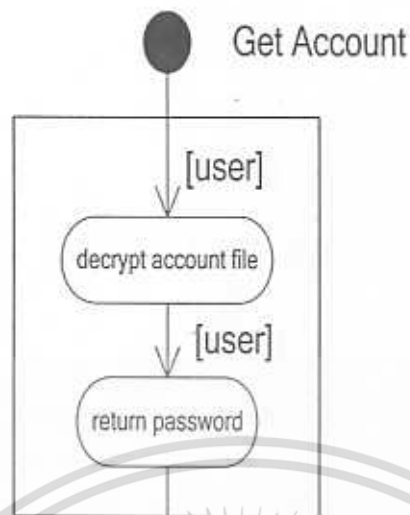


รูปที่ 3.25 แอกทिवิตีโค้ดโปรแกรมสำหรับการสร้างคอนเนกชัน (New Connection)

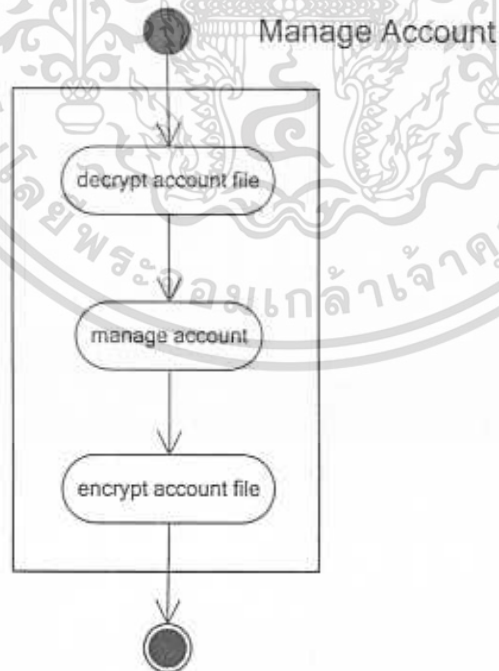


รูปที่ 3.26 แอกทिवิตีโค้ดโปรแกรมสำหรับการฟรีคอนเนกชัน (Free Connection)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

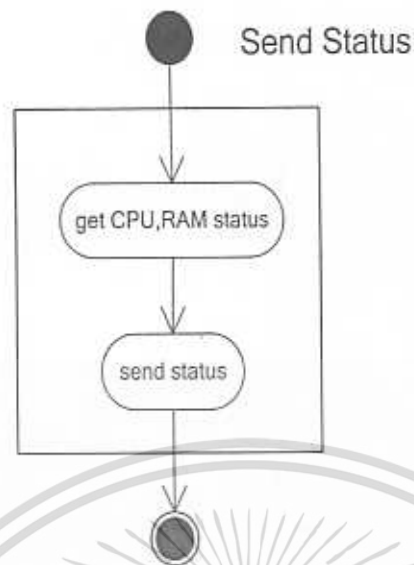


รูปที่ 3.27 แอคทีวิตี้ไดอะแกรมสำหรับการอ่านค่าข้อมูลผู้ใช้ (Get Account)

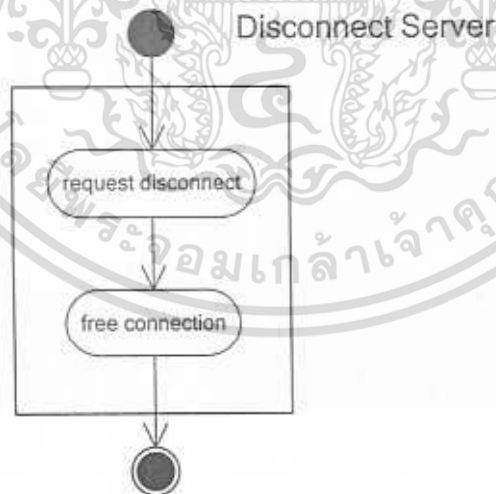


รูปที่ 3.28 แอคทีวิตี้ไดอะแกรมสำหรับการจัดการเกี่ยวกับข้อมูลผู้ใช้ (Manage Account)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

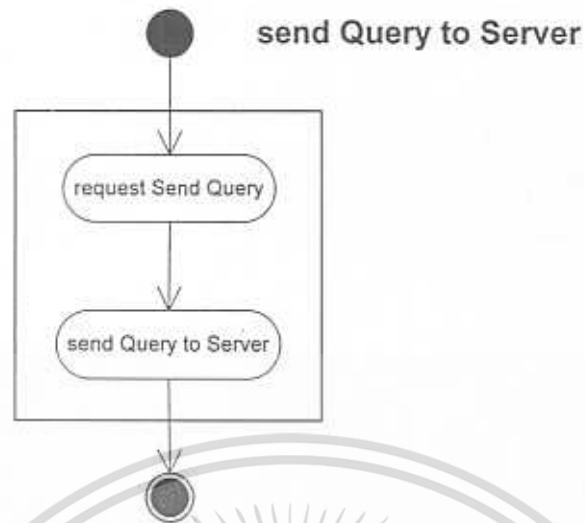


รูปที่ 3.29 แอกทิวิตี้ไดอะแกรมสำหรับการรายงานสถานะ (Send Status)

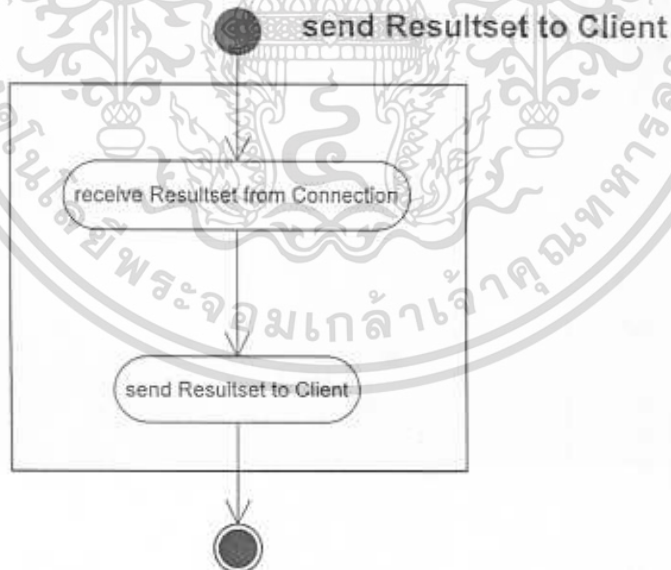


รูปที่ 3.30 แอกทิวิตี้ไดอะแกรมสำหรับการยกเลิกการเชื่อมต่อกับเซิร์ฟเวอร์ (Disconnect Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

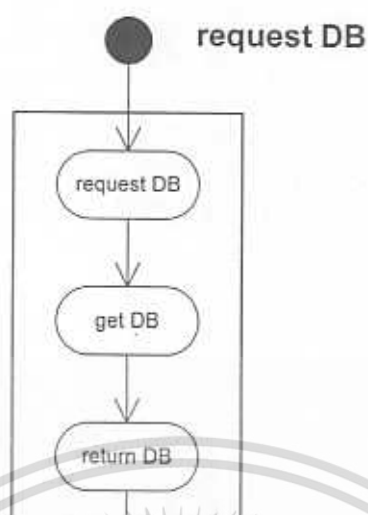


รูปที่ 3.31 แอกทिवิตีไดอะแกรมสำหรับการส่งคำสั่ง Query ไปยังเซิร์ฟเวอร์ (send Query to Server)

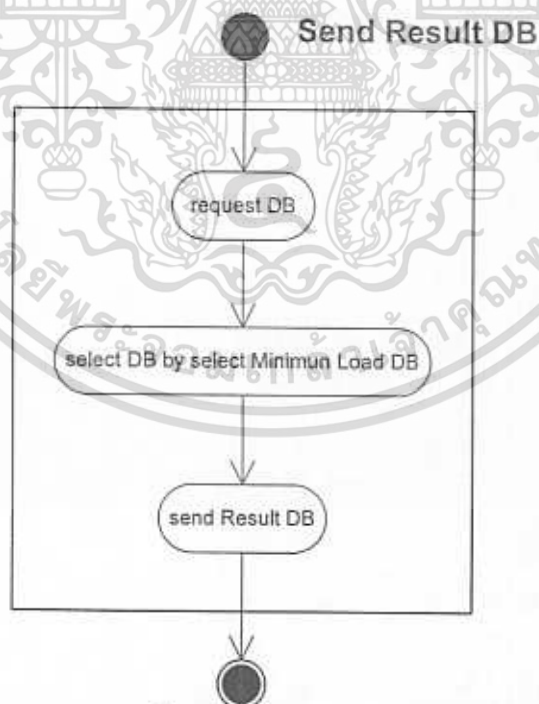


รูปที่ 3.32 แอกทिवิตีไดอะแกรมสำหรับการส่งผลลัพธ์ไปยังไคลเอนต์ (send resultset to Client)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

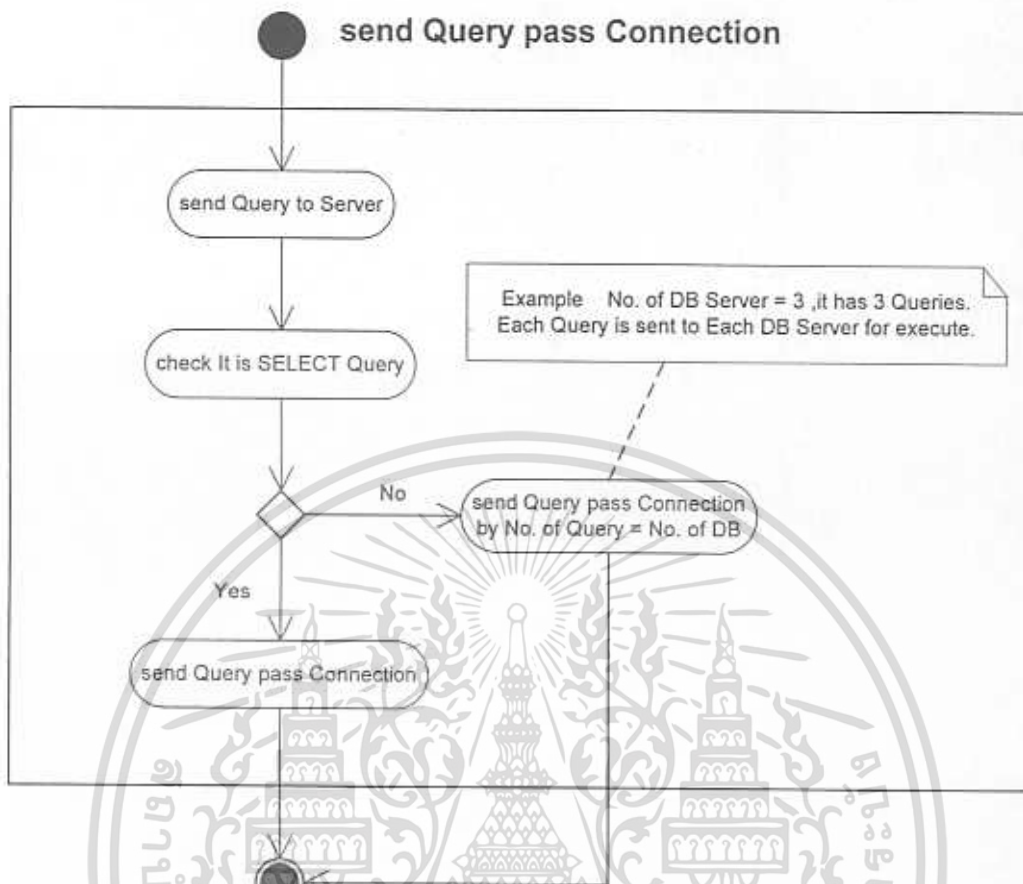


รูปที่ 3.33 แอคทิวิตีไดอะแกรมสำหรับการร้องขอว่าเซิร์ฟเวอร์ฐานข้อมูล
ขณะนี้ตัวใดถูกใช้งานน้อยสุด (request DB)

