

การพัฒนาโปรแกรม การบริหารและจัดการฐานข้อมูลออร์ราเคิล
ผ่านระบบอินเทอร์เน็ต

The Development of an Internet-based DBA Tool for Oracle



H001688

โดย

นายประดิษฐ์ วัฒนสุวรรณ

รหัส 41067141

วัน เดือน ปี	25	ส.ค.	2549
เลขทะเบียน	01688		
เลขเรียกหนังสือ	วท.	ว ๒๖๗๗	๒๕๔๙
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."			

อาจารย์ที่ปรึกษา

รศ.ดร.ศุภมิตร จิตตะยโสธร

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาโปรแกรม การบริหารและจัดการฐานข้อมูลออร์ราเคิลผ่านระบบอินเทอร์เน็ต
นักศึกษา	นายประดิษฐ์ วัฒนสุวรรณ
อาจารย์ที่ปรึกษา	รศ.ดร.ศุภมิตร จิตตะยโสธร
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

ปัจจุบันการใช้งานฐานข้อมูล ได้มีบทบาทกับทุกๆองค์กร จึงทำให้เกิดแนวความคิดในการพัฒนาโปรแกรมประยุกต์ที่ใช้ในการแก้ไขปัญหาของฐานข้อมูลที่เกิดขึ้น โดยที่ผู้ดูแลฐานข้อมูล (DBA) ไม่จำเป็นต้องดำเนินการที่เครื่องที่ทำการติดตั้งฐานข้อมูลนั้น ซึ่งจะมีการนำเทคโนโลยีทางด้านอินเทอร์เน็ตมาผสมผสานในการสร้างโปรแกรม เพื่อให้ผู้ดูแลฐานข้อมูล (DBA) สามารถบริหารจัดการแก้ไขปัญหาที่เกิดขึ้นกับฐานข้อมูลผ่านอินเทอร์เน็ตได้ ยังผลให้ผู้ดูแลฐานข้อมูลสามารถบริหารงานได้ง่ายขึ้น การพัฒนาจะใช้ภาษา Java เป็นหลัก เพื่อให้สามารถทำงานได้กับ Web Server ทุกๆ Platform โดยโปรแกรมที่พัฒนาขึ้นจะเชื่อมต่อกับฐานข้อมูลโดยใช้ IP Address เป็นตัวบ่งชี้ที่ตั้งที่อยู่ของฐานข้อมูล คล้ายกับลักษณะของ 3-Tier Model ที่มีการแยกส่วนของ เครื่องที่เป็น Application Server กับ Database Server ออกจากกัน

Title	The Development of an Internet-base DBA Tool for Oracle
Student	Mr.Pradit Wattanasuwan
Advisor	Assoc.Prof.Dr.Suphamit Chittayasothorn
Level of Student	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

Abstract

In present, Database's role with organization are grow. It make idea develop application software for manage and solve problem that occur when database are run. This software help DBA solve problem from anywhere that can access Internet. In development use Java programming because it can work with every platform. This software are use IP Address for connect database. It is look like 3-Tier Model and this software are in 2's Tier

กิตติกรรมประกาศ

ผลงานการศึกษาและพัฒนาโปรแกรม การบริหารและจัดการฐานข้อมูลออร์ราเคิลผ่านระบบอินเทอร์เน็ต สามารถบรรลุวัตถุประสงค์ได้เป็นอย่างดีนั้น ข้าพเจ้าได้รับการให้คำปรึกษาแนะนำ รวมถึงข้อเสนอแนะต่างๆทั้งหมด จนสามารถผลิตผลงานที่ตรงตามแนวทางที่วางไว้ได้เป็นอย่างดี จากท่านอาจารย์ รศ.ดร.ศุภมิตร จิตตะยโสธร ซึ่งท่านเป็นอาจารย์ที่ปรึกษาในโครงการพัฒนาระบบงานนี้ได้เป็นอย่างดี รวมถึงยังได้รับความร่วมมือจากทางครอบครัวในด้านการสนับสนุนเงินทุนทางด้านการศึกษามาเป็นอย่างดีตลอดระยะเวลาการศึกษา จนสามารถก้าวมาสู่การพัฒนา ระบบงานครั้งนี้ได้ และนอกเหนือจากนี้ยังต้องขอขอบคุณทางคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เปิดโอกาสให้ข้าพเจ้าได้เข้ามาศึกษายังสถาบันแห่งนี้ รวมถึงท่านอาจารย์ทุกท่านที่ได้ถ่ายทอดความรู้ที่ดีตลอดมา จนสามารถนำมาประยุกต์ใช้กับการพัฒนาระบบงานนี้ได้อย่างเต็มประสิทธิภาพ และในท้ายที่สุดข้าพเจ้าขอขอบคุณเพื่อนๆทุกท่านที่ได้ให้คำปรึกษา และเป็นกำลังใจให้มาโดยตลอด ซึ่งหากปราศจากบุคคลดังกล่าวแล้วการพัฒนา ระบบงานครั้งนี้ก็คงจะไม่สามารถดำเนินการ ได้แล้วเสร็จตามที่หวังไว้อย่างแน่นอน

นายประดิษฐ์ วัฒนสุวรรณ

นักศึกษาระดับปริญญาโท สาขาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่	
1. บทนำ	
1.1 ความเป็นมา.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของการพัฒนาระบบงาน.....	2
1.4 เป้าหมายของการพัฒนาระบบงาน.....	2
1.5 องค์ประกอบและเครื่องมือที่ใช้ในการพัฒนาระบบงาน.....	3
2. การวิเคราะห์และออกแบบระบบงาน	
2.1 ความสำคัญของปัญหา.....	4
2.2 แนวทางการแก้ปัญหา.....	4
2.3 ขอบเขตของการวิเคราะห์.....	5
2.4 วิเคราะห์และออกแบบ.....	5
3. จาวาเซิร์ฟเวท	
3.1 แนะนำเกี่ยวกับเซิร์ฟเวทเบื้องต้น.....	8
3.2 สถาปัตยกรรมของเซิร์ฟเวท.....	9
3.3 วงจรชีวิตของเซิร์ฟเวท.....	9
3.4 การเขียนโปรแกรมเซิร์ฟเวท.....	10
3.5 การโต้ตอบกับ Client.....	10
3.6 การใช้จาวาเซิร์ฟเวทในรูปแบบ Web-Based Applications.....	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4. ฐานข้อมูล ออร์ราเคิล	
4.1 ส่วนประกอบของสถาปัตยกรรมออร์ราเคิล.....	14
4.2 โครงสร้างหน่วยความจำ.....	21
4.3 การจัดการในส่วนของ Control Files.....	22
4.4 การจัดการในส่วนของ Redo Log Files.....	23
4.5 ความแตกต่างระหว่างโครงสร้างทาง Logical และ Physical.....	24
4.6 Locking Mechanism.....	26
4.7 การจัดการกับ Sessions.....	28
5. ความปลอดภัยของข้อมูล	
5.1 บทนำ.....	29
5.2 แนะนำหลักการทำงานของ JavaScript เบื้องต้น.....	29
5.3 การเข้ารหัสแบบ Nihilist Substitution.....	32
6. การพัฒนาโปรแกรม	
6.1 ขั้นตอนการเตรียมสถานะแวดล้อมสำหรับการพัฒนาโปรแกรม.....	33
6.2 ขั้นตอนในการพัฒนาโปรแกรมในส่วนของกรอกแบบหน้าจอ.....	35
6.3 การดำเนินการในส่วนการเข้ารหัส.....	45
7. สรุปผลการค้นคว้าและการพัฒนาโปรแกรม	
7.1 ผลการดำเนินการพัฒนาโปรแกรม.....	47
7.2 ประโยชน์ที่ได้รับจากการพัฒนาโปรแกรม.....	47
7.3 ข้อเสนอแนะ.....	48
บรรณานุกรม	49
ภาคผนวก	
ภาคผนวก ก. ตัวอย่างรูปแบบการเขียนโปรแกรมในลักษณะของ Servlet.....	51
ประวัติผู้เขียน	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

4.1. แสดงวิวัฒนาการที่เกี่ยวข้องกับการจัดการฐานข้อมูล	19
6.1. การแทนค่า ASCII CODE ที่ใช้ในการเข้าและถอดรหัส	45



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

ภาพที่

2.1. Dynamic Performance View	6
3.1. รูปแบบการให้บริการของ Servlet	8
3.2. ตัวอย่างการใช้งาน Servlet กับงาน โครงการวิศวกรรม โรงงาน	13
4.1. แสดงไฟล์ที่เป็นส่วนประกอบของ Oracle Database	14
4.2. แสดงลักษณะการมองในรูปแบบ Data Dictionary Views	18
4.3. แสดงลักษณะของ Group และ Member ของ Redo Log files	23
4.4. แสดงโครงสร้างของฐานข้อมูลในเชิง Logical และ Physical	25
5.1. การแบ่งส่วนของ JavaScript	30
6.1. แสดงหน้าจอการ Login เข้าสู่ระบบ	35
6.2. แสดงหน้าจอหลักของโปรแกรม	36
6.3. แสดงหน้าจอส่วนของ Dictionary Table Page	37
6.4. แสดงหน้าจอส่วนของการเรียกดูข้อมูลตารางภายใน GV\$EVENT_NAME	38
6.5. แสดงหน้าจอส่วนของ SQL Worksheet Page	38
6.6. แสดงหน้าจอส่วนผลลัพธ์ที่ได้จากการ Execute SQL Command	39
6.7. แสดงหน้าจอส่วนของ Tablespace's List Page	40
6.8. แสดงหน้าจอส่วนของการเปลี่ยนแปลงขนาด Tablespace	41
6.9. แสดงหน้าจอผลของการเปลี่ยนแปลงขนาด Tablespace	42
6.10. แสดงหน้าจอส่วนของ Redo Log File's List Page	42
6.11. แสดงหน้าจอส่วนของ Deadlocks Process Page	43
6.12. แสดงหน้าจอส่วนของ Session's List Page	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมา

ในปัจจุบันมีการนำคอมพิวเตอร์เข้ามาช่วยในงานด้านต่างๆมากมาย เพื่อให้ประหยัดแรงงานมนุษย์ จึงทำให้เกิดโปรแกรมประยุกต์ขึ้นมาอย่างรวดเร็ว และสิ่งที่จำเป็นอย่างยิ่งในการดำเนินงาน ก็คือการจัดเก็บข้อมูลเพื่อใช้ในงานด้านต่างๆ เช่น ประวัติพนักงาน, รายละเอียดของสินค้า และบริการ เพื่อให้การจัดเก็บ ค้นหา หรือดำเนินการใดๆ เป็นไปได้อย่างสะดวกรวดเร็ว ตอบสนองต่อความต้องการของผู้ใช้งานได้ ดังนั้นจึงจำเป็นที่จะต้องมีผู้ที่ทำการดูแลระบบฐานข้อมูล (DBA) ทำหน้าที่ในการบริหารและจัดการกับข้อมูล เพื่อให้ระบบฐานข้อมูลทำงานได้อย่างถูกต้องตลอดเวลา รวมถึงการแก้ไขปัญหาที่เกิดขึ้นกับฐานข้อมูลได้อย่างรวดเร็ว แต่ปัจจุบันเทคโนโลยีทางด้านอินเทอร์เน็ตได้มีบทบาทกับงานทุกๆด้าน และมีส่วนช่วยให้ผู้คนมีความสะดวกสบายมากขึ้นในการค้นหาหาข้อมูล รวมถึงแนวโน้มของโปรแกรมประยุกต์ในปัจจุบันก็จะคล้อยตามอินเทอร์เน็ตเช่นกัน ทำให้เกิดแนวความคิดที่จะนำเอาเทคโนโลยีทางด้านอินเทอร์เน็ตมาปรับใช้ในงานทางด้านการบริหารและจัดการฐานข้อมูลได้สะดวกขึ้น เพื่อให้ผู้ที่ดูแลฐานข้อมูลสามารถเข้ามาบริหารและจัดการฐานข้อมูลจากสถานที่และเวลาใดก็ได้ที่สามารถเข้าถึงอินเทอร์เน็ตได้

1.2 วัตถุประสงค์

- เพื่อศึกษาหน้าที่หลักที่จำเป็นของผู้ดูแลระบบฐานข้อมูล
- ทำการวิเคราะห์ปัญหาหลักที่เกิดขึ้นกับฐานข้อมูล ที่พบบ่อยครั้งในขณะฐานข้อมูลดำเนินการอยู่ เพื่อให้ผู้ดูแลฐานข้อมูลสามารถ ที่จะดำเนินการแก้ไขปัญหาที่เกิดขึ้นผ่านระบบอินเทอร์เน็ตได้
- พัฒนาโปรแกรมประยุกต์ เพื่อใช้ในการแก้ไขปัญหาที่เกิดขึ้นได้ ตามที่วิเคราะห์ปัญหาหลักได้

1.3 ขอบเขตของการพัฒนาระบบงาน

ในการพัฒนาระบบงานมีหลักการและขอบเขตเพื่อให้ครอบคลุมส่วนต่างๆที่สำคัญคือ

- ระบบที่พัฒนาขึ้นจะทำงานในรูปแบบเว็บแอปพลิเคชัน โดยโปรแกรมที่พัฒนาขึ้นจะอยู่ในส่วนของแอปพลิเคชันเซิร์ฟเวอร์ ที่เป็นตัวกลางระหว่างดาต้าเบสเซิร์ฟเวอร์กับไคลเอนต์ ซึ่งจะมีการเรียกการใช้งานผ่านเว็บ
- ผู้ดูแลระบบฐานข้อมูลสามารถที่จะทำการเรียกใช้ระบบที่พัฒนาขึ้นจากสถานที่ใดก็ได้ที่มี การเชื่อมต่อกับระบบอินเทอร์เน็ต
- ระบบที่พัฒนาขึ้นจะมีหน้าที่หลักในการบริหารและจัดการฐานข้อมูลดังนี้
 - ทำการวิวดู Dictionary ทั้งหมดของฐานข้อมูลที่ดาต้าเบสเซิร์ฟเวอร์ได้
 - ทำการส่งคำสั่ง SQL ที่ต้องการ ไปประมวลผลยังดาต้าเบสเซิร์ฟเวอร์ได้
 - ทำการลดหรือเพิ่มขนาดของ Tablespace ของดาต้าเบสได้ตามต้องการ
 - ทำการลดหรือเพิ่มขนาดของ Redo log files ของดาต้าเบสได้ตามต้องการ
 - จัดการกับ Process ที่เกิดการคอยกันได้ (Deadlock Process)
 - ทำการลบ Session บาง Session ที่ไม่ได้ใช้งานแล้ว แต่ยังคงมีการค้างอยู่ในระบบได้

1.4 เป้าหมายของการพัฒนาระบบงาน

ระบบที่พัฒนาขึ้นก็เพื่อให้ผู้ดูแลระบบฐานข้อมูลสามารถที่จะทำการดูแลฐานข้อมูลจากสถานที่ต่างๆได้ โดยการดำเนินการจะกระทำผ่านเครือข่ายอินเทอร์เน็ต โดยการทำงานจะไม่สามารถที่จะครอบคลุมงานทั้งหมดของผู้ดูแลฐานข้อมูลได้ ซึ่งงานที่สามารถจัดการได้ต้องเป็นงานที่กระทำผ่าน SQL command เท่านั้น เนื่องจากการติดต่อระหว่างแอปพลิเคชันเซิร์ฟเวอร์กับดาต้าเบสเซิร์ฟเวอร์นั้นจะใช้ตัวภาษา SQL ในการสั่งงานจากแอปพลิเคชันเซิร์ฟเวอร์ไปยังดาต้าเบสเซิร์ฟเวอร์เพื่อให้ทำการประมวลผลแล้วทำการส่งค่ากลับมาเป็น Result Set

