

การพัฒนาเครื่องมือเพื่อการจัดการฐานข้อมูลสำรอง



นายคณิต เลิศศรีบัณฑิต

สถาบันวิทยบริการ  
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2545

ISBN 974-17-1412-2

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF A TOOL FOR MANAGING A STANDBY DATABASE



MR. KANIT LERTSRIBANDIT

สถาบันวิทยบริการ

A Thesis Submitted in Partial Fulfillment of the Requirments

จุฬาลงกรณ์มหาวิทยาลัย  
for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2002

ISBN 974-17-1412-2

หัวข้อวิทยานิพนธ์      การพัฒนาเครื่องมือเพื่อการจัดการฐานข้อมูลสำรอง  
โดย                              นายคณิต เลิศศรีบัณฑิต  
สาขาวิชา                      วิทยาศาสตร์คอมพิวเตอร์  
อาจารย์ที่ปรึกษา              ผู้ช่วยศาสตราจารย์ บุญชัย ไสวรรณวงษ์กุล

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้  
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดี คณะวิศวกรรมศาสตร์  
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
(อาจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์)

..... อาจารย์ที่ปรึกษา  
(ผู้ช่วยศาสตราจารย์ บุญชัย ไสวรรณวงษ์กุล)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ วิวัฒน์ วัฒนาวุฒิ)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.พรศิริ หมั่นไชยศรี)

คณิต เลิศศรีบัณฑิต : การพัฒนาเครื่องมือเพื่อการจัดการฐานข้อมูลสำรอง .(DEVELOPMENT OF A TOOL FOR MANAGING A STANDBY DATABASE) อ.ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ บุญชัย ไสวรรณวณิชกุล , 125 หน้า . ISBN 974-17-1412-2.

วิทยานิพนธ์ฉบับนี้ มุ่งหมายที่จะพัฒนาเครื่องมือจัดการฐานข้อมูลสำรอง เพื่อให้ระบบฐานข้อมูลสำรองมีความพร้อมในการทำงานทดแทนฐานข้อมูลหลัก ในกรณีที่มีเหตุการณ์ร้ายแรงต่างๆ เกิดขึ้น ส่งผลทำให้ฐานข้อมูลหลักเกิดความเสียหาย และไม่สามารถทำงานได้ตามปกติ

ในการพัฒนาผู้วิจัยได้ทำการพัฒนาเครื่องมือในการจัดการฐานข้อมูลสำรองของออร่าเคิลบนเครื่องแม่ข่ายซึ่งมีระบบปฏิบัติการโซลาริส และได้นำโปรแกรมอรรถประโยชน์ที่มีอยู่แล้วมาใช้ร่วมกับใช้ภาษา คอรันเชลล์ ที่พัฒนาขึ้นและมีการโต้ตอบกับผู้ใช้เป็นลักษณะเมนู เพื่อให้เกิดความสะดวกต่อการใช้งาน

จากผลการวิจัยทำให้ได้เครื่องมือที่มีความสะดวกในการจัดการฐานข้อมูลสำรองและลดการทำงานที่ซ้ำซ้อนเพื่อลดความผิดพลาดที่อาจจะเกิดขึ้นได้ อีกทั้งช่วยให้เจ้าหน้าที่ดูแลระบบฐานข้อมูลสามารถกู้คืนฐานข้อมูลให้กลับคืนสู่สภาพเดิมได้อย่างรวดเร็ว ลดปัญหาการสูญหายของข้อมูลได้ ซึ่งมีผลทำให้ข้อมูลมีความถูกต้องและมีความน่าเชื่อถือด้วย

สถาบันนวัตกรรมการ  
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา ....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิติ.....  
สาขาวิชา ...วิทยาศาสตร์คอมพิวเตอร์. ลายมือชื่ออาจารย์ที่ปรึกษา.....  
ปีการศึกษา..2545.....

## 4371409121 : MAJOR COMPUTER SCIENCE

KEY WORD: STANDBY DATABASE / RECOVERY STRATEGIES

KANIT LERTSRIBANDIT : DEVELOPMENT OF A TOOL FOR MANAGING A STANDBY  
DATABASE. THESIS ADVISER : ASSISTANT PROFESSOR BOONCHAI  
SOWANWANICHAKUL , 125 pp . ISBN 974-17-1412-2.

The purpose of this thesis is to develop a tool of creating and managing a standby database for preparation to replace primary database when the primary database is destroyed from disaster. Primary database may be lost and cannot work properly

This tools is developed to manage ORACLE standby database on SUN servers using SOLARIS oprating system and existing utilities and new developed shell scripts

The developed tool is helps a database administrators in managing database backup and recovery. the tool also help when a disaster happens.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Department/Program..Computer Engineering. Student's signature.....

Field of study...Computer Science..... Advisor's signature.....

Academic year.....2002.....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยความช่วยเหลืออย่างดียิ่ง ของผู้ช่วยศาสตราจารย์ บุญชัย ไสววรรณวิชกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆแก่ผู้วิจัย และตรวจสอบแก้ไขวิทยานิพนธ์ฉบับนี้ ทำให้มีความถูกต้องและสมบูรณ์มากที่สุด ผู้วิจัยขอกราบขอขอบคุณในความกรุณาเป็นอย่างสูง

ขอขอบพระคุณ ท่านคณะกรรมการสอบวิทยานิพนธ์ ที่ได้ช่วยพิจารณา ให้คำแนะนำตรวจทานแก้ไข และอนุมัติวิทยานิพนธ์ฉบับนี้

ขอขอบคุณเพื่อนๆ สาขาวิทยาศาสตร์คอมพิวเตอร์ทุกท่านโดยเฉพาะอย่างยิ่ง คุณภัทรพร เอกบรรณสิงห์ ที่ให้ความช่วยเหลือ ตลอดจนเป็นกำลังใจแก่ผู้วิจัยตลอดมา

ท้ายสุดนี้ ผู้วิจัย ไคร่กราบขอขอบคุณ บิดา-มารดา ซึ่งเป็นผู้มีพระคุณแก่ผู้วิจัยอย่างหาที่เปรียบมิได้ ซึ่งคอยให้กำลังใจและสนับสนุนตลอดมา คุณความดีที่เกิดจากงานวิทยานิพนธ์ครั้งนี้ ผู้วิจัยขอกราบมอบให้แก่บิดาและมารดา หากมีข้อผิดพลาดประการใด ผู้วิจัยขออภัยมา ณ ที่นี้ด้วย

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	ง
บทคัดย่อภาษาอังกฤษ .....	จ
กิตติกรรมประกาศ .....	ฉ
สารบัญ .....	ช
สารบัญตาราง .....	ฅ
สารบัญภาพ .....	ญ

## บทที่

1. บทนำ .....	1
ความสำคัญและความเป็นมาของปัญหา .....	1
วัตถุประสงค์ของการวิจัย .....	2
ขอบเขตของการวิจัย .....	2
ขั้นตอนและวิธีดำเนินการวิจัย .....	3
ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย .....	3
2. กลยุทธ์ในการกู้ข้อมูลและฐานข้อมูลสำรอง .....	5
กลยุทธ์ในการกู้ข้อมูล .....	6
ฐานข้อมูลสำรอง .....	7
3. การออกแบบ .....	13
ขั้นตอนในการออกแบบเครื่องมือในการสร้างและจัดการระบบฐานข้อมูลสำรอง .....	13
คอนเทกซ์ (Context Diagram) และแผนภาพกระแสข้อมูลระบบ (Data Flow Diagram) .....	20
การออกแบบลักษณะส่วนติดต่อผู้ใช้ (User Interface) .....	30

4. การพัฒนา .....	35
ภาษาที่ใช้ในการพัฒนา .....	35
การทำงานของชุดฟังก์ชัน .....	36
การทำงานของชุดโปรแกรม .....	41
5. การทดสอบเครื่องมือ .....	52
การส่งผ่านแฟ้มบันทึกถาวรและการปรับปรุงข้อมูลบนฐานข้อมูลสำรอง .....	53
การทดสอบการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลหลัก .....	58
การเปลี่ยนแปลงสถานะการทำงานของฐานข้อมูลสำรองเป็นฐานข้อมูลหลัก ...	60
6. สรุปการวิจัยและข้อเสนอแนะ .....	66
สรุปการวิจัย .....	66
เงื่อนไขในกาใช้งาน .....	66
ปัญหาที่พบ .....	66
ข้อเสนอแนะ .....	67
รายการอ้างอิง .....	68
ภาคผนวก .....	69
ภาคผนวก ก. การใช้เครื่องมือ .....	70
ภาคผนวก ข. ตัวอย่างโปรแกรม .....	92
ประวัติผู้เขียน .....	134



## สารบัญตาราง

ตาราง	หน้า
ตารางที่ 4.1 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลหลัก .....	36
ตารางที่ 4.2 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลสำรอง .....	39
ตารางที่ 5.1 ข้อกำหนดของเครื่องแม่ข่ายที่ใช้ในการทดสอบ .....	52
ตารางที่ 5.2 ประเภทของการทดสอบเครื่องมือ .....	52



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญภาพ

ภาพประกอบ	หน้า
รูปที่ 2.1 การลงบันทึกข้อมูล .....	5
รูปที่ 2.2 ความสัมพันธ์ของบล็อกข้อมูล, เอ็กเทนต์, เช็กเมนต์ .....	7
รูปที่ 2.3 ภาพเทเบิลสเปซ บนระบบฐานข้อมูล .....	8
รูปที่ 2.4 ความสัมพันธ์ระหว่าง เทเบิลสเปซและ แฟ้มข้อมูล .....	9
รูปที่ 2.5 การทำงานของฐานข้อมูลหลัก .....	11
รูปที่ 2.6 การทำงานของระหว่างฐานข้อมูลหลักและฐานข้อมูลสำรอง .....	12
รูปที่ 2.7 การทำงานของฐานข้อมูลสำรองเมื่อเกิดเหตุการณ์ร้ายแรง .....	13
รูปที่ 3.1 การสร้างและจัดการฐานข้อมูลสำรอง .....	15
รูปที่ 3.2 โปรแกรมที่กำหนดเวลาทำงานอัตโนมัติ .....	16
รูปที่ 3.3 การทำงานของแฟ้มลงบันทึกทำซ้ำที่มีการสำเนาไปที่เครื่องแม่ข่ายสำรอง .....	17
รูปที่ 3.4 วงจรสถานะการทำงานของฐานข้อมูลสำรอง .....	18
รูปที่ 3.5 สำรองข้อมูลและนำข้อมูลกลับคืนบนฐานข้อมูลสำรอง .....	19
รูปที่ 3.6 ภาพการทำงานของเครื่องมือสร้างและจัดการฐานข้อมูลสำรอง .....	20
รูปที่ 3.7 แผนภาพคอนเท็กซ์.....	21
รูปที่ 3.8 แผนภาพกระแสข้อมูลในระดับที่ 1 .....	22
รูปที่ 3.9 แผนภาพกระแสข้อมูลในระดับที่ 2 .....	23
รูปที่ 3.10 แผนภาพกระแสข้อมูลในระดับที่ 2 .....	24
รูปที่ 3.11 แผนภาพกระแสข้อมูลในระดับที่ 2 .....	25
รูปที่ 3.12 แผนภาพกระแสข้อมูลในระดับที่ 3 .....	26
รูปที่ 3.13 แผนภาพกระแสข้อมูลในระดับที่ 3 .....	27
รูปที่ 3.14 แผนภาพกระแสข้อมูลในระดับที่ 3 .....	28
รูปที่ 3.15 แผนภาพกระแสข้อมูลในระดับที่ 3 .....	29
รูปที่ 3.16 ภาพเมนูรายการหลัก .....	30
รูปที่ 3.17 ภาพเมนูรายการสำหรับตรวจสอบรายละเอียดบนฐานข้อมูลหลัก .....	31
รูปที่ 3.18 ภาพเมนูรายการสำหรับสร้างและลบฐานข้อมูลสำรอง .....	31
รูปที่ 3.19 ภาพเมนูรายการสำหรับจัดการกับฐานข้อมูลสำรอง .....	32
รูปที่ 3.20 ภาพเมนูรายการย่อยสำหรับการเริ่มและหยุดการทำงานของฐานข้อมูลสำรอง ..	32

## สารบัญภาพ (ต่อ)

ภาพประกอบ	หน้า
รูปที่ 3.21 ภาพเมนูรายการย่อยสำหรับแสดงข้อมูลเกี่ยวกับฐานข้อมูลสำรอง .....	33
รูปที่ 3.22 ภาพเมนูรายการย่อยสำหรับการสำรองและการนำฐานข้อมูลสำรองกลับคืน....	33
รูปที่ 3.23 ภาพเมนูรายการย่อยสำหรับการจัดการฐานข้อมูลสำรองหลังจากฐานข้อมูลหลักเสียหาย .....	34
รูปที่ 3.24 ภาพเมนูรายการสำหรับการให้บริการด้านเครือข่ายของฐานข้อมูลสำรอง .....	34
รูปที่ 4.1 ขั้นตอนการทำงานรายการเมนู .....	41
รูปที่ 4.2 ขั้นตอนการสร้างฐานข้อมูลสำรอง.....	42
รูปที่ 4.3 ขั้นตอนการส่งผ่านแฟ้มบันทึกถาวร.....	43
รูปที่ 4.4 ขั้นตอนการเปลี่ยนแปลงโครงสร้างฐานข้อมูลสำรอง .....	44
รูปที่ 4.5 ขั้นตอนการตรวจสอบสถานะของฐานข้อมูลทั้งสองระบบ .....	45
รูปที่ 4.6 ขั้นตอนสำรองข้อมูล .....	47
รูปที่ 4.7 ขั้นตอนการกู้คืนฐานข้อมูลสำรอง .....	48
รูปที่ 4.8 ขั้นตอนการปรับปรุงข้อมูลบนแฟ้มบันทึกทำซ้ำ.....	50
รูปที่ 4.9 ขั้นตอนการเปลี่ยนสถานะจากฐานข้อมูลสำรองเป็นฐานข้อมูลหลัก .....	51
รูปที่ 5.1 การสร้างผู้ใช้ใหม่และทำการกำหนดสิทธิ์ .....	53
รูปที่ 5.2 การสร้างตารางเพื่อรองรับการเปลี่ยนแปลงที่จำลองขึ้น .....	54
รูปที่ 5.3 การเพิ่มข้อมูลในตารางบนฐานข้อมูลหลัก .....	54
รูปที่ 5.4 การปรับปรุงข้อมูลในตารางบนฐานข้อมูลหลัก .....	55
รูปที่ 5.5 การลบข้อมูลในตารางบนฐานข้อมูลหลัก .....	56
รูปที่ 5.6 การสอบถามข้อมูลในตารางบนฐานข้อมูลสำรอง .....	57
รูปที่ 5.7 การเพิ่มเทเบิลสเปซบนฐานข้อมูลหลัก .....	58
รูปที่ 5.8 เครื่องมือในการตรวจสอบสถานะหลังจากเพิ่มจำนวนเทเบิลสเปซบนฐานข้อมูลหลัก .....	59
รูปที่ 5.9 เครื่องมือในการตรวจสอบสถานะหลังปรับปรุงโครงสร้างบนฐานข้อมูลสำรองเรียบร้อยแล้ว .....	59
รูปที่ 5.10 การเพิ่มเทเบิลสเปซข้อมูลในตารางบนฐานข้อมูลหลัก .....	60
รูปที่ 5.11 การสร้างเทเบิลสเปซใหม่ที่สร้างขึ้น .....	60

### สารบัญภาพ (ต่อ)

ภาพประกอบ	หน้า
รูปที่ 5.12 การเพิ่มข้อมูลในตารางบนฐานข้อมูลหลัก .....	61
รูปที่ 5.13 สถานะในกรณีที่ฐานข้อมูลหลักไม่สามารถใช้งานได้ตามปกติ .....	61
รูปที่ 5.14 การปรับปรุงข้อมูลจากแฟ้มบันทึกทำซ้ำ .....	62
รูปที่ 5.15 การปรับปรุงโครงสร้างทางกายภาพ .....	63
รูปที่ 5.16 การเปลี่ยนสถานะฐานข้อมูลสำรองเป็นฐานข้อมูลหลัก .....	64
รูปที่ 5.17 สถานะของฐานข้อมูลสำรองที่สามารถทำงานแทนฐานข้อมูลหลัก .....	65
รูปที่ 5.18 การสอบถามข้อมูลบนฐานข้อมูลสำรองหลังเกิดเหตุการณ์ร้ายแรง .....	65



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 1

### บทนำ

#### ความสำคัญและความเป็นมาของปัญหา

ปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาท และมีความสำคัญต่อระบบธุรกิจเป็นอย่างมาก ซึ่งการแข่งขันในโลกธุรกิจทุกวันนี้ นับวันจะมีเพิ่มมากขึ้นและทวีความรุนแรงยิ่งขึ้น ทำให้ในแต่ละองค์กรหรือแต่ละหน่วยงานต่างต้องรักษาภาพพจน์ที่ดีต่อลูกค้าในทุกๆ ด้าน ดังนั้นจึงมีการนำระบบคอมพิวเตอร์เข้ามาประยุกต์ใช้ เพื่อเพิ่มประสิทธิภาพในการทำงาน และเพื่อเพิ่มความเชื่อมั่นแก่ลูกค้าที่มีต่อองค์กรหรือหน่วยงานนั้นๆ ดังนั้นไม่ว่าจะมีเหตุการณ์ใดๆ ที่ผิดพลาดเกิดขึ้นกับระบบงานขององค์กร ผู้เกี่ยวข้องจะต้องทำให้กลับคืนสู่สภาพเดิมโดยเร็วที่สุด มีความถูกต้องมากที่สุด และให้เกิดความเสียหายน้อยที่สุด ระบบฐานข้อมูลก็เป็นอีกตัวแปรหนึ่งที่มีโอกาสเกิดปัญหาขึ้น ไม่ว่าจะเป็นความเสียหายของตัวเครื่องคอมพิวเตอร์และอุปกรณ์ต่อพ่วง (Hardware and Peripheral Failure) ความเสียหายของชุดคำสั่ง (Software Failure) ความเสียหายของข้อมูล (Data Failure) ซึ่งในกรณีนี้ผู้ดูแลระบบสามารถทำการกู้ข้อมูลเพื่อฟื้นฟูระบบ (Recovery) เพื่อให้สามารถทำงานได้ โดยการนำข้อมูลกลับคืน (Restore) จากอุปกรณ์ที่ใช้ในการสำรองข้อมูล (Backup) ซึ่งอาจใช้เวลาเป็น นาที ชั่วโมง หรือ วัน ขึ้นอยู่กับระดับความเสียหายที่เกิดขึ้น แต่ในกรณีที่ความเสียหายนั้นเกิดจากเหตุการณ์ร้ายแรงต่างๆ (Disaster) อาทิเช่น อัคคีภัย ตลอดจนภัยธรรมชาติต่างๆ ที่อาจส่งผลกระทบต่อสถานที่ตั้งขององค์กรหรือหน่วยงานนั้นๆ เป็นผลให้ผู้ดูแลระบบไม่สามารถนำอุปกรณ์ที่ใช้ในการสำรอง ข้อมูล (Backup) ออกจากที่เกิดเหตุได้ ทำให้ไม่สามารถทำการฟื้นฟูระบบจากความเสียหายได้ ถ้าในแต่ละองค์กรหรือหน่วยงานมีการจัดระบบและวางแผนการฟื้นฟูความเสียหายให้ครอบคลุมต่อเหตุการณ์ต่างๆ ที่อาจจะเกิดขึ้น โดยมีการออกแบบระบบฐานข้อมูลสำรอง (Standby Database) ซึ่งอาจตั้งอยู่ในสถานที่ที่แตกต่างกับระบบฐานข้อมูลหลัก (Primary Database) เพื่อเตรียมความพร้อมสำหรับเหตุการณ์ต่างๆ โดยมีการเขียนขั้นตอนการทำงานไว้อย่างชัดเจน ครอบคลุม แล้วจึงทำการทดสอบระบบดังกล่าว จะทำให้ผู้ใช้ระบบ (User) และผู้ดูแลระบบฐานข้อมูล (Database Administrator) มีความเข้าใจและสามารถปฏิบัติงานในการรองรับสถานการณ์ที่อาจจะเกิดขึ้นโดยไม่ได้คาดการณ์มาก่อนได้ เพื่อเป็นการป้องกันและลดความเสียหายของธุรกิจ

ในการสร้างและจัดการฐานข้อมูลสำรองเพื่อให้สามารถทำงานทดแทนฐานข้อมูลหลักได้นั้น จะต้องมีขั้นตอนการจัดการที่ซับซ้อน โดยจะเริ่มตั้งแต่การสร้างฐานข้อมูลสำรองบนเครื่องแม่ข่าย

ฐานข้อมูลสำรอง การส่งผ่านการเปลี่ยนแปลงข้อมูลที่เกิดขึ้นบนฐานข้อมูลหลักไปยังฐานข้อมูลสำรอง การควบคุมการปรับปรุงข้อมูลบนฐานข้อมูลสำรองให้ใกล้เคียงกับฐานข้อมูลหลัก และการเปลี่ยนแปลงโครงสร้างทางกายภาพของฐานข้อมูลสำรองให้ตรงกับฐานข้อมูลหลัก เพื่อให้สามารถรองรับการปรับปรุงข้อมูลที่ส่งมาจากฐานข้อมูลหลักได้ ซึ่งขั้นตอนต่างๆ ที่กล่าวมาจะสร้างยุ่งยากต่อเจ้าหน้าที่ดูแลระบบฐานข้อมูล อาจจะมีการทำงานที่ซ้ำซ้อน และอาจเกิดข้อผิดพลาดขึ้นได้

### วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาโปรแกรมที่ใช้ในการเตรียมระบบฐานข้อมูลสำรอง เพื่อให้มีความพร้อมในการทำงานทดแทนฐานข้อมูลหลัก

### ขอบเขตของการวิจัย

1. ในการปรับปรุงข้อมูลของรายการที่เปลี่ยนแปลงที่เก็บอยู่ในแฟ้มลงบันทึกเก็บถาวรบนระบบฐานข้อมูลสำรอง จะใช้วิธีการกำหนดเวลาในการปรับปรุง
2. เส้นทาง (Path) ของตำแหน่งที่เก็บโครงสร้างทางกายภาพบนเครื่องแม่ข่ายฐานข้อมูลหลัก และ เครื่องแม่ข่ายฐานข้อมูลสำรองต้องเหมือนกัน
3. รายการที่เปลี่ยนแปลงที่เกิดขึ้นกับฐานข้อมูลหลักจะใช้การจำลองการทำงานโดยให้มีความคำสั่งในการเพิ่มข้อมูล ลบข้อมูล และเปลี่ยนแปลงข้อมูลในแต่ละตาราง โดยมีรายการเปลี่ยนแปลงเกิดขึ้นอย่างละ 50 รายการ
4. ในการทดสอบการทำงานของฐานข้อมูลสำรองหลังจากเกิดเหตุการณ์ร้ายแรงจะใช้การจำลองสถานการณ์โดยการปิดไฟ(Power Off) เครื่องแม่ข่ายฐานข้อมูลหลัก
5. ในการตรวจสอบการทำงานของฐานข้อมูลสำรองจะใช้คำสั่งในการสอบถามข้อมูล (Query) ในแต่ละตารางที่มีการเปลี่ยนแปลง
6. ในการทำวิจัยจะใช้ ระบบคอมพิวเตอร์ (Computer System) รวมถึงชุดคำสั่งระบบ (System Software) ของเครื่องชั้น (SUN) เป็นกรณีศึกษา ซึ่งผู้วิจัยได้ทำการทดสอบกับระบบดังนี้
  - เครื่องแม่ข่ายฐานข้อมูลหลัก และ เครื่องแม่ข่ายฐานข้อมูลสำรองทำงานอยู่บนเครื่องชั้น (SUN) ซึ่งใช้ระบบปฏิบัติการโซลาริส (SOLARIS)

- ระบบจัดการฐานข้อมูลบนฐานข้อมูลหลักและบนฐานข้อมูลสำรอง (Database Management System) ใช้ออราเคิล (ORACLE)
- เครื่องแม่ข่ายฐานข้อมูลหลัก และ เครื่องแม่ข่ายฐานข้อมูลสำรองทำงานอยู่บนเครือข่ายท้องถิ่น(Local Area Network)

### ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษารายละเอียด และการออกแบบระบบฐานข้อมูลให้สามารถฟื้นฟูระบบจากเหตุการณ์ร้ายแรงต่างๆ ได้
2. ศึกษารายละเอียดและออกแบบของการสร้างฐานข้อมูลสำรองเพื่อรองรับการปรับปรุงข้อมูลจากฐานข้อมูลหลัก
3. ศึกษารายละเอียดการส่งผ่าน และปรับปรุงข้อมูลจากแฟ้มลงบันทึกเก็บถาวรเพื่อให้ฐานข้อมูลสำรองมีข้อมูลใกล้เคียงกับฐานข้อมูลหลัก
4. ศึกษารายละเอียด และผลกระทบที่จะเกิดขึ้นกับฐานข้อมูลสำรองถ้ามีการเปลี่ยนแปลงแก้ไขโครงสร้างของระบบฐานข้อมูลหลัก
5. ออกแบบและพัฒนาโปรแกรมที่ใช้ในการจัดการโครงสร้างของฐานข้อมูลสำรอง
6. ทดสอบฟื้นฟูระบบฐานข้อมูลบนเครื่องแม่ข่ายฐานข้อมูลสำรองหลังจากเกิดเหตุการณ์ร้ายแรงได้
7. สรุปผลการวิจัย เสนอแนะ และจัดทำรูปเล่ม

### ประโยชน์ที่คาดว่าจะได้รับการวิจัย

1. ทำให้ระบบฐานข้อมูลสำรองสามารถปรับปรุงข้อมูลให้มีความถูกต้องตรงกับระบบฐานข้อมูลหลัก
2. ทำให้โครงสร้างของระบบฐานข้อมูลสำรองมีความถูกต้องตรงกับระบบฐานข้อมูลหลัก
3. ให้เจ้าหน้าที่ดูแลระบบฐานข้อมูลมีความสะดวกในการจัดการระบบฐานข้อมูลสำรองมากขึ้น
4. ลดการทำงานที่ซ้ำซ้อนของเจ้าหน้าที่ดูแลระบบฐานข้อมูลเพื่อลดความผิดพลาดในการเปลี่ยนแปลงโครงสร้างของระบบฐานข้อมูลสำรองได้
5. ให้เจ้าหน้าที่ดูแลระบบฐานข้อมูลมีแนวทางในการออกแบบระบบฐานข้อมูลสำรองเพื่อรองรับกับเหตุการณ์ต่างๆ ได้



6. ทำให้เจ้าหน้าที่ดูแลระบบฐานข้อมูลมีความเข้าใจถึงการทำงานของระบบฐานข้อมูลหลัก และระบบฐานข้อมูลสำรอง



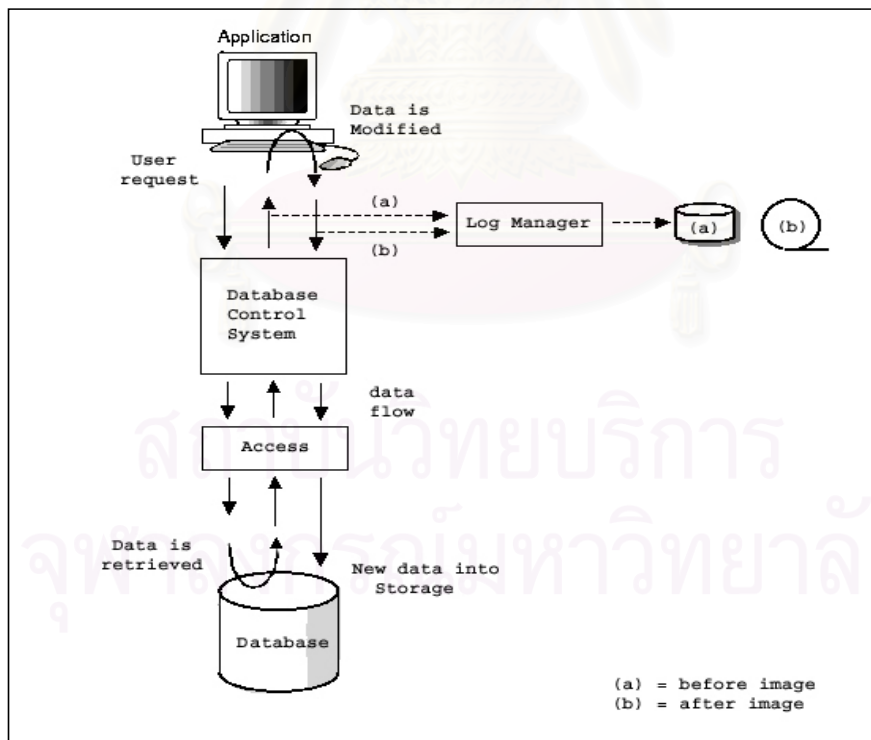
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## บทที่ 2

### การกู้คืนข้อมูลและฐานข้อมูลสำรอง

ฐานข้อมูลที่เกิดข้อผิดพลาดอาจเกิดมาจากสาเหตุหลายประการ เช่น ไม่มีการตรวจสอบการป้อนข้อมูลเข้าระบบ ซึ่งรวมถึงความเสียหายของเครื่องและอุปกรณ์ดังที่กล่าวไว้ในบทที่ 1 โดยที่ความผิดพลาดนี้บางครั้งเราไม่สามารถหลีกเลี่ยงได้ แต่ในระบบการจัดการฐานข้อมูล (Database Management System) จะมีกระบวนการในการจัดการกู้คืนข้อมูล โดยการนำข้อมูลที่มีการทำสำรองไว้มาอยู่ในสถานะที่ถูกต้องหากเกิดปัญหาขึ้น ซึ่งกระบวนการทำงานที่สำคัญก็คือ การลงบันทึก(Logging) ระบบจัดการฐานข้อมูล จะมีกลไกในการลงบันทึกที่จะทำหน้าที่ในการลงบันทึกทะเบียน (Record) ที่มีการปรับปรุงในแต่ละรายการที่เปลี่ยนแปลง (Transaction) ซึ่งการลงบันทึกเป็นสิ่งที่จำเป็นในการกู้คืนฐานข้อมูล การจัดการลงบันทึก (Log Manager) เป็นส่วนที่สำคัญของระบบจัดการฐานข้อมูล ทะเบียนที่มีการปรับปรุงในแต่ละรายการที่เปลี่ยนแปลงจะถูกบันทึกทั้งก่อนและหลังการเปลี่ยนแปลงข้อมูล ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 การลงบันทึกข้อมูล (LOGGING DATA FLOW) [1]

การลงบันทึกจะแบ่งออกเป็น 2 ลักษณะดังนี้

- การลงบันทึกก่อนภาพลักษณ์ (Logging Before Image)

ภาพลักษณ์ก่อนการเปลี่ยนแปลงจะมีการบันทึกข้อมูลเก่าก่อนจะมีการปรับปรุงของแต่ละการเปลี่ยนแปลง ซึ่งการลงบันทึกก่อนภาพลักษณ์จะทำให้สามารถยกเลิกรายการเปลี่ยนแปลงที่ไม่สำเร็จ

- การลงบันทึกหลังภาพลักษณ์ (Logging After Image)

ภาพลักษณ์หลังการเปลี่ยนแปลงจะมีการลงบันทึก หลังจากมีการปรับปรุงข้อมูลในแต่ละรายการเปลี่ยนแปลง ซึ่งจะใช้ในการเขียนซ้ำลงไปในฐานะข้อมูล เพื่อให้ในการกู้ข้อมูลในลักษณะการกู้ข้อมูลแบบคืบหน้า (Roll-forward Recovery)

การสำรองข้อมูลโดยการคัดลอกเพียงบางส่วน หรือ ทั้งหมดของฐานข้อมูล จะเป็นการปฏิบัติขั้นพื้นฐานที่ผู้ดูแลระบบฐานข้อมูลต้องดำเนินการอยู่เสมอ หากเกิดปัญหาหรือมีข้อผิดพลาดเกิดขึ้น การสำรองข้อมูลครั้งสุดท้ายจะถูกนำมาใช้กู้คืนเพื่อให้ข้อมูลมีสถานะที่ถูกต้อง รวดเร็ว และถูกปรับปรุงที่เป็นปัจจุบันมากที่สุด ซึ่งสิ่งที่ต้องพิจารณามีดังนี้

1. กลยุทธ์ในการกู้ฐานข้อมูล

กลยุทธ์ในการกู้ฐานข้อมูลจะมี 2 วิธีคือ การกู้ข้อมูลแบบคืบหน้า และ การกู้ข้อมูลแบบถอยกลับ (Roll-back Recovery) โดยที่วิธีการกู้ข้อมูลจะขึ้นอยู่กับลักษณะของแบบและข้อผิดพลาด

1.1. การกู้ข้อมูลแบบคืบหน้า

วิธีนี้จะกู้คืนข้อมูลโดยการนำสำเนาของฐานข้อมูลที่ทำสำรองไว้กลับคืน ในกรณีที่เกิดข้อผิดพลาดขึ้น หลังจากนั้นจะใช้ภาพลักษณ์หลังการเปลี่ยนแปลงที่มีการลงบันทึกไว้ในแฟ้มมาทำการปรับปรุงซ้ำไปที่สำเนาของฐานข้อมูล เพื่อให้ข้อมูลมีสถานะที่ถูกปรับปรุงให้เป็นปัจจุบันมากที่สุด

1.2 การกู้ข้อมูลแบบถอยกลับ

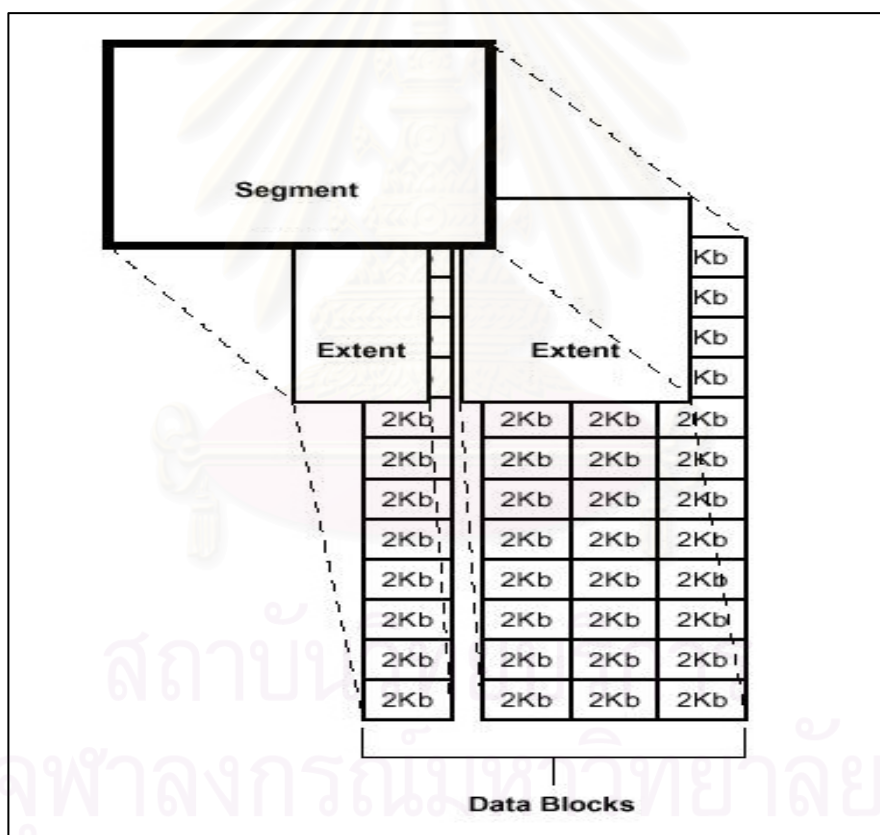
การถอยกลับสามารถยกเลิกการทำงานที่ผ่านมาของรายการที่เปลี่ยนแปลงที่ไม่สำเร็จ เพื่อให้ข้อมูลมีความถูกต้องตรงกัน (Data Integrity) เมื่อฐานข้อมูลมีรายการที่เปลี่ยนแปลงเข้ามาทำงาน จะต้องทำให้สำเร็จทั้งหมด หรือต้องไม่ทำทั้งหมด การทำงานในส่วนนี้จะใช้ ภาพลักษณ์ก่อนการเปลี่ยนแปลง ที่มีการลงบันทึกไว้

## 2. ฐานข้อมูลสำรอง

เป็นฐานข้อมูลแบบเดียวกับที่เครื่องแม่ข่ายฐานข้อมูลหลัก แต่ติดตั้งอยู่บนอีกเครื่องหนึ่ง ในเวลาปกติจะไม่มี การนำขึ้นมาให้บริการ แต่จะคอยปรับปรุงข้อมูลให้ทันสมัยใกล้เคียงกับ ฐานข้อมูลของเครื่องแม่ข่ายฐานข้อมูลหลักให้มากที่สุด เพื่อให้จะให้เข้าใจการทำงานของระบบ ฐานข้อมูลสำรอง จึงขอกล่าวถึงหลักการการทำงานของฐานข้อมูลออราเคิลดังนี้

### 2.1 โครงสร้างของฐานข้อมูลออราเคิล[6] ซึ่งจะมี 2 แบบดังนี้

2.1.1 โครงสร้างของฐานข้อมูลทางตรรก (Logical Structure) จะประกอบด้วย บล็อกข้อมูล (Data Block) เอ็กเทนต์ (Extent) เซ็กเมนต์ (Segment) และ เทเบิลสเปซ (Tablespace) โดยที่ในรูปที่ 2.2 จะแสดงภาพความสัมพันธ์ของ บล็อกข้อมูล เอ็กเทนต์ เซ็ก



เมนต์

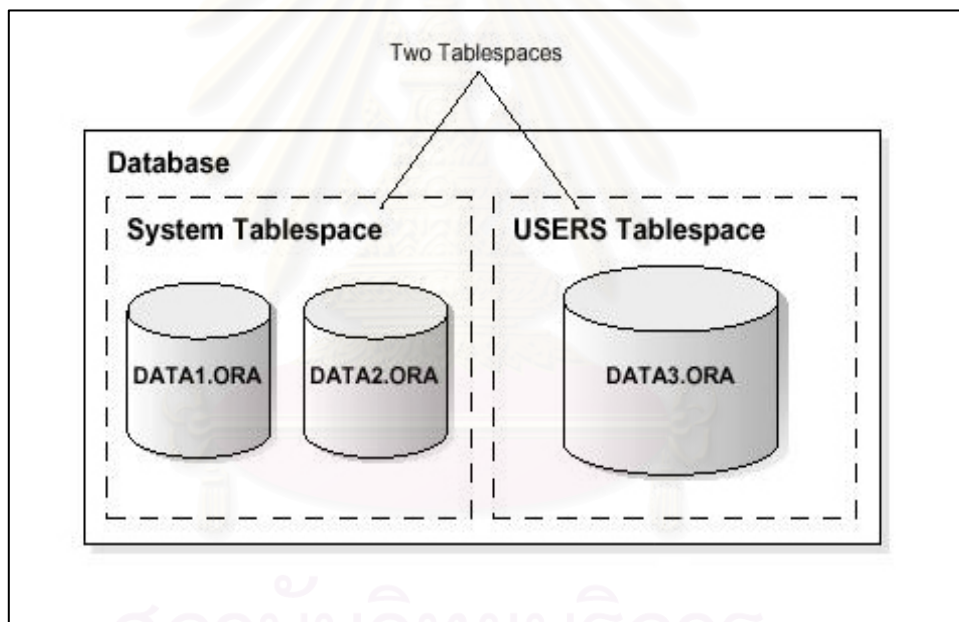
รูปที่ 2.2 ความสัมพันธ์ของบล็อกข้อมูล,เอ็กเทนต์,เซ็กเมนต์ [7]

2.1.1.1 บล็อกข้อมูล จะเป็นหน่วยที่เก็บข้อมูลที่เล็กที่สุดของระบบฐานข้อมูลซึ่งสามารถกำหนดขนาดของบล็อกข้อมูลได้ตอนที่เราสร้างฐานข้อมูล

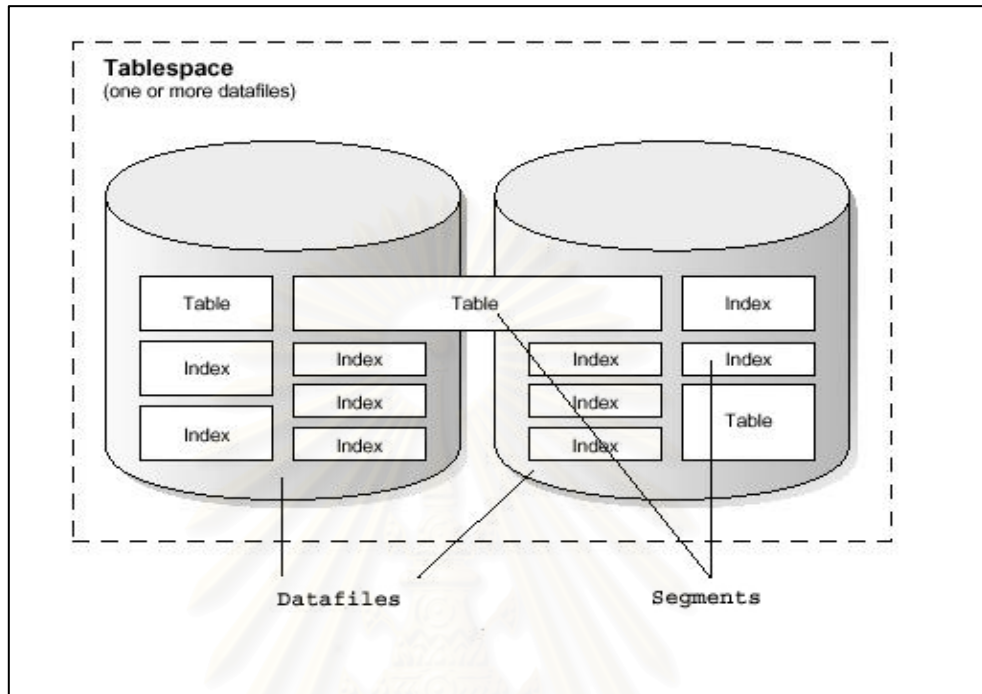
2.1.1.2 เอ็กเทนต์ จะเป็นกลุ่มของบล็อกข้อมูลที่ต่อเนื่องกัน

2.1.1.3 เซ็กเมนต์ จะเป็นกลุ่มของเอ็กเทนต์ที่จะถูกจองเพื่อใช้ในการสร้างเซ็กเมนต์ต่างๆดังนี้ เซ็กเมนต์ข้อมูล (Data Segment) เซ็กเมนต์ดัชนี (Index Segment) เซ็กเมนต์ถอยกลับ (Rollback Segment) เซ็กเมนต์ชั่วคราว (Temporary Segment)

2.1.1.4 เทเบิลสเปซ จะเป็นหน่วยเก็บข้อมูลที่ใหญ่ที่สุดบนฐานข้อมูล จะแบ่งเป็น หนึ่งหรือมากกว่า หน่วยเก็บข้อมูลทางตรรก (Logical Storage Units) เรียกว่า เทเบิลสเปซ ดังแสดงในรูปที่ 2.3 โดยที่ในแต่ละ เทเบิลสเปซ จะประกอบไปด้วยหนึ่งหรือมากกว่า แฟ้มข้อมูล (Data File) และในแต่ละเทเบิลสเปซ จะเป็นที่เก็บของเซ็กเมนต์ต่างๆ ดังแสดงในรูปที่ 2.4



รูปที่ 2.3 เทเบิลสเปซบนระบบฐานข้อมูล [3]



รูปที่ 2.4 ความสัมพันธ์ระหว่าง เทเบิลสเปซ และ แฟ้มข้อมูล [6]

2.1.2 โครงสร้างของฐานข้อมูลทางกายภาพ (Physical Structure) จะประกอบด้วย

2.1.2.1 แฟ้มข้อมูล จะเก็บข้อมูลต่างๆของระบบฐานข้อมูล แฟ้มข้อมูลเหล่านี้จะถูกสร้างตอนที่มีการสร้างฐานข้อมูล และมีการสร้างเทเบิลสเปซใหม่ หรือเพิ่มเนื้อที่ให้เทเบิลสเปซเดิมที่มีอยู่ ซึ่งจะจัดเก็บข้อมูลของผู้ใช้ และข้อมูลของระบบฐานข้อมูล (Data Dictionary) ได้แก่ ตารางข้อมูล ดัชนีข้อมูล

2.1.2.2 แฟ้มควบคุม (Control Files)ฐานข้อมูลออราเคิลทุกๆ ระบบจะต้องมีแฟ้มควบคุมอย่างน้อยหนึ่งแฟ้ม แฟ้มควบคุมเป็นแฟ้มซึ่งมีความสำคัญมากแต่ผู้ใช้ไม่สามารถเข้าไปอ่านได้เอง อาจเทียบได้ว่า แฟ้มควบคุม เปรียบเสมือนการเริ่มต้นทำงาน (Bootstrap) ของระบบปฏิบัติการที่ใช้ในการเปิดเครื่อง ข้อมูลที่เก็บอยู่ภายในได้แก่

- ชื่อ และตำแหน่งที่จัดเก็บของแฟ้มข้อมูล
- ชื่อและตำแหน่งที่จัดเก็บของแฟ้มลงบันทึกทำซ้ำ(Redo Log Files)
- ข้อมูลเกี่ยวกับสถานะของแฟ้มลงบันทึกเก็บถาวร (Archive Log Files)

- หมายเลขลำดับปัจจุบันของแฟ้มลงบันทึกทำซ้ำ
- ข้อมูลเกี่ยวกับแฟ้มลงบันทึกทำซ้ำที่ต้องใช้ในการกู้ฐานข้อมูล
- ข้อมูลเกี่ยวกับเวลาในการการเริ่มและการปิดฐานข้อมูล

ข้อมูลที่เก็บไว้ในแฟ้มควบคุมมีความสำคัญมากหากเกิดความเสียหาย หรือสูญหายไป วิธีแก้ก็คือการสร้างแฟ้มควบคุมขึ้นมาใหม่ หรือสร้างฐานข้อมูลขึ้นมาใหม่ และนำข้อมูลที่สำรองไว้มาลง เพราะความสำคัญนี้เองจึงต้องสร้างสำเนาของ แฟ้มควบคุมไว้หลายที่ด้วยกัน โดยระบุไว้ในตัวแปรที่ชื่อ "CONTROL\_FILES" ในแฟ้ม INIT.ORA

2.1.2.3 แฟ้ม INIT.ORA เป็นแฟ้มข้อมูลที่เก็บรายละเอียด และตัวแปรที่สามารถปรับแต่งได้ ต้องมีอยู่ในฐานข้อมูลออราเคิลทุกตัว แฟ้มนี้จะใช้โดยออราเคิลเพื่อเริ่มเปิดฐานข้อมูล ดังนั้นจะต้องมีตำแหน่งที่จัดเก็บแน่นอน หรือไม่ก็ต้องบอกตำแหน่งของแฟ้มในการเริ่มเปิดใช้งานฐานข้อมูล

2.1.2.4 แฟ้มลงบันทึกทำซ้ำ ฐานข้อมูลออราเคิลต้องมีแฟ้มบันทึกทำซ้ำอย่างน้อยสองแฟ้ม แฟ้มบันทึกทำซ้ำจะถูกบันทึกเข้าไปซ้ำมาระหว่างการดำเนินงานในแต่ละวันโดย "LGWR" โปรเซส ซึ่ง "LGWR" โปรเซสจะมีหน้าที่บันทึกข้อมูลจากหน่วยความจำแคช ที่มีการจองเนื้อที่จากหน่วยความจำหลัก เพื่อใช้เก็บข้อมูลจากรายการที่เปลี่ยนแปลงที่เกิดขึ้นกับฐานข้อมูล หรือที่เรียกว่า รีดบูฟเฟอร์แคช (Redo Buffer Cache) ลงบนแฟ้มข้อมูลบันทึกทำซ้ำ หลังจากที่มีการยอมรับรายการที่เปลี่ยนแปลงที่เกิดขึ้นกับฐานข้อมูลแล้ว ซึ่งจะเขียนลงไปเรื่อยๆ ถ้าแฟ้มบันทึกทำซ้ำสุดท้ายเกิดเต็มก็จะมาบันทึกที่แฟ้มบันทึกทำซ้ำแรกใหม่อีก และถ้าได้มีการเปิดสถานะของฐานข้อมูลเป็นแบบการทำแฟ้มลงบันทึกเก็บถาวรด้วย ก็จะมีการบันทึกแฟ้มบันทึกทำซ้ำนี้ลงเป็นแฟ้มลงบันทึกเก็บถาวรโดยมีหมายเลขลำดับที่ปิดท้าย