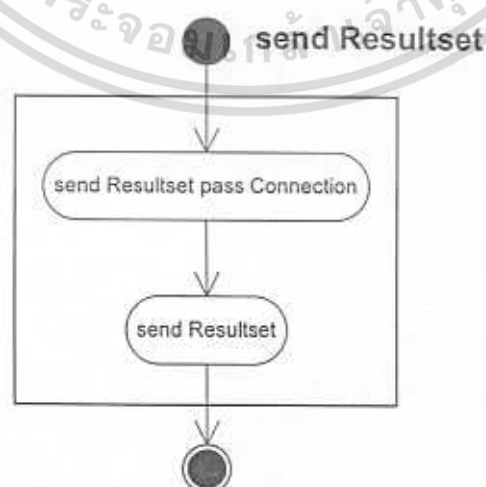


รูปที่ 3.34 แอคทิวิตีไดอะแกรมสำหรับการส่งผลไปแจ้งว่า
ในขณะนี้เซิร์ฟเวอร์ฐานข้อมูลตัวใดถูกใช้งานน้อยสุด (send Result DB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

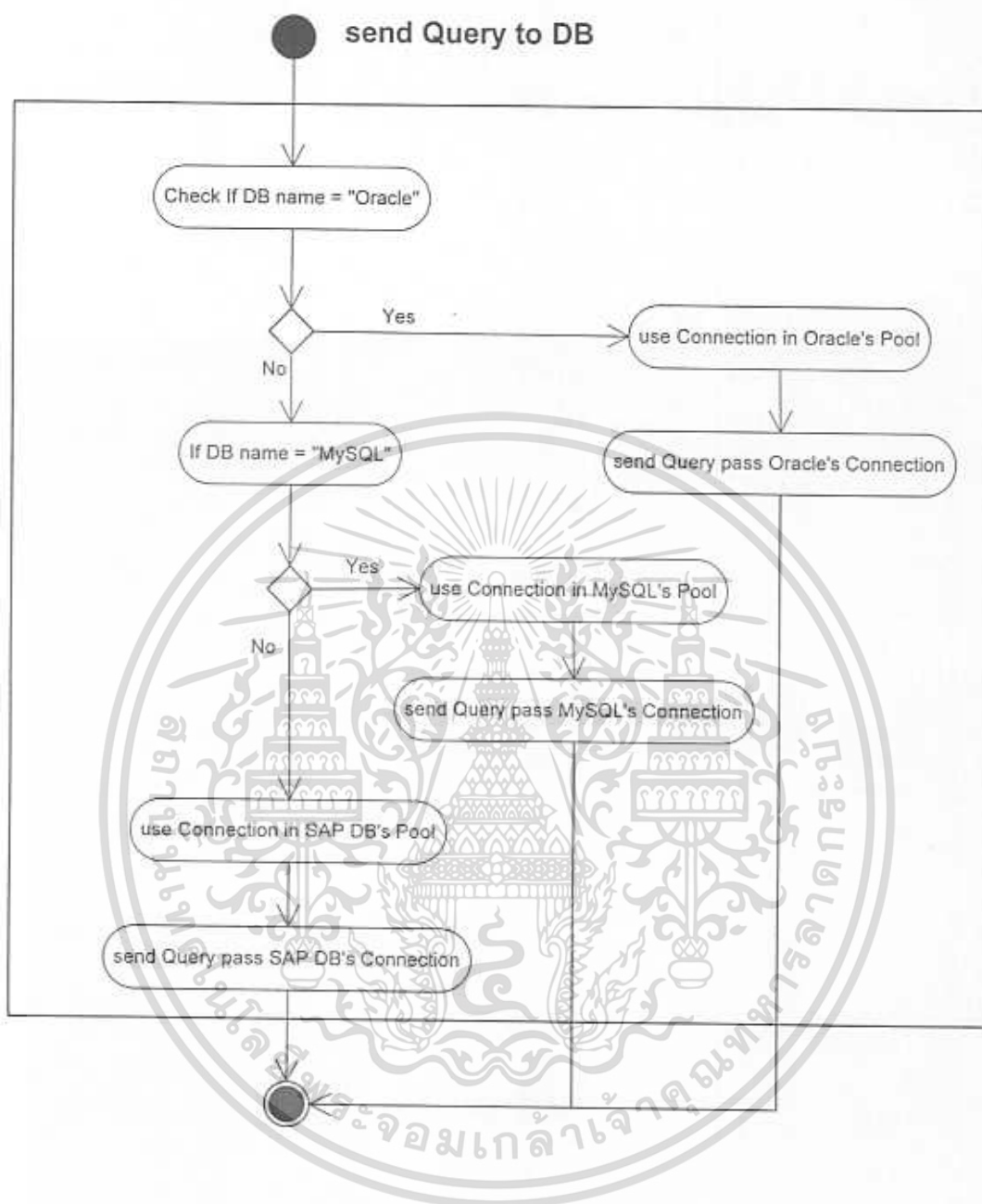


รูปที่ 3.35 แอคทิวิตี้ไดอะแกรมสำหรับการส่งคิวรีผ่านไปในคอนเน็คชัน
(send Query pass Connection)



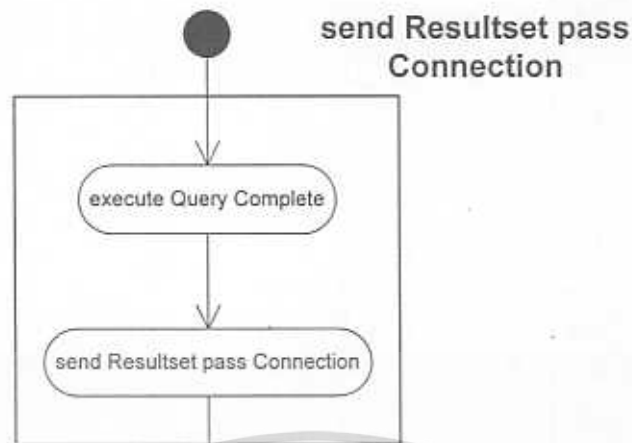
รูปที่ 3.36 แอคทิวิตี้ไดอะแกรมสำหรับการส่งผลลัพธ์ไปยังเซิร์ฟเวอร์ (send Resultset)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

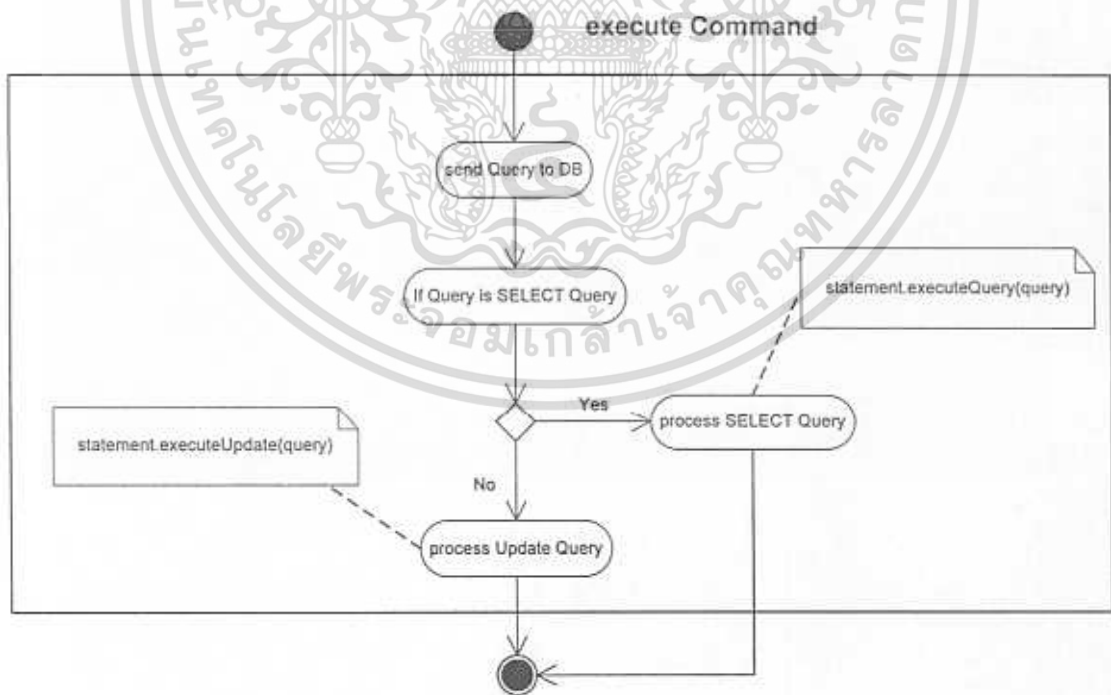


รูปที่ 3.37 แอคทิวิตีไดอะแกรมสำหรับการส่งคิวรีไปยังเซิร์ฟเวอร์ฐานข้อมูล (send Query to DB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



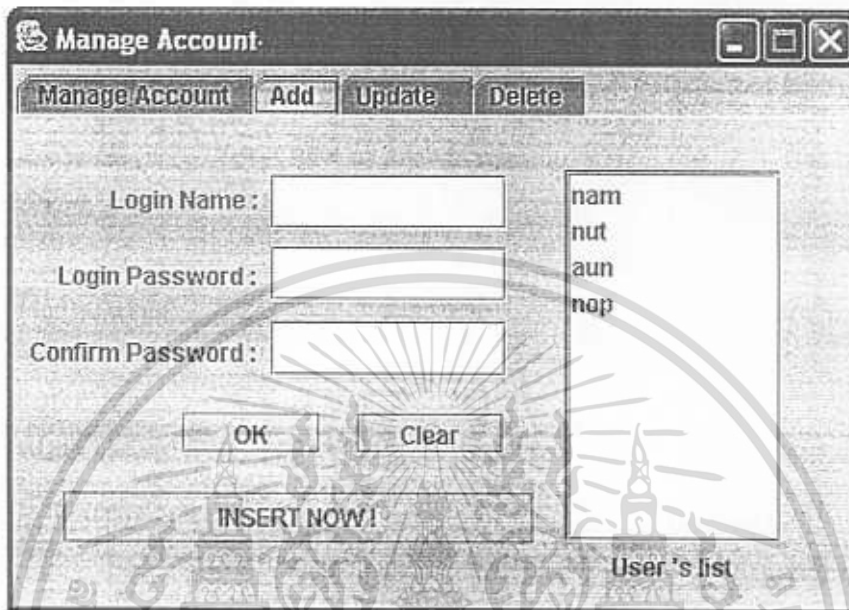
รูปที่ 3.38 แอกทิวิตีไดอะแกรมสำหรับการส่งผลลัพธ์ผ่านไปในคอนเน็คชัน
(send Resultset pass Connection)



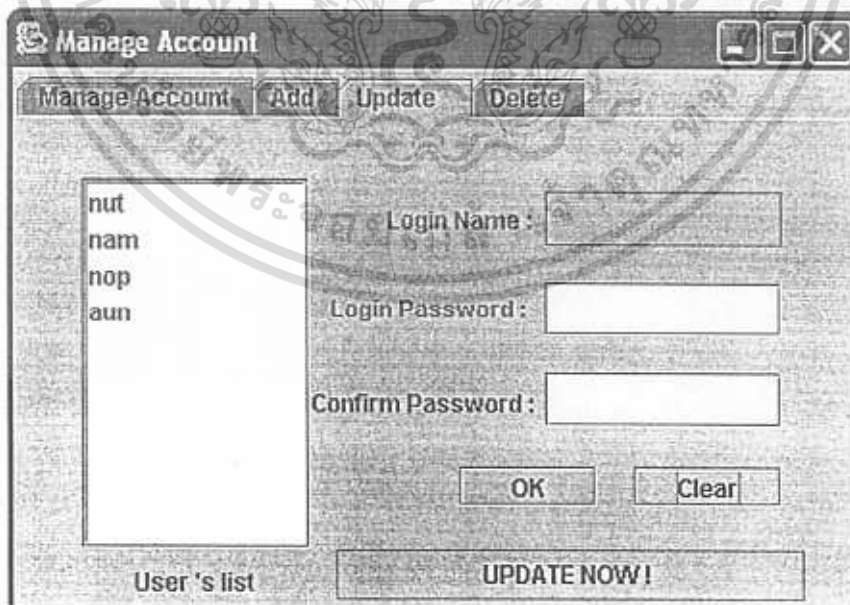
รูปที่ 3.39 แอกทิวิตีไดอะแกรมสำหรับการประมวลผลคำสั่ง (execute Command)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 การออกแบบส่วนติดต่อผู้ใช้ (User Interface)