1.5 องค์ประกอบและเครื่องมือที่ใช้ในการพัฒนาระบบงาน

1.5.1 องค์ประกอบทางด้านฮาร์ดแวร์

- เครื่องคอมพิวเตอร์เชื่อมต่อเครือข่ายสำหรับดำเนินการเป็นดาต้าเบสเซิร์ฟเวอร์
- เครื่องคอมพิวเตอร์เชื่อมต่อเครือข่ายสำหรับดำเนินการเป็นแอปพลิเคชันเซิร์ฟเวอร์ (เครื่องสำหรับติดตั้งระบบงานที่พัฒนาขึ้นมา)
- เครื่องคอมพิวเตอร์เชื่อมต่อเครือข่ายสำหรับทดสอบเรียกดูการใช้งานระบบงาน (Client)

1.5.2 องค์ประกอบทางด้านซอฟต์แวร์ (Software)

- ระบบปฏิบัติการเครือข่าย (Network Operating System)
Microsoft Windows 2000 Advanced Server
- ระบบการให้บริการร้องขอผ่านเว็บ (Web Application Server)
JSWDK (JavaServer Web Development Kit)(Java Servlets)
- ระบบการให้บริการฐานข้อมูล (DBMS)
Oracle Server 8.0.5 for Windows NT
- เครื่องมือหรือซอฟต์แวร์ที่ใช้ในการพัฒนาระบบงาน
 - Microsoft Internet Explorer 5.0
 - Adobe Photoshop 5.0
 - EditPlus 2.0

บทที่ 2

การวิเคราะห์และออกแบบระบบงาน

2.1 ความสำคัญของปัญหา

ปัจจุบันการแก้ไขปัญหามาตรฐานข้อมูลที่เกิดขึ้นอย่างฉับพลันนั้น ในบางครั้งไม่สามารถดำเนินการได้ทันกับความต้องการของผู้ใช้ เนื่องจากสาเหตุหลายๆประการ เช่น ผู้ดูแลระบบฐานข้อมูล ไม่สามารถมายังสถานที่ตั้งของฐานข้อมูลนั้นๆ ทำให้เกิดความเสียหายบางอย่างกับองค์กรนั้นๆ โดยมากแล้วปัญหาเกี่ยวกับฐานข้อมูลออร์ราเคิลที่พบบ่อยมีดังนี้

- Space ของฐานข้อมูลมีไม่เพียงพอต่อความต้องการ ของผู้ใช้งาน แยกออกเป็น
 - Tablespaces (เป็นพื้นที่ในเชิง Logical)
 - Data Files (เป็นพื้นที่ในเชิง Physical)
- Redo log files ที่กำหนดขึ้นมีขนาดเล็กเกินความต้องการที่ฐานข้อมูลดำเนินการ
- เกิดการคอยกันระหว่าง Process ในการเข้าถึงการใช้งานตารางหรือทรัพยากรระบบ
- มี Session บาง Session ที่ไม่ได้ใช้งานแล้วค้างอยู่ในระบบ

2.2 แนวทางการแก้ปัญหา

แนวทางการแก้ปัญหาจะกระทำโดยการวิเคราะห์ถึงเหตุที่ทำให้เกิดปัญหาขึ้นในแต่ละส่วน แยกจากกัน แล้วทำการกำหนดวิธีการแก้ไขให้อยู่ในรูปแบบของภาษา SQL เพื่อให้สามารถที่จะดำเนินการสั่งการทำงานให้แก้ไขปัญหามาตรฐานข้อมูลได้โดยง่าย โดยกำหนดวิธีการแก้ไขปัญหานั้นทั้งหมดได้ดังนี้

- ดำเนินการในส่วนที่จะให้ DBA สามารถที่จะทำการเรียกดูขนาดและรายละเอียดเกี่ยวกับแต่ละ Tablespace เพื่อดำเนินการขยายหรือลดขนาดได้ตามความต้องการ
- ดำเนินการในส่วนที่จะให้ DBA สามารถที่จะทำการเรียกดูขนาดและรายละเอียดเกี่ยวกับแต่ละ Redo log files เพื่อดำเนินการขยายหรือลดขนาดได้ตามความต้องการ
- ดำเนินการในส่วนที่จะให้ DBA สามารถเรียกดู Process ต่างๆที่มีอยู่ในระบบฐานข้อมูลนั้นเพื่อให้สามารถเรียกคืนทรัพยากรบางส่วนได้ในกรณีที่เกิดการรอคอยทรัพยากรกันได้

- ดำเนินการให้ DBA สามารถที่จะทำลาย Session บาง Session ได้ในกรณีที่ Session นั้นไม่มีการใช้งานจริง แต่ได้มีการจองทรัพยากรบางส่วนอยู่

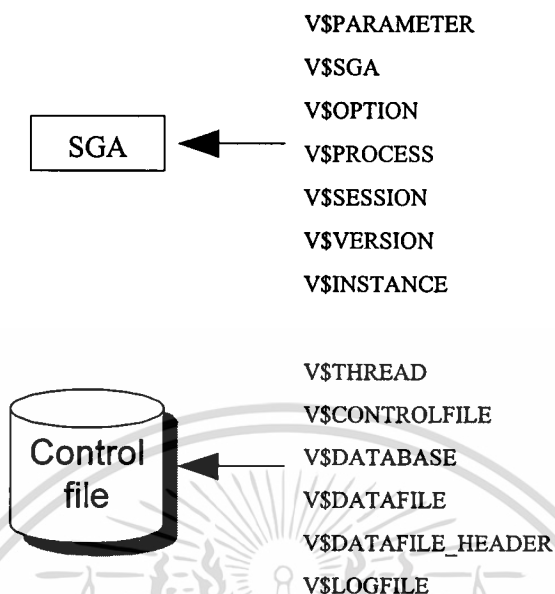
2.3 ขอบเขตของการวิเคราะห์

การวิเคราะห์และออกแบบนี้ ทำขึ้นเพื่อให้ผู้ดูแลระบบฐานข้อมูลสามารถจัดการกับปัญหาที่เกิดขึ้นกับฐานข้อมูล งานส่วนมากของผู้ดูแลระบบฐานข้อมูลจะมีการกระทำไว้ก่อนแล้ว เช่นการสร้าง Instance, การสร้างฐานข้อมูล, การกำหนดสิทธิ์ต่างๆในการเข้าถึงฐานข้อมูล งานเหล่านี้จึงไม่จำเป็นต้องดำเนินการผ่าน Internet ในการออกแบบให้ดำเนินการผ่าน Internet ครั้งนี้ จึงเน้นหนักไปในทิศทางของการแก้ไขปัญหาที่เกิดขึ้นในขณะที่ฐานข้อมูลถูกใช้งานอยู่ โดยมากแล้วปัญหาส่วนใหญ่จะเกิดจากการที่มีผู้ใช้งานเข้าไปใช้งานระบบแล้วไม่สามารถดำเนินการตามวัตถุประสงค์ที่ต้องการ หรือประสิทธิภาพไม่ดีเท่าที่ควร ส่วนใหญ่แล้วผู้ดูแลจะรู้ถึงปัญหาที่เกิดขึ้นจากผู้ใช้งานระบบ ทำให้ผู้ดูแลระบบสามารถที่จะมุ่งเข้าไปแก้ปัญหาได้อย่างถูกต้องประสงค์ ปัญหาที่สามารถแก้ไขผ่าน Internet ได้นั้นจำเป็นอย่างยิ่งที่จะต้องสามารถแก้ไขได้โดยการใช้ภาษา SQL

2.4 วิเคราะห์และออกแบบ

2.4.1 วิเคราะห์

ในการดำเนินการเกี่ยวกับการแก้ไขปัญหาของฐานข้อมูลที่เกิดขึ้นนั้นจำเป็นอย่างยิ่งที่จะต้องทราบถึงปัญหา และสถานะต่างของระบบที่จะเกี่ยวข้องกับปัญหา โดยที่ประสิทธิภาพและการดำเนินการ รวมถึงพารามิเตอร์ต่างๆของ Oracle Database นั้นจะเก็บข้อมูลเป็นลักษณะของตาราง ซึ่งผู้ที่เป็นผู้ดูแลระบบสามารถเข้ามาดูข้อมูลต่างๆเพื่อใช้ในการวิเคราะห์ และตัดสินใจในการแก้ไขปัญหาที่เกิดขึ้น ตารางดังกล่าวจะเก็บในชื่อที่ขึ้นต้นด้วย V\$ นำหน้า อย่างเช่น V\$FIXED_TABLE นี้จะเป็นการบอกถึง dynamic performance view ทั้งหมดของระบบ



ภาพที่ 2.1 Dynamic Performance View

และยังมีตารางพื้นฐานและ Data Dictionary View บางส่วนที่ต้องใช้งานในการตัดสินใจแก้ปัญหา

Data Dictionary View :

- View simplify the base table information
- Create with the catalog.sql script

Base Tables:

- Normalized
- Create with the sql.bsq script

นอกจากนี้แล้วยังต้องดูในส่วนของ Diagnostic Information

Trace Files:

- Alert log file
- Background process trace files
- User trace files

Events:

- Oracle wait events
- OEM events

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ออกแบบ

การออกแบบนั้นจะออกแบบให้ผู้ดูแลระบบสามารถแก้ไขปัญหาที่ได้รับการแจ้งมาจากผู้ใช้ระบบฐานข้อมูล โดยลำดับความสามารถของโปรแกรมให้สามารถทำงานได้ดังนี้

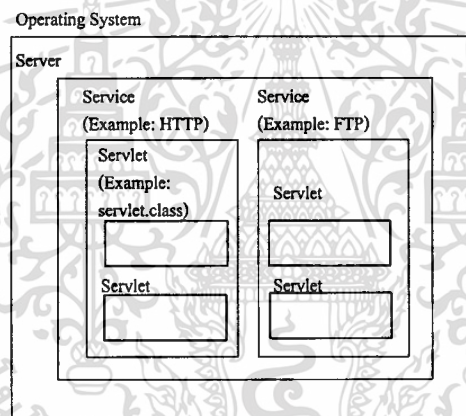
- List ดู Dictionary Table ทั้งหมดของระบบได้ เพื่อเป็นรายละเอียดในการตัดสินใจ
- สามารถส่งคำสั่ง SQL ได้ต่อกับระบบได้ เพื่อการทำงานเฉพาะบางอย่าง (คล้ายกับ SQL Worksheet)
- List ดูขนาดของ Tablespace ทั้งหมดพร้อมทั้งสามารถทำการขยายพื้นที่ได้
- List ดูขนาดของ Redo log files ทั้งหมดที่มีอยู่ในระบบ พร้อมทั้งขยายพื้นที่ได้
- List ดู Process ทั้งหมดที่เกิดการคอยทรัพยากรเดียวกัน พร้อมกับแก้ไข
- List ดู Session ทั้งหมดที่เกิดขึ้น พร้อมทั้งสามารถ Kill Session ที่ค้างอยู่ได้

บทที่ 3

จาวา เซอร์ฟเวท

3.1 แนะนำเกี่ยวกับเซอร์ฟเวทเบื้องต้น

ในตระกูลผลิตภัณฑ์ JavaServer™ นั้น สามารถที่จะจัดการให้ทำการรันการให้บริการได้มากกว่า 1 การให้บริการ โดยในส่วนของ Code สามารถที่จะดำเนินการ application-level protocol เช่น FTP, DHCP หรือ SMTP ได้ ตัว Servlets คือส่วนหนึ่งซึ่งทำการรันภายในส่วนของ request/response-oriented services



ภาพที่ 3.1 รูปแบบการให้บริการของ Servlet

ตัวอย่างบางส่วนจากหลายๆ applications สำหรับ Servlets

- บางส่วนของการประมวลผลข้อมูลบน HTTPS ซึ่งจะใช้รูปแบบ HTML เช่นการสั่งสินค้า ผ่านการใช้งาน credit card โดยตัว Servlet สามารถที่จะทำการบันทึกเกี่ยวกับระบบการจัดซื้อ พร้อมกับมีการบันทึกข้อมูลลงบนฐานข้อมูล รวมถึงการจ่ายเงินแบบ on-line ได้ด้วย
- อนุญาตให้หลายๆคนทำงานร่วมกันได้ โดยตัว servlet สามารถรองรับการร้องขอแบบหลายๆการร้องขอได้ อีกทั้งยังสามารถ Synchronize ระหว่าง requests ได้ เช่นระบบ on-line conferencing
- Servlet สามารถส่งต่อการร้องขอจาก service servlet ที่หนึ่งไปยังอีกที่หนึ่งได้ ทำให้สามารถทำ Load Balance ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 สถาปัตยกรรมของเซิร์ฟเวท

ในตัว JSWDK นั้นจะมี Servlet Interface อยู่ โดยที่ทุกๆ Servlets ที่สร้างขึ้นมาจาก Interface นี้ส่วนใหญ่จะดำเนินการ extending มาจาก HttpServlet ซึ่งตัว Servlet interface จะมีการจัดไว้สำหรับ method ซึ่งใช้ในการจัดการการติดต่อสื่อสารกับ Client เมื่อ Servlet ได้รับการร้องขอจาก client แล้ว ตัว servlet เองจะได้รับ 2 object ขึ้นมา ได้แก่ ServletRequest และ ServletResponse

- ServletRequest class จะใช้ในการติดต่อจากตัวเครื่องที่เป็น Client ไปยังเครื่องที่เป็น Server
- ServletResponse class จะใช้ในการติดต่อจากตัวเครื่องที่เป็น Server ตอบกลับไปยังเครื่องที่เป็น Client

ServletRequest interface จะอนุญาตให้ servlet สามารถทำการเข้าถึงข้อมูลต่างๆ ที่ส่งมาจาก Client เช่น Protocol ที่ Client ใช้ในการติดต่อกับ Server, ชื่อของ parameter ที่ส่งมาจากเครื่อง Client รวมถึงชื่อของทั้ง Server และ Client และยังมีจัดการ servlet สำหรับการ input stream ด้วย โดยที่ตัว ServletInputStream จะเป็นทางสำหรับการดึงข้อมูลจาก clients ซึ่งมีการใช้ application protocol เช่น HTTP POST หรือ PUT methods ส่วน subclasses ของ ServletRequest นั้นจะอนุญาตให้ servlet สามารถดึงข้อมูลโดยการระบุ Protocol มากขึ้น เช่น HttpServletRequest จะมีการบรรจุ method สำหรับการเข้าถึง HTTP-specific header information

ServletResponse interface จะให้ servlet method สำหรับการตอบกลับจาก Server ไปยัง Client โดยจะอนุญาตให้ servlet ทำการกำหนด content length and mine type สำหรับการตอบกลับ รวมถึงการจัดเตรียม output stream ที่เรียกว่า ServletOutputStream ในการดำเนินการส่งข้อมูลกลับไปโดยตรงได้ ส่วน subclass ของ ServletResponse จะให้ servlet มีการระบุ Protocol ได้ เช่น HttpServletResponse จะมีการบรรจุ method ซึ่งอนุญาตให้ servlet สามารถดำเนินการจัดการกับ HTTP-specific header information ได้

3.3 วงจรชีวิตของเซิร์ฟเวท

เมื่อ Service ทำการโหลดและรัน Servlets ซึ่งจะสามารถรองรับกับการร้องขอได้มากกว่า 1 การร้องขอ และหลังจากดำเนินการเสร็จเรียบร้อยแล้วก็จะทำการเคลื่อนย้ายตัว Servlets ออกจากระบบ

ทันทีที่ Service ทำการโหลด servlet ก็จะมีการรัน init method เพื่อให้สามารถกำหนดค่าเริ่มต้นต่างๆให้กับระบบ โดยจะดำเนินการเพียงครั้งเดียวในตอนเริ่มต้น หรือสามารถสั่งให้มันดำเนินการได้โดยการ Reload

หลังจากทำการดำเนินการตาม init method เรียบร้อยแล้วก็จะทำการ handle client request ในกระบวนการนี้จะกระทำในส่วนของ Service method (etc. doPost, doGet) โดยจะทำการรันใน thread ของตัวเอง ในตัว method นี้จะทำหน้าที่รองรับการร้องขอจาก client แล้วทำการตอบกลับจาก server ไปยัง client ด้วย