แฟ้มบันทึกทำซ้ำเหล่านี้ออราเคิลจะใช้เพื่อบันทึกสิ่งที่ได้เกิดขึ้นกับฐานข้อมูล ซึ่งจะบันทึกทุกเหตุการณ์ที่เกิดขึ้นกับฐานข้อมูลแยกออกจากแฟ้มข้อมูล ดังนั้นแฟ้มบันทึกทำซ้ำเหล่านี้สามารถใช้เพื่อกู้คืนฐานข้อมูลได้ในกรณีที่แฟ้มข้อมูลเกิดเสียหายได้

2.1.2.5 แฟ้มลงบันทึกเก็บถาวร เนื้อหาของแฟ้มบันทึกทำซ้ำและแฟ้มลงบันทึกเก็บถาวรจะเหมือนกัน เมื่อได้มีการบันทึกแฟ้มบันทึกทำซ้ำเสร็จ จะมี "ARCH" โปรเซส ทำการบันทึกแฟ้มที่มีเนื้อหาเหมือนกับแฟ้มบันทึกทำซ้ำเรียกว่า "แฟ้มลงบันทึกเก็บถาวร" ซึ่งจะมีหมายเลขลำดับเหมือนกันโดยจะเก็บเป็นประวัติการเปลี่ยนแปลงที่เกิดขึ้นกับฐานข้อมูลในกรณีที่แฟ้มบันทึกทำซ้ำถูกลบมาเขียนทับ ซึ่งจะใช้แฟ้มลงบันทึกเก็บถาวรเหล่านี้เพื่อทำการกู้คืนฐานข้อมูลที่เสียหายได้ โดยการลงข้อมูลที่สำรองไว้ล่าสุด แล้วทำการกู้คืนด้วยแฟ้มลงบันทึกเก็บ

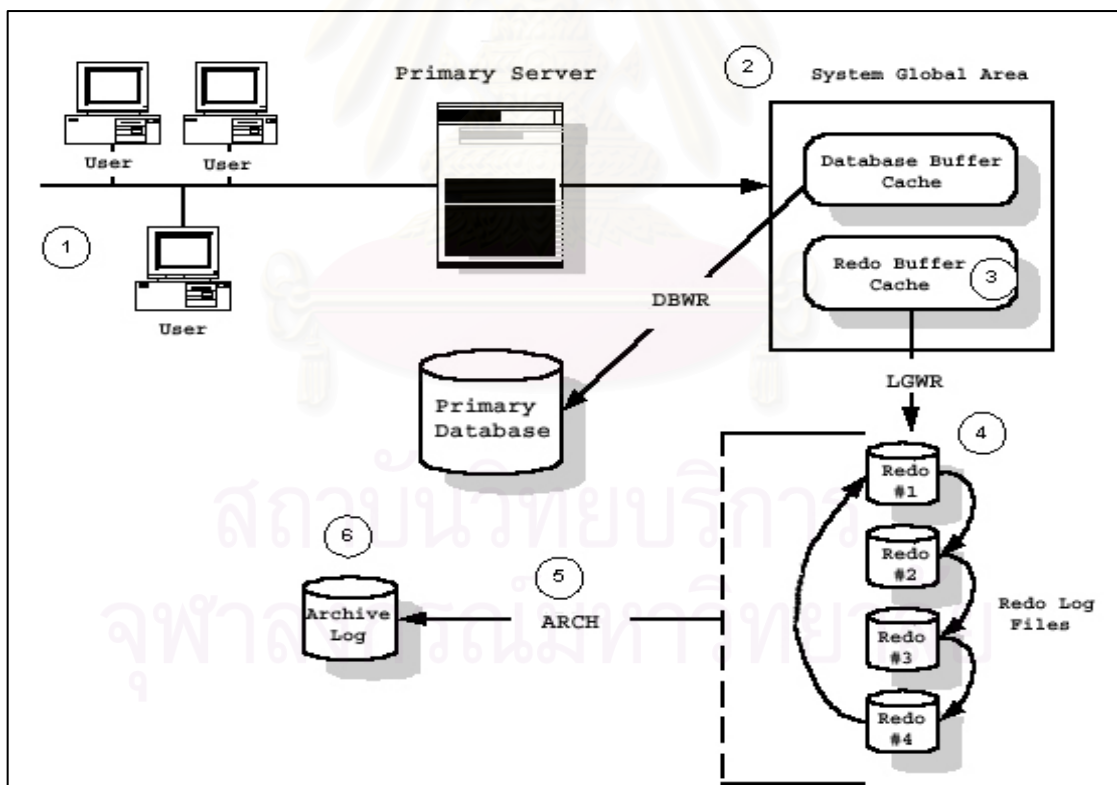


ถาวร ที่เกิดขึ้นทั้งหมดตั้งแต่ทำสำรองครั้งสุดท้ายเพื่อทำการปรับปรุงข้อมูลให้ทันสมัยจนถึงเวลาที่ระบบฐานข้อมูลเสียหาย[4]

## 2.2 การทำงานของฐานข้อมูลหลัก[4]

ขั้นตอนต่อไปนี้จะแสดงวิธีการทำงานของระบบฐานข้อมูลหลัก โดยเริ่มตั้งแต่การทำงานของ ผู้ใช้ การสร้างแฟ้มลงบันทึกทำซ้ำจนมาถึงการสร้างแฟ้มลงบันทึกเก็บถาวร เพื่อส่งไปยังระบบ ฐานข้อมูลสำรองเพื่อทำการปรับปรุงให้ทันสมัยใกล้เคียงกับฐานข้อมูลหลัก

ลำดับที่ 1. ผู้ใช้งานทำงานประจำวัน (เพิ่ม ลบ และมีการปรับปรุงข้อมูล) ผ่านโปรแกรมที่ ติดตั้งไว้ที่เครื่องลูกข่าย ซึ่งงานเหล่านี้เรียกรวมว่า รายการที่เปลี่ยนแปลงลำดับที่ 1. ในขณะที่ผู้ใช้ ทำงาน จะมีการบันทึกประวัติการทำงานไว้ที่ รีดิวฟ์เฟอร์แคช ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 การทำงานของฐานข้อมูลหลัก [5]

ลำดับที่ 2. ประวัตินงานจะถูกเก็บไว้ในรีดิวซ์ฟเฟอร์แคช จนกระทั่งได้มีการบันทึกยอมรับการเปลี่ยนแปลงนั้น (Commit) จากนั้นข้อมูลดังกล่าวจะถูกบันทึกลงในแฟ้มบันทึกทำซ้ำโดยการทำงานของ “LGWR” โปรเซส

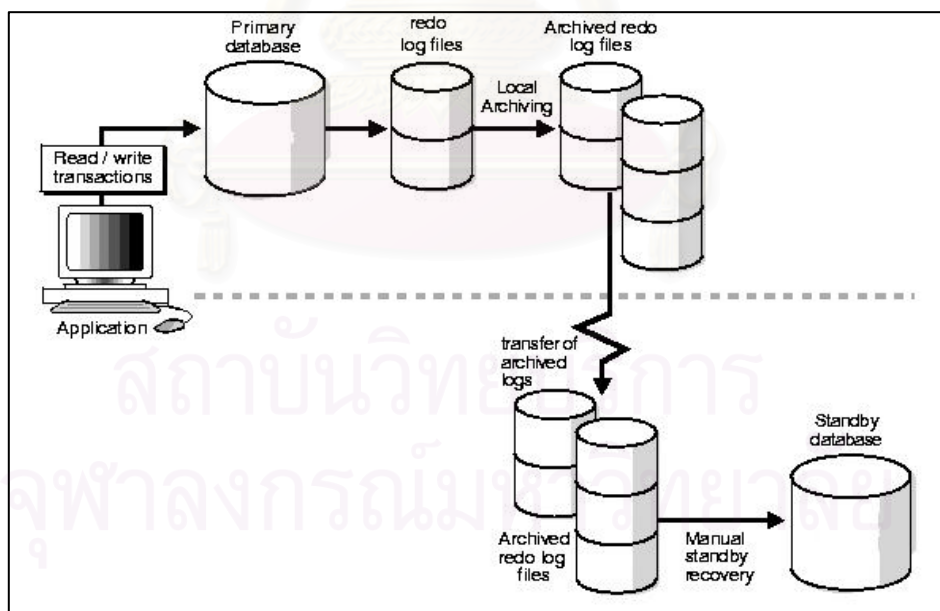
ลำดับที่ 3. ประวัตินงานของรายการที่เปลี่ยนแปลงจะถูกบันทึกลงในแฟ้มบันทึกทำซ้ำจนกว่าจะเต็ม หรือได้รับการสั่งให้ทำการบันทึกใส่แฟ้มบันทึกเก็บถาวร

ลำดับที่ 4. แฟ้มบันทึกเก็บถาวรจะถูกสร้างโดย “ARCH” โปรเซส

ลำดับที่ 5. แฟ้มบันทึกเก็บถาวรจะถูกบันทึกเรียงตามลำดับหมายเลข และรอการส่งผ่านไปยังเครื่องแม่ข่ายฐานข้อมูลสำรองต่อไป

### 2.3 การทำงานของฐานข้อมูลสำรอง [5]

เป็นฐานข้อมูลแบบเดียวกับที่เครื่องแม่ข่ายฐานข้อมูลหลัก แต่ติดตั้งอยู่บนอีกเครื่องหนึ่งในเวลาปกติจะไม่ได้มีการนำขึ้นมาให้บริการ แต่จะคอยปรับปรุงข้อมูลให้ทันสมัยใกล้เคียงกับฐานข้อมูลของเครื่องแม่ข่ายฐานข้อมูลหลักให้มากที่สุด

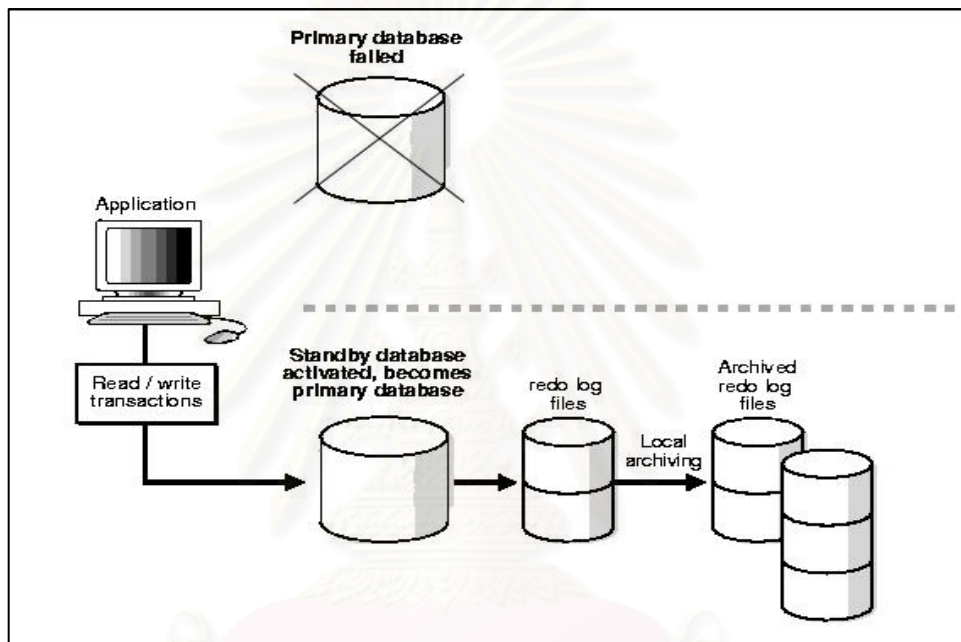


รูปที่ 2.6 การทำงานของระหว่างฐานข้อมูลหลักและฐานข้อมูลสำรอง [5]

ในเวลาปกติฐานข้อมูลหลักมีรายการที่เปลี่ยนแปลงเกิดขึ้น จะมีการสร้างแฟ้มบันทึกเก็บถาวรตามที่อธิบายไปแล้วนั้น แฟ้มบันทึกเก็บถาวรเหล่านี้จะถูกส่งไปยังเครื่องแม่ข่ายฐานข้อมูลสำรองที่มีการติดตั้งฐานข้อมูลสำรองไว้แล้ว และจะทำการปรับปรุงข้อมูลที่เก็บอยู่ในแฟ้มบันทึก



เก็บถาวรที่ละเพิ่มตามลำดับ เพื่อที่จะให้การปรับปรุงข้อมูลบนฐานข้อมูลของทั้งสองใกล้เคียงกันมากที่สุด ดังแสดงไว้ในรูปที่ 2.6 และเมื่อใดก็ตามที่ระบบฐานข้อมูลหลักเกิดปัญหาขึ้นจนไม่สามารถให้บริการได้ตามปกติ ฐานข้อมูลสำรองจะถูกเปลี่ยนเป็นฐานข้อมูลหลักเพื่อให้บริการแทน โดยที่ข้อมูลจะมีการปรับปรุงให้ใกล้เคียง หรือเทียบเท่ากับฐานข้อมูลหลักจะขึ้นอยู่กับแฟ้มลงบันทึกเก็บถาวรตัวสุดท้ายที่ส่งมาอย่างสมบูรณ์ หลังจากนั้นก็เปลี่ยนการทำงานของผู้ใช้ ให้มาใช้ที่ฐานข้อมูลสำรอง ที่เปลี่ยนเป็นฐานข้อมูลหลักแทน ดังแสดงไว้ในรูปที่ 2.7



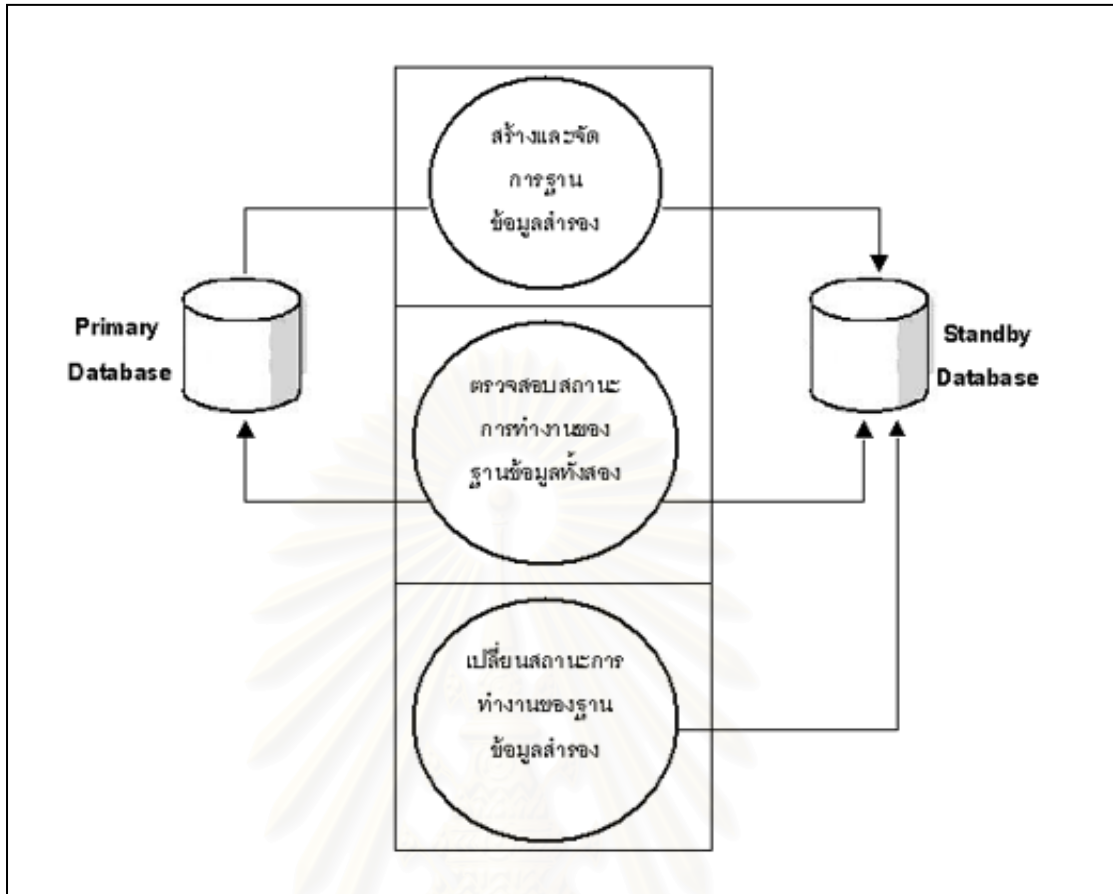
รูปที่ 2.7 การทำงานของฐานข้อมูลสำรองเมื่อเกิดเหตุการณ์ร้ายแรง [5]

### บทที่ 3 การออกแบบ

จากแนวความคิดและทฤษฎีที่เกี่ยวข้องดังที่กล่าวมาแล้วในบทที่ 2 จะพบว่าในการออกแบบระบบฐานข้อมูลสำรองเพื่อให้สามารถรองรับเหตุการณ์ร้ายแรงต่างๆ ได้นั้น การจัดการระบบฐานข้อมูลสำรองมีความสำคัญมากไม่ว่าจะเป็น การปรับปรุงข้อมูลของรายการที่เปลี่ยนแปลงที่เก็บอยู่ในแฟ้มลงบันทึกถาวร การตรวจสอบความถูกต้อง และที่สำคัญก็คือ โครงสร้างทางกายภาพของฐานข้อมูล ในการปรับปรุงข้อมูลของรายการที่เปลี่ยนแปลงที่เก็บอยู่ในแฟ้มลงบันทึกถาวรนั้น โครงสร้างทางกายภาพของฐานข้อมูลสำรองต้องมีความพร้อมในการรองรับกับการทำงานที่เกิดขึ้น

#### 3.1 ขั้นตอนในการออกแบบเครื่องมือในการสร้างและจัดการระบบฐานข้อมูลสำรอง

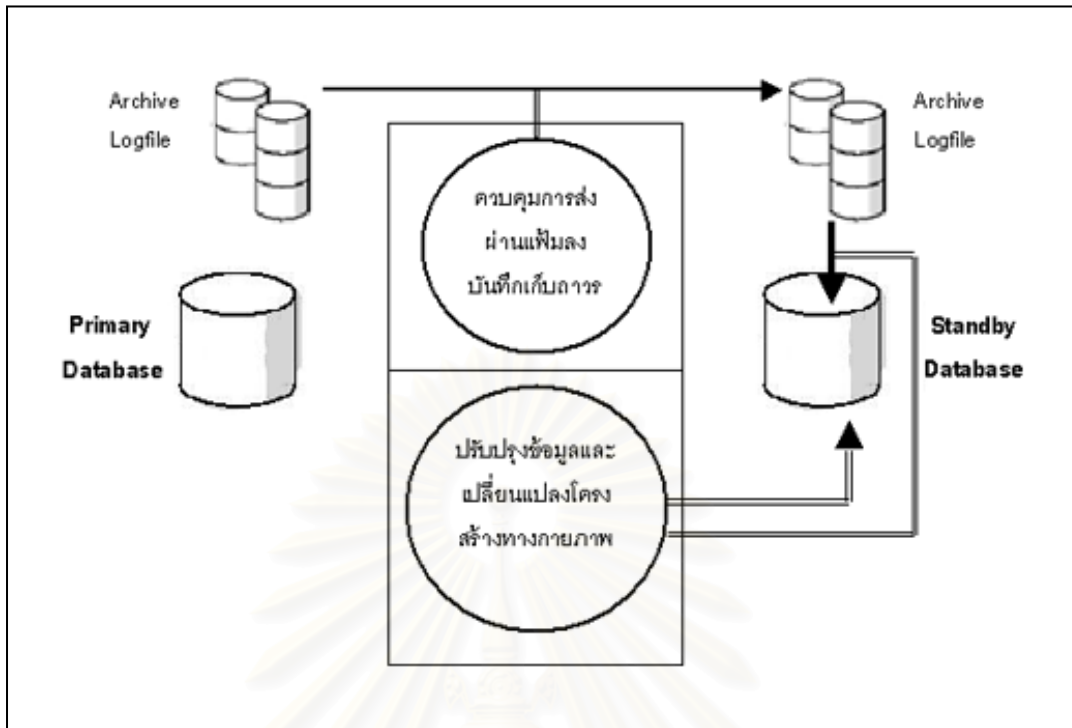
งานวิจัยนี้มีการออกแบบและพัฒนาเครื่องมือ โดยที่เครื่องมือจะแบ่งการทำงานเป็นสองส่วนหลักๆ โดยที่ในส่วนแรกจะทำหน้าที่ในส่วนของการจัดการระบบฐานข้อมูลสำรองโดยจะเริ่มตั้งแต่การสร้างระบบฐานข้อมูลสำรองบนเครื่องแม่ข่ายฐานข้อมูลสำรอง รวมถึงการตรวจสอบสถานะการทำงานและข้อผิดพลาดต่างๆที่เกิดขึ้น โดยจะทำการติดตั้งเครื่องมือบนเครื่องแม่ข่ายฐานข้อมูลสำรองแล้ว จะมีการเข้าไปตรวจสอบโครงสร้างทางกายภาพในฐานข้อมูลหลัก เพื่อทำสำเนาโครงสร้างทางกายภาพของฐานข้อมูลหลัก และสร้างแฟ้มควบคุมสำหรับฐานข้อมูลสำรองเพื่อทำการสร้างฐานข้อมูลสำรองบนเครื่องแม่ข่ายฐานข้อมูลสำรองต่อไป โดยที่เครื่องมือยังสามารถตรวจสอบสถานะการทำงานของฐานข้อมูลของทั้งสองตัวได้ ในกรณีที่ฐานข้อมูลหลักเกิดปัญหาขึ้น เจ้าหน้าที่ดูแลระบบฐานข้อมูลจะสามารถทำให้ฐานข้อมูลสำรองอยู่ในสภาพพร้อมใช้งานผ่านเครื่องมือนี้ได้ โดยในส่วนของการทำงาน (Application) ต้องทำการเปลี่ยนแฟ้มกำหนดการใช้ (Configuration File) เพื่อให้มาติดต่อใช้งานที่ฐานข้อมูลสำรองที่เปลี่ยนมาเป็นฐานข้อมูลหลักแทน ดังรูปที่ 3.1



รูปที่ 3.1 การสร้างและจัดการฐานข้อมูลสำรอง

ในส่วนที่สองนี้จะใช้การกำหนดเวลาทำงานอัตโนมัติ โดยอาศัยโปรแกรมอรรถประโยชน์ (Program Utility) ที่ชื่อว่า ครอนแทบ (Crontab) ซึ่งเป็นโปรแกรมอรรถประโยชน์ที่ติดตั้งมาในระบบปฏิบัติการเซลาริส โดยทุกๆ 5 นาที ครอนแทบ จะไปเรียกให้โปรแกรม 2 โปรแกรมที่พัฒนาขึ้นทำงาน ซึ่งโปรแกรมทั้งสองตัวมีการทำงานดังรูปที่ 3.2