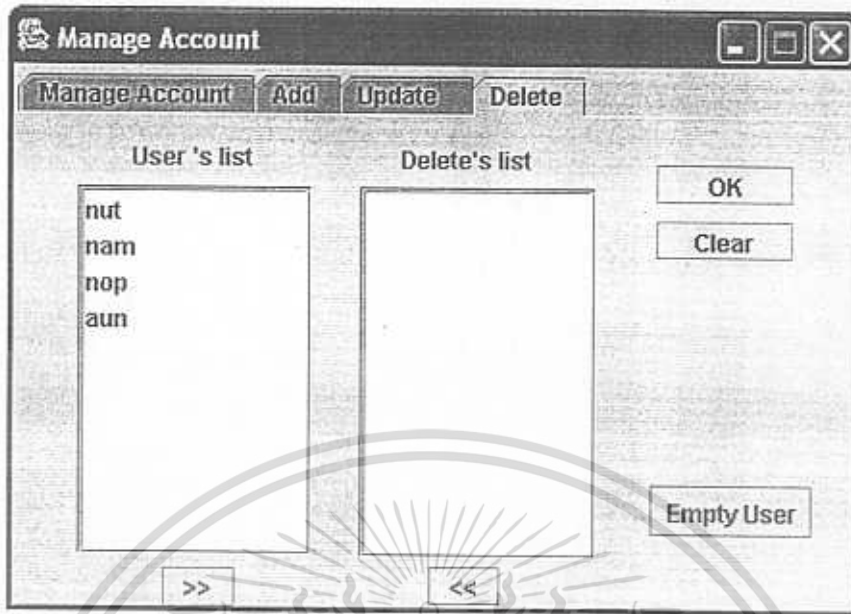


รูปที่ 3.40 หน้าที่ใช้ในการเพิ่มข้อมูลใหม่เข้าสู่ระบบ

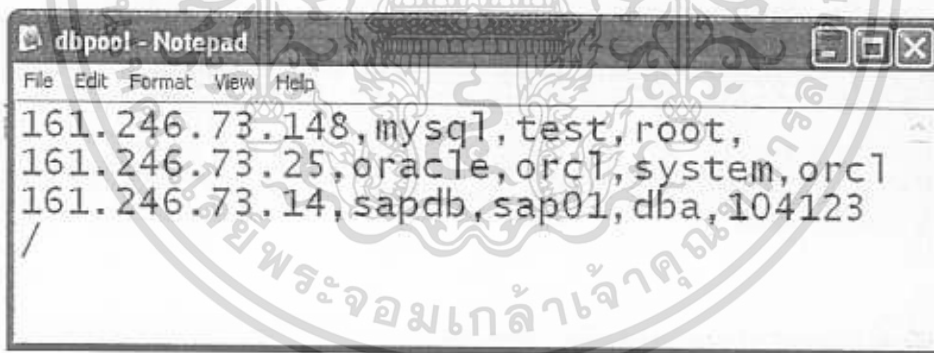


รูปที่ 3.41 หน้าที่ใช้ในการเปลี่ยนแปลงรหัสผ่านของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

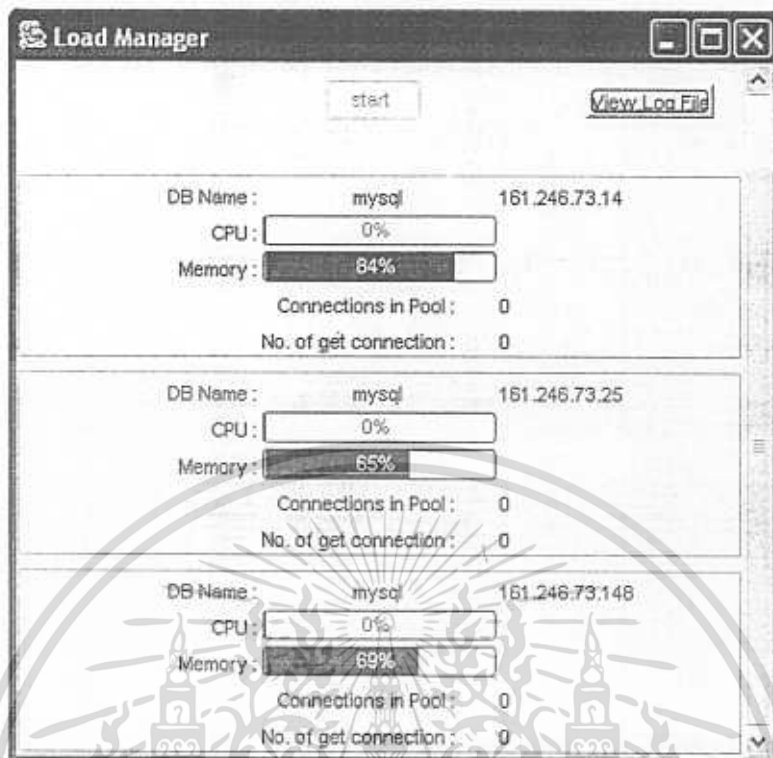


รูปที่ 3.42 หน้าที่ใช้ในการลบชื่อผู้ใช้ออกจากระบบ



รูปที่ 3.43 หน้าที่ใช้ในการกำหนดค่าให้กับเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.44 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด
กรณีที่ใช้เซิร์ฟเวอร์ฐานข้อมูลทั้งหมด 3 เครื่อง

The screenshot shows a 'Database Load Balancing's Log' window with a list of log entries. The log includes timestamps, IP addresses, and detailed descriptions of database activities such as pool initialization, connection management, and SQL query execution.

```

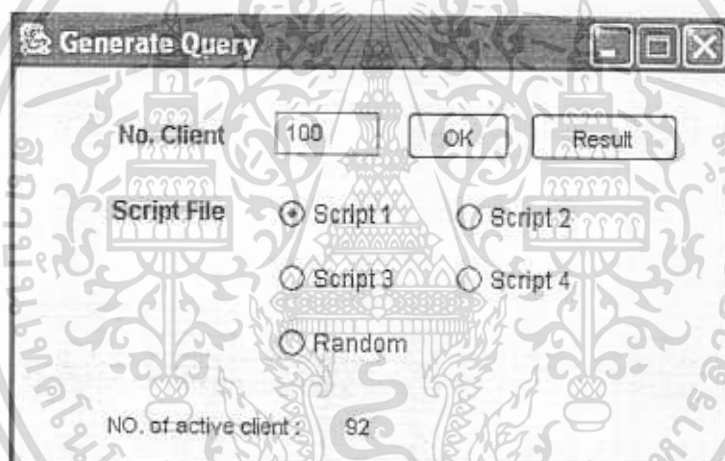
Refresh
Apr 3, 2004 2:57:19 PM (98959) : Sat Apr 03 14:57:19 GMT+07:00 2004 INITIALIZED POOL NO. 0 : mysql
Apr 3, 2004 2:57:19 PM (20) : Sat Apr 03 14:57:19 GMT+07:00 2004 INITIALIZED POOL NO. 1 : mysql
Apr 3, 2004 2:57:19 PM (0) : Sat Apr 03 14:57:19 GMT+07:00 2004 INITIALIZED POOL NO. 2 : mysql
Apr 3, 2004 2:57:37 PM (17515) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:37 PM (31) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:37 PM (180) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:37 PM (0) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:37 PM (110) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:37 PM (0) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:37 PM (511) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:37 PM (90) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:38 PM (90) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:38 PM (0) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:38 PM (461) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:38 PM (0) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:38 PM (70) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:38 PM (20) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:38 PM (140) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:38 PM (0) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
Apr 3, 2004 2:57:38 PM (150) : Executed Command : Select * From tl Where Subject = 'Project2' and Name = 'Aur
Apr 3, 2004 2:57:38 PM (20) : Executed Command : Select * From tl Where Subject = 'Project2' and Name = 'Aur
Apr 3, 2004 2:57:38 PM (20) : Executed Command : Select * From tl Where Subject = 'Project2' and Name = 'Aur
Apr 3, 2004 2:57:38 PM (0) : Executed Command : Select * From tl Where Subject = 'Project2' and Name = 'Aur
Apr 3, 2004 2:57:38 PM (10) : Executed Command : Select * From tl Where Subject = 'Project2' and Name = 'Aur
Apr 3, 2004 2:57:38 PM (150) : Executed Command : Select * From tl Where Subject = 'Project2' and Name = 'Aur
Apr 3, 2004 2:57:38 PM (80) : Executed Command : Select * From tl Where Subject = 'Project2' and Name = 'Aur
Apr 3, 2004 2:57:39 PM (351) : Executed Command : Select Subject_ID,Name From tl Where Age =21 and Subject =
Apr 3, 2004 2:57:39 PM (60) : Executed Command : Select Subject_ID,Name From tl Where Age =21 and Subject =
Apr 3, 2004 2:57:39 PM (40) : Executed Command : Select Subject_ID,Name From tl Where Age =21 and Subject =
Apr 3, 2004 2:57:39 PM (0) : Create new connection : Connection in mysql's pool = 0, Connection 1
Apr 3, 2004 2:57:39 PM (0) : Send Out Connection : Connections in pool no. 0 DB Name mysql = 0, Connect
  
```

รูปที่ 3.45 หน้าที่ใช้สำหรับแสดงข้อมูลการทำงาน (log file)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

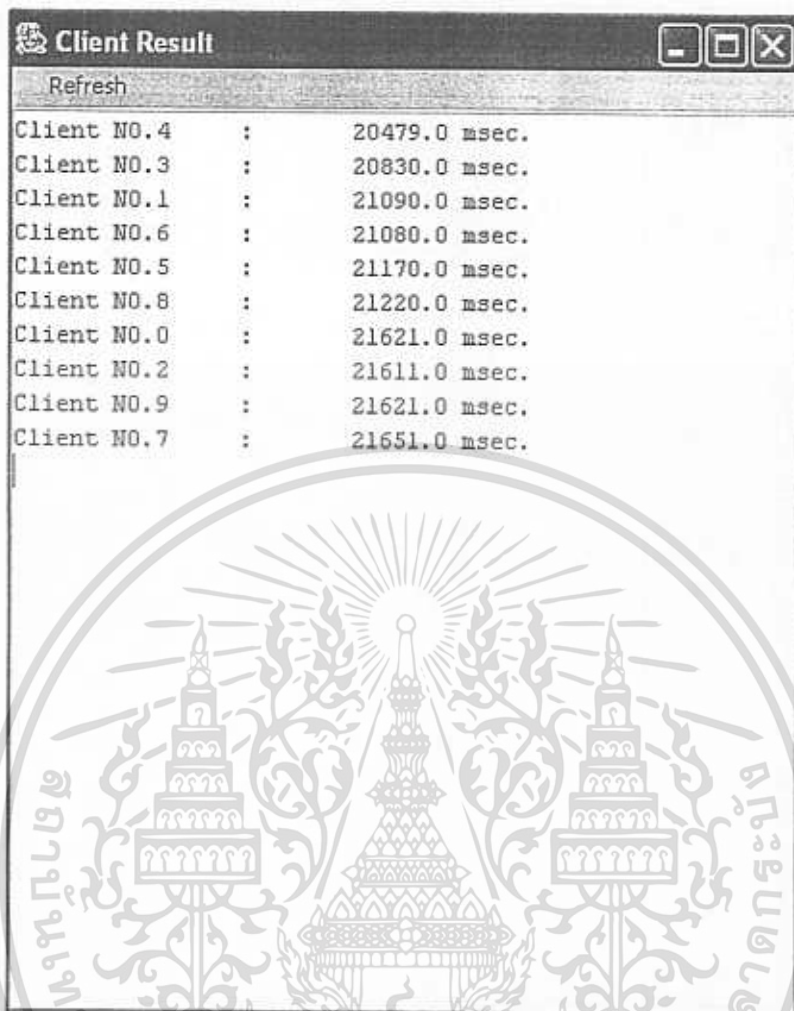


รูปที่ 3.46 หน้าที่ใช้ในการทำการรายงานค่าสถานะมายังเซิร์ฟเวอร์



รูปที่ 3.47 หน้าที่ใช้ในการทดสอบประสิทธิภาพของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



| Refresh | | |
|-------------|---|---------------|
| Client NO.4 | : | 20479.0 msec. |
| Client NO.3 | : | 20830.0 msec. |
| Client NO.1 | : | 21090.0 msec. |
| Client NO.6 | : | 21080.0 msec. |
| Client NO.5 | : | 21170.0 msec. |
| Client NO.8 | : | 21220.0 msec. |
| Client NO.0 | : | 21621.0 msec. |
| Client NO.2 | : | 21611.0 msec. |
| Client NO.9 | : | 21621.0 msec. |
| Client NO.7 | : | 21651.0 msec. |