สุดท้ายหลังจากดำเนินการตาม service method เรียบร้อยแล้วก็จะทำการเคลื่อนย้ายตัว servlet ออก โดยก่อนที่จะทำการเคลื่อนย้ายตัว servlet ออกนั้นจะมีการดำเนินการรัน destroy method เพื่อที่จะให้สามารถดำเนินการใดๆก็ได้ก่อนที่จะเคลื่อนย้าย servlet ออก อย่างเช่น การ synchronized ทรัพยากรบางส่วนก่อนที่จะคืนให้กับระบบ

3.4 การเขียนโปรแกรมเซอร์ฟเวท

Servlets จะดำเนินจาก javax.servlet.Servlet interface โดยส่วนมากแล้ว servlets จะ extend มาจาก web servers ซึ่งใช้ HTTP Protocol ในการติดต่อกับ clients ส่วนใหญ่แล้วจะใช้ javax.servlet.http.HttpServlet class สันนิษฐานมาตรฐาน HTTP/1.1 ตัวอย่างเช่น

```
Public class SurveyServlet extends HttpServlet
    Implements SingleThreadModel{
        /* typical servlet code, with no threading concerns
        * in the service method. No extra code for the
        * SingleThreadModel interface. */
    }
```

3.5 การโต้ตอบกับ client

Servlet writers ที่พัฒนา HTTP servlets ซึ่งบ่งชี้ไปถึง HttpServlet class นั้นจะต้องมีการ override บาง method เพื่อให้รองรับกับ HTTP interactions ซึ่ง handle โดย servlet ได้ โดย method ต่างๆจะมีดังนี้

- doGet, สำหรับการรองรับกับการร้องขอที่มี GET เป็น Header
- doPost, สำหรับการรองรับการร้องขอแบบ POST
- doPut, สำหรับการรองรับการร้องขอแบบ PUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- doDelete, สำหรับการรองรับการร้องขอแบบ Delete

HttpServletRequest object จะจัดหาวิธีการเข้าถึงข้อมูล HTTP header ดังเช่น cookies ต่างๆที่พบจากการร้องขอ นอกจากนี้แล้วในส่วนของการร้องขอยังมีข้อมูลต่างๆที่ต้องการอีกมากมายที่ส่งมาพร้อมกับการร้องขอ เช่น พารามิเตอร์ที่ส่งมาด้วย ดังนั้นจึงต้องมีวิธีการดึงข้อมูลต่างๆเหล่านี้มาใช้งานได้ตาม method เหล่านี้

- `getParameterValues` method จะให้ผลลัพธ์กลับมาเป็นค่าของชื่อ parameter นั้นๆ
- `getParameterNames` method จะให้ผลลัพธ์กลับมาเป็นชื่อของ parameter นั้นๆ
- `getQueryString` method ใช้ในการดึงสายข้อมูลอักขระที่มากับ HTTP GET method
- `getReader`, `getInputStream` method ใช้ในการอ่านข้อมูลจาก `BufferedReader` และ `ServletInputStream` ที่มากับ method POST, PUT, DELETE

HttpServletResponse object จะมีการส่งค่ากลับได้ 2 วิธี ดังนั้นจึงมี 2 method คือ

- `getWriter` method ใช้ในการส่งข้อมูลจาก server กลับไปยัง client ในรูปแบบข้อมูลที่เป็น text
- `getOutputStream` method ใช้ในการส่งข้อมูลจาก server กลับไปยัง client ในรูปแบบข้อมูลที่เป็น binary

/**

* This is a simple example of an HTTP Servlet. It responds to the GET

* and HEAD methods of the HTTP protocol.

*/

```
public class SimpleServlet extends HttpServlets{
```

```
    public void doGet(HttpServletRequest req, HttpServletResponse res)
```

```
        throws ServletException, IOException
```

```
    {
```

```
        // set header field first
```

```
        res.setContentType("text/html");
```

```
        // then get the writer and write the response data
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    PrintWriter out = res.getWriter();
    out.println("<HEAD><TITLE>SimpleServlet
    Output</TITLE></HEAD><BODY>");
    out.println("<h1> SimpleServlet Output</h1>");
    out.println("<p>This is output is from SimpleServlet.");
    out.println("</Body>");
    out.close();
}
public String getServletInfo(){
    return "A simple servlet";
}
}

```

3.6 การใช้จาวาเซิร์ฟเวทในรูปแบบ Web-Based Applications

Servlet สามารถเป็นกุญแจสำคัญในโปรแกรมประยุกต์หลายๆตัว โดยใช้ Java และ Internet Technology เข้ามามีส่วนร่วม

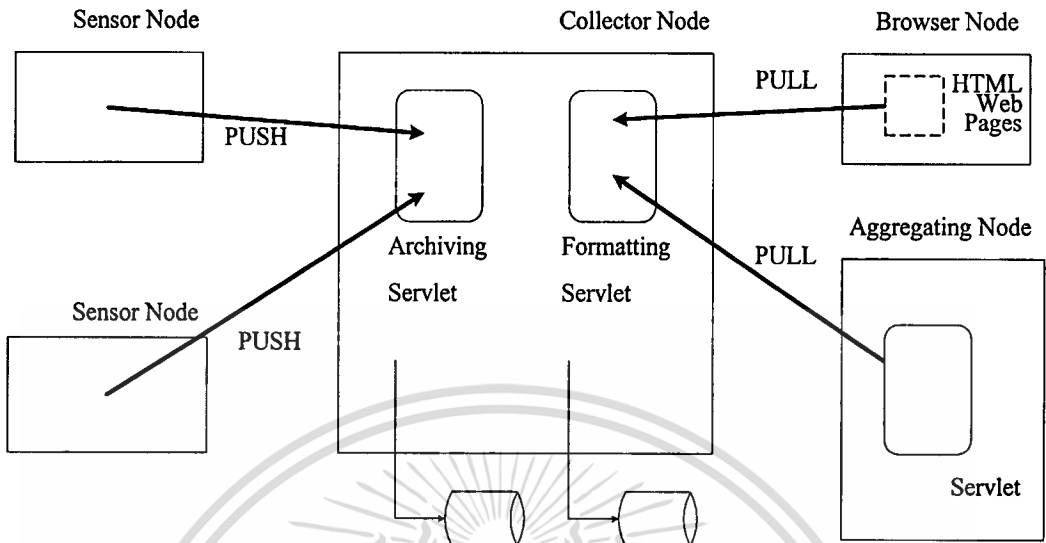
Three Tier Applications

ในปัจจุบันหลายๆองค์กรมีความต้องการในการใช้ multi-tier applications

Tier ที่ 1: ทำหน้าที่ในการแสดงผลโดย application ที่จะใช้ก็อย่างเช่น java-enabled browsers หรืออาจเป็น browser อื่นๆก็ได้ ซึ่งจะใช้เครื่องคอมพิวเตอร์ที่ Personal Computer หรือ Workstations ในการเรียกการใช้งานจะใช้ tier นี้เป็นตัวการในการติดต่อข้อมูลไปกลับกับ tier ที่ 2

Tier ที่ 2: ทำหน้าที่เป็น application server ในการเก็บ business rules and logic ของ application ต่างๆที่นำมาเก็บไว้เพื่อให้ tier ที่ 1 เรียกใช้งาน

Tier ที่ 3: ทำหน้าที่เป็นระบบการให้บริการทางด้านข้อมูล (Database System) เช่น relational database โดยส่วนมากแล้วจะทำการติดต่อผ่าน JDBC หรืออาจเป็น Interface อื่นๆที่สนับสนุนการติดต่อกับ legacy data ก็ได้



ภาพที่ 3.2 ตัวอย่างการใช้งานเซิร์ฟเวทกับงานโครงการวิศวกรรมโรงงาน

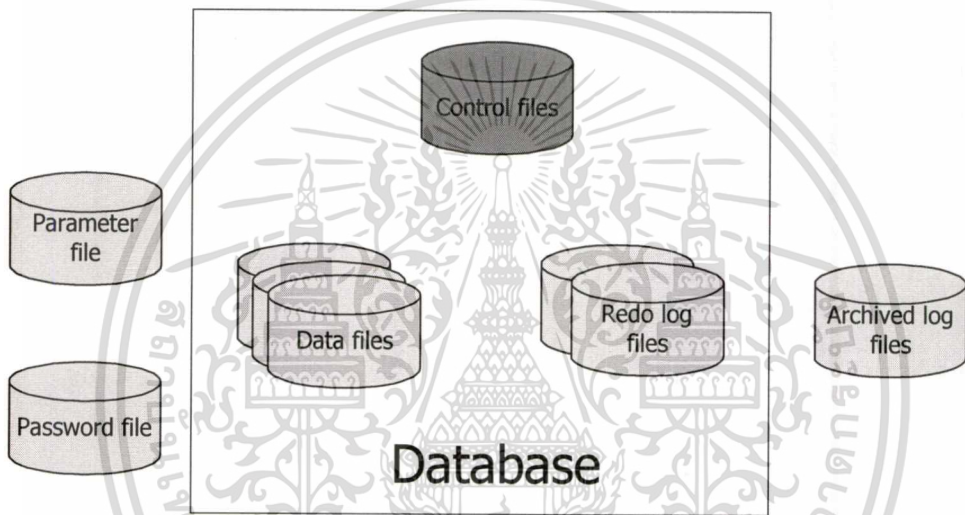
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ฐานข้อมูล ออร์ราเคิล

4.1 ส่วนประกอบของสถาปัตยกรรมออร์ราเคิล

4.1.1 ไฟล์ของฐานข้อมูล ออร์ราเคิล



ภาพที่ 4.1 แสดงไฟล์ที่เป็นส่วนประกอบของ Oracle Database

Oracle Database ถูกบ่งชี้โดยชื่อของฐานข้อมูล (DB_NAME) แทนด้วยโครงสร้างทางกายภาพ คือจะเป็นส่วนประกอบของ Operating System Files โดยสามารถใช้ชื่อฐานข้อมูลที่แตกต่าง Instance name ได้ แต่ Oracle แนะนำให้ใช้ชื่อเดียวกัน เพื่อการบริหารและจัดการที่ง่าย

4.1.2 Database Files

ไฟล์ที่ประกอบเป็นดาต้าเบสนั้นจะเก็บข้อมูลผู้ใช้งาน และส่วนประกอบรายละเอียดเพิ่มเติมที่ต้องการ เพื่อให้แน่ใจว่าในกระบวนการฐานข้อมูล Oracle Database ประกอบด้วยส่วนประกอบของไฟล์ตามชนิดต่อไปนี้

- Data Files: เก็บ Data Dictionary, User Objects โดยที่หนึ่งดาต้าเบส จะ

ต้องมีอย่างน้อย 1 Data File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Redo log Files: เก็บบันทึกการเปลี่ยนแปลงค่า database เพื่อให้แน่ใจว่าข้อมูลจะไม่เสียหาย (1 ค่า database ต้องการอย่างน้อย 2 Redo log files)
- Control Files: บรรจुरายละเอียดข้อมูลที่ต้องการดูแลและตรวจสอบความถูกต้อง (Data integrity) 1 ค่า database ต้องการอย่างน้อย 1 Control file

นอกจากส่วนประกอบของ Database Files แล้วยังต้องประกอบไปด้วยไฟล์อื่นๆ อีกดังนี้

- Parameter File: ใช้ในการกำหนดคุณสมบัติของ Oracle instance
- Password File: ใช้ในการตรวจสอบสิทธิ์ของ Database Users
- Archived redo log files: เป็นส่วนคัดลอกของ redo log files ซึ่งอาจจะจำเป็นในการ Recovery จากสื่อที่เสียหายไป

4.1.3 Rollback Segment

ก่อนการทำการเปลี่ยนแปลงข้อมูลนั้น จะมีการเซฟค่าเดิมลงใน rollback segment โดยเก็บค่าเดิมไว้เพื่อประโยชน์

- เปลี่ยนค่ากลับเมื่อ transaction rolled back
- แน่ใจว่า transactions อื่น ไม่สามารถเห็นค่าที่ยังไม่ Commit ได้ (เพื่อความถูกต้องในการอ่านข้อมูล)
- Recover Database เมื่อเกิดการ fail

Rollback Segment จึงทำหน้าที่คล้ายกับตารางและ indexes แทนบางส่วนของ data files ลงใน database buffer cache เมื่อต้องการ

4.1.4 Redo log Buffer

Server process จะทำการบันทึกการเปลี่ยนแปลงที่เกิดขึ้นโดย instance ใน redo log buffer ซึ่งเป็นส่วนประกอบหนึ่งของ SGA โดยที่ redo log buffer มีคุณลักษณะดังนี้

- ขนาดจะกำหนดโดย Log_Buffer parameter
- เก็บ redo record ที่มีการเปลี่ยนแปลง
- Redo log buffer ใช้แบบลำดับในการเก็บ transaction ที่เปลี่ยนแปลง
- มีคุณลักษณะเป็น Circular buffer ซึ่งจะนำกลับมาใช้ใหม่ได้ แต่จะใช้ได้ก็ต่อเมื่อ old redo entries บันทึก record ลงใน redo log files เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 Database Writer (DBWR)

Server process จะบันทึกการเปลี่ยนแปลงลงใน rollback และ data block ใน buffer cache โดยที่ DBWR จะทำหน้าที่เขียนข้อมูลจาก database buffer cache ไปยัง data files เพื่อเป็นการ clear buffers

การที่ Server process บันทึกข้อมูลการเปลี่ยนแปลงลง buffer เพียงอย่างเดียวก็เพื่อความรวดเร็ว และ DBWR ก็จะทำหน้าที่เขียนข้อมูลลงไป ใน data files จนกระทั่งเหตุการณ์ใดเหตุการณ์หนึ่งเกิดขึ้น

- จำนวน buffer ที่จะเขียนถึงจุดสูงสุดที่กำหนดไว้
- พื้นที่ buffer เต็ม
- Timeout occurs
- DBWR checkpoint สามารถ triggered โดยเหตุการณ์ที่หลากหลายเป็น database ปิด (checkpoint มีความหมายถึงความสอดคล้องและสัมพันธ์กันระหว่าง database buffer cache กับ data file)

4.1.6 Log Writer (LGWR)

LGWR คือ background process ซึ่งถูกเขียนจาก redo log buffer ลงใน redo log files โดย LGWR ทำหน้าที่เขียนข้อมูลตามลำดับไปยัง redo log file ภายใต้ข้อกำหนดดังนี้

- เมื่อ redo log buffer ที่ 1 ถึง 3 เต็ม
- เมื่อ timeout ปรากฏขึ้น (ทุกๆ 3 วินาที)
- ก่อนที่ DBWR เปลี่ยนแปลง block ข้อมูลใน database buffer cache ไปยัง data files
- เมื่อ transaction commits

4.1.7 Dynamic Performance Views

มีการเปลี่ยนแปลงอย่างต่อเนื่องขณะที่ database ถูกปิดและใช้งาน เนื้อหาจะสอดคล้องกับประสิทธิภาพ จัดหาข้อมูลเกี่ยวกับโครงสร้างภายใน disk และโครงสร้างหน่วยความจำและการเข้าถึง database administrator

DPV ถูกบ่งชี้โดยตัวนำหน้าด้วย V_\$, แต่ Oracle จัดให้เป็นสาธารณะกับคำนำหน้าด้วย VS โดยที่ทุกตัวของ DPV จะแสดงไว้ใน V\$FIXED_TABLE

4.1.8 Data Dictionary Structure

data dictionary จัดหารายละเอียดเกี่ยวกับ

- logical & physical database structure
- ซึ่งนิยาม และพื้นที่เหลือของ schema object
- กฎข้อบังคับต่างๆ
- Database users และสิทธิการเข้าถึงต่างๆ
- การตรวจสอบ

Data Dictionary คือส่วนสำคัญส่วนหนึ่งของ oracle database โดย data dictionary คือกลุ่มของตารางและมุมมองซึ่งใช้อ้างอิงในการจัดการรายละเอียดเกี่ยวกับ database โดยที่ data dic table สร้างโดย script file sql.bsq ระหว่างการสร้างของ database โดย data dictionary คือศูนย์กลางข้อมูลของ Oracle Server และสำหรับ database administrators และ database users