จุฬาลงกรณ์มหาวิทยาลัย

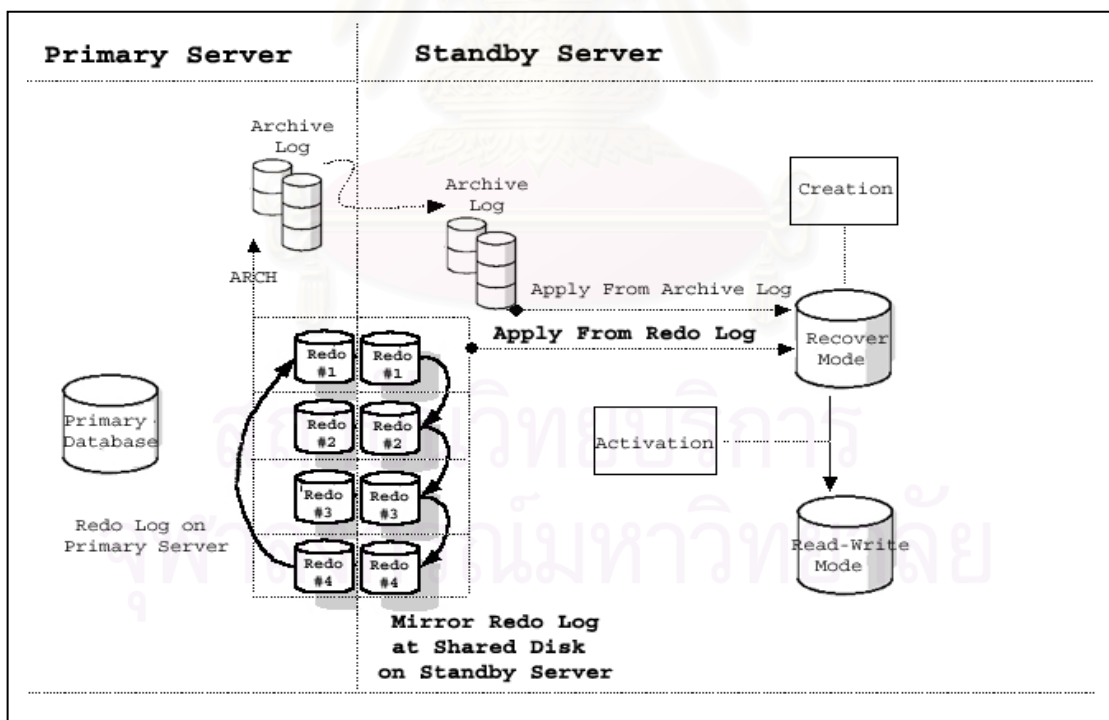


รูปที่ 3.2 โปรแกรมที่กำหนดเวลาทำงานอัตโนมัติ

โปรแกรมที่ทำงานอัตโนมัติตัวที่ 1 ทำหน้าที่ในการควบคุมการส่งผ่านแฟ้มลงบันทึกเก็บถาวรจากฐานข้อมูลหลักมายังฐานข้อมูลสำรอง โดยก่อนที่จะทำการส่งผ่านแฟ้มลงบันทึกเก็บถาวรจะมีการสั่งให้ฐานข้อมูลหลักทำการเขียนข้อมูลที่อยู่บนแฟ้มบันทึกทำซ้ำลงบนแฟ้มลงบันทึกเก็บถาวรก่อน (Switch Logfile) เพื่อให้รายการที่เปลี่ยนแปลงที่อยู่บนแฟ้มบันทึกทำซ้ำถูกนำไปปรับปรุงที่ฐานข้อมูลสำรองด้วย แต่ในการส่งผ่านแฟ้มลงบันทึกเก็บถาวรจะไม่นำแฟ้มลงบันทึกเก็บถาวรตัวใหม่ล่าสุดในขณะนั้นไปด้วย เนื่องจากอาจจะมีการบันทึกข้อมูลที่อยู่บนแฟ้มบันทึกทำซ้ำลงบนแฟ้มลงบันทึกเก็บถาวรไม่เสร็จสมบูรณ์ ทำให้ไม่สามารถนำข้อมูลที่อยู่บนแฟ้มลงบันทึกเก็บถาวรไปปรับปรุงได้

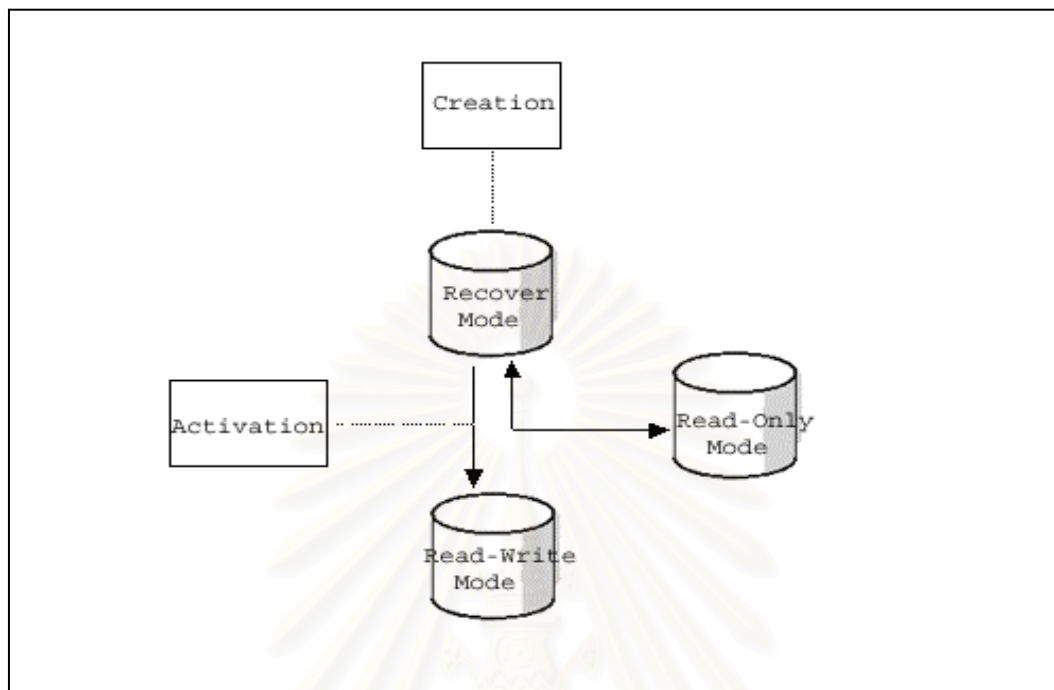
โปรแกรมที่ทำงานอัตโนมัติตัวที่ 2 การปรับปรุงข้อมูลจากแฟ้มลงบันทึกเก็บถาวรลงบนฐานข้อมูลสำรอง และการเปลี่ยนแปลงโครงสร้างทางกายภาพของระบบฐานข้อมูลสำรอง หลังจากนั้นก็จะเป็นส่วนในการปรับปรุงข้อมูลจากแฟ้มลงบันทึกเก็บถาวรเรียงตามลำดับลงบนฐานข้อมูลสำรอง เพื่อให้มีข้อมูลใกล้เคียงกับฐานข้อมูลหลักมากที่สุด และในกรณีที่ฐานข้อมูลหลักมีการเปลี่ยนแปลงโครงสร้างทางกายภาพ จะมีขั้นตอนในการตรวจสอบ และหลังจากนั้นก็ทำการเปลี่ยนแปลงโครงสร้างทางกายภาพของระบบฐานข้อมูลสำรองให้ เพื่อให้ฐานข้อมูลสำรองสามารถรองรับการปรับปรุงข้อมูลของรายการที่เปลี่ยนแปลงที่ส่งมาจากระบบฐานข้อมูลหลักได้

ในส่วนของการทำงานบนฐานข้อมูลหลักดังกล่าวมาแล้วในบทที่ 2 ประสิทธิภาพของรายการที่เปลี่ยนแปลงจะถูกบันทึกลงในแฟ้มลงบันทึกทำซ้ำจนกว่าจะเต็มจึงจะทำการบันทึกไปที่แฟ้มลงบันทึกเก็บถาวร แต่ในกรณีที่เครื่องแม่ข่ายหลักเกิดเหตุการณ์ร้ายแรงต่างๆเกิดขึ้น ในขณะที่แฟ้มลงบันทึกทำซ้ำยังไม่เต็ม ข้อมูลต่างๆที่มีการยืนยันการบันทึกแล้วก็จะหายไป เนื่องจากยังไม่มีทำการบันทึกไปเป็นแฟ้มลงบันทึกเก็บถาวรและไม่สามารถกู้คืนบนฐานข้อมูลสำรองได้ ในกรณีนี้จึงมีการออกแบบให้เพิ่มในส่วนของการจัดการเรื่อง แฟ้มลงบันทึกทำซ้ำที่มีการบันทึกพร้อมกัน (Mirror Redo Log) มายังเครื่องแม่ข่ายสำรองเพื่อลดปัญหาดังกล่าวได้ โดยเจ้าหน้าที่ดูแลระบบฐานข้อมูลสามารถกู้คืนข้อมูลจากแฟ้มลงบันทึกทำซ้ำตัวปัจจุบันได้ โดยในการปรับปรุงข้อมูลจะทำการอ้างเส้นทางไปยังแฟ้มลงบันทึกทำซ้ำตัวปัจจุบัน อีกชุดหนึ่งที่อยู่บนเครื่องแม่ข่ายสำรอง ดังแสดงในรูปที่ 3.3 ที่เครื่องแม่ข่ายหลักและเครื่องแม่ข่ายสำรองจะต้องมีการกำหนดร่วมกัน โดยให้เครื่องแม่ข่ายสำรองมีการแบ่ง(Shared) เนื้อที่บางส่วนของฮาร์ดดิสก์ให้เครื่องแม่ข่ายหลักสามารถทำการบันทึกข้อมูลลงไปได้ หลังจากนั้นก็จะทำการกำหนดในส่วนของฐานข้อมูลหลักให้สร้างจำนวนสมาชิก (Member) ของแฟ้มลงบันทึกทำซ้ำเพิ่มให้กับทุกกลุ่ม (Group) โดยที่เส้นทางกายภาพของสมาชิกแฟ้มลงบันทึกทำซ้ำที่เพิ่มจะถูกบันทึกลงตำแหน่งบนเนื้อที่ของฮาร์ดดิสก์ที่แม่ข่ายสำรองแบ่งไว้เพื่อเป็นประโยชน์ในการกู้คืนข้อมูลที่อยู่บนแฟ้มบันทึกทำซ้ำ



รูปที่ 3.3 การทำงานของแฟ้มลงบันทึกทำซ้ำที่มีการสำเนาไปที่เครื่องแม่ข่ายสำรอง

ฐานข้อมูลสำรองที่สร้างเสร็จเรียบร้อยแล้วจะมีสถานะการทำงานเป็นดังรูปที่ 3.4 โดยที่แต่ละสถานะอธิบายได้ดังนี้



รูปที่ 3.4 สถานะการทำงานของฐานข้อมูลสำรอง

1. สถานะรอการกู้คืน (Recover Mode) คือ สถานะที่ฐานข้อมูลสำรองรอแฟ้มลงบันทึกเก็บถาวรที่ส่งมาจากฐานข้อมูลหลักเพื่อทำการปรับปรุง(Apply) ให้มีข้อมูลใกล้เคียงกับฐานข้อมูลหลัก

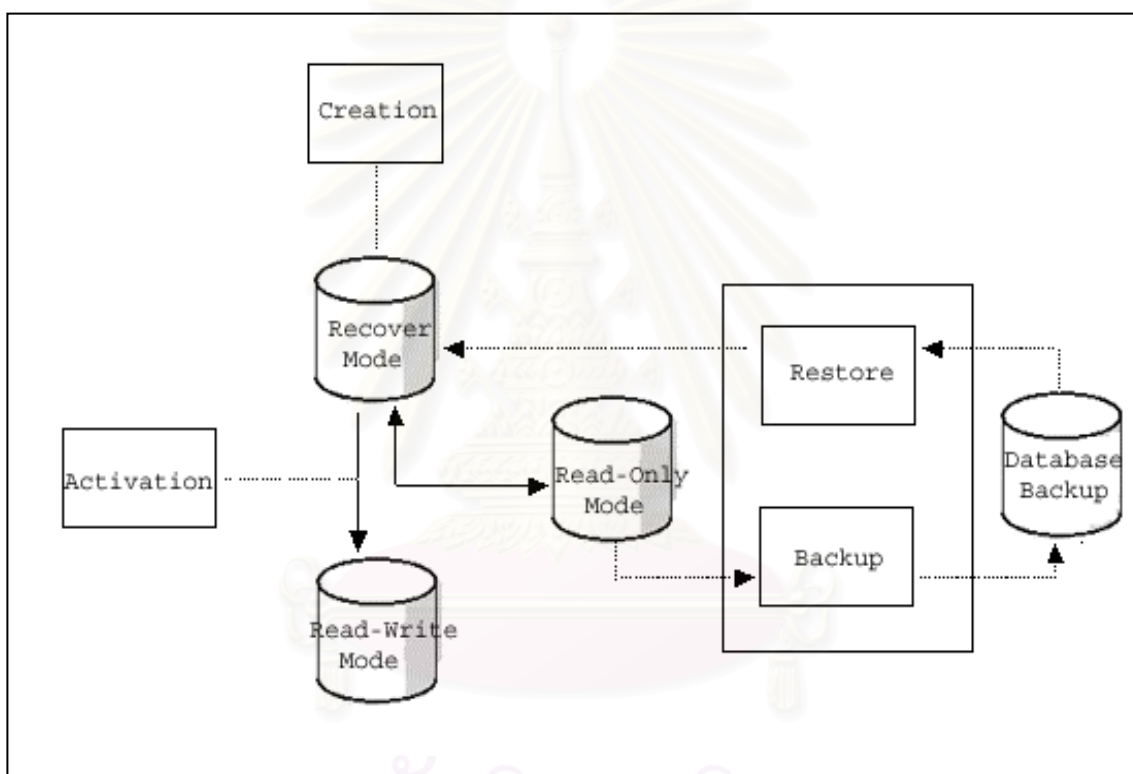
2. สถานะอ่านได้อย่างเดียว (Read-Only Mode) คือ สถานะที่ฐานข้อมูลสำรองสามารถเปิดขึ้นมาทำงานในลักษณะที่อ่านได้อย่างเดียว ไม่สามารถแก้ไขข้อมูลต่างๆ ได้ สถานะนี้สามารถนำไปประยุกต์ใช้งานในลักษณะการสอบถามข้อมูลเพื่อออกรายงานต่างๆได้

3. สถานะอ่านเขียน (Read-Write Mode) คือ สถานะที่ฐานข้อมูลสำรองถูกเปิดขึ้นมาอยู่ในสภาพพร้อมใช้งานเหมือนฐานข้อมูลหลัก

จากสถานะของฐานข้อมูลสำรองที่กล่าวมาแล้วนั้น ฐานข้อมูลสำรองจะสามารถเปลี่ยนจากสถานะใดสถานะหนึ่งไปเป็นอีกสถานะหนึ่งได้ 3 รูปแบบดังนี้

- เปลี่ยนสถานะจากการรอการกู้คืน ไปเป็นสถานะอ่านได้อย่างเดียว
- เปลี่ยนสถานะอ่านได้อย่างเดียวไปเป็นสถานะจากการรอการกู้คืน

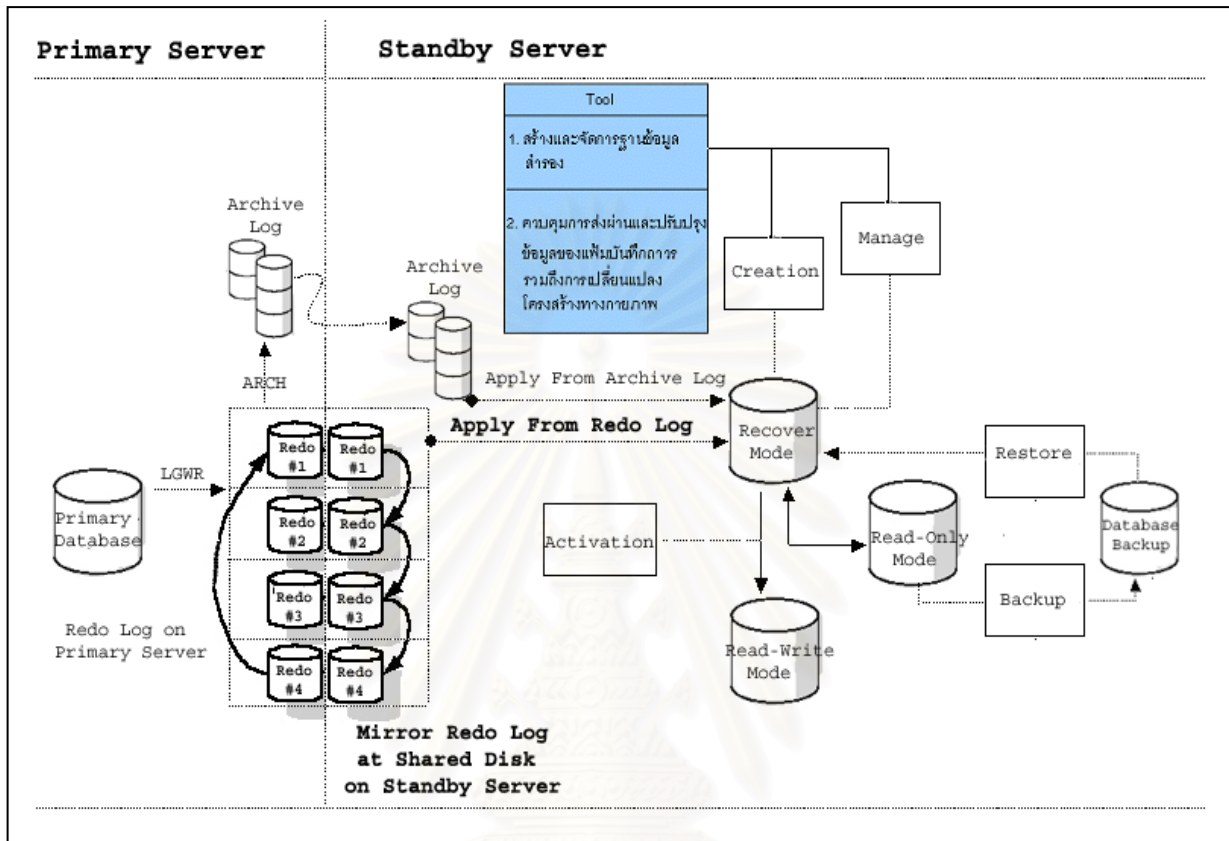
- เปลี่ยนสถานะจากการรอกู้คืนไปเป็นสถานะอ่านเขียนเหมือนฐานข้อมูลหลัก  
 ฐานข้อมูลสำรองจะมีข้อจำกัดว่าในกรณีที่มีการเปลี่ยนสถานะจากการรอกู้คืนไปเป็นสถานะอ่านเขียนเพื่อให้อยู่ในสภาพพร้อมใช้เหมือนฐานข้อมูลหลักแล้ว จะไม่สามารถเปลี่ยนสถานะกลับเป็นสถานะการรอกู้คืนได้อีก ดังนั้นในกรณีที่ต้องการให้ฐานข้อมูลอยู่ในสถานะการรอกู้คืนอีกครั้งจึงจำเป็นต้องทำการสร้างฐานข้อมูลสำรองใหม่อีกครั้ง ซึ่งช่วงเวลาในการสร้างแต่ละครั้งจะมีผลกระทบต่อประสิทธิภาพของฐานข้อมูลหลักลดลง ในทางวิจยนี้จึงมีการออกแบบให้เพิ่มในส่วนของการจัดการทำการสำรองข้อมูลและการนำข้อมูลกลับคืนบนฐานข้อมูลสำรอง เพื่อให้สามารถนำฐานข้อมูลสำรองกลับคืนมาอยู่ในสถานะการรอกู้คืนอีกครั้ง ดังรูปที่ 3.5



รูปที่ 3.5 สำรองข้อมูลและนำข้อมูลกลับคืนบนฐานข้อมูลสำรอง



ในการออกแบบงานวิจัยครั้งนี้เมื่อนำแต่ละส่วนมาประกอบกันจะมีภาพการทำงานของเครื่องมือสร้างและจัดการฐานข้อมูลสำรองดังรูปที่ 3.6

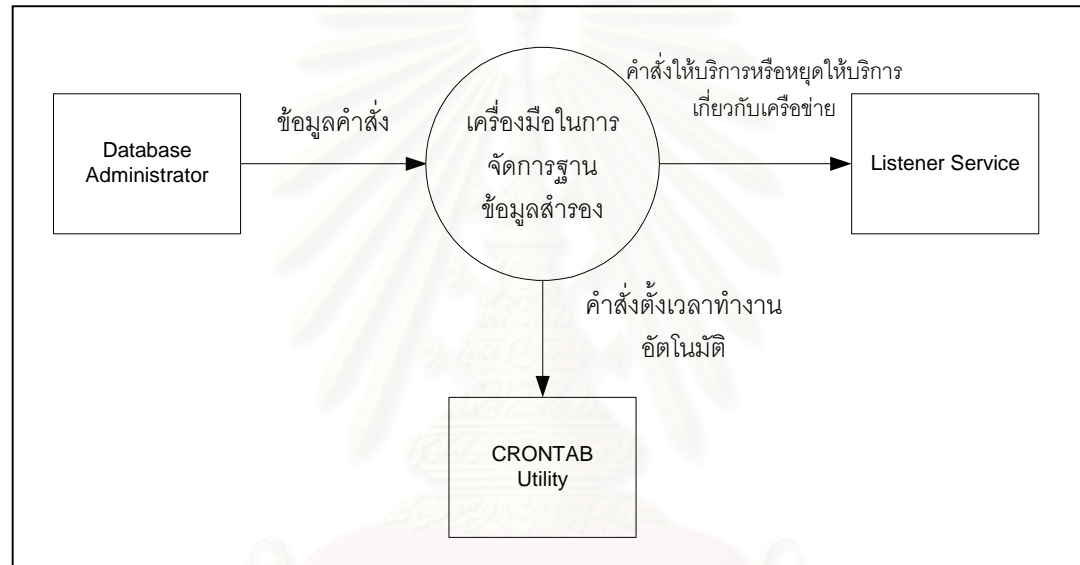


รูปที่ 3.6 การทำงานของเครื่องมือสร้างและจัดการฐานข้อมูลสำรอง

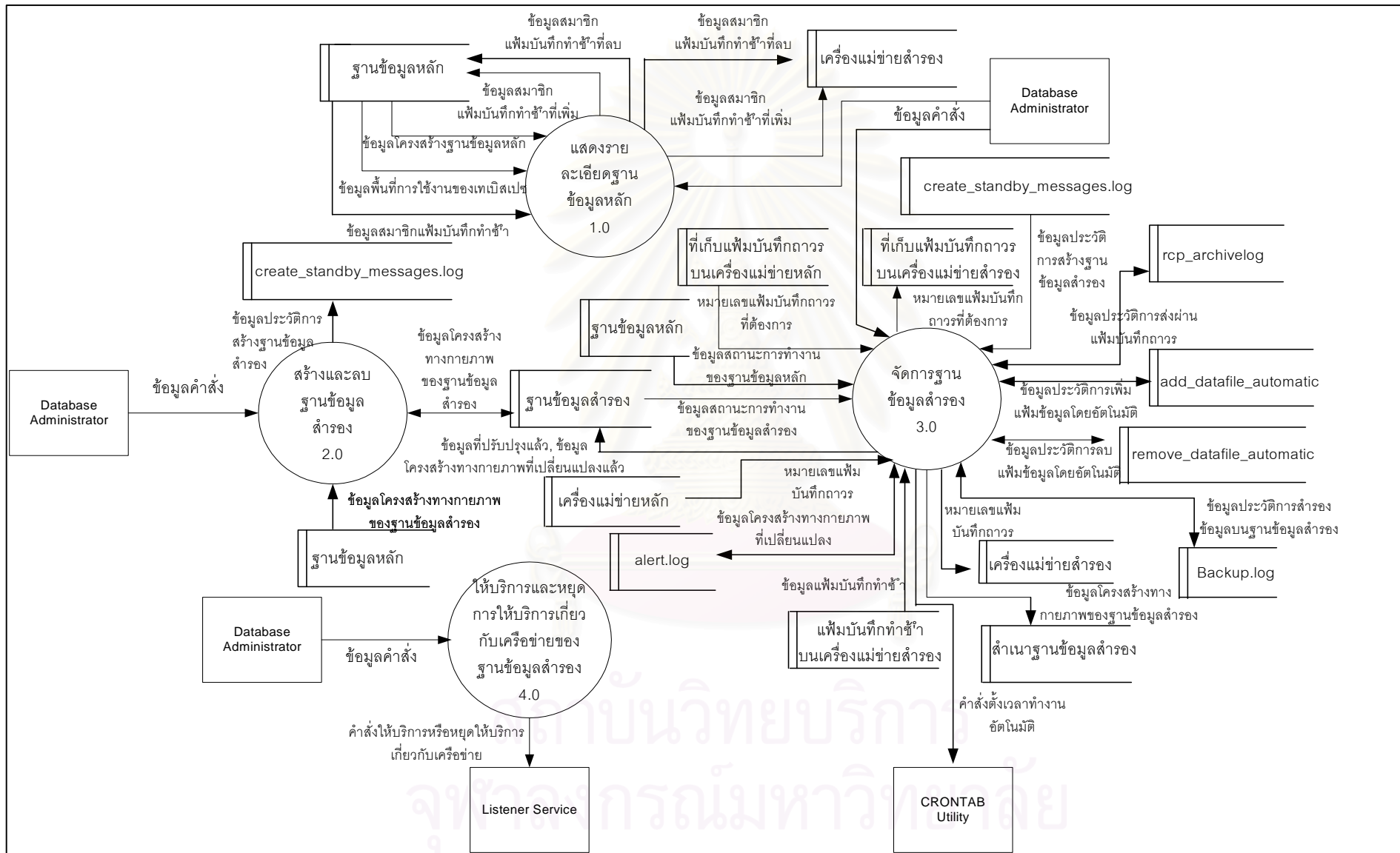
3.2 แผนภาพคอนเท็กซ์ (Context Diagram) และแผนภาพกระแสข้อมูลระบบ (Data Flow Diagram)