3.48 หน้าที่ใช้ในการแสดงเวลาที่ใช้ในการประมวลผลของแต่ละไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ฟังก์ชันสำหรับผู้ดูแลระบบ (Administrator and Moderator)

ฟังก์ชันสำหรับผู้ดูแลระบบ จะเป็นฟังก์ชันเพื่อใช้ในการตั้งค่า หรือแก้ไขค่าต่าง ๆ ของระบบ เช่น การเพิ่ม การลบ การแก้ไขข้อมูลของผู้ใช้ในระบบ ซึ่งมีหน้าที่หลัก ๆ ด้วยกัน 3 อย่างคือ

- สำหรับการเพิ่มผู้ใช้ใหม่เข้าไปในระบบ
- สำหรับการลบข้อมูลของผู้ใช้เดิมออกไปจากระบบ
- สำหรับการเปลี่ยนแปลงรหัสผ่าน (password) ของผู้ใช้งานในระบบ

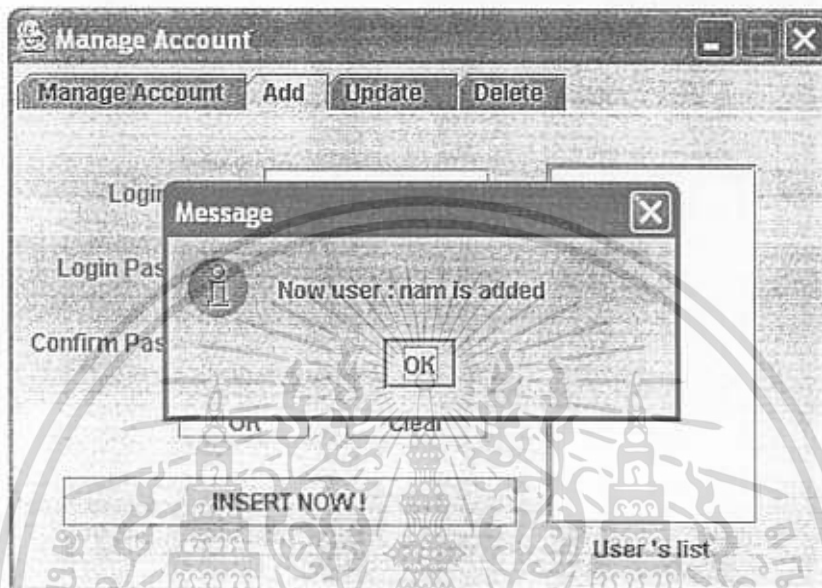
4.1.1 ฟังก์ชันสำหรับเพิ่มผู้ใช้ใหม่เข้าไปในระบบ

รูปที่ 4.1 หน้าที่ใช้ในการเพิ่มชื่อผู้ใช้ใหม่เข้าสู่ระบบ

ผู้ดูแลระบบจะต้องทำการป้อนชื่อผู้ใช้งานใหม่รวมทั้งรหัสผ่าน เพื่อให้ผู้ใช้งานใหม่สามารถเข้าใช้งานระบบฐานข้อมูลได้ ดังรูปที่ 4.1 ซึ่งแบ่งเป็น 3 กรณี คือ

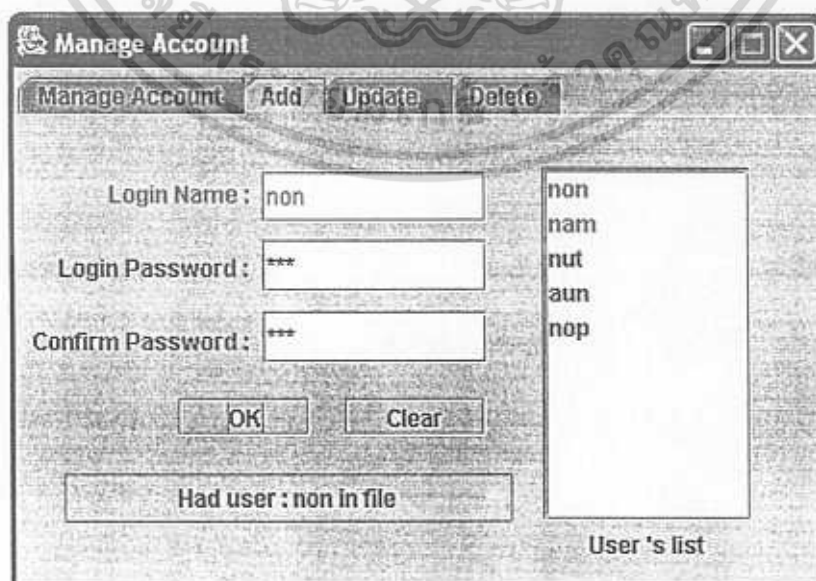
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กรณีที่ผู้ดูแลระบบป้อนชื่อผู้ใช้ที่ยังไม่เคยมีในระบบ และป้อนรหัสผ่าน และยืนยันรหัสผ่าน (confirm password) ตรงกัน ระบบก็จะทำการเพิ่มชื่อผู้ใช้นั้นเข้าไป และมีข้อความเตือนแสดงขึ้น (pop up) ว่าได้เพิ่มชื่อผู้ใช้นั้นเข้าไปแล้ว ดังรูปที่ 4.2



รูปที่ 4.2 หน้าเพิ่มชื่อผู้ใช้ใหม่เข้าสู่ระบบเมื่อผู้ดูแลระบบป้อนข้อมูลอย่างถูกต้อง

2. กรณีที่ผู้ดูแลระบบได้ทำการป้อนชื่อผู้ใช้ที่มีอยู่แล้วในระบบ ระบบจะแสดงข้อความว่ามีชื่อผู้ใช้นั้นใช้งานอยู่ในระบบอยู่แล้ว ดังรูปที่ 4.3



รูปที่ 4.3 หน้าเพิ่มชื่อผู้ใช้ใหม่เข้าสู่ระบบเมื่อผู้ดูแลระบบป้อนข้อมูลชื่อผู้ใช้ที่มีอยู่แล้วในระบบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กรณีที่ผู้ดูแลระบบป้อนรหัสผ่าน และยืนยันรหัสผ่าน ไม่ตรงกัน ระบบจะแสดงข้อความว่ารหัสผ่าน และยืนยันรหัสผ่าน ไม่ตรงกัน ดังรูปที่ 4.4

รูปที่ 4.4 หน้าเพิ่มชื่อผู้ใช้งานใหม่เมื่อผู้ดูแลระบบป้อนรหัสผ่านและยืนยันรหัสผ่าน ไม่ตรงกัน

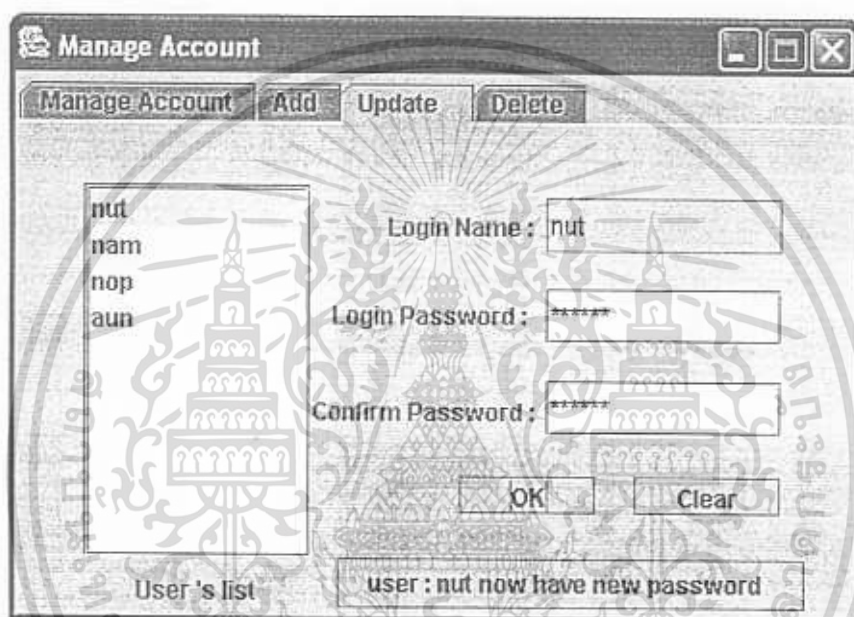
4.1.2 ฟังก์ชันสำหรับเปลี่ยนแปลงรหัสผ่านของผู้ใช้ภายในระบบ

รูปที่ 4.5 หน้าที่ใช้ในการเปลี่ยนแปลงรหัสผ่านของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ดูแลระบบจะต้องทำการเลือกชื่อผู้ใช้ที่ต้องการจะเปลี่ยนแปลงรหัสผ่าน จากนั้นทำการแก้ไขเปลี่ยนแปลงรหัสผ่านของผู้ใช้ที่ต้องการ ดังรูปที่ 4.5 ซึ่งแบบเป็น 2 กรณีคือ

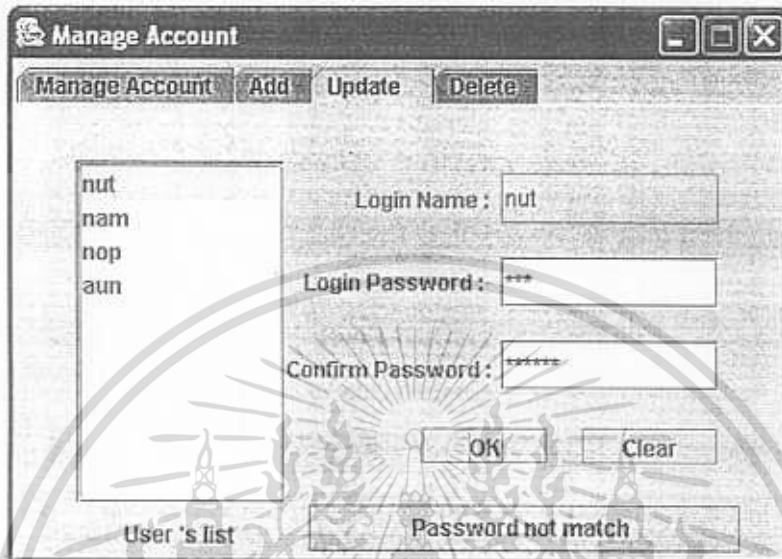
1. กรณีที่ผู้ดูแลระบบป้อนข้อมูลรหัสผ่าน และยืนยันรหัสผ่านตรงกัน ระบบจะทำการเปลี่ยนรหัสผ่านเป็นรหัสผ่านใหม่ จากนั้นก็จะแสดงข้อความว่าได้เปลี่ยนรหัสผ่านให้ผู้ใช้คนดังกล่าวเรียบร้อยแล้ว ดังรูปที่ 4.6