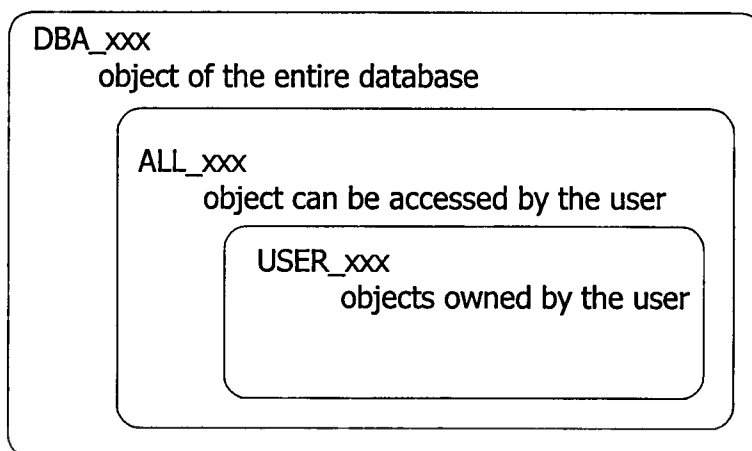
Data Dictionary ถูก update โดย Oracle Server เมื่อ DDL command ถูก executed ในส่วนเพิ่มเติม DML Commands ดังเช่นผลลัพธ์ในตารางเพื่อขยายความสามารถในการ update data dictionary โดยที่ data dictionary สร้างเก็บไว้ใน SYSTEM Tablespace และเป็นเจ้าของโดย User SYS

Base Tables

พื้นฐานของ data dictionary คือกลุ่มของพื้นฐาน tables โดยที่ Oracle Serve เขียนและอ่าน table นี้ เช่นการเรียกดู IND\$ table เพื่อเก็บรายละเอียดเกี่ยวกับ indexes ซึ่งกำหนดไว้ใน database หรือ Select จาก Obj\$ table ในการแสดง object ที่กำหนดใน database โดยที่ไม่สามารถใช้ DML Command เช่น INSERT, UPDATE และ DELETE เปลี่ยนแปลงแก้ไข data dictionary table ได้โดยตรง ยกเว้นแต่ AUD\$ table

Data Dictionary Views

สร้างโดยการรัน catalog.sql script จะเก็บและรวบรวมรายละเอียดต่างๆใน base data dictionary tables โดยชื่อที่เป็นสาธารณะจะถูกเก็บใน data dictionary views ซึ่งผู้ใช้งานสามารถเข้าถึงได้โดยง่าย



ภาพที่ 4.2 แสดงลักษณะการมองในรูปแบบ Data Dictionary Views

Data Dictionary View Categories

Data Dictionary view ถูกแยกเก็บไว้ 3 ส่วน ในหลายๆกรณี ตัว views เองอาจมีข้อมูลที่คล้ายๆกัน และสามารถบ่งชี้จาก Prefixes โดย

The Prefix USER

การ views สามารถเข้าถึงได้จากหลายๆ user เช่น USER_TABLES บรรจุ information บนทุกตารางที่มีผู้ใช้งานเป็นเจ้าของ การให้ค่ากลับของ view คือกลุ่มของ information ในมุมมอง ALL_

The Prefix ALL

มุมมองกับ Prefix ALL จะเข้าถึงโดยผู้ใช้งานที่เป็นเจ้าของ การมองในลักษณะนี้ให้ผลลัพธ์ข้อมูลเกี่ยวกับ Object ซึ่ง user เข้าถึงได้โดยผ่าน public หรือ explicit โดยการอนุญาตและกฎ รวมถึงเจ้าของ Object นี้

The Prefix DBA

มุมมองที่มี DBA นำหน้าให้รายละเอียดบนทุกๆ Object ใน database และรวมถึง Column OWNER โดยที่ view นี้จะ query โดย database admin หรือทุก user ซึ่งได้รับสิทธิอนุญาตโดย SELECT ANY TABLE

มุมมอง	คำอธิบาย
dictionary dict_columns	General overview
dba_tables dba_objects dba_lobs dba_tab_columns dba_constraints	รายละเอียดเกี่ยวกับ user object อย่างเช่น ตาราง กฎข้อบังคับ และ columns
dba_users dba_sys_privs dba_roles	รายละเอียดเกี่ยวกับสิทธิ์และบทบาทการใช้งานของผู้ใช้งาน
dba_extents dba_free_space dba_segments	การจองพื้นที่ของ database object
dba_rollback_segs dba_data_files dba_tablespace	ข้อมูลโครงสร้างทั่วไปของ database
dba_audit_trail dba_audit_objects	รายละเอียดข้อมูลเกี่ยวกับการตรวจสอบ

ตารางที่ 4.1 แสดงวิวกางส่วนที่เกี่ยวข้องกับการจัดการฐานข้อมูล

ในการแนะนำเบื้องต้นของ data dictionary views และ dynamic performance views คุณสามารถ query view จาก DICTIONARY หรือ DICT_COLUMNS ดังตัวอย่างเช่น

```
SVRMGR>SELECT *
```

```
2>FROM dictionary
```

```
3>WHERE table_name LIKE '%TABLE%';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE_NAME	COMMENTS
ALL_ALL_TABLES	Description of all object and relational tables accessible to the user
ALL_NESTED_TABLES	Description of nested table in tables
ALL_OBJECT_TABLES	Description of all object tables accessible to the user
ALL_PART_TABLES	
ALL_TABLES	Description of relational tables accessible to the user
ALL_UPDATABLE_COLUMNS	Description of all update table columns
DBA_ALL_TABLES	Description of all object and relational tables in the database
DBA_NESTED_TABLES	Description of nested tables contained in all tables
DBA_OBJECT_TABLES	Description of all object tables in the database
...	

```
SVRMGR>SELECT column_name, comments
2>FROM dict_columns
3>WHERE table_name='DBA_TABLES'
```

COLUMN_NAME	COMMENTS
OWNER	Owner of the table
TABLE_NAME	Name of the table
TABLESPACE_NAME	Name of the tablespace containing the table
CLUSTER_NAME	Name of the cluster, if any, to which the table belongs
IOT_NAME	Name of the index-only table, if any, to which the overflow entry belongs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โครงสร้างหน่วยความจำ

ส่วนแรกที่จะดูคือในส่วนของโครงสร้างการจัดการหน่วยความจำ (Memory Structure) ทุกครั้งที่ Oracle Database Server เริ่มการทำงานจะมีการจองพื้นที่ในหน่วยความจำสร้างเป็นส่วนที่เรียกว่า System Global Area (SGA) หรือบางครั้งก็เรียกว่า Shared Global Area เป็นโครงสร้างของหน่วยความจำที่มีการใช้งานร่วมกันเก็บ ข้อมูล และส่วนที่เป็นการควบคุมการใช้ข้อมูล 1 SGA จะใช้กับ 1 Oracle Database เท่านั้น นอกจากนี้ SGA จะต้องอยู่ใน memory ที่ Non-Paged และ Non-Swapped เท่านั้น หน่วยความจำส่วนนี้จะมีการใช้งานร่วมกันระหว่างผู้ใช้หลายๆคน

ใน SGA จะแบ่งออกเป็นส่วนย่อยๆเรียก Shared Pool ภายในจำลองเป็น memory ส่วนย่อยๆที่ใช้เก็บ Shared SQL Area และ Data Dictionary

4.2.1 Shared SQL Area (SQA)

แต่ละ SQA จะเก็บข้อมูลที่ใช้ในการทำงาน (Execute) ของ 1 คำสั่ง SQL เสมือนกับว่ามีคำสั่ง SQL นั้นทำงานอยู่เพียง 1 Process ใน memory ที่แบ่งไว้ส่วนนี้ การจากหน่วยความจำของแต่ละคำสั่ง SQL นี้จะใช้เทคนิคของ Hashing Function คือจะมีเพียง 1 SQL Statement เท่านั้นที่ ได้ใช้ตำแหน่งของหน่วยความจำที่เดียวกัน บางครั้งเราเรียก SQA นี้ว่า Library Cache

4.2.2 Data Dictionary Cache (DDC)

เป็นส่วนที่เก็บข้อมูลของฐานข้อมูล (Database), Table, View มีการอ้างถึงความสัมพันธ์ระหว่าง ฐานข้อมูล โครงสร้างของฐานข้อมูลและผู้ที่ใช้งานฐานข้อมูล สิ่งที่เก็บใน DDC คือ

- ชื่อของ Table, View ทั้งหมดในฐานข้อมูล
- ชื่อของชนิดของข้อมูล แต่ละคอลัมน์ในฐานข้อมูล
- สิทธิการเข้าถึงข้อมูล (Privileges) ของผู้ใช้งานฐานข้อมูล Oracle

4.2.3 Database Buffer Cache (DBC)

จะเป็นส่วนที่เก็บข้อมูลที่เหมือนกันกับใน Data Block ผู้ใช้งานฐานข้อมูลทั้งหมดที่เข้าถึงระบบรวมกันอยู่จะมีการแบ่งกันงาน Database Buffer Cache นี้การเข้าถึงข้อมูลจาก cache จะมี 2 แบบคือ

- Cache Miss
- Cache Hit

4.2.4 Redo Log Buffer (RLB)

เป็น Buffer ที่เก็บแบบ Circular Buffer เก็บข้อมูลเกี่ยวกับการเปลี่ยนแปลงที่เกิดขึ้นกับฐานข้อมูล

4.3 การจัดการในส่วนของ Control Files

Control file คือรูปแบบ file ในรูปแบบ binary ซึ่งมีความจำเป็นสำหรับ database ในการ start โดยทุกครั้งที่ instance mount ตัว Oracle Database จะทำการอ่าน Control file ว่าตำแหน่ง data files และ online redo log files อยู่ที่ใด

Control file มีการเปลี่ยนแปลงอย่างต่อเนื่อง ขณะที่ database ใช้อยู่ และจะใช้งานได้เมื่อ database mount หรือเปิดอยู่ ซึ่งจะมีการจัดหาข้อมูลเกี่ยวกับ database ในการใช้งานระหว่าง การฟื้นคืนสภาพ ดังนั้นถ้า Control file มีการเสียหาย หรือไม่สามารถทำงานได้จะมีผลทำให้ database ไม่สามารถใช้งานบาง function ได้

ส่วนประกอบของ Control files

- ชื่อฐานข้อมูล และการบ่งชี้
- ชื่อและตำแหน่งของ Data files และ Redo log file
- ชื่อ Tablespaces
- การบันทึกเวลาของการสร้างฐานข้อมูล
- ลำดับของ log ปัจจุบันที่ใช้งานอยู่
- ข้อมูลเกี่ยวกับ checkpoint
- Log history
- รายละเอียดเกี่ยวกับการทำ Backup ของ Recovery Manager Utility

Control file แยกออกเป็น 2 ส่วนหลักๆดังนี้

- Control file ที่สามารถนำกลับมาใช้ใหม่ได้ (Reusable) นำกลับมาใช้ได้โดย Recovery Manager เท่านั้น
- Control file ที่ไม่สามารถนำกลับมาใช้ใหม่ได้ (Not reusable)

ขนาดของ Control file นั้นจะขึ้นอยู่กับ Parameter ดังนี้

- MAXLOGFILES
- MAXLOGMEMBERS
- MAXLOGHISTORY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- MAXDATAFILES
- MAXINSTANCES

View ที่ทำการเก็บข้อมูลเกี่ยวกับ Control file ได้แก่

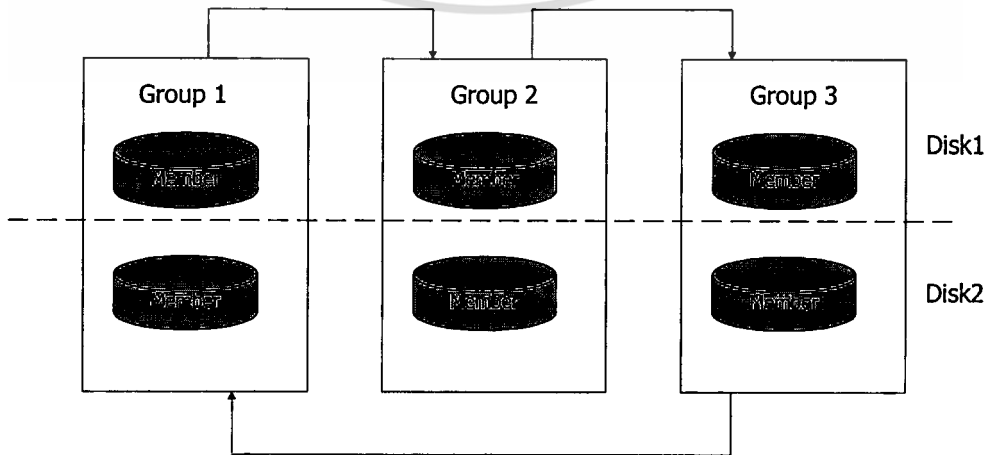
- V\$CONTROLFILE
 - NAME
- V\$PARAMETER
 - NAME (Control file)
 - VALUE
- V\$CONTROLFILE_RECORD_SECTION
 - TYPE
 - RECORD_SIZE
 - RECORD_TOTAL
 - RECORD_USED

4.4 การจัดการในส่วนของ Redo Log Files

Oracle Server จะมีการดูแล redo log เพื่อให้เกิดการสูญเสียของข้อมูลในฐานข้อมูลน้อยที่สุด โดยใช้ในการบันทึกการเปลี่ยนแปลงข้อมูลที่เกิดขึ้นใน database buffer cache ยกเว้นบางอย่าง เช่นกรณีที่มีการเขียนลงไฟล์โดยตรง

Redo log จะใช้ในกรณีที่ instance failure และจะนำ committed data ที่เสร็จเรียบร้อยแล้วแต่ยังไม่ได้เขียนลง data files มาทำการใช้ในการฟื้นคืนสภาพ

4.4.1 สมาชิกและกลุ่มของ Redo log



ภาพที่ 4.3 แสดงลักษณะของ Group และ Member ของ Redo Log files

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการกรอกข้อมูลเท่านั้น เมื่อผู้ดูแลระบบใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ดูแลระบบฐานข้อมูลสามารถกำหนดชุดของ online redo log files ได้เพื่อเป็นการป้องกันการสูญเสียข้อมูลของฐานข้อมูล แม้ว่า redo log ที่จุดหนึ่งจะเสียไปก็ตาม

กลุ่มของ Online Redo Log

- คือกลุ่มของ online redo log file
- Background process LGWR จะเขียนข้อมูลเหมือนกันในทุกๆสมาชิกในกลุ่ม
- Oracle Server ต้องการ 2 online redo log groups เป็นอย่างน้อยสำหรับกระบวนการปกติของฐานข้อมูล

สมาชิกของ Online Redo Log

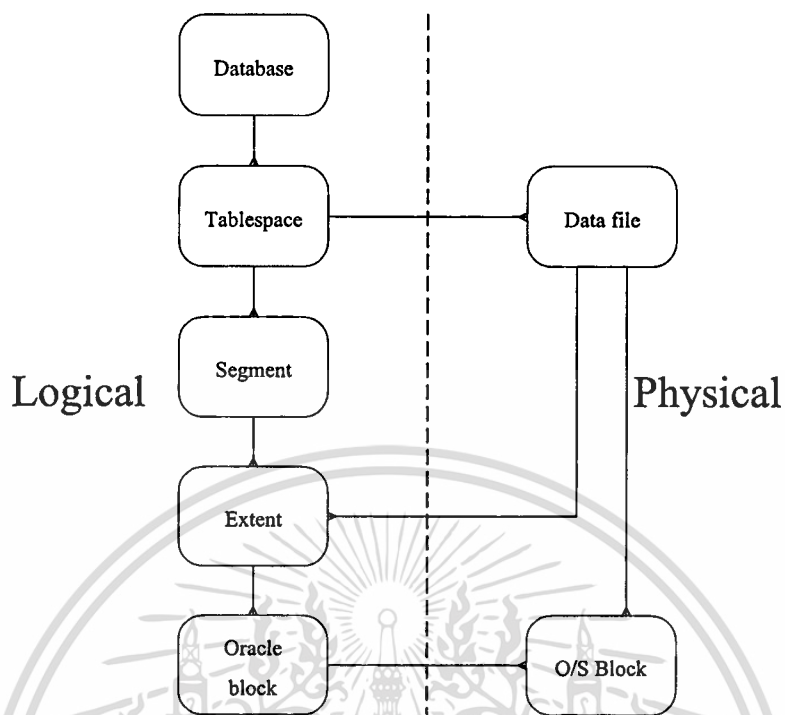
- แต่ละ Online Redo Log File ในกลุ่มเรียกว่าสมาชิก
- แต่ละสมาชิกในกลุ่มถูกกำหนดโดย log sequence number ที่มีขนาดเดียวกัน

4.4.2 Log Switches

LGWR จะเขียน online redo log ตามลำดับ ดังนั้นเมื่อ online redo log group ที่ทำงานอยู่เกิดเต็ม LGWR จะขยับไปใช้กลุ่มอื่นแทน และเมื่อวนจนสุดแล้วก็จะวนกลับมาใช้ตัวแรกใหม่อีกครั้ง ดังนั้น Log switch เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ LGWR หยุดเขียน online redo log group และ เริ่มเขียนตัวถัดไป โดยที่ผู้ดูแลระบบฐานข้อมูลสามารถสั่งให้เกิดการทำ log switch ได้ เมื่อ log switch ปรากฏแล้วจะเกิด checkpoint ขึ้นทันที

4.5 ความแตกต่างระหว่างโครงสร้างทาง Logical และ Physical

สถาปัตยกรรมฐานข้อมูลประกอบด้วยโครงสร้าง logical & physical ซึ่งสร้างโดยฐานข้อมูล โดยที่โครงสร้างทาง physical ประกอบด้วยกลุ่มของ Control files, online redo log files และ data files ส่วนโครงสร้างทาง logical ประกอบด้วย tablespace, segment, extent และ data block



ภาพที่ 4.4 แสดงโครงสร้างของฐานข้อมูลในเชิง Logical และ Physical

ความสัมพันธ์ระหว่าง Database, Tablespace และ Data Files

Oracle database สามารถตัดเป็นส่วนหลักๆที่เรียกว่า tablespaces ได้

Tablespaces

- 1 tablespace เป็นได้แค่ 1 database
- แต่ละ tablespace สามารถประกอบด้วยหลายๆ OS files ได้
- Tablespace ต้องมีการ online ในขณะที่ database รันอยู่
- ยกเว้น SYSTEM tablespace กับ active rollback segment แล้ว tablespace สามารถ offline จาก database ได้ขณะที่ database รันอยู่
- Tablespace สามารถปรับ mode เป็นอ่านและเขียน หรือเขียนอย่างเดียวได้

Datafile

แต่ละ tablespace ใน Oracle Database จะประกอบไปด้วยหลายๆไฟล์ได้เรียกว่า data files ซึ่งเป็น physical structure โดยที่จะอยู่ในระดับ OS ขณะที่ Oracle Server กำลังรันอยู่

- Data file หนึ่ง data file จะเป็นได้แค่ 1 tablespace

- จำนวนสูงสุดของ data file กำหนดโดย MAXDATAFILES โดยที่ขนาดจะมากขึ้นตามขนาดของ tablespace

ชนิดของ tablespace

Database หนึ่งตัว ต้องประกอบด้วยอย่างน้อย 1 tablespace เรียกว่า SYSTEM tablespace เอาไว้สำหรับการควบคุมและการบำรุงรักษาในระยะยาว โดยการแยกชนิดของ tablespace สามารถแยกได้ 2 ส่วนหลักๆคือ

SYSTEM Tablespace ประกอบด้วย

- รายละเอียดเกี่ยวกับ Data dictionary
- System rollbacksegment

Non-System ประกอบด้วย

- Rollback segments
- Temporary segments
- Application data
- Application indexes

4.6 Locking Mechanism

Oracle Server มีการจัดการ Locking แบบอัตโนมัติ โดยมีค่าปกติที่ตั้งไว้ให้ทำการล็อกที่ระดับต่ำเพื่อให้เกิด Data Consistency ขณะที่ให้เกิดระดับสูงสุดของ Data Concurrency

Data Concurrency

การออกแบบการล็อกก็เพื่ออนุญาตให้มี Data Concurrency สูงสุด นั่นคือจะสามารถให้ผู้ใช้งานเข้าถึงข้อมูลเดียวกัน และเวลาเดียวกันได้อย่างปลอดภัย โดยที่การดำเนินการของ Data manipulation Language (DML) จะทำการล็อกในระดับแถวข้อมูล

Data Consistency

Oracle Server ออกแบบมาให้สามารถทำการปรับระดับของ Data Consistency ได้ เพื่อให้เกิดภาพการมองที่คงที่ แม้ว่าข้อมูลจะถูกเปลี่ยนไปจากผู้ใช้งานคนอื่นแล้วก็ตาม

Locking Modes

Exclusive lock mode: จะเป็นการล็อกเพื่อป้องกันทรัพยากรที่สามารถใช้งานร่วมกัน ได้กับ transaction อื่นๆ จนกระทั่งมีการปลด Exclusive lock ออก

Share lock mode: โดยทั่วไปแล้ว transactions สามารถเข้าสู่ Share lock ได้ บนทรัพยากรเดียวกัน

DML Locks

วัตถุประสงค์ของการทำ DML lock ก็เพื่อรับประกันความมั่นคงของข้อมูลที่สามารถเข้าถึงได้พร้อมๆกันจากหลายๆผู้ใช้งาน

DDL Locks

วัตถุประสงค์ของการทำ DDL lock ก็เพื่อป้องกันนิยามของโครงสร้างของวัตถุ ขณะที่วัตถุนั้นขึ้นกับการอ้างถึงโดย DDL operation ในการทำงานนั้น Oracle server จะดำเนินการทำ DDL lock แบบอัตโนมัติ เพื่อป้องกัน DDL operation ที่ทำการเปลี่ยนแปลงหรืออ้างถึง โครงสร้างของวัตถุตัวเดียวกัน

ในบางครั้งการดำเนินการล๊อคอาจเกิดจากการกระทำโดยผู้ใช้งานโดยตรงได้ โดยสามารถใช้คำสั่ง Lock Table ได้