การออกแบบเครื่องมือจัดการฐานข้อมูลสำรองสามารถแสดงการทำงานได้ดังแผนภาพคอนเท็กซ์ รูปที่ 3.7 และ แผนภาพกระแสข้อมูลในระดับที่ 1 ระดับที่ 2 และ ระดับที่ 3 ดังในแผนภาพรูปที่ 3.8 – 3.16

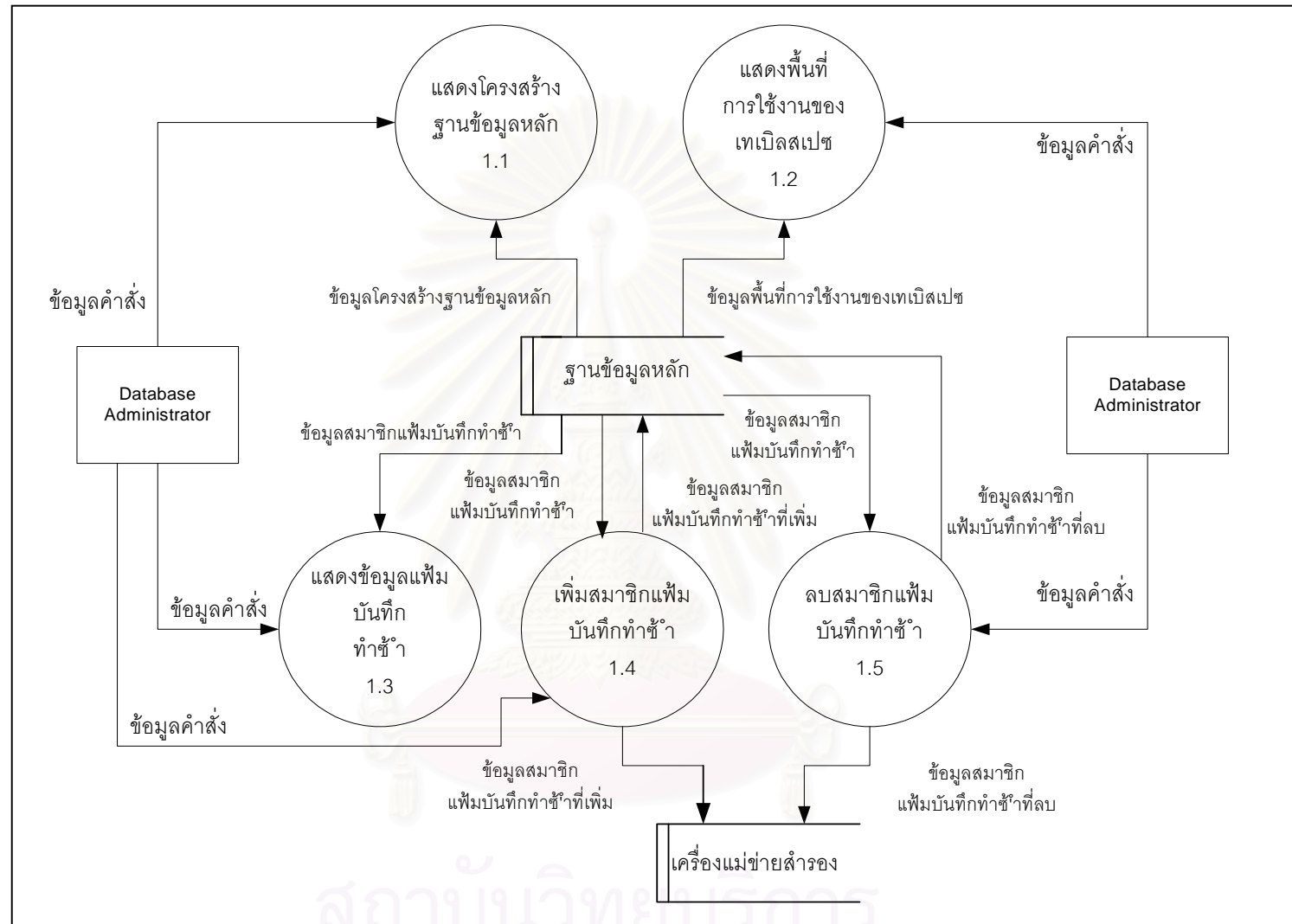




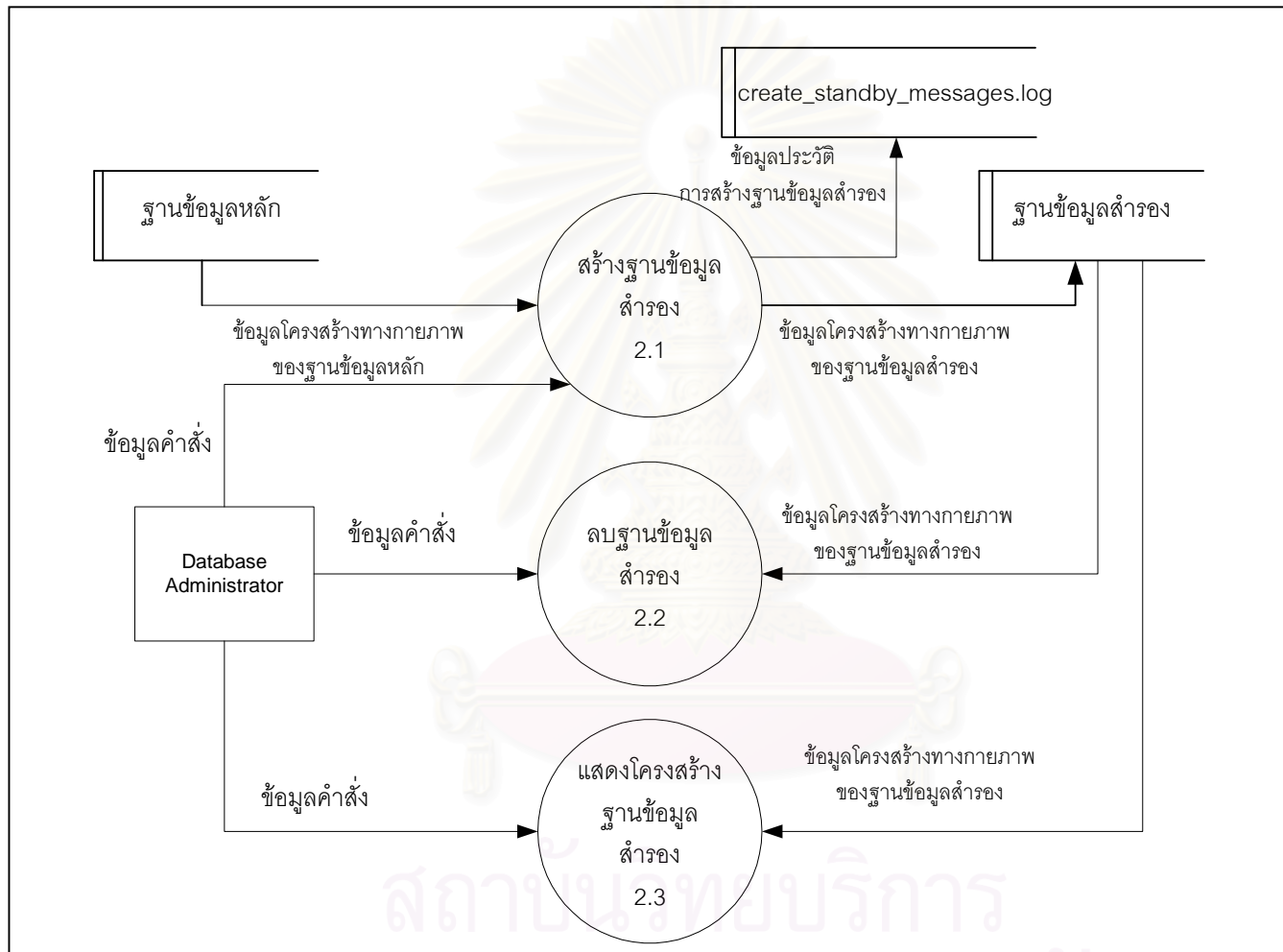
รูปที่ 3.7 แผนภาพคอนเทกซ์



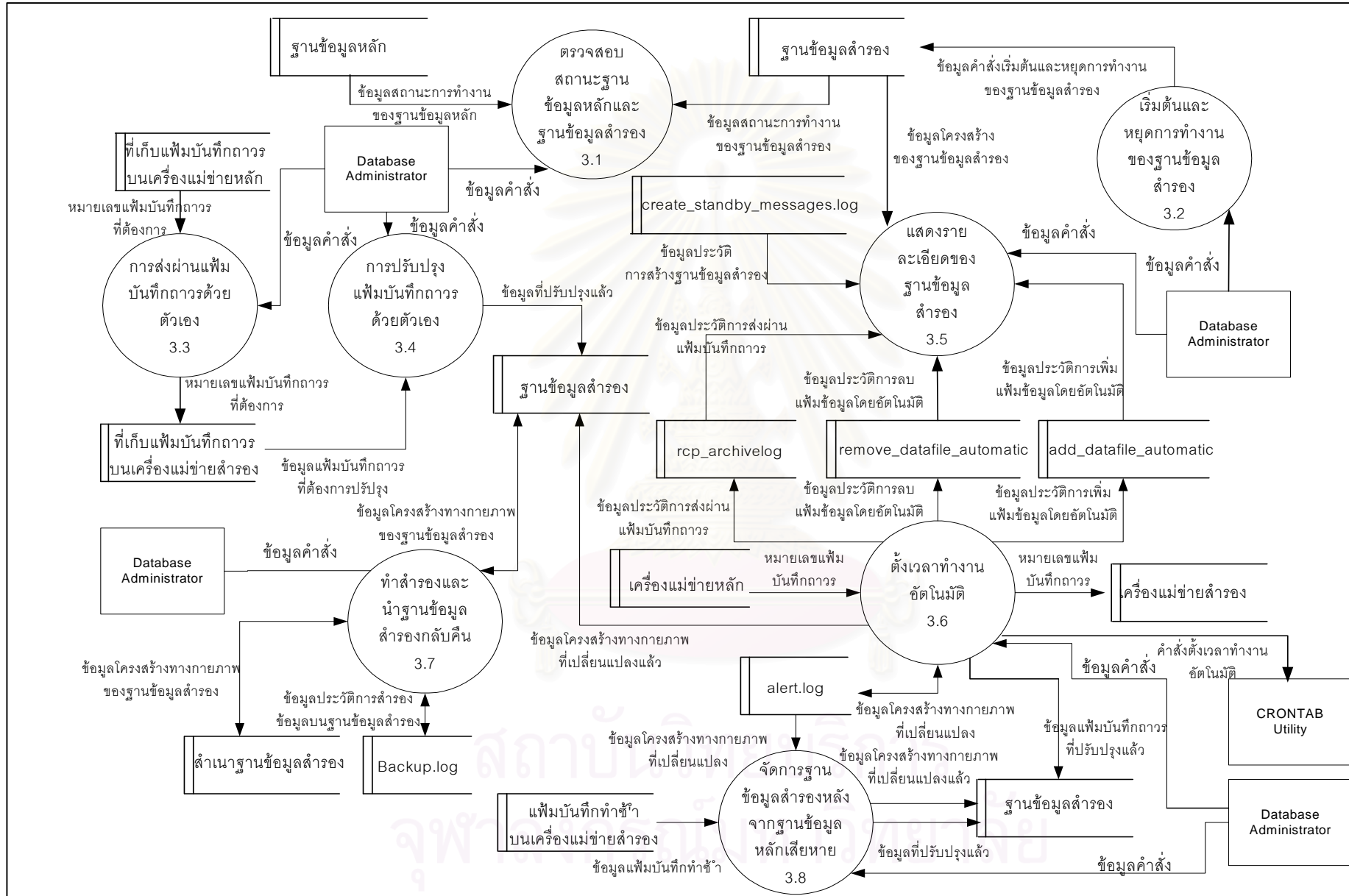
รูปที่ 3.8 แผนภาพกระแสข้อมูลในระดับที่ 1



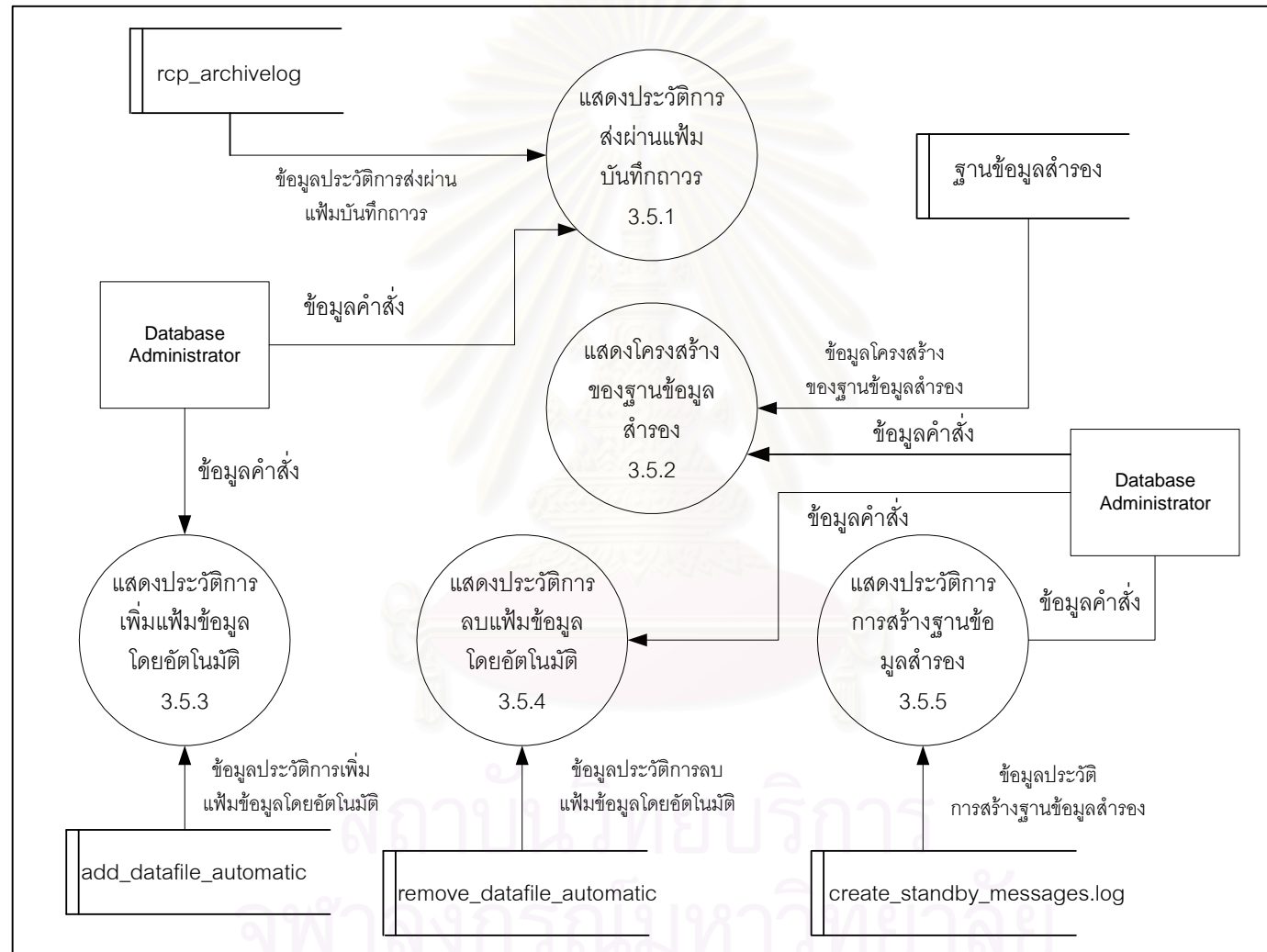
รูปที่ 3.9 แผนภาพกระแสข้อมูลในระดับที่ 2



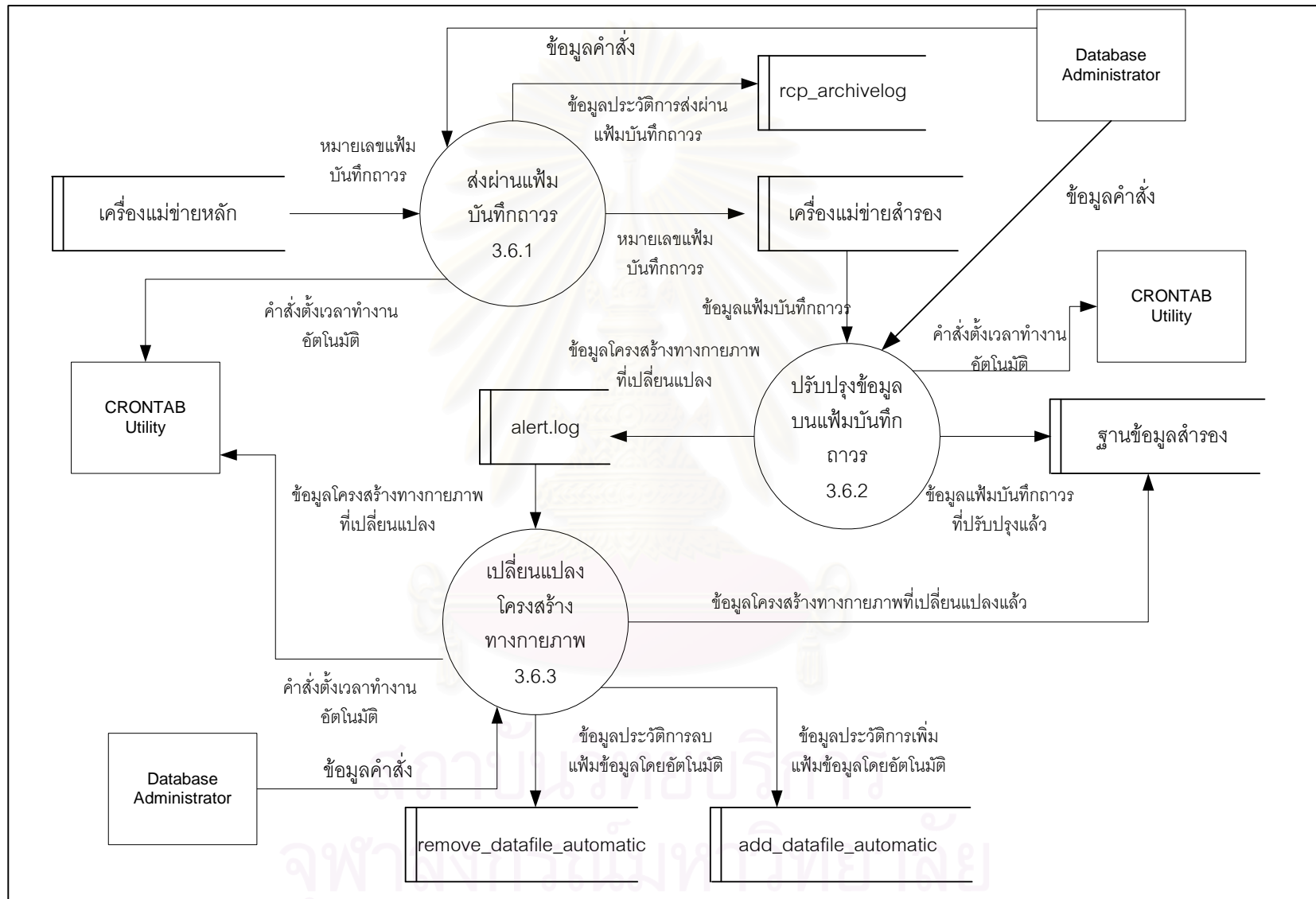
รูปที่ 3.10 แผนภาพกระแสข้อมูลในระดับที่ 2



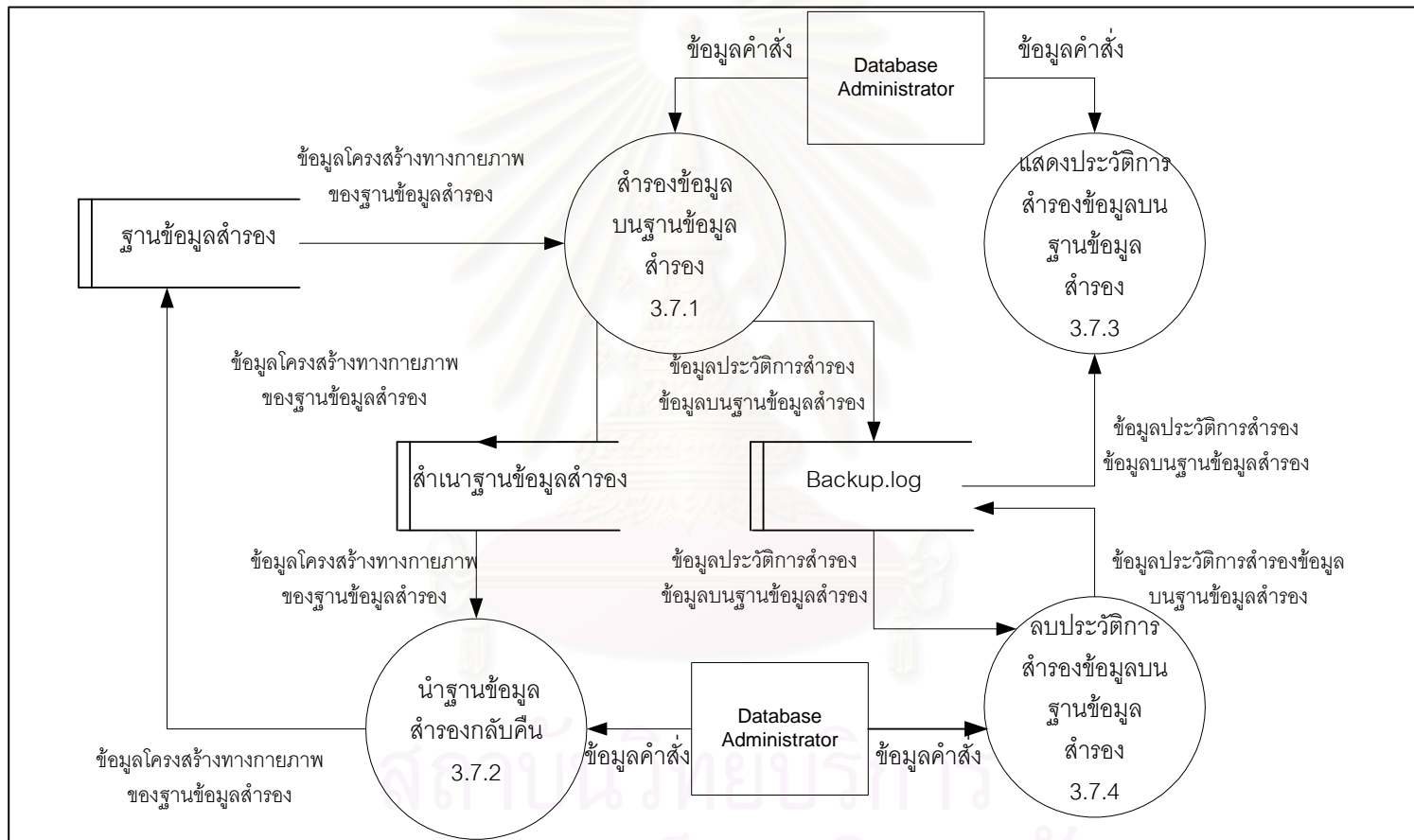
รูปที่ 3.11 แผนภาพกระแสข้อมูลในระดับที่ 2