รูปที่ 4.6 หน้าเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในกรณีที่รหัสผ่านและยืนยันรหัสผ่านตรงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

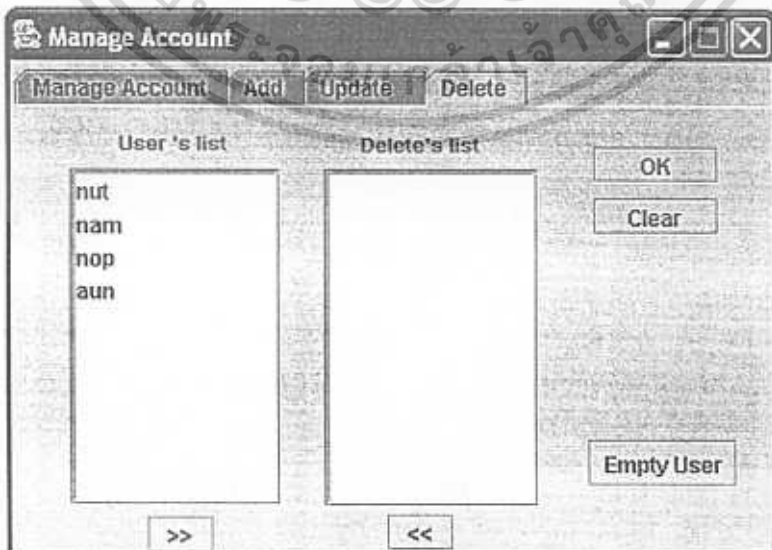
2. กรณีที่ผู้ดูแลระบบป้อนรหัสผ่าน และยืนยันรหัสผ่านไม่ตรงกัน ระบบจะทำการแสดงข้อความเตือนว่ารหัสผ่าน และยืนยันรหัสผ่านไม่ตรงกัน ดังรูปที่ 4.7



รูปที่ 4.7 หน้าเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในกรณีที่รหัสผ่านและยืนยันรหัสผ่านไม่ตรงกัน



4.1.3 ฟังก์ชันสำหรับลบชื่อผู้ใช้ภายในระบบ

ผู้ดูแลระบบจะทำการลบรายชื่อผู้ใช้ที่ไม่ต้องการออกจากระบบได้ โดยทำการเลือกจากรายชื่อผู้ใช้ภายในระบบทั้งหมด โดยสามารถทำการลบออกทีละรายชื่อหรือทีละหลายรายชื่อได้ ดังรูปที่ 4.8



รูปที่ 4.8 หน้าที่ใช้ในการลบชื่อผู้ใช้ออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ดูแลระบบจะทำการเลือกชื่อผู้ใช้ที่ต้องการทำการลบออกจากระบบจากรายชื่อผู้ใช้ (User's list) จากนั้นทำการกด  เพื่อทำการเพิ่มชื่อผู้ใช้ที่ต้องการลบเข้าไปในรายชื่อผู้ใช้ที่ต้องการลบ (Delete's list) และสามารถนำรายชื่อออกจากรายชื่อผู้ใช้ที่ต้องการลบโดยการกด  และเมื่อต้องการลบรายชื่อผู้ใช้ที่มีอยู่ในรายชื่อผู้ใช้ที่ต้องการลบออกจากระบบก็ทำการกดปุ่ม โอเค (OK) แต่ถ้าต้องการยกเลิกข้อมูลทั้งหมดที่มีอยู่ในรายชื่อผู้ใช้ที่ต้องการจะลบก็สามารถกดปุ่ม เคลียร์ (clear) ได้ ส่วนในกรณีที่ต้องการลบรายชื่อผู้ใช้ทั้งหมดที่มีอยู่ก็สามารถกดปุ่ม เอ็มตี้ยูสเซอร์ (Empty User) ได้

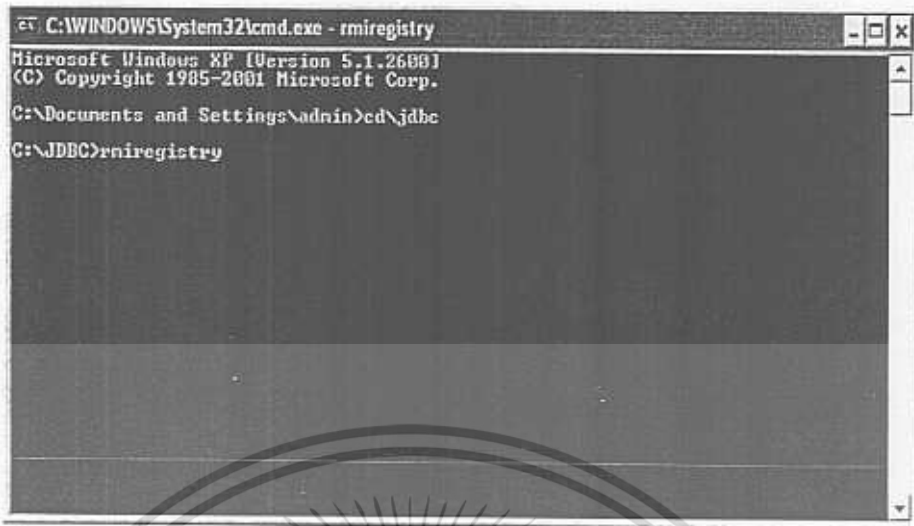
4.2 ฟังก์ชันสำหรับการทำงานของระบบ

ฟังก์ชันสำหรับการทำงานของระบบจะเป็นฟังก์ชันที่เกี่ยวกับการลงทะเบียนเพื่อรองรับการติดต่อจากระยะไกลจากเครื่อง โคลเอนต์ การกำหนดค่าเซิร์ฟเวอร์ฐานข้อมูลที่ใช้งานให้กับเซิร์ฟเวอร์ การเริ่มต้นการใช้งานของเซิร์ฟเวอร์และการแสดงค่าสถานะของเซิร์ฟเวอร์ฐานข้อมูล การรายงานค่าสถานะมายังเซิร์ฟเวอร์ และการทดสอบประสิทธิภาพของระบบ

4.2.1 ฟังก์ชันในการลงทะเบียนเพื่อรองรับการติดต่อระยะไกลจากเครื่อง โคลเอนต์ (RMI Registry)

อาร์เอ็มไอเป็นชุดคำสั่งภาษาจาวาที่ทำให้โปรแกรมภาษาจาวาค้างแค่สอง โปรแกรมขึ้นไปสามารถทำงานร่วมกันได้บนระบบเครือข่าย โดยการนี้จะทำให้โปรแกรมภาษาจาวาสอง โปรแกรมทำงานร่วมกันได้นั้น ต้องทำให้อยู่ในรูปแบบที่โปรแกรมที่ทำหน้าที่เป็น โคลเอนต์เรียกเมธอดของ โปรแกรมที่ทำหน้าที่เป็นเซิร์ฟเวอร์เสมอ ดังนั้นเราจะทำการรีจิสเตอร์อาร์เอ็มไอ ที่เซิร์ฟเวอร์เพื่อให้ โคลเอนต์สามารถเรียกเมธอดของเซิร์ฟเวอร์ได้ ดังรูปที่

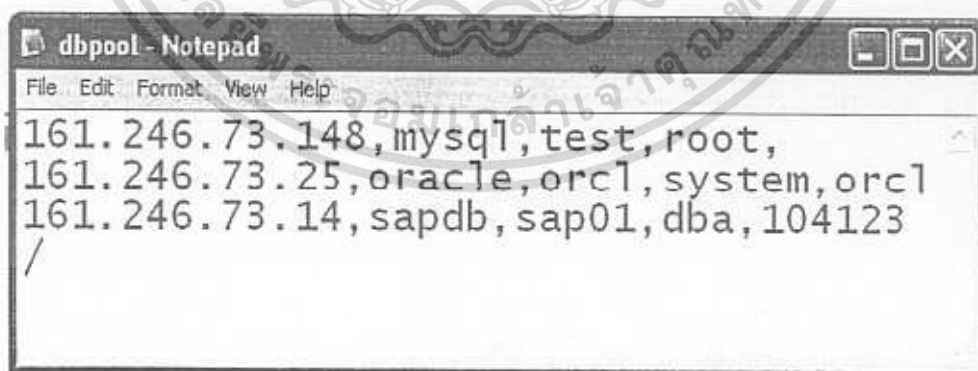
4.9



รูปที่ 4.9 หน้าที่ใช้ในการลงทะเบียนเพื่อรองรับการติดต่อระยะไกลจากเครื่องไคลเอนต์

4.2.2 ฟังก์ชันในการกำหนดค่าเซิร์ฟเวอร์ฐานข้อมูลที่ใช้งานให้กับเซิร์ฟเวอร์

เป็นการกำหนดว่ามีเซิร์ฟเวอร์ฐานข้อมูลเครื่องใดบ้างที่ใช้งานกับระบบกระจาย โหลดฐานข้อมูลซึ่งจะเป็นการกำหนดให้กับเซิร์ฟเวอร์ โดยเซิร์ฟเวอร์จะทำการอ่านค่ามาจากไฟล์ ข้อมูลที่อยู่ในไฟล์จะต้องกำหนดอยู่ในรูปแบบ หมายเลข IP, ชื่อซีบีเอ็มเอสที่ใช้, ชื่อฐานข้อมูล, ชื่อผู้ใช้ที่ล็อกอินเข้าฐานข้อมูล, รหัสผ่าน โดยที่มีเครื่องหมาย “/” เป็นตัวบอจุดสิ้นสุดของไฟล์ ดังรูปที่ 4.10

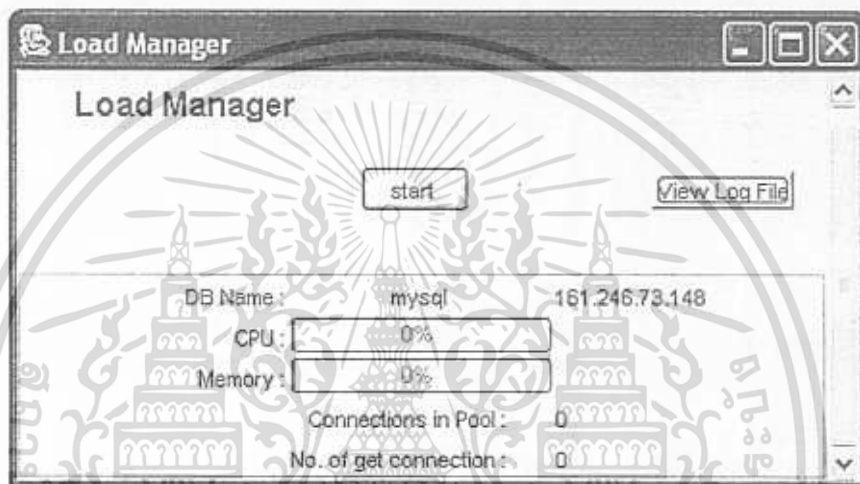


รูปที่ 4.10 หน้าที่ใช้ในการกำหนดค่าให้กับเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