```
SQL> LOCK TABLE s_emp IN exclusive MODE;
```

Table(s) Locked.

ทำให้ในบางครั้งผู้ใช้งานคนอื่นไม่สามารถทำการใช้งานข้อมูลได้ จนกว่าจะมีการปลดล๊อคออกจากผู้ดำเนินการล๊อคเอง ซึ่งอาจก่อให้เกิดปัญหาการเข้าถึงข้อมูลไม่ได้ ดังนั้นจึงมีส่วนของการมองเพื่อให้เห็นว่าการล๊อคเกิดขึ้นอย่างไรบ้างจากการมองที่ V\$LOCK, V\$LOCKED_OBJECT ดังเช่นทำการค้นหาชื่อตารางที่เกี่ยวข้องกับทรัพยากร ID 1 ของ V\$LOCK โดย

```
SQL> SELECT owner, object_id, object_name, object_type, v$lock.type
FROM dab_objects, v$lock
WHERE object_id = v$lock.id1 and object_name = table_name;
```

Deadlocks

Deadlock สามารถเกิดได้เมื่อมีการรอคอยข้อมูลที่ถูกล๊อคจากผู้ใช้งานที่มากกว่าหรือเท่ากับ 2 คน ซึ่งโดยปกติแล้ว Oracle Server จะทำการตรวจจับการเกิด deadlocks และหาวิธีการแก้ปัญหาเอง โดยทำการ rolling back statement ที่ตรวจพบการเกิด deadlocks ในการเกิด deadlock แต่ครั้งจะมีการเขียนบันทึกไว้ใน trace file ใน USER_DUMP_DEST directory ซึ่งจะใช้ชื่อไฟล์ว่า <SID>_ora_<PID>.trc และ Ora<PID>.trc

4.7 การจัดการกับ Sessions

ในบางครั้งผู้ดูแลระบบฐานข้อมูลอาจต้องการเข้าสู่ Restricted session เพื่อประโยชน์เช่น การบำรุงรักษาฐานข้อมูลให้ดีขึ้น หรือการทำ Import/Export ฐานข้อมูล จึงจำเป็นต้องทำการเปลี่ยน ฐานข้อมูลให้เข้าสู่สถานะของ restricted mode เพื่อให้ผู้ดูแลระบบฐานข้อมูลเพียงคนเดียวเท่านั้นที่ จะมีสิทธิในการใช้งานฐานข้อมูล ในการเปลี่ยนแปลงสถานะของฐานข้อมูลนั้นสามารถกระทำได้ โดยใช้คำสั่ง ALTER SYSTEM {ENABLE|DISABLE} RESTRICTED SESSION ซึ่งจะเป็นการ อนุญาตให้ผู้ใช้งานที่เป็น Restricted Session เท่านั้นที่สามารถเข้าถึงฐานข้อมูลได้

การ Terminating Sessions

หลังจากที่ทำการเปลี่ยนสถานะของฐานข้อมูลเป็น restricted แล้ว จำเป็นต้องทำการ Kill All Current Session ทั้งหมด ก่อนที่จะดำเนินการกระทำงานของผู้ดูแลระบบฐานข้อมูล โดยการใช้คำสั่ง ALTER SYSTEM KILL SESSION 'sid, serial'

ผลกระทบหลังจากการ Terminating Sessions

หลังจากทำการ Terminate Session แล้ว background process PMON จะทำการตาม ขั้นตอนดังนี้

- ทำการ roll back transaction ที่ผู้ใช้งานกระทำอยู่
- ทำการลบตารางที่ถืออยู่พร้อมกับสิ่งที่ทำการถืออยู่ทั้งหมด
- คืนทรัพยากรทั้งหมดที่ทำการครอบครองโดยผู้ใช้งานคนนั้น

ในการ Terminating Active Session นั้นจะทำให้ transaction เกิดการ roll back และแจ้งข้อ ความไปทันที “ORA-00028: you session has been killed” ส่วนในการดำเนินการ Terminating Inactive Session นั้นจะไม่เกิดการแจ้งไปทันที แต่ Status Column ของ V\$SESSION เปลี่ยนแปลงเป็น killed และเมื่อ Session นั้นจะพยายามใช้งานอีกทีก็จะเกิด การแจ้งข้อความ “ORA-00028: you session has been killed” กลับไป พร้อมกับทำการลบ แถวที่มีสถานะเป็น killed ออกจาก V\$SESSION ด้วย

บทที่ 5

ความปลอดภัยของข้อมูล

5.1 บทนำ

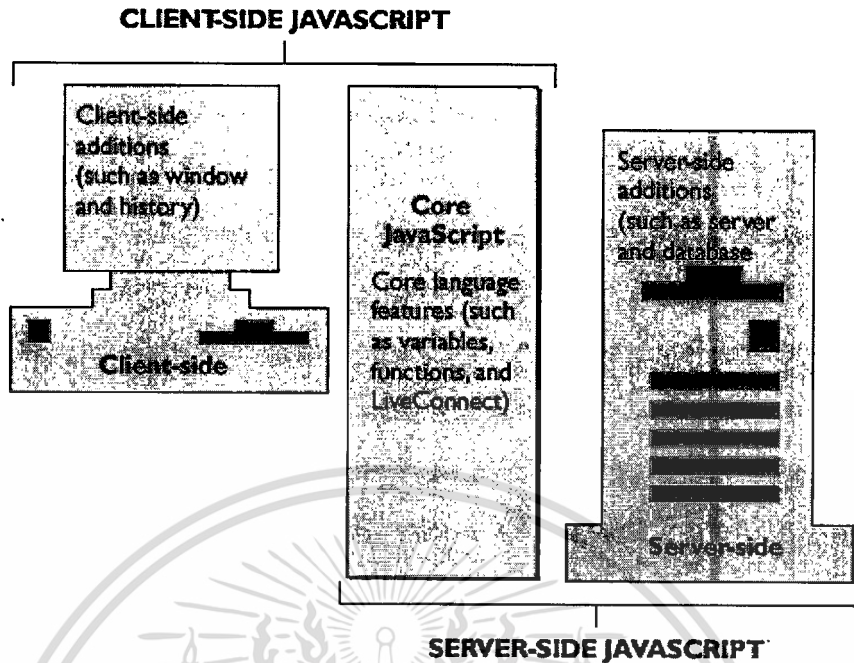
ในการดำเนินการผ่าน Web นั้น สิ่งที่น่าจะให้ความสำคัญมากที่สุดก็คือ รหัสลับของผู้ดูแลระบบฐานข้อมูลที่ส่งออกไปนั้นมีความปลอดภัยมากน้อยเพียงใด เนื่องจากการดำเนินการครั้งนี้ใช้ Java เป็นภาษาหลัก โดยที่ Java Script นั้นจะทำการดำเนินการที่ฝั่ง Client ดังนั้นเราจึงสามารถที่จะทำการเขียน Algorithm เพื่อทำการ Encrypt และ Decrypt โดยจะเป็นข้อตกลงกันระหว่าง Client กับ Server ซึ่งทำให้รหัสลับที่ส่งไปออกไปนั้นถูก Encrypt อยู่ ดังนั้นในการดำเนินการครั้งนี้ ในส่วนของการส่งรหัสผู้ใช้งาน และรหัสลับที่เป็นของผู้ดูแลระบบฐานข้อมูลไปยัง Server จะมีการเข้ารหัสข้อมูลไว้ก่อนที่จะทำการส่งออกไป และเมื่อถึงปลายทางก็จะทำการ Decrypt ด้วยวิธีการเดียวกันกับการเข้ารหัส โดยใช้เทคนิคของการเข้ารหัสแบบ Nihilist Substitution

5.2 แนะนำหลักการทำงานของ JavaScript เบื้องต้น

JavaScript นั้นจะทำให้คุณสามารถสร้างโปรแกรมประยุกต์ที่รันบน Internet ได้ ซึ่งอยู่ในรูปแบบพื้นฐานของ Object ในลักษณะของ Script ที่เป็นภาษาสำหรับ Client/Server โดยที่สามารถสร้าง dynamic HTML และทำการเชื่อมโยงกับ relational database ได้ โดยที่ JavaScript นั้นแบ่งแยกได้เป็น 2 ส่วนคือ

- Client-Side
- Server-Side

โดยที่ทั้ง 2 ฝั่งนั้นจะใช้รูปแบบภาษาเดียวกัน ตัวภาษานั้นบรรจุกลุ่มของ Object หลักๆไว้เช่น Array, Date Object



ภาพที่ 5.1 การแบ่งส่วนของ JavaScript

5.2.1 Client-Side JavaScript

Web Browser เช่น Netscape Navigator 2.0 หรือใหม่กว่า สามารถตีความ Client-Side JavaScript ที่ฝังใน HTML page ได้ เมื่อ Client ทำการร้องขอไปยัง Server จากนั้น Server ก็ทำการส่งเนื้อหาของเอกสารทั้งหมดซึ่งประกอบไปด้วย HTML และ JavaScript มาด้วย โดยผ่านทางเครือข่ายกลับมายัง Client และเมื่อ Client ทำการอ่านข้อความภายในเอกสารจากบนลงล่าง เพื่อจะแสดงผลลัพธ์ของ HTML และพร้อมกับการประมวลผลคำสั่งของ JavaScript ด้วย ดังตัวอย่าง Page HTML ที่ฝัง JavaScript ไว้

```
<HEAD><TITLE>A Simple Document</TITLE>
<SCRIPT>
function update(form) {
    alert("Form being updated")
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="myform" ACTION="start.htm"
METHOD="get">
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Enter a value:

...