รูปที่ 3.12 แผนภาพกระแสข้อมูลในระดับที่ 3



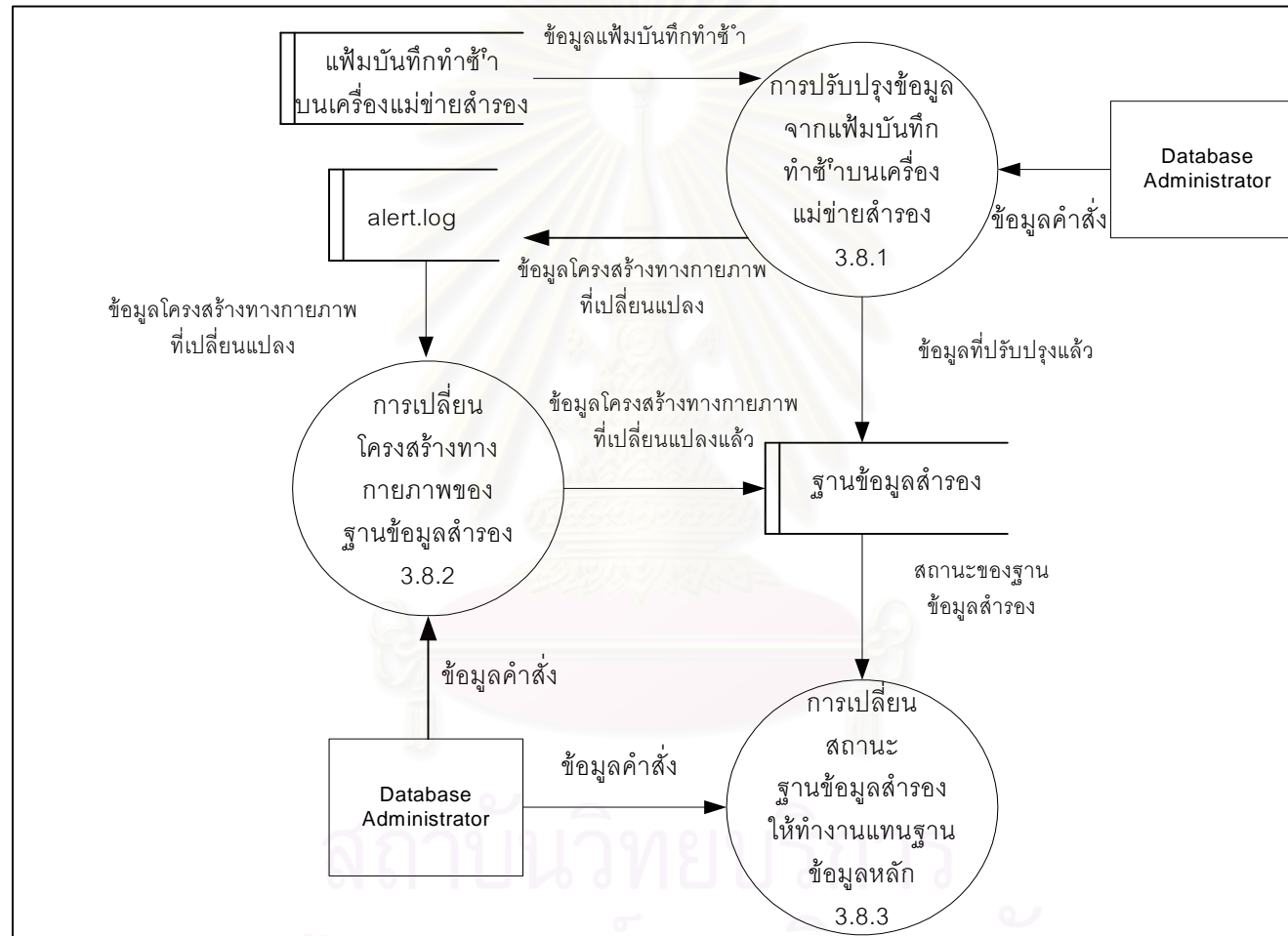
รูปที่ 3.13 แผนภาพกระแสข้อมูลในระดับที่ 3



รูปที่ 3.14 แผนภาพกระแสข้อมูลในระดับที่ 3

จุฬาลงกรณ์มหาวิทยาลัย





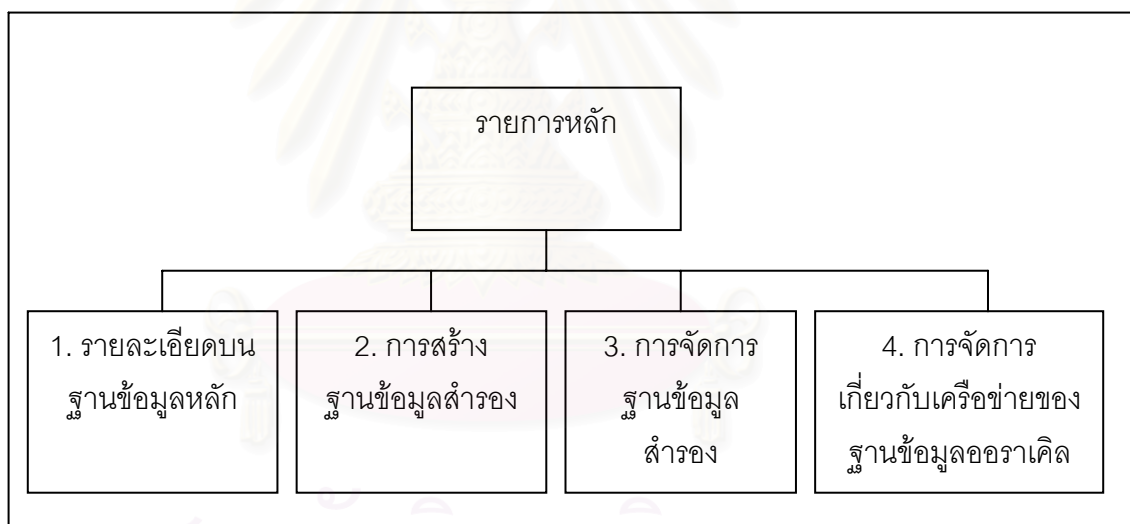
รูปที่ 3.15 แผนภาพกระแสข้อมูลในระดับที่ 3

### 3.3. ขั้นตอนในการออกแบบลักษณะส่วนติดต่อผู้ใช้ (user interface)

ในขั้นตอนการออกแบบเครื่องมือที่ใช้ในการสร้างและจัดการฐานข้อมูลสำรองในส่วนของลักษณะส่วนติดต่อผู้ใช้ เป็นลักษณะรายการเมนู โดยมีลักษณะการใช้งานดังนี้

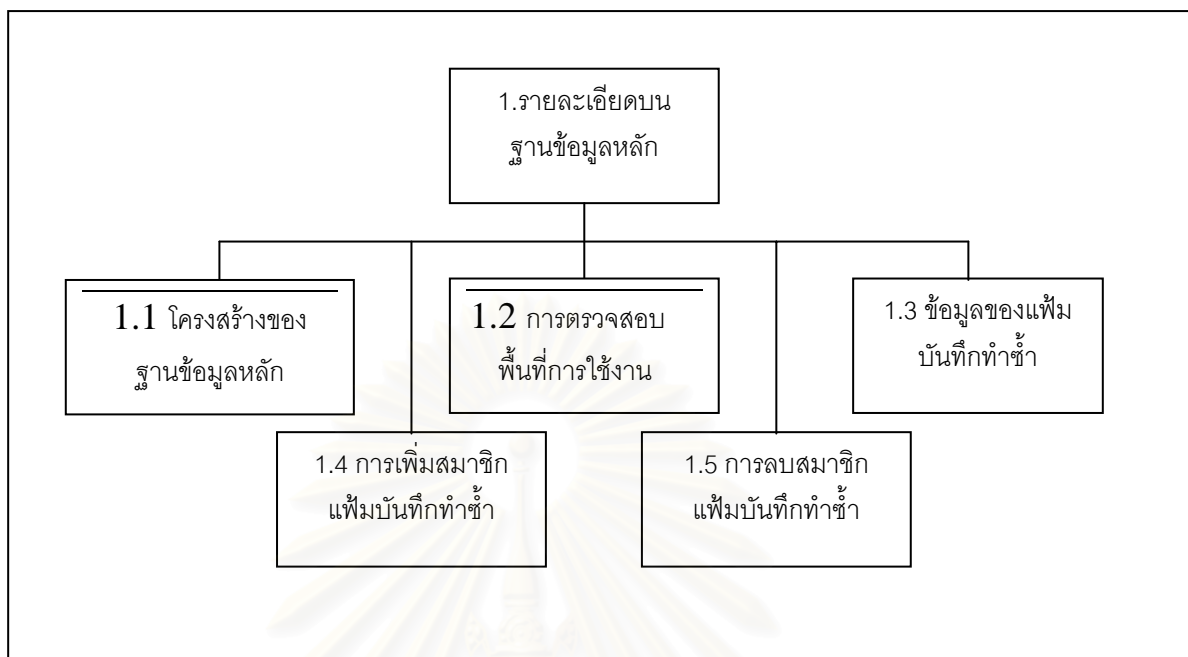
- ให้ผู้ใช้เลือกทำรายการ โดยใส่หมายเลขรายการที่ต้องการจากนั้นผู้ใช้ต้องกดปุ่ม enter แล้วโปรแกรมจะทำงาน
- หลังจากผู้ใช้เลือกหัวข้อที่ทำงานและโปรแกรมทำงานเสร็จแล้วจะมีข้อความแนะนำ ซึ่งเมื่อผู้ใช้ทำตามคำแนะนำแล้ว โปรแกรมจะกลับเข้าสู่รายการตามเดิม
- จะมีรายการให้เลือกในกรณีต้องการออกจากระบบ ถ้าอยู่ในรายการย่อยจะกลับมายังรายการหลัก หรือรายการก่อนหน้า

โดยที่ในแต่ละเมนูรายการจะมีรายการแสดงดังรูปที่ 3.16 – รูปที่ 3.24



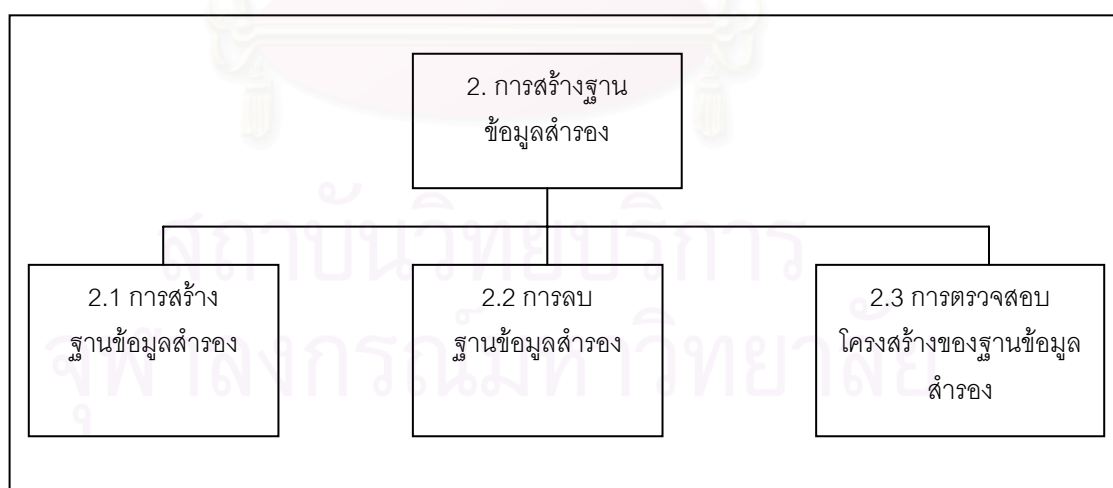
รูปที่ 3.16 เมนูรายการหลัก

### 3.2.1 เมนูรายการรายสำหรั้ตรวจสอบรายละเอียดบนฐานข้อมูลหลัก



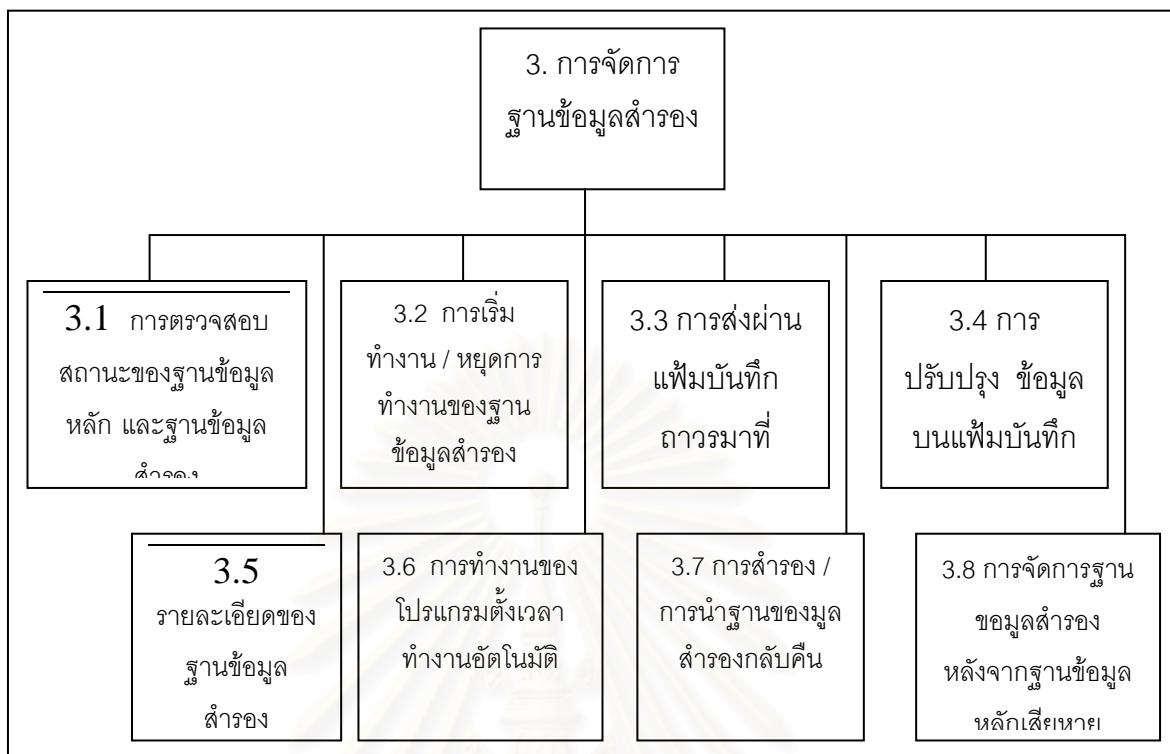
รูปที่ 3.17 การรายสำหรั้ตรวจสอบรายละเอียดบนฐานข้อมูลหลัก

### 3.2.2 เมนูรายการสำหรั้การสร้างและลบฐานข้อมูลสำรอง



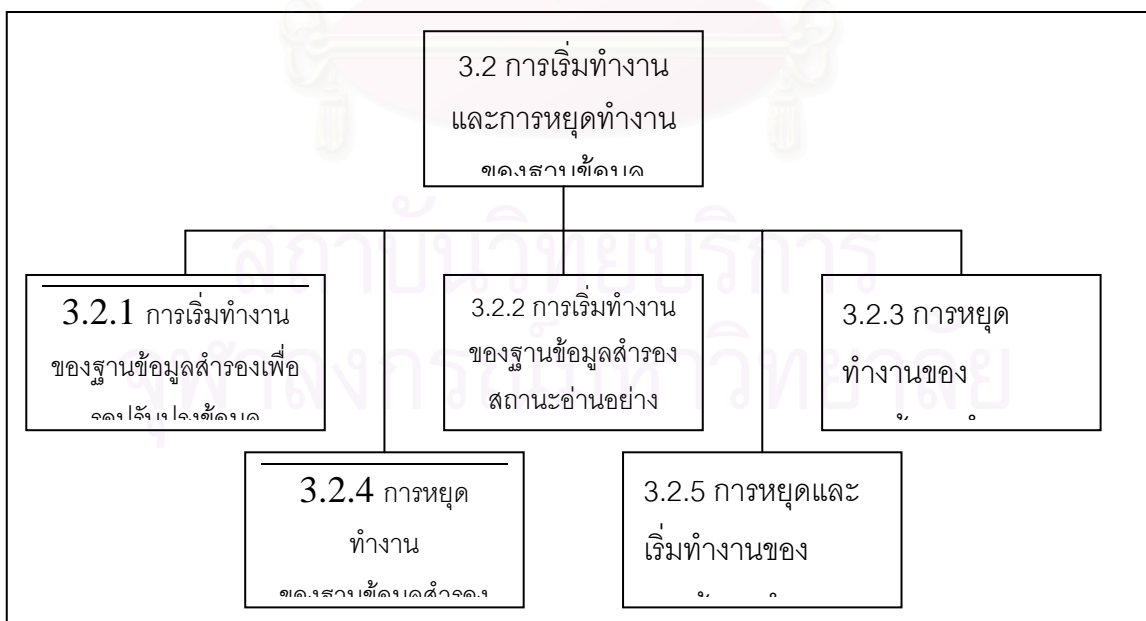
รูปที่ 3.18 รายการสร้างและลบฐานข้อมูลสำรอง

## 3.2.3 เมนูรายการสำหรับการจัดการฐานข้อมูลสำรอง



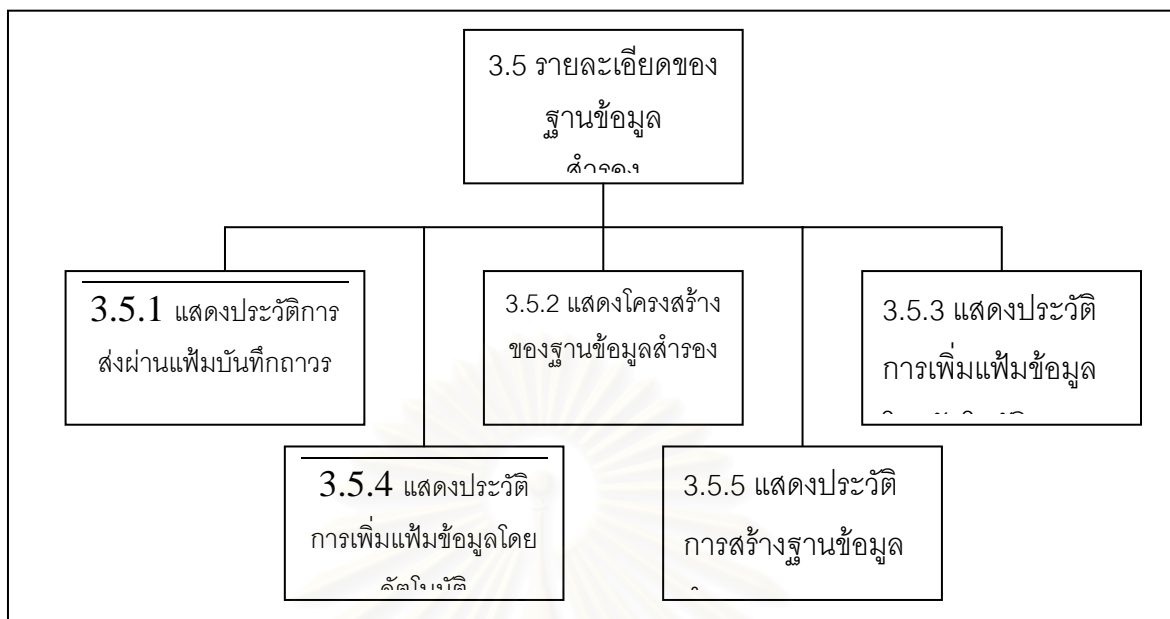
รูปที่ 3.19 รายการรายการสำหรับการจัดการฐานข้อมูลสำรอง

## 3.2.3.1 เมนูรายการย่อยสำหรับการเริ่มการทำงานและหยุดการทำงานฐานข้อมูลสำรอง



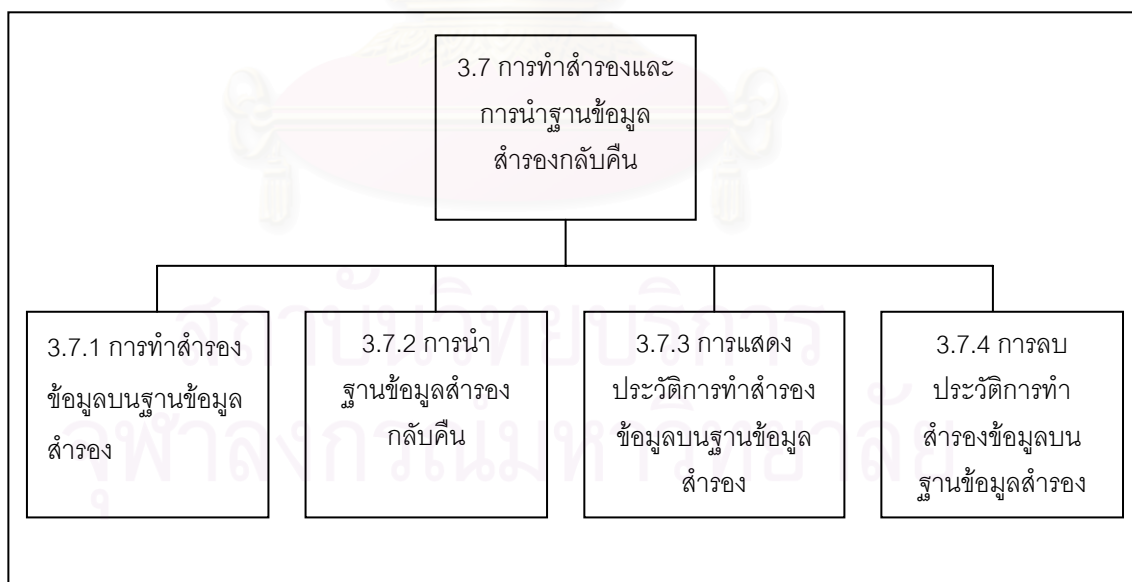
รูปที่ 3.20 รายการย่อยสำหรับการเริ่มการทำงานและหยุดการทำงานฐานข้อมูลสำรอง

1.1.1.1.1 3.2.3.2 เมนูรายการย่อยสำหรับการแสดงข้อมูลเกี่ยวกับฐานข้อมูลสำรอง



รูปที่ 3.21 รายการย่อยสำหรับการแสดงข้อมูลเกี่ยวกับฐานข้อมูลสำรอง

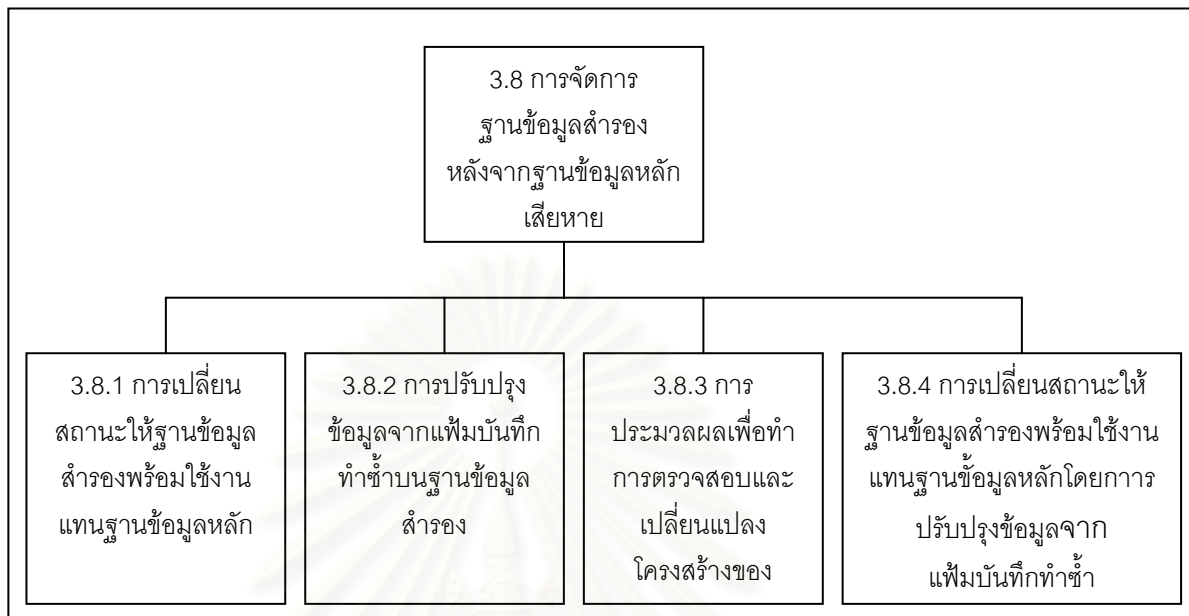
3.2.3.3 เมนูรายการย่อยสำหรับการสำรองและการนำฐานข้อมูลสำรองกลับคืน