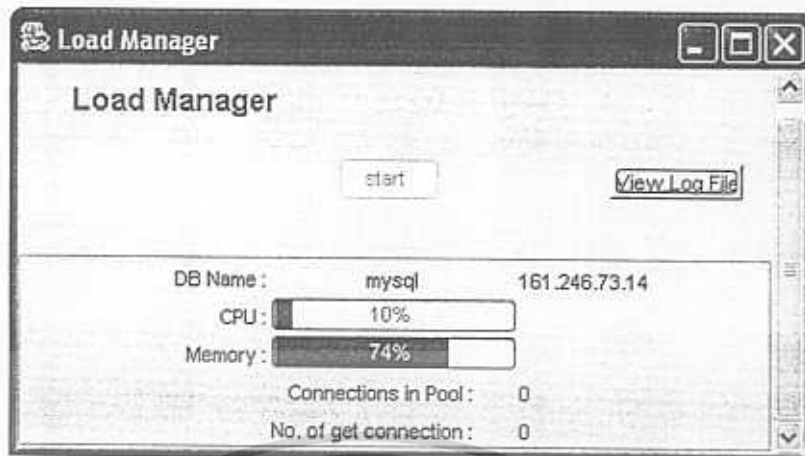
4.2.3 ฟังก์ชันในการเริ่มต้นการใช้งานเซิร์ฟเวอร์ (Start Load Balance Server) และการแสดงค่าสถานะของเซิร์ฟเวอร์ฐานข้อมูล

ระบบจะเริ่มต้นการทำงานเมื่อมีการกดปุ่มสตาร์ทซึ่งจะทำให้เซิร์ฟเวอร์เริ่มการทำงานขึ้น โดยจะรอรับการรายงานค่ามาจากเอเจนต์ (Agent) (เอเจนต์เป็นคอมพิวเตอร์ที่ฝังตัวอยู่ในเครื่องเซิร์ฟเวอร์ฐานข้อมูลแต่ละตัว) ซึ่งเอเจนต์ จะทำการรายงานค่าเปอร์เซ็นต์ของซีพียูและหน่วยความจำที่ใช้ไปมาให้ยังเซิร์ฟเวอร์ ดังรูปที่ 4.11

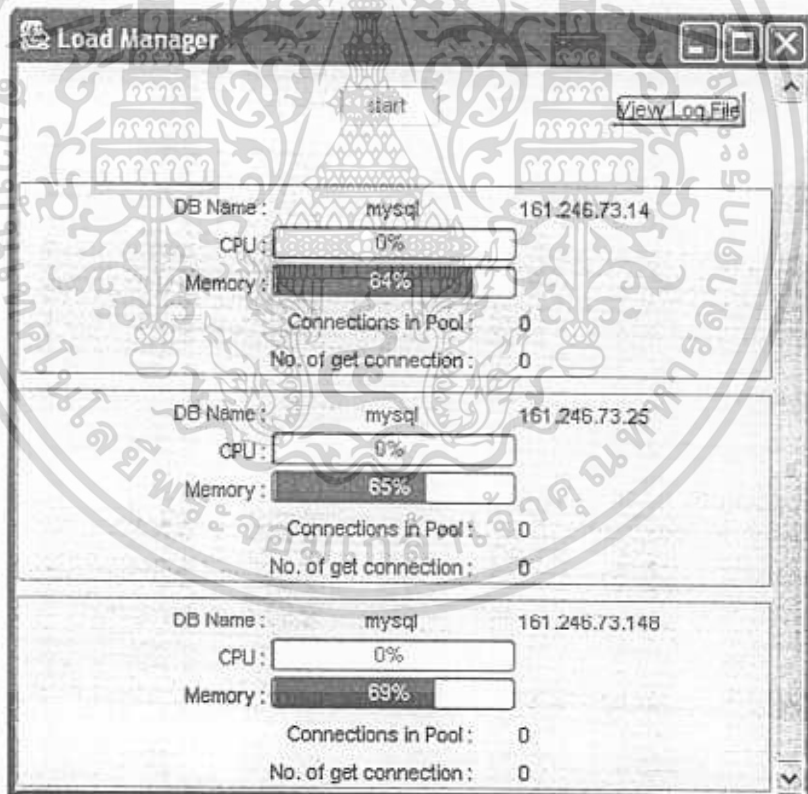


รูปที่ 4.11 หน้าที่ใช้ในการเริ่มต้นการใช้งานเซิร์ฟเวอร์

หลังจากที่ได้ทำการเริ่มต้นการใช้งานเซิร์ฟเวอร์ และส่วนของเอเจนต์ ได้มีการรายงานค่าส่งมายังเซิร์ฟเวอร์แล้ว เซิร์ฟเวอร์จะทำการตรวจสอบและแสดงผลภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด รวมทั้งรอรับการเชื่อมต่อที่จะมีมาจากส่วนของผู้ใช้ด้วย เมื่อผู้ใช้งานมีการเรียกใช้งานเซิร์ฟเวอร์ฐานข้อมูล เซิร์ฟเวอร์จะทำการตรวจสอบว่าในขณะนั้นเซิร์ฟเวอร์ฐานข้อมูลเครื่องใดที่มีภาระงานน้อยที่สุดและตัดสินใจเลือกเครื่องที่มีภาระงานน้อยที่สุด จากนั้นก็จะส่งคำสั่งไปให้เซิร์ฟเวอร์ฐานข้อมูลเครื่องนั้นทำการประมวลผล โดยจะทำการบันทึกข้อมูลการทำงานไว้ในล็อกไฟล์ด้วย เพื่อทำการแสดงและเก็บสถิติการใช้งาน ส่วนปุ่มแสดงล็อกไฟล์มีไว้สำหรับแสดงข้อมูลการทำงานที่ถูกเก็บไว้ในล็อกไฟล์

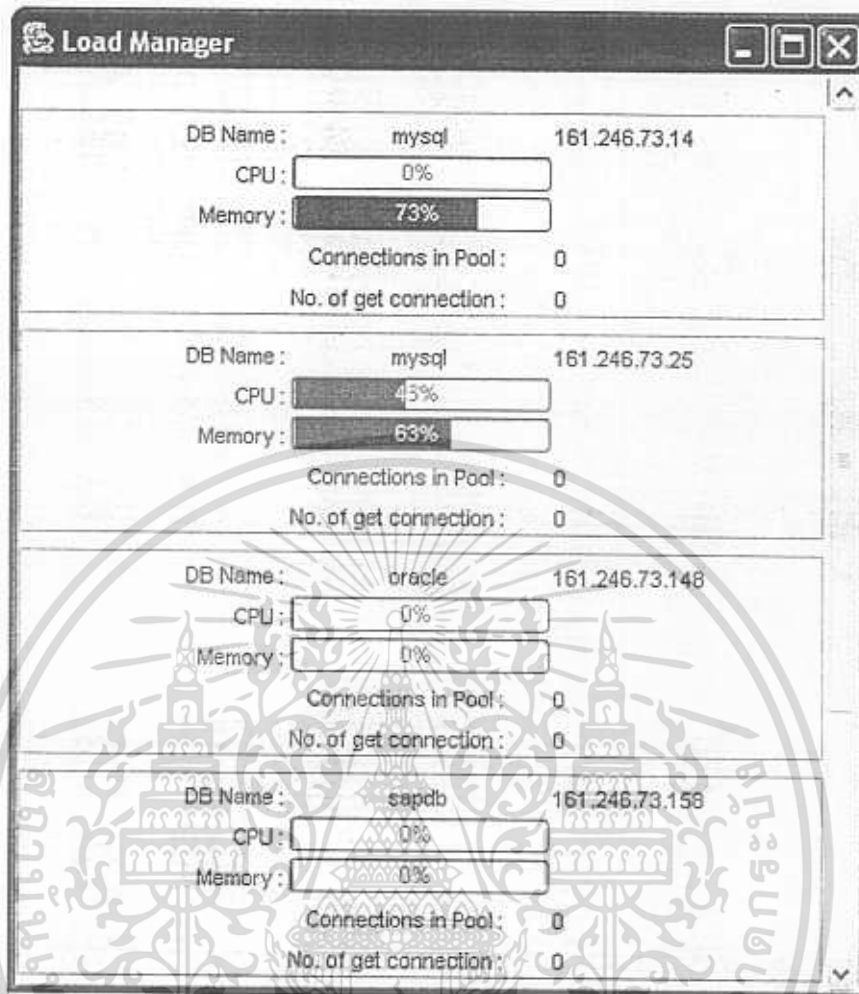


รูปที่ 4.12 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด
กรณีที่ใช้เซิร์ฟเวอร์ฐานข้อมูลเพียงเครื่องเดียว



รูปที่ 4.13 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด
กรณีที่ใช้เซิร์ฟเวอร์ฐานข้อมูลทั้งหมด 3 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



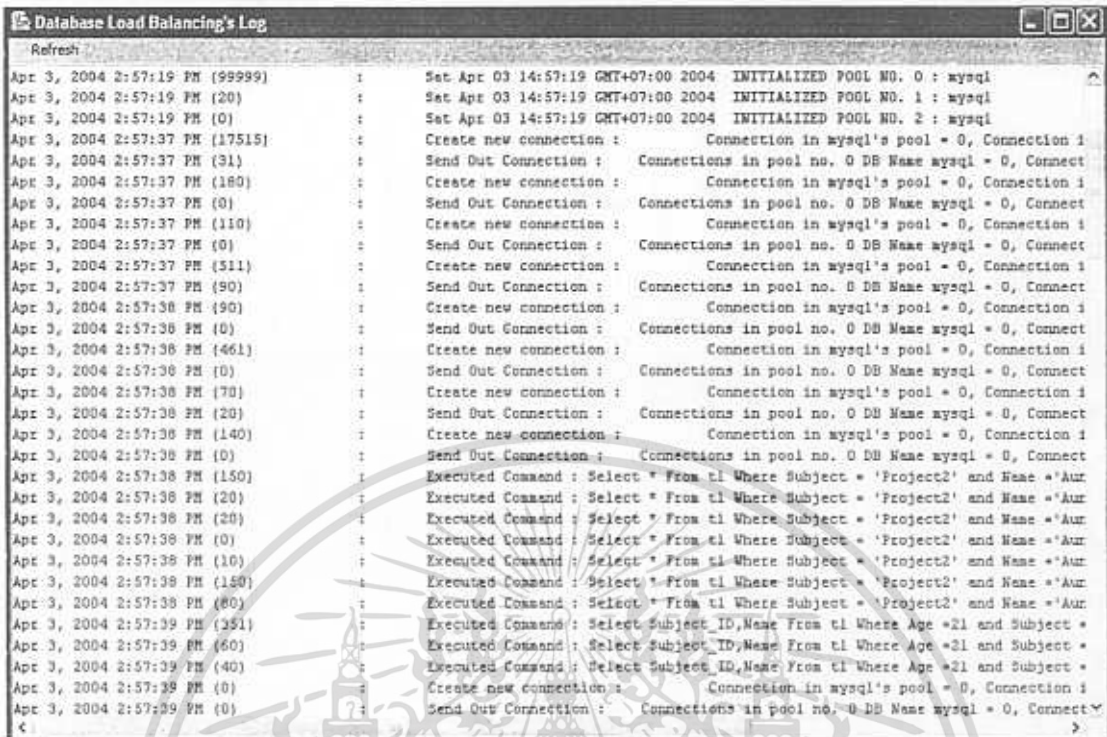
รูปที่ 4.14 หน้าที่ใช้แสดงค่าภาระงานของเซิร์ฟเวอร์ฐานข้อมูลทั้งหมด
กรณีที่ใช้เซิร์ฟเวอร์ฐานข้อมูล 6 เครื่อง

จากรูป จะเห็นได้ว่าระบบของเราสามารถใช้งานกับฐานข้อมูลได้หลากหลาย
อย่างเช่น Oracle, MySQL, SAP DB และสามารถเพิ่มจำนวนของเครื่องเซิร์ฟเวอร์
ฐานข้อมูลได้ อย่างเช่นในรูปที่ 4.12, 4.13 และ 4.14 มีการใช้งานเครื่องเซิร์ฟเวอร์
ฐานข้อมูลจำนวน 1 เครื่อง, 3 เครื่อง และ 6 เครื่องตามลำดับ