</FORM>

</BODY>

5.2.2 Server-Side JavaScript

ที่ Server จะสามารถมี JavaScript ฝังใน HTML Page ได้ ทำให้สามารถติดต่อกับ Relational Database จากหลายตัวแทนจำหน่ายได้ โดยลักษณะของ Script จะทำอยู่ในรูปแบบ Bytecode Executable Files โดยที่สามารถใช้งานในรูปแบบ Client-Side และ Server-Side ร่วมกันได้ การดำเนินการเช่นนี้จำเป็นที่ Web Server ต้องมี JavaScript Runtime Engine เพื่อเหตุผลในการสร้าง JavaScript Application โดยมีกระบวนการทั้งหมด 2 กระบวนการ

กระบวนการที่ 1

ทำการสร้าง HTML Page และ JavaScript Files โดยทำการ Compile files ทั้งหมดลงในส่วนเดียวกัน ที่สามารถทำการประมวลผลได้ เช่นตัวอย่างไฟล์ที่มีนามสกุลเป็น JS นั้นจะทำการแปลงให้อยู่ในรูปของ ByteCode แล้วทำการเก็บไว้ในหน่วยความจำก่อน

กระบวนการที่ 2

เมื่อมีการร้องขอโดย Client แล้วตัว Runtime Engine จะใช้ส่วนที่ได้จากกระบวนการ 1 เป็นตัวดำเนินการเพื่อให้สามารถส่งผลลัพธ์กลับไปยัง Client ได้ โดยสามารถรันทั้ง คำสั่ง JavaScript บน Server-Side ที่พบบน Page แล้วให้ผลลัพธ์ไปยัง HTML ใหม่ หรือว่าเป็นในลักษณะที่ทำการส่งไปยัง HTML Page เพื่อส่งผลลัพธ์ไปบนเครือข่ายแล้วให้ Client ทำการตีความที่ฝั่ง Client เอง

5.3 การเข้ารหัสแบบ Nihilist Substitution

ในการเข้ารหัสแบบ Nihilist Substitution นั้นจะมีรูปแบบของการเข้ารหัสในลักษณะของ Matrix โดยจะถูกกำหนดด้วยแถวและสดมภ์ให้เพียงพอต่อความต้องการของตัวอักษรที่ต้องการใช้ทั้งหมด ดังตัวอย่างแสดงการกำหนดตำแหน่งของตัวอักษรที่ถูกบ่งชี้ด้วยแถวและสดมภ์

	1	2	3	4	5	6
1	U	N	I	T	E	J
2	D	S	A	O	F	-
3	M	R	C	B	G	-
4	H	K	L	P	Q	-
5	V	W	X	Y	Z	-

ในการวางตำแหน่งของตัวอักษรนั้นจะทำการวางในรูปแบบที่สุมขึ้นมา แต่ทั้งนี้และทั้งนั้นจะต้องเข้าใจตรงกันระหว่างผู้ที่ทำการเข้ารหัสและผู้ที่ทำการถอดรหัส

Plain text: G O T
 Cipher text: 35 24 14

นอกจากนี้แล้วยังอาจมีการกำหนด Key เพื่อนำไปใช้ร่วมกับการดำเนินการเข้ารหัสด้วยได้เช่น

Let the KW = 32 13
 Plain: T H E
 Numerical: 14 41 15
 Key: 32 13 32
 Cipher: 46 54 47

จากนั้นจะทำการจัดกลุ่มเป็นชุดของข้อมูลดังเช่น 1 ชุดข้อมูลใช้ 3 ตัวเลข

465 447

บทที่ 6

การพัฒนาโปรแกรม

6.1 ขั้นตอนการเตรียมสภาวะแวดล้อมสำหรับการพัฒนาโปรแกรม

ในการจะทำการพัฒนาโปรแกรมนั้นจำเป็นต้องทำอย่างยั้งที่จะต้องทำการติดตั้งและทำการกำหนดสภาวะแวดล้อมต่างๆของอุปกรณ์ทุกอย่างที่ต้องการใช้งานให้พร้อมก่อนการติดตั้งโดยจะแยกเป็น 3 ส่วนหลักๆในการดำเนินการ โดยติดตั้งและกำหนดแต่ละส่วนตาม tier ทั้งสาม

6.1.1 การติดตั้งและกำหนดค่าให้กับ tier ที่ 1 (Client)

การติดตั้งและกำหนดค่าให้กับ tier ที่ 1 นั้นกระทำได้ง่ายกว่า tier ที่ 2 และ tier ที่ 3 โดยกระทำการติดตั้งให้เครื่องคอมพิวเตอร์ที่ต้องการใช้งานนั้นสามารถดำเนินการเข้าสู่ Internet ได้ (อาจผ่านทาง การ Dial up หรือทาง LAN ก็ได้) โดยผ่านทาง browser ตัวใดตัวหนึ่งก็ได้ เช่น Internet Explorer, Netscape Navigator (ในการดำเนินการครั้งนี้จะทำการติดตั้ง Internet Explorer 5.0) ในการติดต่อกับ tier ที่ 2 ที่เป็นตัวเก็บโปรแกรมนั้น จะกระทำการเรียกผ่าน Web Browser โดยการพิมพ์ URL ซึ่งประกอบด้วยชื่อและ port ของเครื่องที่ต้องการทำการติดต่อ เช่น <http://202.28.159.25:7777> เป็นต้น

6.1.2 การติดตั้งและกำหนดค่าให้กับ tier ที่ 2 (Application Server)

การดำเนินการพัฒนาโปรแกรมนั้น หลังจากดำเนินการพัฒนาเสร็จเรียบร้อยแล้ว จะนำ code ที่ได้จากการดำเนินการมาเก็บไว้ที่ tier นี้ ดังนั้น tier นี้จึงต้องทำหน้าที่เป็น Server ตัวหนึ่งเพื่อให้ tier ที่ 1 เรียกใช้งานได้ จะดำเนินการตามขั้นตอนดังนี้

- ทำการติดตั้ง Microsoft Windows2000 Advanced Server ลงบนเครื่องคอมพิวเตอร์ที่ต้องการจะทำการเป็น tier ที่ 2 ซึ่งทำการกำหนด IP ของเครื่องเป็นดังนี้ 202.28.159.25 Subnet คือ 255.255.255.0 เพื่อทำการจำลองตัวเองเป็น Server ใน tier ที่ 2
- ทำการติดตั้ง Java Development Kit version 1.2.2 for windows ลงบน Windows 2000 สามารถหา download ได้ฟรีจาก www.javasoft.com ซึ่งมีชื่อไฟล์ว่า “jdk1_2_2-win.exe” โดยให้กำหนด Path สำหรับการติดตั้งคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- “D:\jdk122” จากนั้นทำการกำหนด System Environment Path ให้สามารถ execute javac.exe ได้ (Path=D:\jdk122\bin)
- ทำการติดตั้ง JavaServer Web Development Kit version 1.0.1 for windows ลงบน Windows 2000 สามารถหา download ได้ฟรีจาก www.javasoft.com ซึ่งมีชื่อไฟล์ว่า “jswdk1_0_1-win.zip” โดยมีวิธีการติดตั้งโดยทำการ Extract จาก zip file ลงบนพื้นที่ส่วนของ Hard disk ในที่นี้จะกำหนด Path เป็น “D:\jswdk” และจากนั้นให้ทำการแก้ Initial file ที่ชื่อ “webserver.xml” ตามที่ต้องการเช่น

```
<WebServer id="webServer">
  <Service id="service0" port="7777">
    <WebApplication id="oratoool" mapping="/oratoool"
      docBase="oratoool"/>
  </Service>
</WebServer>
```

และเมื่อต้องการทำการ start jswdk service ก็สามารถกระทำได้โดยการเรียกใช้งานไฟล์ที่ชื่อ startserver.bat ในทำนองเดียวกัน เมื่อต้องการทำการ stop jswdk service ก็ทำการเรียกใช้งานไฟล์ที่ชื่อ stopserver.bat

6.1.3 การติดตั้งและกำหนดค่าให้กับ tier ที่ 3 (Database Server)

การติดตั้งและเตรียมความพร้อมของระบบสำหรับส่วน tier ที่ 3 นั้นจะเป็นส่วนของการให้บริการฐานข้อมูล โดยฐานข้อมูลที่ใช้ก็จะเป็น Oracle Server 8.0.5 for Windows NT สามารถกระทำได้โดยการดำเนินการหลักๆ 2 ขั้นตอน

- ทำการติดตั้ง Microsoft Windows2000 Advanced Server ลงบนเครื่องคอมพิวเตอร์ที่ต้องการจะทำเป็น tier ที่ 3 ซึ่งทำการกำหนด IP ของเครื่องเป็นดังนี้ 202.28.159.26 Subnet คือ 255.255.255.0 เพื่อทำการจำลองตัวเองเป็น Database Server ใน tier ที่ 3
- ทำการติดตั้ง DBMS (Oracle Server 8.0.5 for Windows NT) ไว้ที่ Path “E:\orant” ซึ่งในการติดตั้งจะทำการใช้ค่า default ทุกอย่าง จากนั้นให้ทำการ Restart เครื่องใหม่ เพียงเท่านั้นเครื่องก็พร้อมสำหรับการรองรับการร้องขอจาก tier ที่ 2

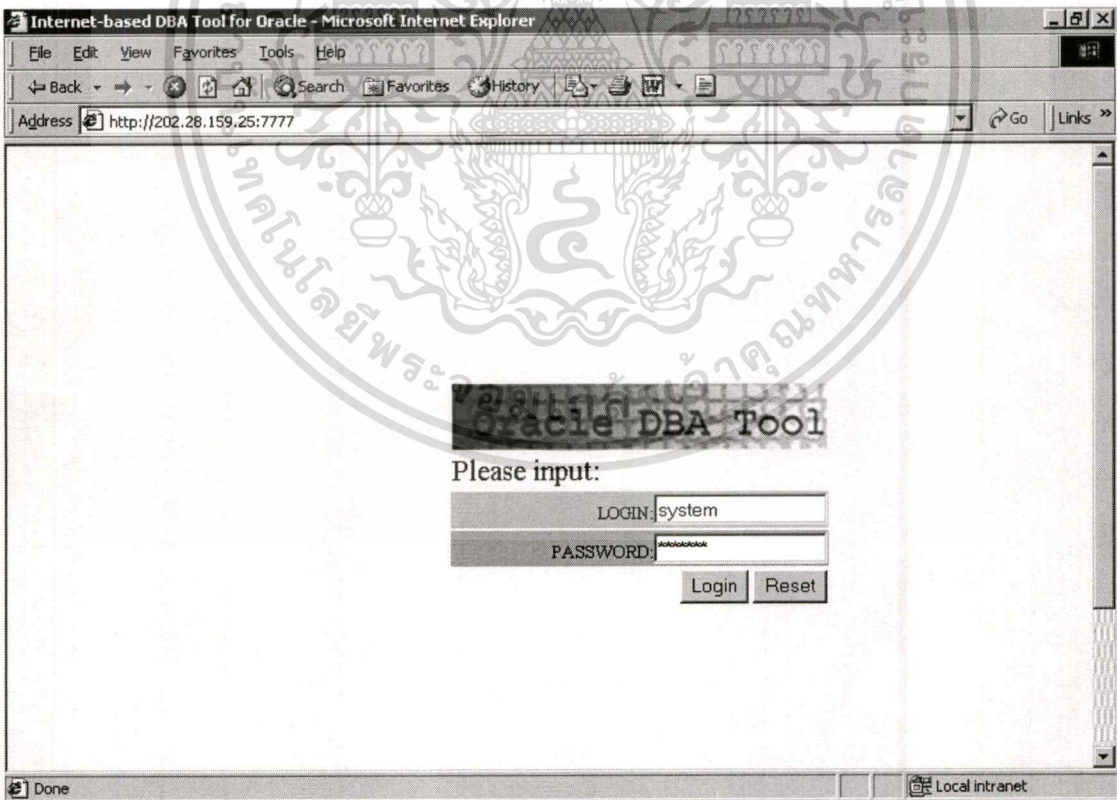
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการติดต่อและการสั่งการต่างๆจาก tier ที่ 2 ไปยัง tier ที่ 3 นั้นจะใช้ JDBC Driver เป็นตัว Engine หลัก แสดงส่วนของโปรแกรมในการเชื่อมต่อได้ดังนี้

```
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@202.28.159.26:1521:orcl",login,
passwd);
Statement stmt = conn.createStatement ();
```

6.2 ขั้นตอนในการพัฒนาโปรแกรมในส่วนของการออกแบบหน้าจอ

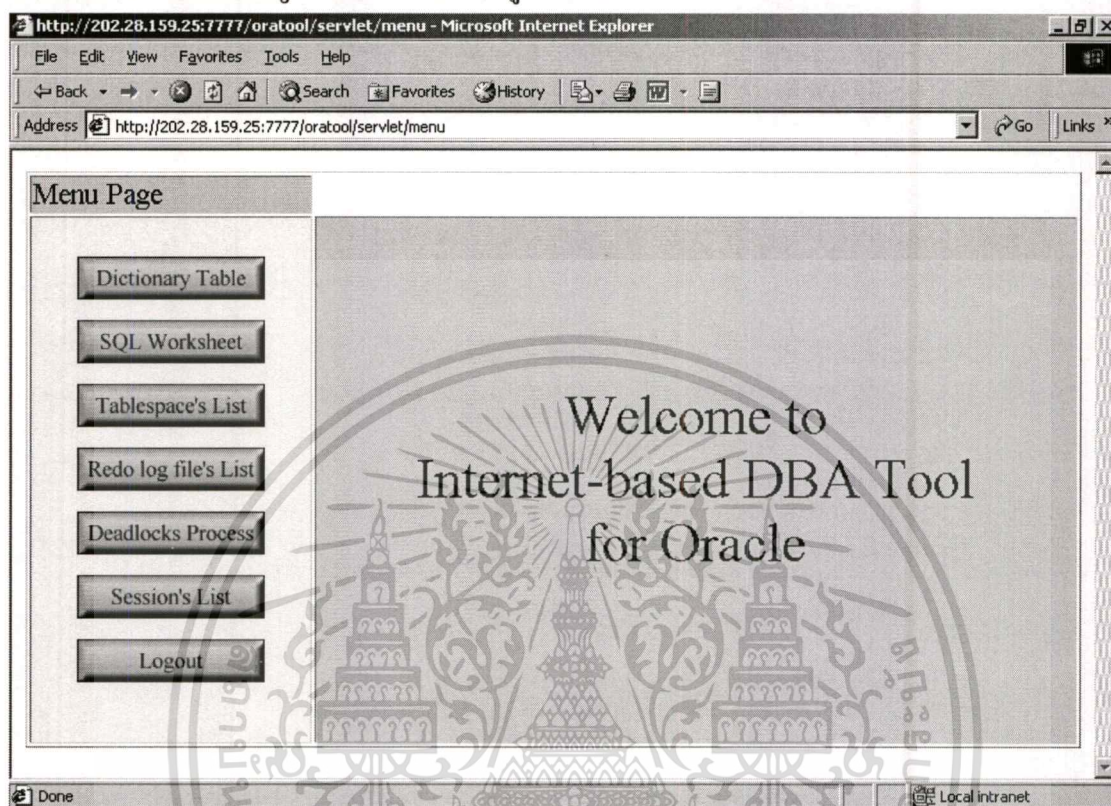
เมื่อทำการเรียกโปรแกรม Oracle DBA Tool โดยใส่ URL ผ่านทาง Web Browser จะได้หน้าจอแรกของการเข้าสู่ระบบ ดังแสดงในภาพที่ 6.1 ซึ่งมีหน้าจอเหมือนกับระบบทั่วไปที่จะต้องมีกรใส่ Login และ Password เพื่อให้แน่ใจว่าผู้ที่เข้ามาใช้งานนั้นเป็นผู้ที่มีสิทธิ์ในการใช้งานระบบ โดย Login และ Password ที่ใช้จะเป็นของ DBA เท่านั้น