รูปที่ 3.22 รายการย่อยสำหรับการสำรองและการนำฐานข้อมูลสำรองกลับคืน

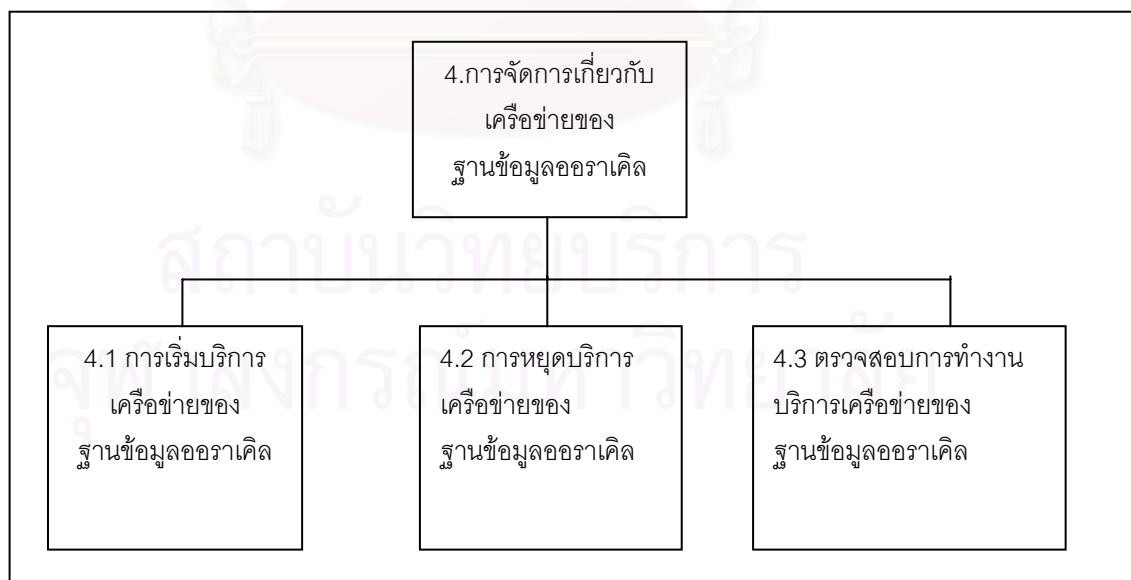
### 3.2.3.4 เมนูรายการย่อยการจัดการกับฐานข้อมูลสำรองหลังจากฐานข้อมูลหลัก

เสียหาย



รูปที่ 3.23 การย่อยสำหรับการจัดการกับฐานข้อมูลสำรองหลังจากฐานข้อมูลหลักเสียหาย

### 3.2.4 เมนูรายการสำหรับการจัดการให้บริการด้านเครื่องข่ายของฐานข้อมูลสำรอง



รูปที่ 3.24 เมนูรายการสำหรับการจัดการให้บริการด้านเครื่องข่ายของฐานข้อมูลสำรอง

## บทที่ 4

### การพัฒนา

ขั้นตอนในการพัฒนาเครื่องมือสร้างและจัดการฐานข้อมูลสำรอง ได้ทำการพัฒนาบนเครื่อง  
ชั้น รุ่น อัลตราทู (Ultra 2) ซึ่งใช้โซลาริสเป็นระบบปฏิบัติการ

#### 4.1 ภาษาที่ใช้ในการพัฒนา

ในการพัฒนาเครื่องมือในการสร้างและจัดการฐานข้อมูลสำรองผู้วิจัยเลือกใช้ภาษาคอรัน  
เชลล์ (Korn Shell) ซึ่งมีเหตุผลดังนี้

1. เป็นเชลล์(Shell)มาตรฐานที่มีการพัฒนาล่าสุดของระบบยูนิกซ์ (Unix)
2. มีลักษณะเป็นภาษาแบบแปลย่อย (Interpretive) ทำให้สะดวกต่อการพัฒนา และการ  
ดูแลรักษาโปรแกรม
3. สามารถนำฟังก์ชันเชลล์และโปรแกรมที่สร้างด้วยภาษาเชลล์มาใช้งานได้เหมือน  
โปรแกรมอรรถประโยชน์ในระบบยูนิกซ์ ทำให้สะดวกในการทำโปรแกรมโครงสร้างระบบ
4. ใช้อำนวยต่อการนำโปรแกรมอรรถประโยชน์มาทำงานร่วมกันโดยใช้ไปป์ การทำงาน  
เบื้องหลังและมีโครงสร้างควบคุมทิศทางการทำงานของอินพุต และเอาต์พุต

ในการพัฒนาจะทำการแบ่งโปรแกรมเป็นส่วนๆ เพื่อความสะดวกในการพัฒนา ซึ่งในแต่ละ  
ส่วนมีการแบ่งออกเป็นไดเรกทอรีย่อย (Sub Directory) ตามลักษณะของงานดังนี้

- ไดเรกทอรี ENV จะเป็นที่เก็บแฟ้มสภาพแวดล้อม ตัวแปร และพารามิเตอร์ต่างๆ ที่จำเป็น  
ในการทำงาน
- ไดเรกทอรี MENU จะเป็นที่เก็บแฟ้มเกี่ยวกับการแสดงรายการเมนูหลักและเมนูย่อย  
ต่างๆ
- ไดเรกทอรี SCRIPT จะเป็นที่เก็บแฟ้มโปรแกรมในการทำงานหลักๆ ของเครื่องมือที่  
พัฒนาขึ้น
- ไดเรกทอรี SQL จะเป็นที่เก็บแฟ้มในส่วนที่มีการทำงานกับฐานข้อมูล ซึ่งแฟ้มเหล่านี้จะมี  
นามสกุลเป็นเอสคิวแอล



- ไดรคทอรี PHYSICAL จะทำงานในส่วนของการเปลี่ยนแปลงโครงสร้างฐานข้อมูลสำรองในกรณีที่ฐานข้อมูลหลักมีการเพิ่มและลบเพิ่มข้อมูล
- ไดรคทอรี ONLINE จะทำงานในส่วนของการตรวจสอบสถานะของฐานข้อมูลหลักและฐานข้อมูลสำรอง
- ไดรคทอรี LOG จะเป็นที่สำคัญสำหรับเก็บแฟ้มล็อกต่างๆ

#### 4.2 การทำงานของชุดฟังก์ชัน

ในหัวข้อนี้จะกล่าวถึงการทำงานในส่วนของชุดฟังก์ชันที่สำคัญซึ่งจะมีการเรียกใช้จากเมนูรายการต่างๆ ดังที่กล่าวมาแล้วในบทที่ 3 ดังนี้

##### 4.2.1 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลหลัก

ในส่วนนี้จะแสดงส่วนของชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลหลักที่มีชื่อว่า `base_command_primary.sh` จะเป็นชุดคำสั่งของออรากเคิล-เอสคิวแอล (Oracle SQL) สำหรับติดต่อกับฐานข้อมูลหลัก โดยที่ชุดฟังก์ชันเหล่านี้จะถูกเรียกใช้งานจากโปรแกรมในส่วนต่างๆ ซึ่งจะมีการเรียกใช้จากเมนูรายการ และทำหน้าที่ดังตารางที่ 4.1

ตารางที่ 4.1 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลหลัก

ชื่อฟังก์ชัน	หน้าที่	เรียกจากเมนูรายการ
<code>add_redo_remote_group</code>	ทำหน้าที่ในการเพิ่มจำนวนสมาชิกเพิ่มบนที่กทำซ้ำบนฐานข้อมูลหลักที่มีการเขียนมาที่เครื่องแม่ข่ายสำรอง	1.4
<code>archivelog_current</code>	ทำหน้าที่ในการสั่งให้ฐานข้อมูลหลักทำการบันทึกข้อมูลที่อยู่บนเพิ่มบันทึกทำซ้ำตัวที่ใช้งานอยู่ลงบนเพิ่มบันทึกถาวร	3.6
<code>create_standby_controlfile</code>	ทำหน้าที่ในการสร้างแฟ้มควบคุมสำหรับฐานข้อมูลสำรอง	2.1

ตารางที่ 4.1 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลหลัก (ต่อ)

ชื่อฟังก์ชัน	หน้าที่	เรียกจากเมนูรายการ
get_chk_space_db_sqlplus	ทำหน้าที่ในการสอบถามข้อมูลเกี่ยวกับการใช้งานของเนื้อที่ของเทเบิลสเปซบนฐานข้อมูลหลัก	1.2
gen_hot_backup_sqlplus	ทำหน้าที่ในการสร้างโปรแกรมที่ใช้ในการเตรียมการสำรองข้อมูลบนฐานข้อมูลหลักเพื่อที่จะใช้ในการสร้างฐานข้อมูลสำรองต่อไป	2.1
get_datafiles	ทำหน้าที่ในการค้นหาชื่อ และตำแหน่งของแฟ้มข้อมูล	2.1
get_info_primary_sqlplus	ทำหน้าที่ในการสอบถามข้อมูลเกี่ยวกับโครงสร้างของฐานข้อมูลหลัก	1.1
get_info_tbs_dbf_sqlplus	ทำหน้าที่ในการนับจำนวนของเทเบิลสเปซและแฟ้มข้อมูลบนบนฐานข้อมูลหลัก	3.1
max_new_log	ทำหน้าที่ในการหาแฟ้มบันทึกถาวรตัวสุดท้ายเพื่อใช้เป็นตัวเปรียบเทียบการส่งผ่านแฟ้มบันทึกถาวรจากเครื่องแม่ข่ายหลักมาที่เครื่องแม่ข่ายสำรอง	3.6
primary_drop_remote_redo log	ทำหน้าที่ในการลบจำนวนสมาชิกแฟ้มบันทึกทำซ้ำบนฐานข้อมูลหลักที่มีการเขียนมาที่เครื่องแม่ข่ายสำรอง	1.5
primary_find_current_redo log	ทำหน้าที่ในการสืบค้นแฟ้มบันทึกทำซ้ำตัวปัจจุบันที่มีการใช้งานอยู่บนฐานข้อมูลหลัก	1.4,1.5

ตารางที่ 4.1 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลหลัก (ต่อ)

ชื่อฟังก์ชัน	หน้าที่	เรียกจากเมนูรายการ
primary_find_remote_redolog	ทำหน้าที่ในการสืบค้นการเพิ่มจำนวนสมาชิกแฟ้มบันทึกทำซ้ำบนฐานข้อมูลหลัก	1.4
primary_redo_infor_sqlplus	ทำหน้าที่ในการสอบถามข้อมูลเกี่ยวกับสมาชิกทั้งหมดของแฟ้มบันทึกทำซ้ำ	1.3
rcp_archive_log	ทำหน้าที่ในการส่งผ่านแฟ้มบันทึกถาวรจากเครื่องแม่ข่ายหลักมาที่เครื่องแม่ข่ายสำรอง	3.6
rcp_control_file_sqlplus	ทำหน้าที่ในการส่งผ่านแฟ้มควบคุมสำหรับฐานข้อมูลสำรองที่สร้างขึ้นมาที่เครื่องแม่ข่ายสำรอง	2.1,3.6
redo_group_count	ทำหน้าที่ในการนับจำนวนกลุ่มของแฟ้มบันทึกทำซ้ำบนฐานข้อมูลหลัก	1.1
run_hot_backup_sqlplus	ทำหน้าที่ในการสั่งให้โปรแกรมที่ใช้ในการเตรียมการสำรองข้อมูลบนฐานข้อมูลหลักเพื่อที่จะใช้ในการสร้างฐานข้อมูลสำรองทำงาน	2.1
switchlog	ทำหน้าที่ในการสั่งให้ฐานข้อมูลหลักทำการสลับการใช้งานแฟ้มบันทึกทำซ้ำไปที่ตัวใหม่	3.6

## 4.2.2 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลสำรอง

ในส่วนนี้จะแสดงส่วนของชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลสำรองที่มีชื่อว่า

base\_command\_standby.sh จะเป็นชุดคำสั่งของออราเคิล-เอสคิวแอล สำหรับติดต่อกับฐานข้อมูลสำรอง โดยที่ชุดฟังก์ชันเหล่านี้จะมีการเรียกใช้จากเมนูรายการ และทำหน้าที่ดังตารางที่ 4.2 ดังนี้

ตารางที่ 4.2 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลสำรอง

ชื่อฟังก์ชัน	หน้าที่	เรียกจากเมนูรายการ
activate_standby_sqlplus	ทำหน้าที่ในการเปลี่ยนสถานะของฐานข้อมูลสำรองให้อยู่ในสถานะเขียนอ่านและพร้อมทำงานแทนฐานข้อมูลหลัก	3.8.1,3.8.4
drop_remote_redolog	ทำหน้าที่ในการลบสมาชิกแฟ้มบันทึกทำซ้ำที่เพิ่มขึ้นมาใหม่ออกจากฐานข้อมูลสำรองหลังจากเปลี่ยนสถานะเป็นฐานข้อมูลหลักแล้ว	3.8
file_need_recover_sqlplus	ทำหน้าที่ในการการค้นหาชื่อและตำแหน่งของแฟ้มข้อมูลของฐานข้อมูลสำรองที่ต้องการการกู้คืน	3.6,3.8.2
find_current_redolog	ทำหน้าที่ในการตรวจสอบจำนวนสมาชิกแฟ้มบันทึกทำซ้ำที่กำลังใช้งานอยู่เพื่อที่จะทำการลบออกจากฐานข้อมูลสำรองหลังจากเปลี่ยนสถานะเป็นฐานข้อมูลหลักแล้ว	3.8
find_remote_redolog	ทำหน้าที่ในการตรวจสอบจำนวนสมาชิกแฟ้มบันทึกทำซ้ำเพื่อที่จะทำการลบออกจากฐานข้อมูลสำรองหลังจากเปลี่ยนสถานะเป็นฐานข้อมูลหลักแล้ว	3.8
gen_standby_backup_sqlplus	ทำหน้าที่ในการทำหน้าที่ในการสร้างโปรแกรมที่ใช้ในการเตรียมการสำรองข้อมูลบนฐานข้อมูลหลักเพื่อที่จะใช้ในการสร้างฐานข้อมูลสำรองต่อไป	3.7

ตารางที่ 4.2 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลสำรอง (ต่อ)

ชื่อฟังก์ชัน	หน้าที่	เรียกจากเมนูรายการ
get_dbfiles_standby_sqlplus	ทำหน้าที่ในการค้นหาชื่อ และตำแหน่งของแฟ้มข้อมูลของฐานข้อมูลสำรอง	2.1,3.7
get_info_standby_sqlplus	ทำหน้าที่ในการตรวจสอบโครงสร้างของฐานข้อมูลสำรอง	3.5
recover_remote_redolog_sqlplus	ทำหน้าที่ในการปรับปรุงข้อมูลบนแฟ้มบันทึกทำซ้ำที่มีการบันทึกมาที่เครื่องแม่ข่ายสำรอง	3.8
recover_standby_sqlplus	ทำหน้าที่ในการปรับปรุงข้อมูลจากแฟ้มบันทึกถาวร	3.4,3.6
shutdown_normal_sqlplus	ทำหน้าที่ในการหยุดการทำงานของฐานข้อมูลสำรอง	3.2
standby_listener_start	ทำหน้าที่ในการเริ่มต้นให้บริการด้านเครือข่ายของฐานข้อมูลสำรอง	4.1
standby_listener_status	ทำหน้าที่ในการตรวจสอบสถานะการทำงานของกรให้บริการด้านเครือข่ายของฐานข้อมูลสำรอง	4.3
standby_listener_stop	ทำหน้าที่ในการหยุดให้บริการด้านเครือข่ายของฐานข้อมูลสำรอง	4.2
standby_max_log_to_backup	ทำหน้าที่ในการหาแฟ้มบันทึกถาวรตัวสุดท้ายเพื่อใช้เป็นตัวเปรียบเทียบในการทำการสำรองข้อมูลบนฐานข้อมูลสำรอง	3.7
standby_switchlog	ทำหน้าที่ในการสั่งให้ฐานข้อมูลหลักทำการสลับการใช้งานแฟ้มบันทึกทำซ้ำไปที่ตัวใหม่	3.6
startup_mount_standby_sqlplus	ทำหน้าที่ในการให้ฐานข้อมูลสำรองเริ่มต้นทำงาน และให้อยู่ในสถานะรอการปรับปรุงข้อมูลจากแฟ้มบันทึกถาวร	3.2

ตารางที่ 4.2 ชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลสำรอง (ต่อ)

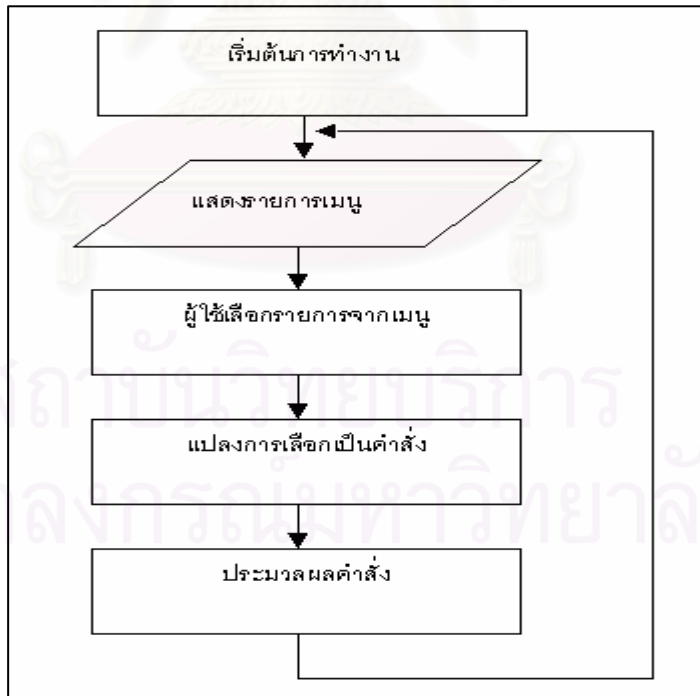
ชื่อฟังก์ชัน	หน้าที่	เรียกจากเมนูรายการ
startup_readonly_sqlplus	ทำหน้าที่ในการให้ฐานข้อมูลสำรองเริ่มต้นทำงานและให้อยู่ในสถานะอ่านอย่างเดียว	3.2

#### 4.3 การทำงานของชุดโปรแกรม

การทำงานของชุดโปรแกรมในส่วนนี้จะกล่าวถึงชุดโปรแกรมบางส่วนที่สำคัญที่พัฒนาขึ้น โดยจะมีการแสดงขั้นตอนการทำงานดังนี้

##### 4.3.1 โปรแกรมรายการเมนู (Main\_Menu\_STDBY.sh)

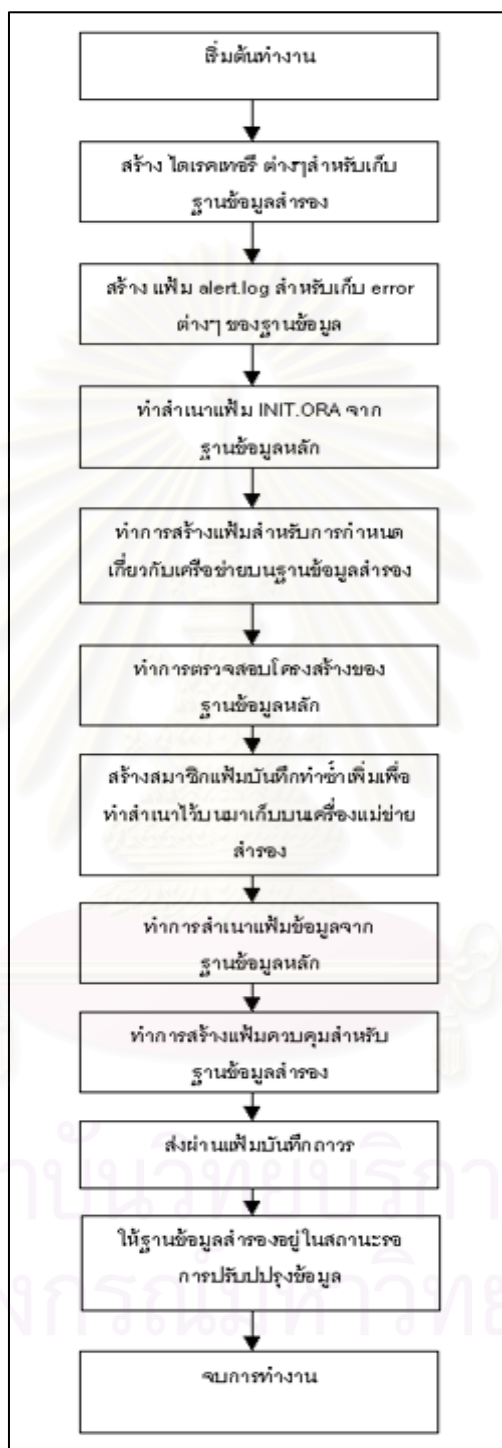
ทำหน้าที่ในการแสดงรายการต่างๆ ให้ผู้ใช้เลือกทำรายการหลังจากนั้นก็ไปเรียกโปรแกรมในส่วนอื่นทำงาน ในการพัฒนาในส่วนของรายการเมนูต่างๆ จะใช้หลักการที่คล้ายคลึงกัน ดังนั้นจึงเลือกแสดงเฉพาะรายการเมนูหลัก ดังแสดงขั้นตอนการทำงานดังรูปที่ 4.1



รูปที่ 4.1 ขั้นตอนการทำงานรายการเมนู

##### 4.3.2 โปรแกรมสร้างฐานข้อมูลสำรอง (CREATE\_STANDBY\_DB.sh)

ทำหน้าที่ในการสร้างฐานข้อมูลสำรองรวมถึงการสร้างเกี่ยวกับเครื่องข่ายการให้บริการบนฐานข้อมูลสำรอง ดังแสดงขั้นตอนการทำงานดังรูปที่ 4.2



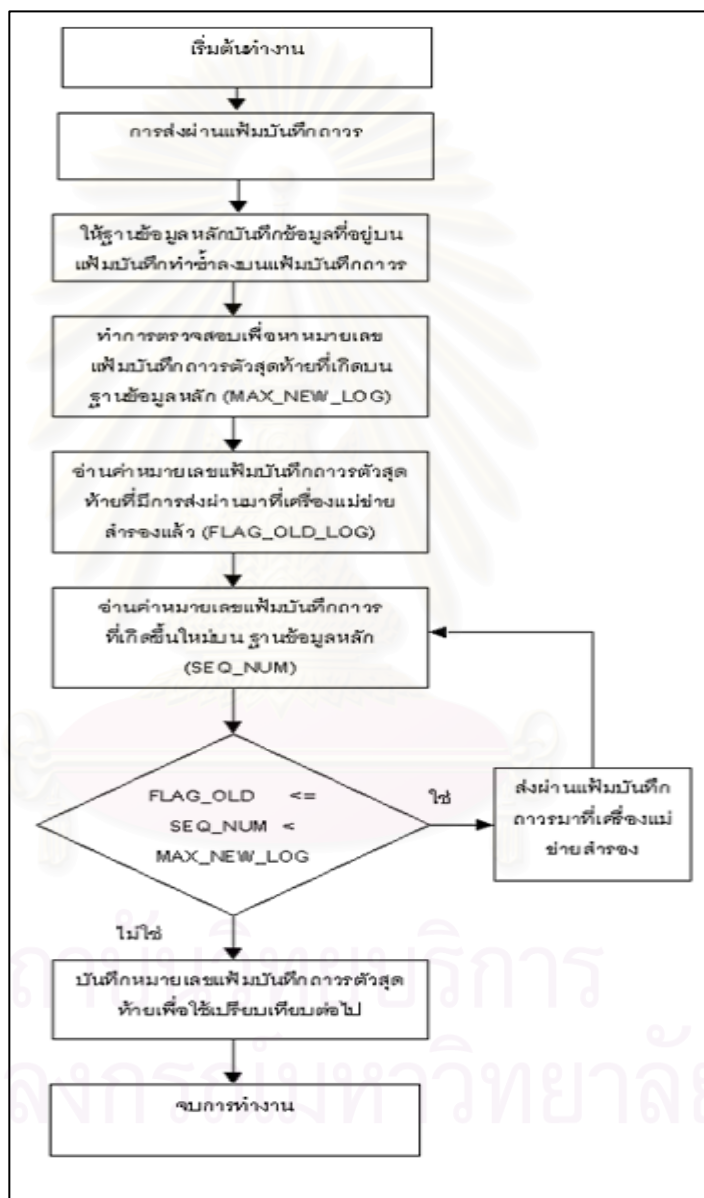
รูปที่ 4.2 ขั้นตอนการสร้างฐานข้อมูลสำรอง