เราสามารถดูข้อมูลการทำงานได้โดยการกดปุ่มวิวล็อกไฟล์ (view log file)
จากนั้นมันก็จะแสดงหน้าของล็อกไฟล์ซึ่งเก็บบันทึกข้อมูลในการทำงานขึ้นมา ดังรูปที่

4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 หน้าที่ใช้สำหรับแสดงข้อมูลการทำงาน

4.2.4 ฟังก์ชันในการรายงานค่าสถานะมายังเซิร์ฟเวอร์

เอเจนต์จะทำการรายงานค่าเปอร์เซ็นต์การใช้งานของซีพียูและหน่วยความจำของเครื่องเซิร์ฟเวอร์ฐานข้อมูลส่งไปให้เซิร์ฟเวอร์ โดยที่เราจะทำการระบุหมายเลขไอพีของเครื่องเซิร์ฟเวอร์ และกลุ่มสตาร์ท ดังรูปที่ 4.16 ซึ่งส่วนของเอเจนต์จะต้องมีอยู่ในเครื่องเซิร์ฟเวอร์ฐานข้อมูลทุกเครื่องเพื่อให้สามารถรายงานค่าสถานะมายังเซิร์ฟเวอร์ได้

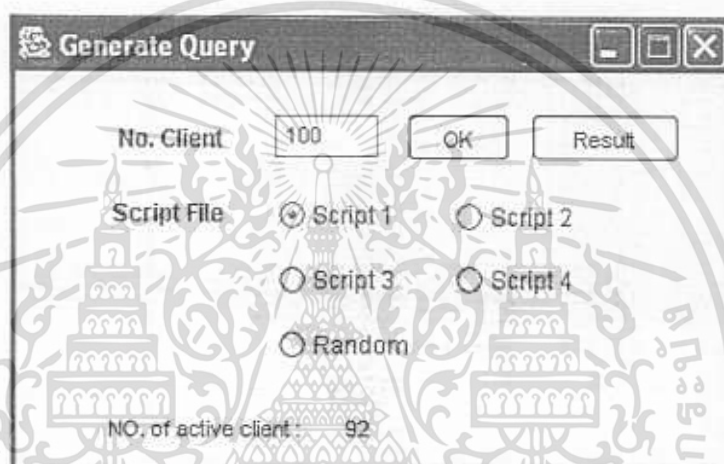


รูปที่ 4.16 หน้าที่ใช้ในการทำการรายงานค่าสถานะมายังเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 ฟังก์ชันในการทดสอบประสิทธิภาพของระบบ

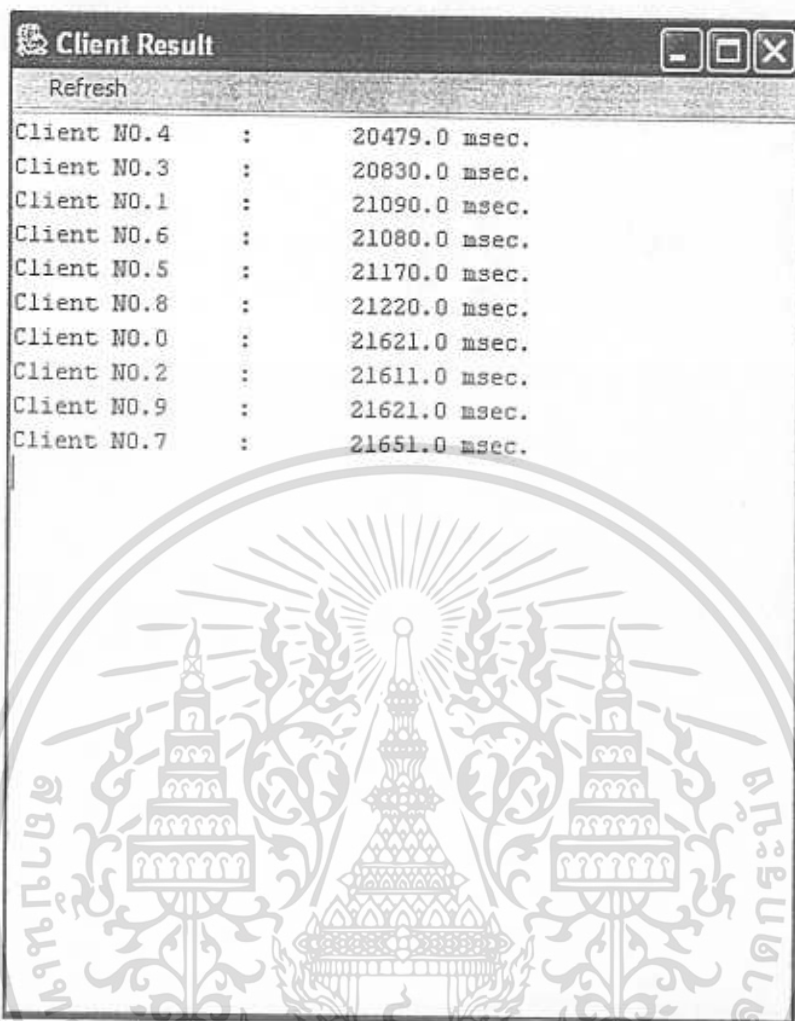
ในการใช้งานโดยปกติแล้วผู้ใช้สามารถเรียกใช้งานมายังฐานข้อมูลพร้อมๆ กันได้ โดยหากมีการเรียกใช้งานฐานข้อมูลหนึ่งจากผู้ใช้งานจำนวนมากพร้อมกันจะส่งผลให้ประสิทธิภาพในการทำงานของเซิร์ฟเวอร์ฐานข้อมูลลดน้อยลง เราจึงได้ทำการพัฒนาระบบกระจายโหลดฐานข้อมูลขึ้นมาเพื่อแก้ปัญหาดังกล่าว ซึ่งในการทดสอบเพื่อเปรียบเทียบประสิทธิภาพของระบบ เราจะทำการจำลองเหตุการณ์ในการใช้งานจริงขึ้นมาตามขั้นตอนดังนี้



รูปที่ 4.17 หน้าที่ใช้ในการทดสอบประสิทธิภาพของระบบ

ทำการระบุจำนวน โคลนเอนต์ที่ต้องการคิดต่อใช้งานกับฐานข้อมูล จากนั้นก็จะทำการเลือกสคริปต์ไฟล์ (Script File) ซึ่งก็คือคำสั่งที่ใช้กับฐานข้อมูลโดยแต่ละสคริปต์ก็จะมีคำสั่งที่แตกต่างกันในที่นี้มี สคริปต์การทำงานอยู่ 4 สคริปต์ หรือเราอาจเลือกแบบสุ่มสคริปต์ (Random) ก็ได้ หลังจากที่เราระบุจำนวนโคลนเอนต์และกำหนดสคริปต์เรียบร้อยแล้ว จากนั้นกดปุ่ม OK มันก็จะทำการประมวลผลคำสั่ง และแสดงค่าเวลาที่ใช้ในการประมวลผลโดยกดปุ่มรีซัลต์ (result) จะได้ค่าเวลาที่ใช้ในการประมวลผลออกมาดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



| Refresh | | |
|-------------|---|---------------|
| Client NO.4 | : | 20479.0 msec. |
| Client NO.3 | : | 20830.0 msec. |
| Client NO.1 | : | 21090.0 msec. |
| Client NO.6 | : | 21080.0 msec. |
| Client NO.5 | : | 21170.0 msec. |
| Client NO.8 | : | 21220.0 msec. |
| Client NO.0 | : | 21621.0 msec. |
| Client NO.2 | : | 21611.0 msec. |
| Client NO.9 | : | 21621.0 msec. |
| Client NO.7 | : | 21651.0 msec. |

รูปที่ 4.18 หน้าที่ใช้ในการแสดงเวลาที่ใช้ในการประมวลผลของแต่ละไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

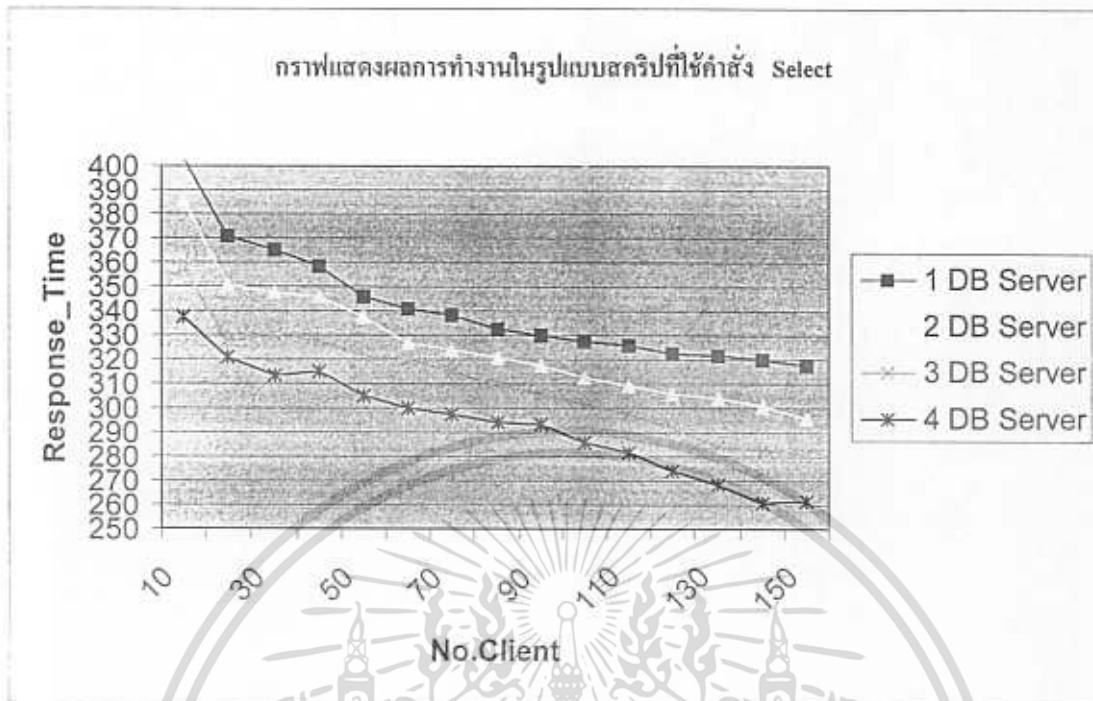
ระบบกระจายโหนดฐานข้อมูลนี้ แบ่งฟังก์ชันการทำงานออกเป็น 3 ส่วน คือ ส่วนทดสอบระบบ, ส่วนระบบกระจายโหนด และส่วนการรายงานผลสถานะจากฐานข้อมูล โดยฟังก์ชันในส่วนทดสอบระบบนั้นสามารถสร้างการเชื่อมต่อเพื่อจำลองเหตุการณ์ได้เป็นอย่างดี ครอบคลุมตามที่ได้ออกแบบไว้ และส่วนของระบบการกระจายโหนดนั้นสามารถทำให้ผู้ดูแลระบบตั้งค่าเริ่มต้นต่าง ๆ รวมถึงสร้างบัญชีรายชื่อผู้ใช้บนระบบกระจายโหนดฐานข้อมูลได้ และในส่วนของการรายงานผลสถานะจากฐานข้อมูล ก็สามารถคำนวณสถานะโหนดและรายงานมายังระบบกระจายโหนดได้อย่างมีประสิทธิภาพ และนอกจากนี้เรายังได้ทำการเก็บสถิติการทำงานของระบบกระจายโหนดฐานข้อมูลในรูปแบบต่าง ๆ ไว้ดังนี้