ภาพที่ 6.1 แสดงหน้าจอการ Login เข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้าสู่ระบบได้ ก็จะพบกันหน้าจอเมนูหลักในการทำงานของโปรแกรม โดยแบ่งเป็นเมนูหลักๆ สำหรับการจัดการกับปัญหาที่เกิดขึ้น แบ่งเมนูออกได้ดังแสดงดังภาพที่ 6.2



ภาพที่ 6.2 แสดงหน้าจอหลักของโปรแกรม

สามารถแบ่งเมนูหลักๆ ได้ดังนี้

- Dictionary Table
- SQL Worksheet
- Tablespace's List
- Redo log file's List
- Deadlocks Process
- Session's List
- Logout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dictionary Table Page

ทำหน้าที่ในการแสดงชื่อตารางทั้งหมดที่อยู่ภายใต้ตาราง Dict ของ Oracle แสดงดังภาพที่ 6.3 ซึ่งตารางต่างๆเหล่านี้จะเก็บค่า Configuration ต่างๆ พร้อมกับสถานการณ์การทำงานของระบบไว้

Column1	Column2	Column3
ALL_ALL_TABLES	ALL_ARGUMENTS	ALL_CATALOG
ALL_CLUSTERS	ALL_CLUSTER_HASH_EXPRESSIONS	ALL_COLL_TYPES
ALL_COL_COMMENTS	ALL_COL_PRIVS	ALL_COL_PRIVS_MADE
ALL_COL_PRIVS_RECD	ALL_CONSTRAINTS	ALL_CONS_COLUMNS
ALL_DB_LINKS	ALL_DEF_AUDIT_OPTS	ALL_DEPENDENCIES
ALL_DIRECTORIES	ALL_ERRORS	ALL_INDEXES
ALL_IND_COLUMNS	ALL_IND_PARTITIONS	ALL_LIBRARIES
ALL_LOBS	ALL_METHOD_PARAMS	ALL_METHOD_RESULTS
ALL_NESTED_TABLES	ALL_OBJECTS	ALL_OBJECT_TABLES
ALL_PART_COL_STATISTICS	ALL_PART_HISTOGRAMS	ALL_PART_INDEXES
ALL_PART_KEY_COLUMNS	ALL_PART_TABLES	ALL_PROBE_OBJECTS
ALL_REFRESH	ALL_REFRESH_CHILDREN	ALL_REFS
ALL_REGISTERED_SNAPSHOTS	ALL_REPAIDIT_ATTRIBUTE	ALL_REPAIDIT_COLUMN

ภาพที่ 6.3 แสดงหน้าจอส่วนของ Dictionary Table Page

DBA สามารถที่จะคลิกที่ชื่อของตารางใดก็ได้เพื่อให้ปรากฏข้อมูลภายในตารางนั้นๆ ดังแสดงในภาพที่ 6.4 เพื่อให้ DBA สามารถที่จะพิจารณาสถานะต่างๆของระบบก่อนที่จะดำเนินการแก้ปัญหาได้

Dictionary Table dicShow from doPost - Microsoft Internet Explorer

Address http://202.28.159.25:7777/oratool/servlet/dicShow

Table GV\$EVENT_NAME have 6 columns show on below.

INST_ID	EVENT#	NAME	PARAMETER1	PARAMETER2	PARAMETER3
1	0	Null event	null	null	null
1	1	latch activity	address	number	process#
1	2	latch free	address	number	tries
1	3	free process state object	null	null	null
1	4	pmon timer	duration	null	null
1	5	inactive session	session#	waited	null
1	6	lock manager wait for remote message	waittime	null	null
1	7	wait for DLM latch	latchtype	gets	immediate
1	8	wait for influx DLM latch	latchtype	latchaddr	null
1	9	DFS lock handle	type mode	id1	id2
1	10	lock manager wait for dynamic allocation	waittime	null	null
1	11	lock manager wait for dead process cleanup from LMON	waittime	null	null

Done Local intranet

ภาพที่ 6.4 แสดงหน้าจอส่วนของการเรียกดูข้อมูลตารางภายใน GV\$EVENT_NAME

SQL Worksheet Page

ทำหน้าที่ในการเรียกดูข้อมูลต่างๆที่นอกเหนือจากในตารางที่มีใน dict table (หรืออาจเป็นการเรียกดูแบบมีเงื่อนไขตามที่ DBA ต้องการ) ด้วยคำสั่ง SQL แสดงดังภาพที่ 6.5

SQL Worksheet - Microsoft Internet Explorer

Address http://202.28.159.25:7777/oratool/servlet/sqlMenu

SQL Worksheet Page

Please input SQL Command

select * from dictionary

Execute

Done Local intranet

ภาพที่ 6.5 แสดงหน้าจอส่วนของ SQL Worksheet Page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายหลังจากการใส่ SQL Command แล้วทำการ Execute ก็จะได้แสดงผลดังภาพที่ 6.6

Your SQL command is "select * from dictionary"

TABLE_NAME	COMMENTS
ALL_ALL_TABLES	Description of all object and relational tables accessible to the user
ALL_ARGUMENTS	Arguments in object accessible to the user
ALL_CATALOG	All tables, views, synonyms, sequences accessible to the user
ALL_CLUSTERS	Description of clusters accessible to the user
ALL_CLUSTER_HASH_EXPRESSIONS	Hash functions for all accessible clusters
ALL_COLL_TYPES	Description of named collection types accessible to the user
ALL_COL_COMMENTS	Comments on columns of accessible tables and views
ALL_COL_PRIVS	Grants on columns for which the user is the grantor, grantee, owner, or an enabled role or PUBLIC is the grantee
ALL_COL_PRIVS_MADE	Grants on columns for which the user is owner or grantor
ALL_COL_PRIVS_RECD	Grants on columns for which the user, PUBLIC or enabled role is the grantee
ALL_CONSTRAINTS	Constraint definitions on accessible tables
ALL_CONS_COLUMNS	Information about accessible columns in constraint definitions

Done Local intranet

ภาพที่ 6.6 แสดงหน้าจอส่วนผลลัพธ์ที่ได้จากการ Execute SQL Command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tablespace's List Page

ทำหน้าที่ในการแสดงให้เห็นถึง จำนวน tablespace ทั้งหมดที่มีอยู่ในระบบฐานข้อมูล แสดงในภาพที่ 6.7 พร้อมกับ Edit เพื่อใช้ในการแก้ไขขนาดของ Tablespace โดยให้ DBA ใส่ขนาดใหม่ที่ต้องการลงไปในช่วง Size ดังแสดงในภาพที่ 6.8 และแสดงผลลัพธ์ของการขยายขนาดแสดงดังภาพที่ 6.9

Tablespace's List Page

TABLESPACE_NAME	DATAFILE_NAME	AUTOEXTENSIBLE	INCREMENT_BY	STATUS	MEGABYTES
USER_DATA	E:\ORANT\DATABASE\USRSR1ORCL.ORA	YES	2560	AVAILABLE	10
ROLLBACK_DATA	E:\ORANT\DATABASE\RSR1ORCL.ORA	YES	2560	AVAILABLE	10
TEMPORARY_DATA	E:\ORANT\DATABASE\SE\TMP1ORCL.ORA	YES	2560	AVAILABLE	3
SYSTEM	E:\ORANT\DATABASE\SYSTEM1ORCL.ORA	YES	5120	AVAILABLE	59

	AUTOEXTENSIBLE	INCREMENT BY	STATUS	MEGABYTES	BLOCKS	MAXMEGABYTES	MAXBLOCKS	
ORA	YES	2560	AVAILABLE	10	5120	110	56320	Edit Row
ORA	YES	2560	AVAILABLE	10	5120	150	76800	Edit Row
ORA	YES	2560	AVAILABLE	3	1536	150	76800	Edit Row
ORA	YES	5120	AVAILABLE	59	30208	199	101888	Edit Row

MENU

ภาพที่ 6.7 แสดงหน้าจอส่วนของ Tablespace's List Page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The image displays two screenshots of a web application interface for managing tablespaces. The top screenshot shows a table titled "Tablespace's List for update page" with the following data:

TABLESPACE_NAME	DATAFILE_NAME	AUTOEXTENSIBLE	INCREMENT_BY	STATUS	MEGABYTES
USER_DATA	E:\ORANT\DATABASE\USR1\ORCL.ORA	YES	2560	AVAILABLE	10

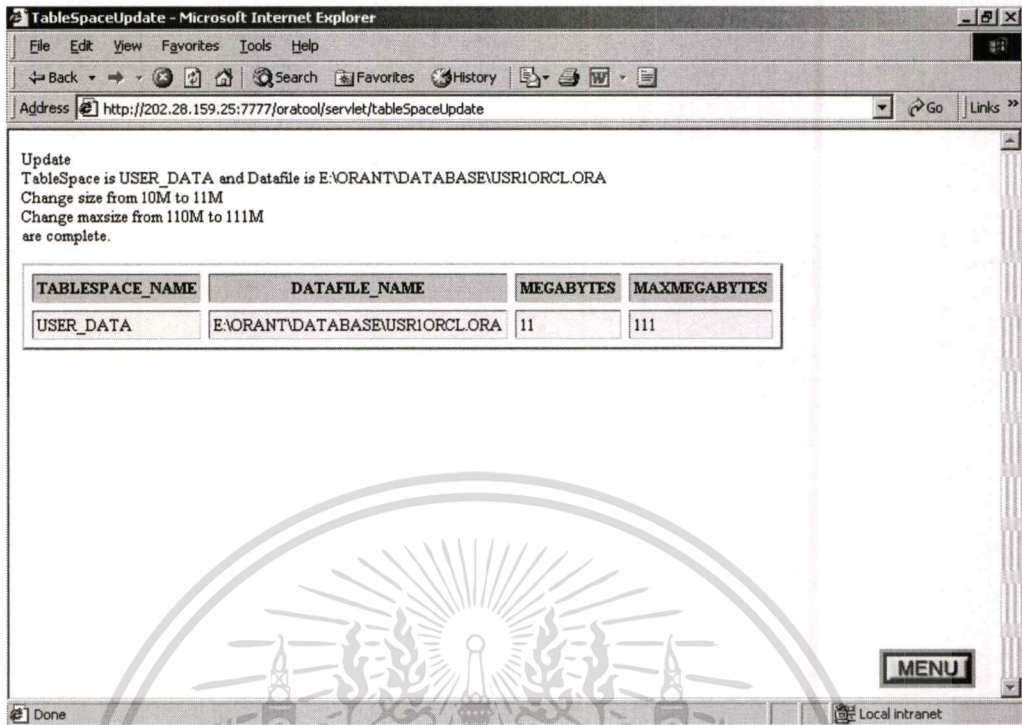
The bottom screenshot shows a detailed view of the tablespace with the following data:

AUTOEXTENSIBLE	INCREMENT_BY	STATUS	MEGABYTES	BLOCKS	MAXMEGABYTES	MAXBLOCKS
YES	2560	AVAILABLE	<input type="text"/>	5120	<input type="text"/>	56320

Both screenshots include a "MENU" button and a status bar at the bottom indicating "Local intranet".

ภาพที่ 6.8 แสดงหน้าจอส่วนของการเปลี่ยนแปลงขนาด Tablespace

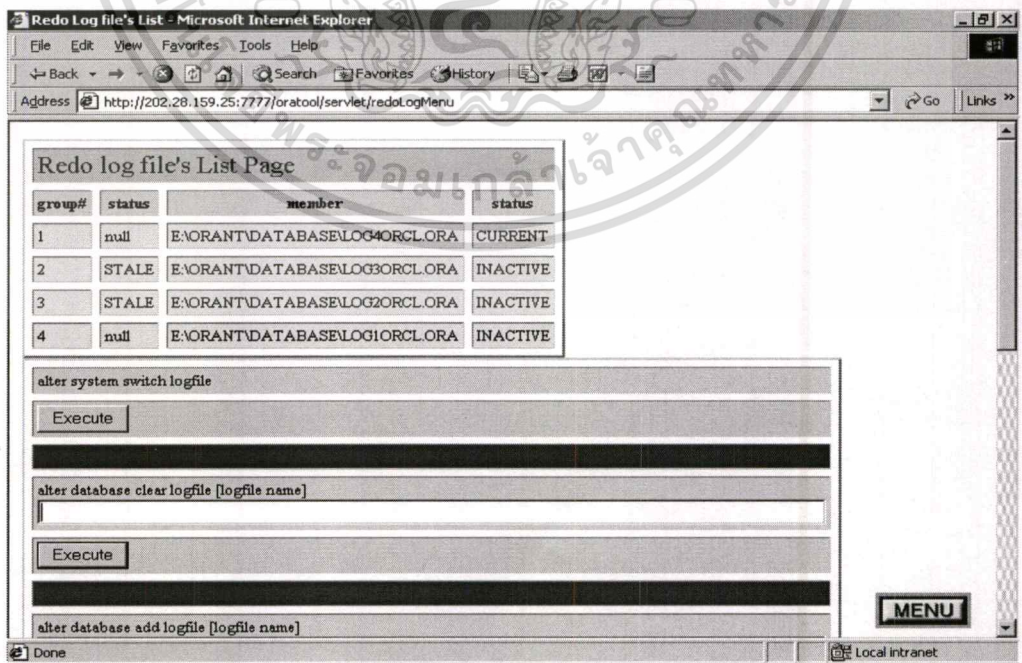
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.9 แสดงหน้าจอผลของการเปลี่ยนแปลงขนาด Tablespace

Redo log file's List Page

ทำหน้าที่ในการแสดงให้เห็นถึงจำนวน Redo log ทั้งหมดที่มีอยู่พร้อมกับขนาดของ Redo log แต่ละตัวแสดงดังภาพที่ 6.10



ภาพที่ 6.10 แสดงหน้าจอส่วนของ Redo Log File's List Page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dead Lock Process Page

ทำหน้าที่ในการดูเหตุการณ์ที่ทำให้เกิด Dead Lock ภายในระบบ โดยเหตุการณ์ที่เกิดขึ้นนั้นมีได้หลายกรณี จึงมีส่วนที่ให้ DBA สามารถใส่ SQL Command เพื่อใช้ในการแก้ปัญหาเฉพาะเจาะจงลงไป ดังแสดงในภาพที่ 6.11

The screenshot shows the 'Deadlocks Process Page' in a Microsoft Internet Explorer browser window. The address bar shows the URL: `http://202.28.159.25:7777/oratool/servlet/deadLockMenu`.

The main content area contains a table titled 'Deadlocks Process Page' with the following columns: EVENT, TOTAL_WAITS, TOTAL_TIMEOUTS, TIME_WAITED, and AVERAGE_WAIT. Below the table is a 'MENU' button.

EVENT	TOTAL_WAITS	TOTAL_TIMEOUTS	TIME_WAITED	AVERAGE_WAIT
Null event	1	1	0	0
latch free	10	9	0	0
pmon timer	7750	7746	0	0
process startup	8	0	0	0
rdbms ipc reply	14	0	0	0
rdbms ipc message	30057	29956	0	0
enqueue	1	0	0	0
control file sequential read	397	0	0	0
control file parallel write	7784	0	0	0
refresh controlfile command	38	0	0	0
checkpoint completed	1	0	0	0
buffer busy waits	1	0	0	0

Below the table is a 'MENU' button.

The second screenshot shows the 'Deadlocks Process Page' in a Microsoft Internet Explorer browser window. The address bar shows the URL: `http://202.28.159.25:7777/oratool/servlet/deadLockMenu`.

The main content area contains a table with the following columns: SID, SEQ#, EVENT, WAIT_TIME, and STATE. Below the table is an 'Input SQL for solve problem' section with a text input field and an 'Execute' button. A 'MENU' button is also present.