จากรูปที่ 5.1 คือสคริปต์ไฟล์ที่มีการทำงานในรูปแบบคำสั่งซีเลค (Select) หลังจากที่เราทำการทดสอบประสิทธิภาพของระบบกระจายโหนดฐานข้อมูลโดยการเลือกการทดสอบไปที่สคริปต์ไฟล์นี้ ประสิทธิภาพการทำงานของระบบสามารถพิจารณาได้จากรูปที่ 5.2 ซึ่งจะแสดงประสิทธิภาพการทำงานในกรณีที่ผู้ใช้เวิร์กเวอร์ฐานข้อมูล 1, 2, 3 และ 4 เครื่องตามลำดับ

```
Select * From t1 Where Subject = 'Project2' and Name = 'Aun' and Singer = 'Blue'
Select Subject_ID, Name From t1 Where Age = 21 and Subject = 'Internet System' and Sang = 'Better Man'
Select * From t2 Where Food = 'Hamburger' and Fruit = 'Rambutan'
Select Food From t2 Where Price Between 45 and 65 and Fruit = 'Durian' and Animal = 'Ant'
Select * From t3 Where Country = 'Thailand' and Thing = 'Pen' and Occupation = 'Engineer'
Select Distinct Country From t3 Where City in ('Bangkok', 'Taipei')
Select * From t4 Where Project = 'DB Load Balance' and Number in ('One') and Province = 'Chonburi' and Name = 'Aun'
Select Project, Name From t4 Where Color = 'Red' and Number = 'Two' and Province = 'Phuket'
Select * From t5 Where Name = 'Somsri' and Model = 'City' and Sport = 'Tennis' and Movie = 'Notting Hill'
Select Distinct Model, Price From t5 Where Brand = 'Honda' Order by Price desc
```

รูปที่ 5.1 หน้าที่ใช้แสดงสคริปต์ไฟล์ที่มีการทำงานในรูปแบบคำสั่งซีเลค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 หน้าที่ใช้แสดงประสิทธิภาพการทำงานของระบบกระจายโหนดฐานข้อมูลในกรณีที่มีการทำงานในรูปแบบคำสั่งซีเล็ค

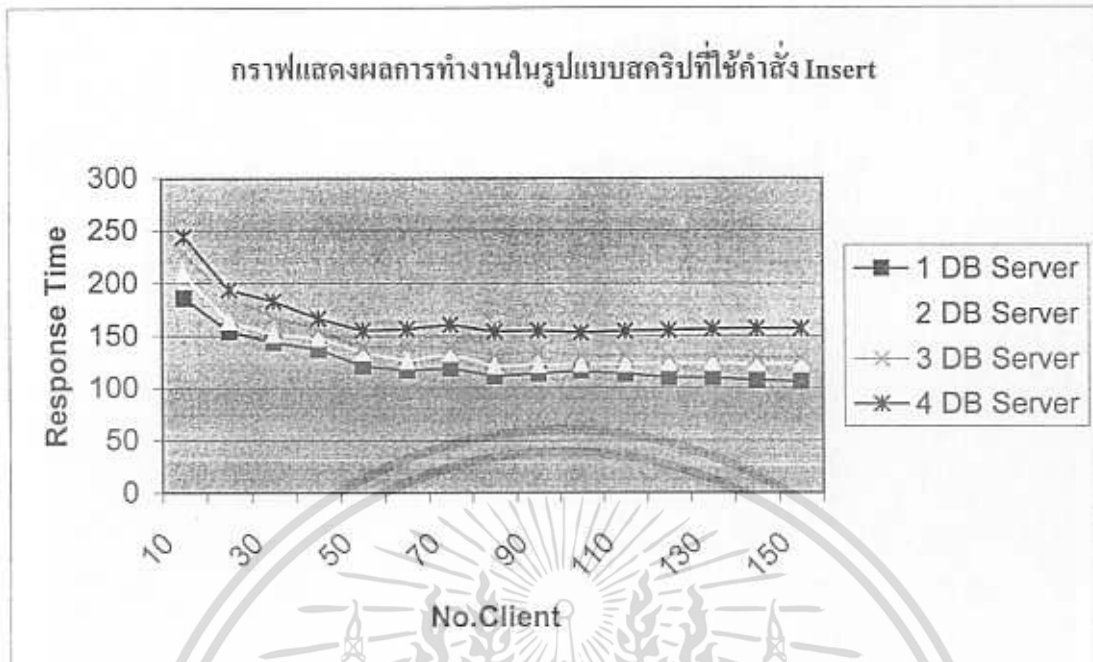
จากรูปที่ 5.3 คือสคริปไฟล์ที่มีการทำงานในรูปแบบคำสั่งอินเซิร์ต (Insert) หลังจากที่เราทำการทดสอบประสิทธิภาพของระบบกระจายโหนดฐานข้อมูล โดยการเลือกการทดสอบไปที่สคริปไฟล์นี้ ประสิทธิภาพการทำงานของระบบสามารถพิจารณาได้จากรูปที่ 5.4 ซึ่งจะแสดงประสิทธิภาพการทำงานในกรณีที่ใช้เซิร์ฟเวอร์ฐานข้อมูล 1, 2, 3 และ 4 เครื่องตามลำดับ

```

Insert - Notepad
File Edit Format View Help
insert into t1 values ('11111111','Insert1',1,'11111111','AAA',1,'AAAAA','AAAAAAA')
insert into t1 values ('22222222','Insert2',2,'22222222','BBB',2,'BBBBB','BBBBBBB')
insert into t2 values ('Insert3',3,'CCC','CCCCC')
insert into t2 values ('Insert4',4,'DDD','DDDDD')
insert into t3 values ('Insert5','EEE','EEE','EEE','EEE','EEE',5,'EEE',5)
insert into t3 values ('Insert6','FFF','FFF','FFF','FFF','FFF',6,'FFF',6)
insert into t4 values ('Insert7','GGG','GGGGG','GGG','GGGGG',7,'GGG')
insert into t4 values ('Insert8','HHH','HHHHH','HHH','HHHHH',8,'HHH')
insert into t5 values ('009','Insert9','III','III','III',9,'IIII','III','IIII')
insert into t5 values ('010','Insert10','JJJ','JJJ','JJJ',10,'JJJJ','JJJ','JJJJ')
  
```

รูปที่ 5.3 หน้าที่ใช้แสดงสคริปไฟล์ที่มีการทำงานในรูปแบบคำสั่งอินเซิร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 หน้าที่ใช้แสดงประสิทธิภาพการใช้งานระบบกระจายโหนดฐานข้อมูล
ในกรณีที่มีการทำงานในรูปแบบคำสั่งอินเจิร์ต

จากรูปที่ 5.5 คือสคริปไฟล์ที่มีการทำงานในรูปแบบคำสั่งจอย (Join) หลังจากที่เรทำการทดสอบประสิทธิภาพของระบบกระจายโหนดฐานข้อมูล โดยการเลือกการทดสอบไปที่ สคริปไฟล์นี้ ประสิทธิภาพการทำงานของระบบสามารถพิจารณาได้จากรูปที่ 5.6 ซึ่งจะแสดงประสิทธิภาพการทำงานในกรณีที่ใช้เซิร์ฟเวอร์ฐานข้อมูล 1, 2, 3 และ 4 เครื่องตามลำดับ

```

Join - Notepad
File Edit Format View Help
select Distinct t1.Subject from t1,t4 where t1.name = t4.name and singer = 'Dido' and units in (2,3) ;
select Distinct t3.occupation,t4.project from t3,t4 where t3.province = t4.province and country in ('Thailand','Japan') ;
select Distinct singer from t1,t4 where (t1.name = 'Aun' and t4.name = 'Aun') and project = 'DB Load Balance' and
subject = 'Project2' and Region = 'South' and Song = 'So Yesterday' and Subject='Project2';
select Distinct t3.place,t4.project from t3,t4 where t3.province = t4.province and t4.region = 'North' and
Currency = 'Yen' and Thing = 'Pen' and Occupation = 'Engineer' and Color = 'Blue';
select * from t3,t4 where (t3.city = 'Bangkok' and t4.province = 'Bangkok' ) and color = 'red' and Number = 'Three' and
Salary = 20000 and Place = 'Wat Pho' and Thing = 'Pencil';

```

รูปที่ 5.5 หน้าที่ใช้แสดงสคริปไฟล์ที่มีการทำงานในรูปแบบคำสั่งจอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 หน้าที่ใช้แสดงประสิทธิภาพการใช้งานระบบกระจายโหนดฐานข้อมูล
ในกรณีที่มีการทำงานในรูปแบบคำสั่ง Join

สำหรับการทำงานของระบบกระจายโหนดฐานข้อมูลนี้ สามารถให้บริการการเชื่อมต่อฐานข้อมูลได้อย่างถูกต้อง ซึ่งหลังจากที่ได้ศึกษาและทำโครงการนี้แล้วนั้น ทำให้มีความเข้าใจในสถาปัตยกรรมเจดีบีซี และสถาปัตยกรรมอาร์เอ็มไอเป็นอย่างดี รวมทั้งมีความเข้าใจหลักการทำงานและการออกแบบระบบที่มีการกระจายโหนดได้เป็นอย่างดี และสามารถนำมาใช้พัฒนาระบบอื่น ๆ ได้ต่อไปในอนาคต

5.2 ปัญหาที่เกิดขึ้นระหว่างการทดลอง

- ความเข้าใจในภาษาจาวา ซึ่งเป็นเรื่องใหม่ที่จะต้องศึกษา ทำให้ต้องใช้เวลาในการศึกษาและค้นคว้าข้อมูล
- ความเข้าใจในการออกแบบเชิงวัตถุ ซึ่งเป็นเรื่องใหม่ทำให้ต้องใช้เวลาในการทำ ความเข้าใจก่อนที่จะเริ่มทำการออกแบบจริง
- จำนวนเครื่องคอมพิวเตอร์ เนื่องจากเป็นระบบที่ทำการกระจายโหนด จึงจำเป็นต้องใช้เครื่องคอมพิวเตอร์มากกว่าระบบโดยทั่วไป
- ระบบฐานข้อมูลแต่ละชนิด มีขั้นตอนในการติดตั้งและใช้งานที่ต่างกัน ทำให้ต้องใช้เวลาในการศึกษาการใช้งานฐานข้อมูลแต่ละชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สถาปัตยกรรมเจคิบีซี และอาร์เอ็มไอ เป็นสถาปัตยกรรมที่มีความซับซ้อน ซึ่งยากแก่การทำความเข้าใจในระยะเวลาอันสั้น
- คอมพิวเตอร์ที่ใช้ค่อนข้างเก่า จึงทำให้ประสิทธิภาพการใช้งานของระบบไม่สามารถใช้ได้อย่างเต็มที่

5.3 แนวทางการพัฒนาโครงการ

- ทำการเก็บสถิติการใช้งาน แล้วพัฒนารูปแบบการกระจายโหนดให้มีประสิทธิภาพมากยิ่งขึ้น
- ออกแบบและพัฒนาส่วนของการใช้งานของผู้ใช้ให้เป็นระบบที่ใช้งานจริง
- พัฒนาให้สามารถรองรับใช้งานฐานข้อมูลได้หลากหลายชนิดยิ่งขึ้น
- พัฒนาให้รองรับการใช้งานระบบฐานข้อมูลขั้นสูง
- พัฒนาให้รองรับการจัดกลุ่ม หรือ ระดับของผู้ใช้ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Dr. Harvey M. Deitel and Paul J. Deitel, Java How to Program FIFTH EDITION, Deitel & Associates, Inc., 2003.
2. ดร. วีระศักดิ์ ชิงदार, JAVA PROGRAMMING Volume I, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), พ.ศ. 2543.
3. ดร. วีระศักดิ์ ชิงदार, JAVA PROGRAMMING Volume II, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), พ.ศ. 2543.
4. สมนึก เข็มเจริญเดช, เจาะลึก Oracle9i รหัส 2, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), พ.ศ. 2546.
5. สุกัญญา เบญจวนิช, คู่มืออ้างอิงชุดคำสั่ง MySQL, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), พ.ศ. 2545.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้