SID	SEQ#	EVENT	WAIT_TIME	STATE
1	7751	pmon timer	0	WAITING
2	8401	rdbms ipc message	0	WAITING
3	8511	rdbms ipc message	0	WAITING
6	21	rdbms ipc message	0	WAITING
8	1	rdbms ipc message	0	WAITING
7	1	rdbms ipc message	0	WAITING
4	16229	rdbms ipc message	0	WAITING
5	191	smon timer	0	WAITING
9	42	SQL*Net message from client	-2	WAITED UNKNOWN TIME

Below the table is an 'Input SQL for solve problem' section with a text input field and an 'Execute' button. A 'MENU' button is also present.

ภาพที่ 6.11 แสดงหน้าจอส่วนของ Deadlocks Process page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Session's List Page

ทำหน้าที่ดู session ทั้งหมดที่ทำการเชื่อมต่อกับฐานข้อมูล โดยที่ DBA สามารถที่จะทำการลบตัว session นั้นออกจากระบบได้ถ้าต้องการ ดังแสดงในภาพที่ 6.12

The screenshot shows a web browser window titled 'Session's List - Microsoft Internet Explorer'. The address bar shows the URL: <http://202.28.159.25:7777/oratool/servlet/sessionMenu>. The main content area displays a table with the following data:

SID	SERIAL#	USERNAME	MACHINE	TERMINAL	PROGRAM	LOGON_TIME	
9	237	SCOTT	COM\PRADIT	PRADIT	D:\Documents and Settings\All Users\Start Menu\Programs\Oracle E	2000-09-19 18:50:16.0	Kill
10	105	SYSTEM	D_STUDENT\STUDENT	STUDENT	SQLPLUSW.EXE	2000-09-19 18:51:15.0	Kill
11	299	SYSTEM	COM\PRADIT	PRADIT	D:\Documents and Settings\All Users\Start Menu\Programs\Oracle F	2000-09-19 18:50:28.0	Kill
14	11	SYSTEM	COM\PRADIT	PRADIT	D:\Documents and Settings\All Users\Start Menu\Programs\Oracle E	2000-09-19 18:51:48.0	Kill

At the bottom right of the page, there is a button labeled 'MENU'.

ภาพที่ 6.12 แสดงหน้าจอส่วนของ Session's List Page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การดำเนินการในส่วนการเข้ารหัส

การดำเนินการในส่วนนี้จะเป็นการดำเนินการเพื่อให้รหัสลับของผู้ดูแลระบบฐานข้อมูลถูกเข้ารหัสก่อนที่จะทำการส่งไปยัง Server โดยจะทำการสร้าง Array 2 มิติ ขนาด 16*16 เพื่อให้สามารถเก็บ ASCII Code ได้เป็นจำนวน 256 ตัว (0-255) โดยที่ค่าภายใน Array จะเก็บอยู่ในลักษณะตัวเลขดังนี้

J\I	0	1	2	3	4	5	...	15
0	0	16	32	48	64	80	...	240
1	1	17	33	49	65	81	...	241
2	2	18	34	50	66	82	...	242
3	3	19	35	51	67	83	...	243
4	4	20	36	52	68	84	...	244
5	5	21	37	53	69	85	...	245
...
15	15	31	47	63	79	95	...	255

ตารางที่ 6.1 การแทนค่า ASCII CODE ที่ใช้ในการเข้าและถอดรหัส

ในการใช้งานจริงจะมีการกำหนด Key ให้กับ I, J และ Key สำหรับความยาวของ Login และ Password ด้วย เมื่อจะทำการเข้ารหัส Login และ Password ก็จะทำการเก็บค่าความยาวของ Login และ Password ไว้ในตัวแปรก่อน แล้วจึงทำการรวม Login และ Password เข้าด้วยกัน จากนั้นจะทำการแยกค่าเป็นตัวอักษรแล้วจึงทำการแปลงเป็น ASCII CODE เพื่อนำไปเปรียบเทียบกับค่าในตาราง ASCII ด้านบนเพื่อทำการหาค่า I, J ของแต่ละตัวอักษร โดยดำเนินการทีละตัวอักษรจนครบทั้งหมด ซึ่งหลังจากทำการเข้ารหัสแล้วจะได้สายอักขระที่เข้ารหัสเรียบร้อยแล้วเพียงเส้นเดียวที่จะทำการส่งออกไป ตัวอย่างเช่น

Key ของความยาวคือ 109

Key ของ I คือ 109

Key ของ J คือ 1011

ดังนั้นสายอักขระที่ได้จากการเข้ารหัสของ

Login: system

Password: manager

หลังจากการเข้ารหัสคือ

115116116101411610201161014116101511510161151024115102411510121151025115101211
5101811510161161013



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการค้นคว้าและการพัฒนาโปรแกรม

7.1 ผลการดำเนินการพัฒนาโปรแกรม

การดำเนินการในครั้งนี้ได้บรรลุวัตถุประสงค์ตามที่ตั้งไว้ทุกประการ โดยได้ทำการพัฒนาโปรแกรมตามความต้องการที่กำหนดไว้ ซึ่งทำให้ได้โปรแกรมที่สามารถให้ผู้ดูแลระบบฐานข้อมูลได้ใช้ในการจัดการบริหารและจัดการฐานข้อมูลได้บางส่วนของงานทั้งหมดของผู้ดูแลระบบฐานข้อมูลที่สามารถดำเนินการผ่านเครือข่ายอินเทอร์เน็ตได้ โดยส่วนของงานหลักๆทั้งหมดที่มีจะประกอบด้วย

- สามารถเข้าดูข้อมูลในส่วนของ Dictionary Table ทั้งหมดของระบบได้
- สามารถส่งคำสั่ง SQL เพื่อทำการสั่งงานด้วยภาษา SQL ไปยังเครื่องที่ทำเป็นฐานข้อมูลได้
- สามารถเรียกดูขนาดของ Tablespaces ทั้งหมดที่ประกอบกันเป็นฐานข้อมูลได้ พร้อมทั้งทำการปรับขนาดได้ตามความเหมาะสม
- สามารถเรียกดูขนาดของ Redo Log Files ทั้งหมดที่เป็นส่วนประกอบของฐานข้อมูลได้ พร้อมทั้งทำการปรับขนาดได้ตามความเหมาะสม
- สามารถเรียกดู Process ทั้งหมดที่เกิดการคอยกันของทรัพยากรได้ พร้อมกับทำการแก้ไขด้วยคำสั่ง SQL ได้
- สามารถเรียกดู Sessions ทั้งหมดที่เกิดขึ้นในระบบที่ผู้ดูแลระบบฐานข้อมูลสามารถทำการ Terminate ได้

7.2 ประโยชน์ที่ได้รับจากการพัฒนาโปรแกรม

ประโยชน์ที่ผู้พัฒนาได้รับ

ในการดำเนินการครั้งนี้ข้าพเจ้าได้ประโยชน์จากการศึกษาถึงหน้าที่การทำงานของผู้ดูแลระบบฐานข้อมูล ทำการศึกษาถึงโครงสร้างของการเขียนโปรแกรมในรูปแบบภาษา Java Servlet เพื่อให้สามารถใช้งานในการเขียนโปรแกรมในครั้งนี้ได้เป็นอย่างดี โดยเน้นหนักไปทางด้านการศึกษาเกี่ยวกับฐานข้อมูล Oracle Server รวมถึงหลักการต่างๆเกี่ยวกับการดำเนินการเขียนโปรแกรมในรูปแบบ Web Application และการเข้ารหัสแบบง่ายที่สามารถนำมาประยุกต์ใช้งานได้จริง เพื่อความปลอดภัยในการส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตได้ในระดับหนึ่ง

ประโยชน์จากโปรแกรมที่พัฒนา

โปรแกรมที่พัฒนาขึ้นมานั้นสามารถนำไปใช้งานได้จริงในเชิงปฏิบัติตามองค์กรและบริษัทต่างๆ ซึ่งทำให้ผู้ดูแลระบบฐานข้อมูลสามารถแก้ไขปัญหาได้อย่างรวดเร็วโดยใช้เครือข่ายอินเทอร์เน็ตเป็นสื่อในการเข้าถึงข้อมูลและทำการแก้ไขปัญหาได้ โดยที่ในปัจจุบันจะเห็นว่าอินเทอร์เน็ตมีบทบาทมากขึ้น ดังนั้นจึงทำให้ไม่จำเป็นที่จะต้องมีการมีผู้ดูแลระบบฐานข้อมูลคอยดูแลอยู่ตลอดเวลา

7.3 ข้อเสนอแนะ

ในการพัฒนาครั้งนี้ได้ทำการดำเนินการพัฒนาในส่วนที่ได้ทำการวิเคราะห์และดำเนินการในส่วนหน้าที่หลักๆของผู้ดูแลระบบฐานข้อมูลเท่านั้น เนื่องจากในการศึกษาและดำเนินการครบทุกหน้าที่ของผู้ดูแลระบบฐานข้อมูลนั้นจำเป็นจะต้องใช้เวลาในการดำเนินการค่อนข้างยาวนาน ข้าพเจ้าจึงได้ทำการตัดงานบางส่วนออก ดังเช่น การทำการสำรองข้อมูลบนDISK (Backup) ฯลฯ ดังนั้นถ้าผู้ที่สนใจในการดำเนินการครั้งนี้ สามารถดำเนินการเพิ่มเติมหรือปรับเปลี่ยนในส่วนที่ต้องการได้ เพื่อให้ได้โปรแกรมตรงตามวัตถุประสงค์ที่ต้องการใช้งานภายในองค์กรของท่านได้เป็นอย่างดี

บรรณานุกรม

Jim, D. et. al. 1998. Oracle8: Performance Tuning Workshop.

Karanjit S. and James L. 1997. Inside JAVA. IN.: New Riders Publishing.

Merlin, H. et. al. 1997. Java Network Programming. CT.: Manning Publications Co.

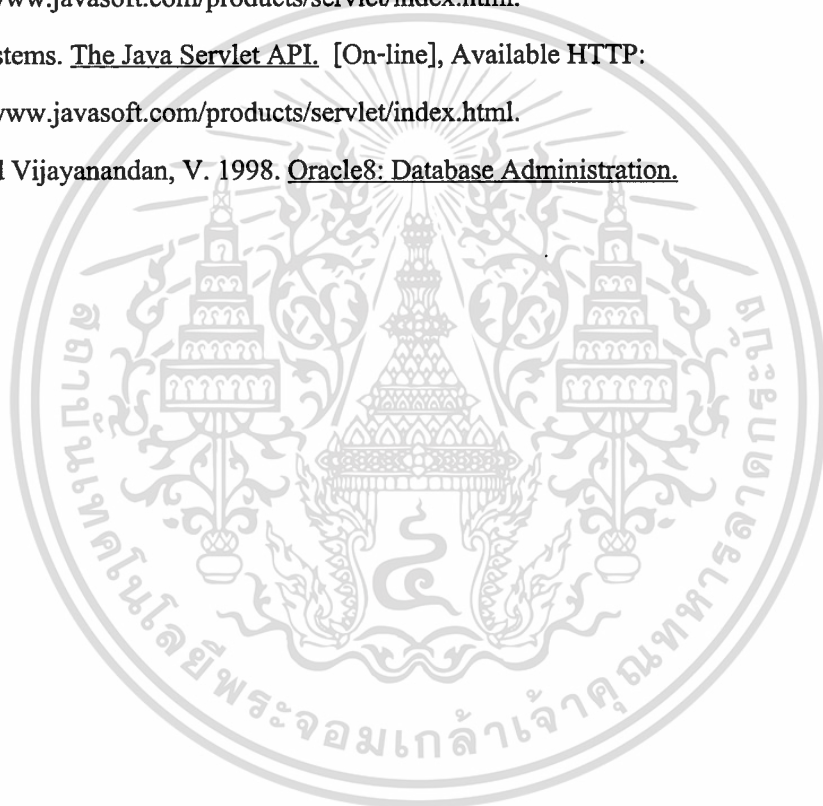
Sun Microsystems. Introduction to Servlets. [On-line], Available HTTP:

<http://www.javasoft.com/products/servlet/index.html>.

Sun Microsystems. The Java Servlet API. [On-line], Available HTTP:

<http://www.javasoft.com/products/servlet/index.html>.

Ulrike, S. and Vijayanandan, V. 1998. Oracle8: Database Administration.





ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก. ตัวอย่างรูปแบบการเขียนโปรแกรมในลักษณะของ Servlet

Source Code for Hello World Example

```

/* $Id: HelloWorldExample.java,v 1.2 1999/03/17 02:21:36 duncan Exp $
 *
 */
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
/**
 * The simplest possible servlet.
 *
 * @author James Duncan Davidson
 */
public class HelloWorldExample extends HttpServlet {

    ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");

    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("<html>");

        out.println("<body bgcolor=\"white\">");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out.println("<head>");
```

```
String title = rb.getString("helloworld.title");
```

```
out.println("<title>" + title + "</title>");
```

```
out.println("</head>");
```

```
out.println("<body>");
```

```
// note that all links are created to be relative. this
```

```
// ensures that we can move the web application that this
```

```
// servlet belongs to to a different place in the url
```

```
// tree and not have any harmful side effects.
```

```
out.println("<a href='../servlets/helloworld.html'>");
```

```
out.println("<img src='../images/code.gif' height=24 " +
```

```
"width=24 align=right border=0 alt='view code'></a>");
```

```
out.println("<a href='../servlets/index.html'>");
```

```
out.println("<img src='../images/return.gif' height=24 " +
```

```
"width=24 align=right border=0 alt='return'></a>");
```

```
out.println("<h1>" + title + "</h1>");
```

```
out.println("</body>");
```

```
out.println("</html>");
```

```
}
```

```
}
```

Source Code for Request Info Example

```

/* $Id: RequestInfoExample.java,v 1.2 1999/03/17 02:36:51 duncan Exp $
 *
 */

import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
 * Example servlet showing request information.
 *
 * @author James Duncan Davidson <duncan@eng.sun.com>
 */
public class RequestInfoExample extends HttpServlet {

    ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

out.println("<html>");
out.println("<body>");
out.println("<head>");

String title = rb.getString("requestinfo.title");
out.println("<title>" + title + "</title>");
out.println("</head>");
out.println("<body bgcolor=\"white\">");

// img stuff not req'd for source code html showing
// all links relative!

out.println("<a href=\"../servlets/reqinfo.html\">");
out.println("<img src=\"../images/code.gif\" height=24 " +
    "width=24 align=right border=0 alt=\"view code\"></a>");
out.println("<a href=\"../servlets/index.html\">");
out.println("<img src=\"../images/return.gif\" height=24 " +
    "width=24 align=right border=0 alt=\"return\"></a>");

out.println("<h3>" + title + "</h3>");
out.println("<table border=0><tr><td>");
out.println(rb.getString("requestinfo.label.method"));
out.println("</td><td>");
out.println(request.getMethod());
out.println("</td></tr><tr><td>");
out.println(rb.getString("requestinfo.label.requesturi"));
out.println("</td><td>");
out.println(request.getRequestURI());
out.println("</td></tr><tr><td>");
out.println(rb.getString("requestinfo.label.protocol"));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

out.println("</td><td>");
out.println(request.getProtocol());
out.println("</td></tr><tr><td>");
out.println(rb.getString("requestinfo.label.pathinfo"));
out.println("</td><td>");
out.println(request.getPathInfo());
out.println("</td></tr><tr><td>");
out.println(rb.getString("requestinfo.label.remoteaddr"));
out.println("</td><td>");
out.println(request.getRemoteAddr());
out.println("</table>");
}

public void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws IOException, ServletException
{
    doGet(request, response);
}
}

```

ประวัติผู้เขียน

ชื่อผู้เขียน นายประดิษฐ์ วัฒนสุวรรณ
 วันเกิด 4 มิถุนายน 2518
 สถานที่เกิด กรุงเทพมหานคร
 วุฒิกการศึกษาระดับปริญญาตรี วท.บ. (วิทยาการคอมพิวเตอร์)
 คณะวิทยาศาสตร์
 มหาวิทยาลัยศิลปากร

ตำแหน่ง ผู้ช่วยผู้อำนวยการฝ่ายสารสนเทศ
 สถานที่ทำงาน โครงการวิทยาลัยการจัดการ มหาวิทยาลัยมหิดล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้