#### 4.3.3 โปรแกรมที่มีการตั้งเวลาทำงานอัตโนมัติ



การพัฒนาในส่วนนี้จะมีโปรแกรมทำงานอยู่ 2 โปรแกรมคือ transfer\_archived\_control.sh และ find\_physical\_structure.sh ซึ่งจะมีหน้าที่ ดังนี้

4.3.3.1 โปรแกรม transfer\_archived\_control.sh จะทำหน้าที่ในการส่งผ่านแฟ้มบันทึกถาวรจากเครื่องแม่ข่ายหลักมาเครื่องแม่ข่ายสำรองเพื่อรอการปรับปรุงข้อมูลข้อมูลบนฐานข้อมูลสำรองต่อไป ดังแสดงขั้นตอนการทำงานดังรูปที่ 4.3



รูปที่ 4.3 ขั้นตอนการส่งผ่านแฟ้มบันทึกถาวร

4.3.3.2 โปรแกรม find\_physical\_structure.sh จะทำหน้าที่ในการตรวจสอบ

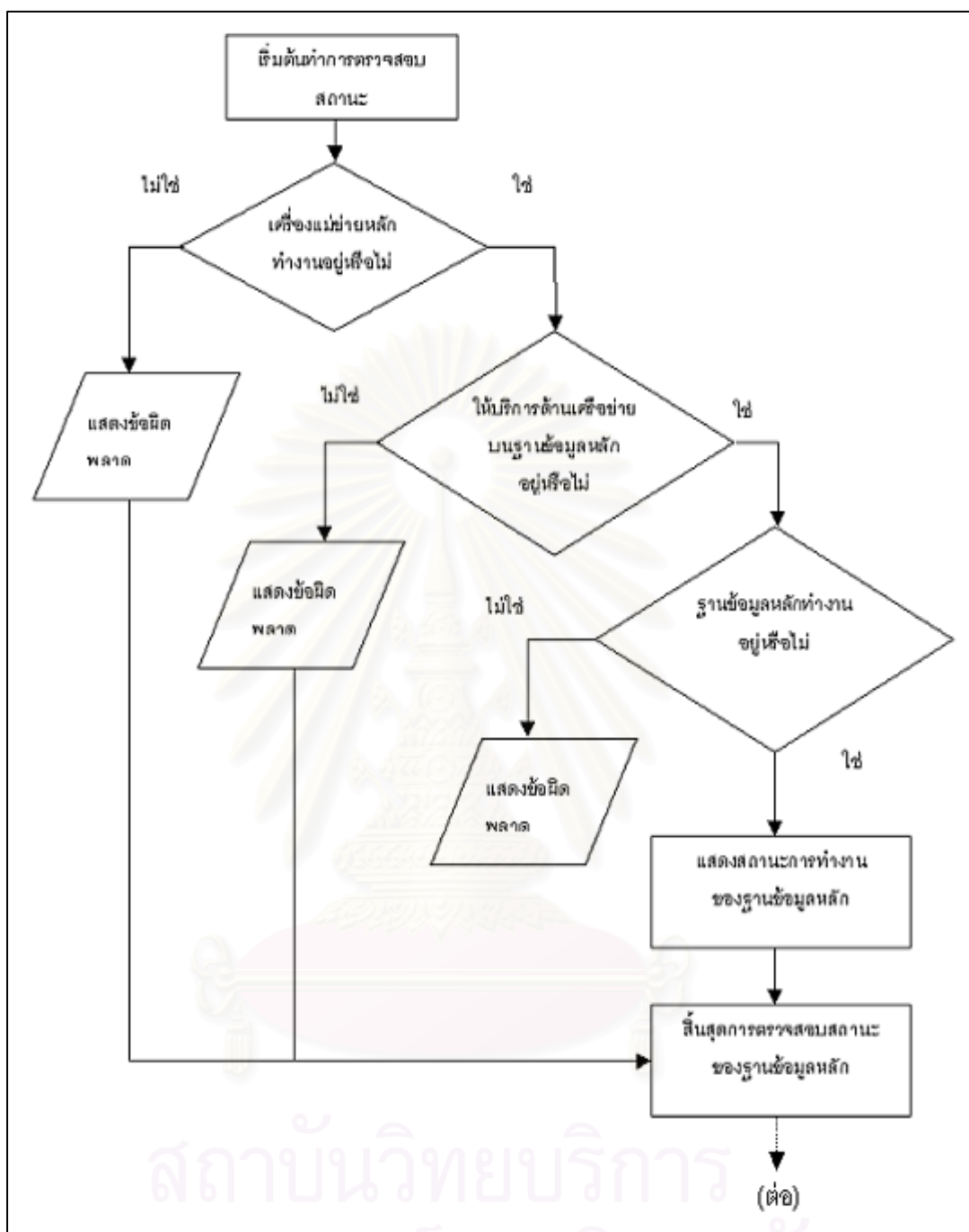
การเพิ่มและลบแฟ้มข้อมูลบนฐานข้อมูลหลัก และจะทำหน้าที่ในการปรับปรุงข้อมูลบนฐานข้อมูลสำรองจากแฟ้มบันทึกถาวร ดังแสดงขั้นตอนการทำงานดังรูปที่ 4.4

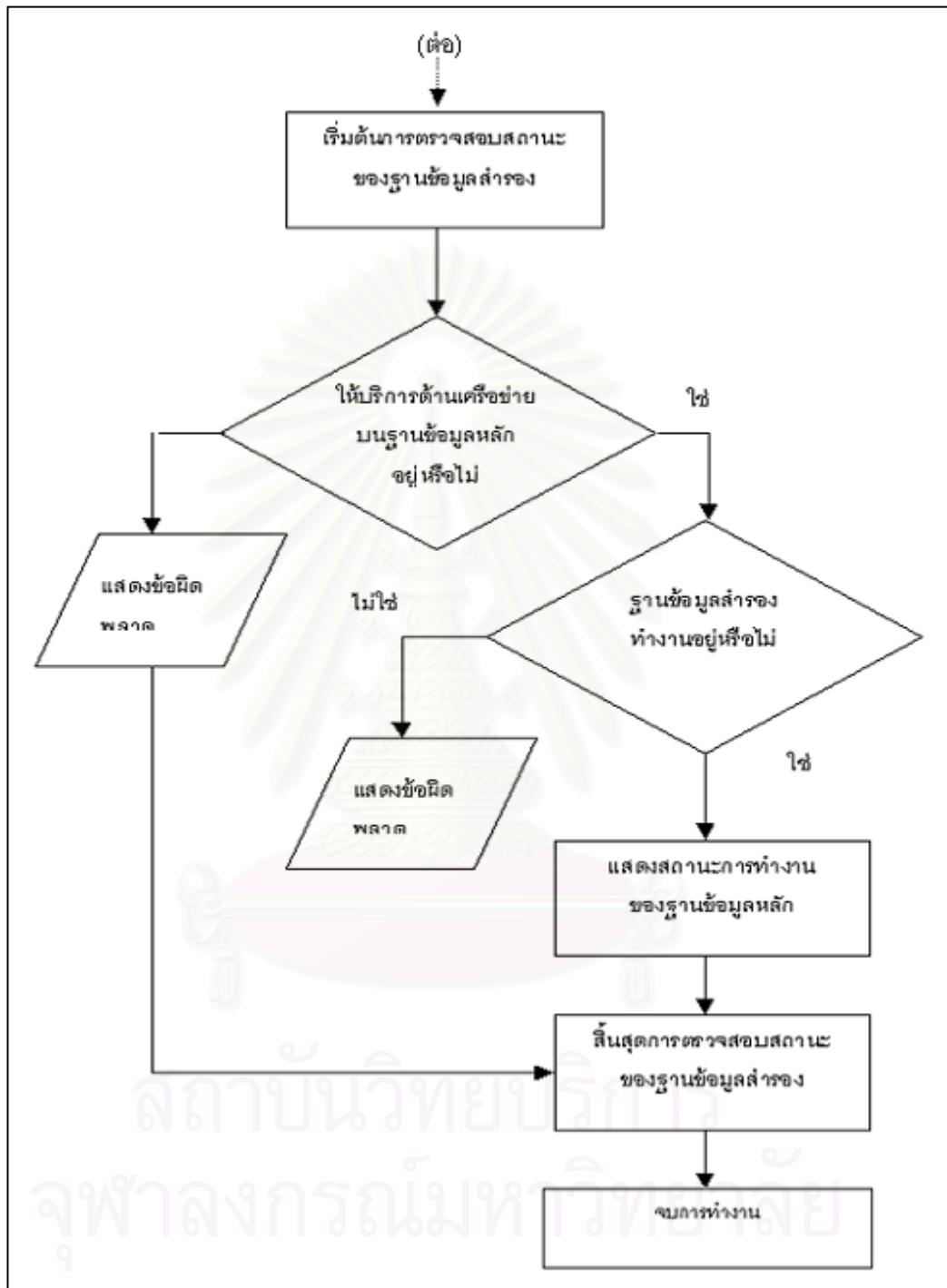


รูปที่ 4.4 ขั้นตอนการเปลี่ยนแปลงโครงสร้างฐานข้อมูลสำรอง

โปรแกรม `transfer_archived_control.sh` และ โปรแกรม `find_physical_structure.sh` ทั้งโปรแกรมจะถูกเรียกให้ทำงานอัตโนมัติ จากโปรแกรมอรรถประโยชน์ของระบบปฏิบัติการคอร์นแทป ดังที่กล่าวมาแล้วในบทที่ 3

4.3.4 โปรแกรมที่ใช้ในการตรวจสอบสถานะของฐานข้อมูลหลักและฐานข้อมูลสำรอง ( `online_monitoring.sh` ) เพื่อใช้ตรวจสอบการทำงานของฐานข้อมูลทั้งสองระบบมีขั้นตอนการทำงานดังรูปที่ 4.5





รูปที่ 4.5 ขั้นตอนการตรวจสอบสถานะของฐานข้อมูลทั้งสองระบบ

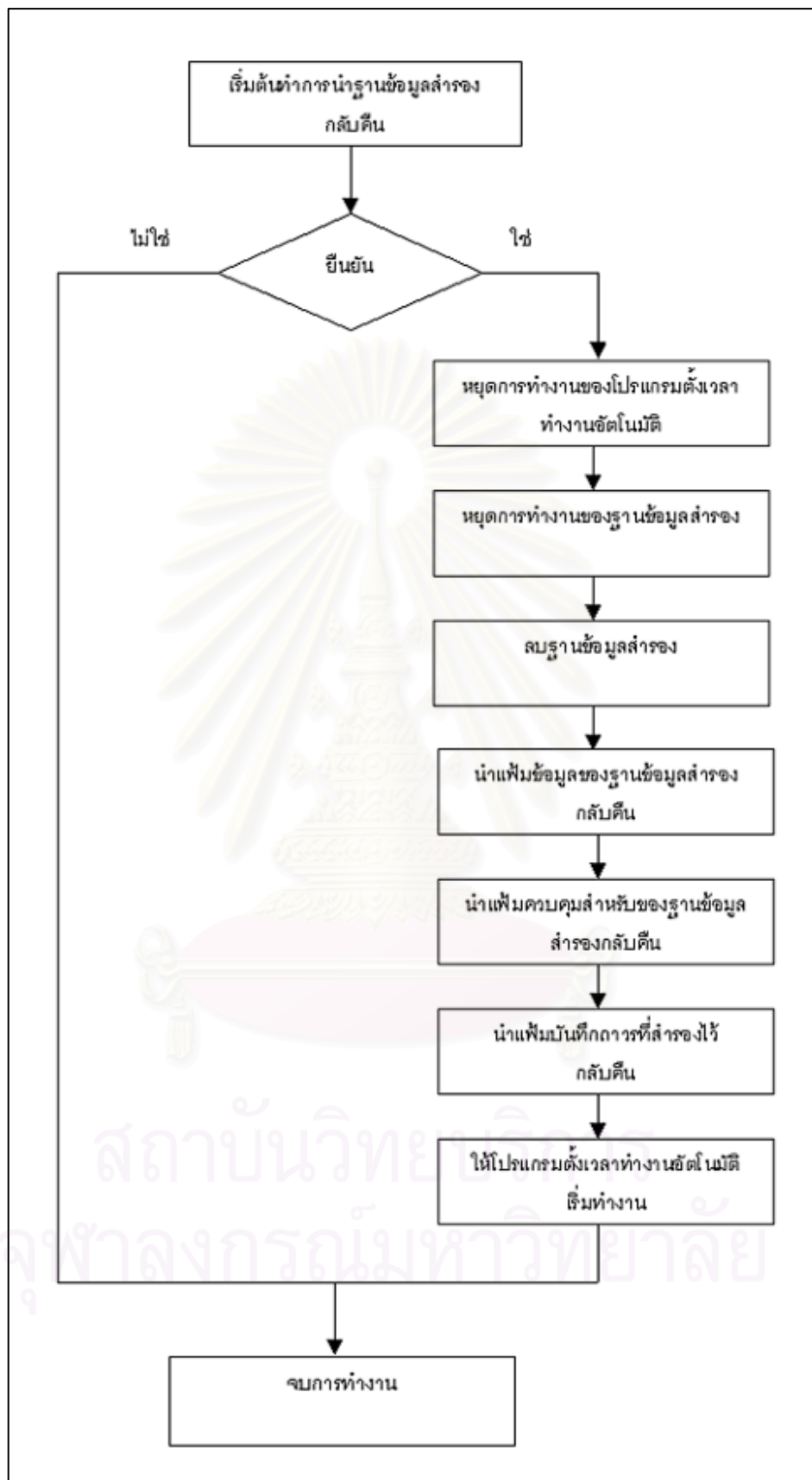
4.3.5 โปรแกรมที่ใช้ในการทำสำรองและนำข้อมูลกลับคืน มีดังนี้

4.3.5.1 โปรแกรมที่ใช้ในการทำสำรองข้อมูล (Backup\_standby.sh) มีขั้นตอนการทำงานดังรูปที่ 4.6

4.3.5.2 โปรแกรมที่ใช้ในการกู้คืนฐานข้อมูลสำรอง (Gen\_restore\_standby.sh) มีขั้นตอนการทำงานดังรูปที่ 4.7



รูปที่ 4.6 ขั้นตอนสำรองข้อมูล



รูปที่ 4.7 ขั้นตอนการกู้คืนฐานข้อมูล

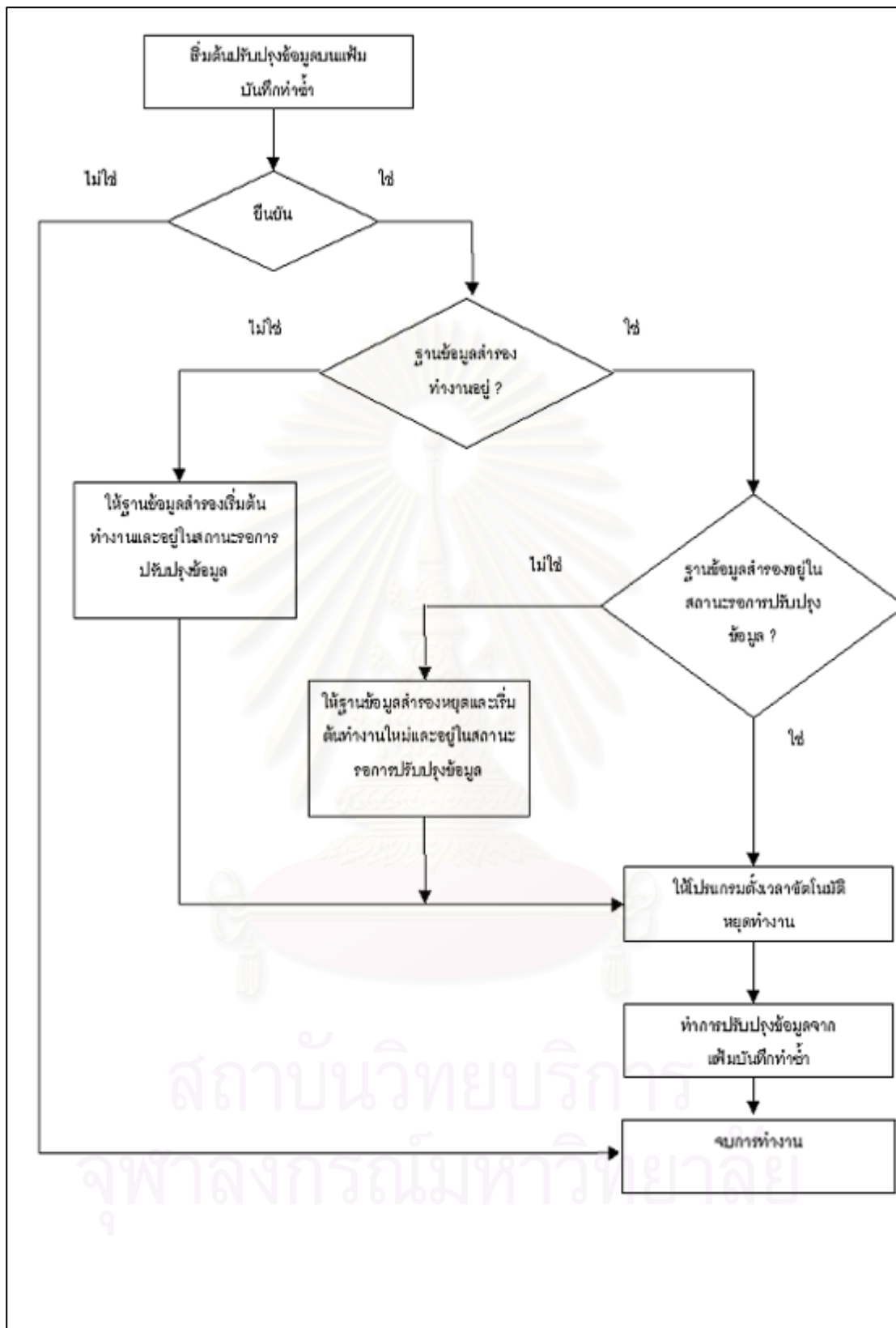
4.3.6 โปรแกรมที่ใช้ในการเปลี่ยนฐานข้อมูลสำรองให้อยู่ในสถานะพร้อมใช้เพื่อทำงานแทนฐานข้อมูลหลัก

4.3.6.1 `apply_remote_redolog.sh` ทำหน้าที่ในการปรับปรุงข้อมูลจากแฟ้มลงบันทึกทำซ้ำที่อยู่บนเครื่องแม่ข่ายสำรอง เพื่อให้ประวัติรายการเปลี่ยนแปลงล่าสุดไม่สูญหาย มีขั้นตอนการทำงานดังในรูปที่ 4.8

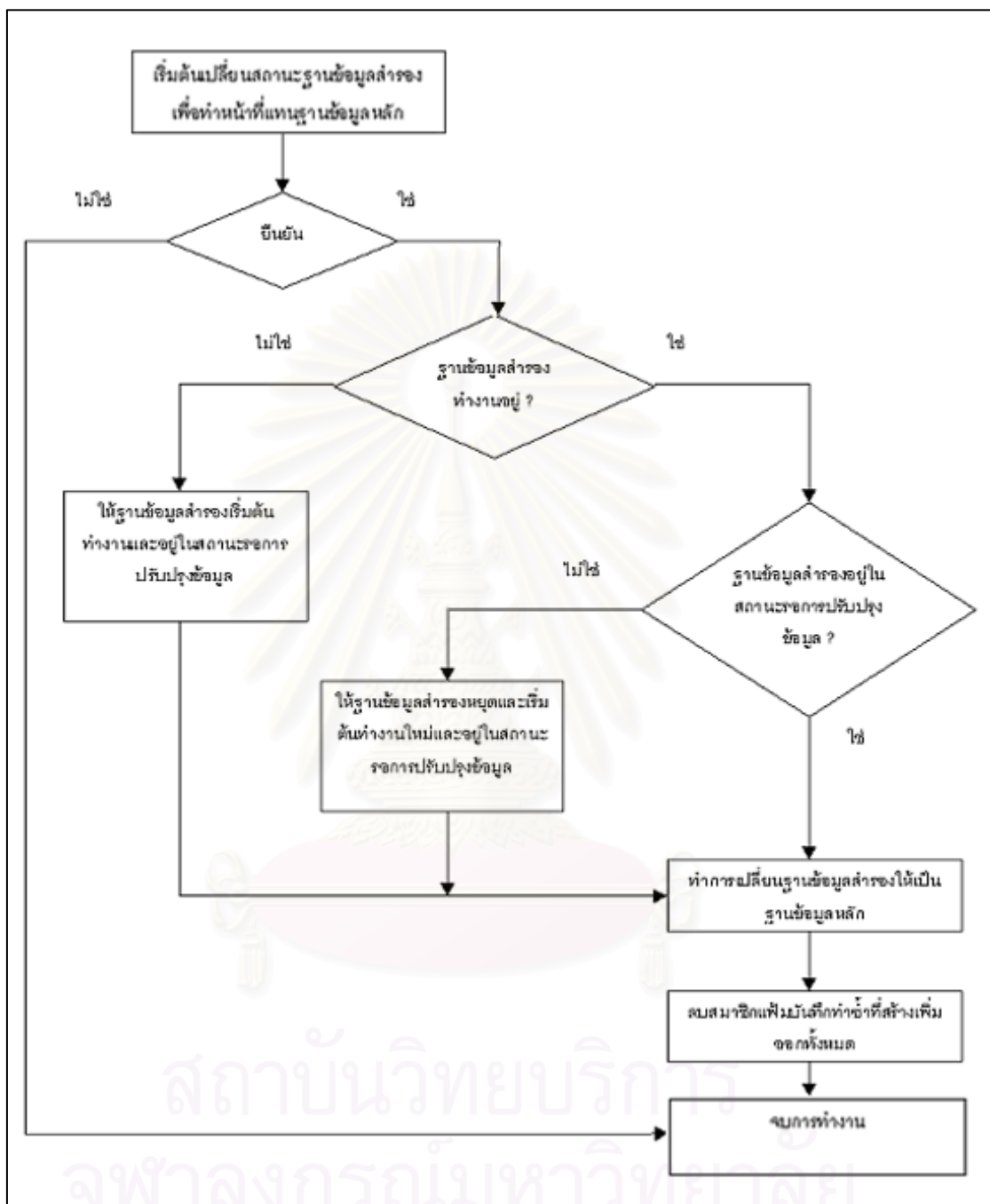
4.3.6.2 `find_physical_structure.sh` ทำการเรียกโปรแกรมนี้เองอีกครั้งหลังจากที่เราเรียกโปรแกรม `apply_remote_redolog.sh` ไปแล้ว เพื่อเป็นการป้องกันในกรณีที่มีเพิ่มหรือลบแฟ้มข้อมูล ที่ฐานข้อมูลแต่ที่ฐานข้อมูลหลัก แต่ประวัติการเปลี่ยนแปลงโครงสร้างยังอยู่บนแฟ้มบันทึกทำซ้ำ

4.3.6.3 `Activate_remote_redo_STDBY.sh` ทำหน้าที่ในการเปลี่ยนฐานข้อมูลสำรองให้อยู่ในสถานะเขียนอ่าน และพร้อมใช้งานเพื่อทำงานแทนฐานข้อมูลหลัก มีขั้นตอนการทำงานดังในรูปที่ 4.9





รูปที่ 4.8 ขั้นตอนการปรับปรุงข้อมูลบนแฟ้มบันทึกทำซ้ำ



รูปที่ 4.9 ขั้นตอนการเปลี่ยนสถานะจากฐานข้อมูลสำรองเป็นฐานข้อมูลหลัก

## บทที่ 5

### การทดสอบเครื่องมือ

ผู้วิจัยได้ทำการทดสอบการทำงานของเครื่องที่พัฒนาขึ้น โดยมีข้อกำหนดของเครื่องแม่ข่ายที่ใช้ในการทดสอบดังตารางที่ 5.1 ดังนี้

ตารางที่ 5.1 ข้อกำหนดของเครื่องแม่ข่ายที่ใช้ในการทดสอบ

เครื่องแม่ข่ายหลัก	เครื่องแม่ข่ายสำรอง
1. เครื่อง Sun Sparc20 - 2 CPU x 176 MHz - หน่วยความจำหลัก(memory) 512 เมกะไบต์ - หน่วยความจำสำรอง (Harddisk) 4000 เมกะไบต์	1. เครื่อง Sun Sparc20 - 1 CPU x 176 MHz - หน่วยความจำหลัก(memory) 512 เมกะไบต์ - หน่วยความจำสำรอง (Harddisk) 4000 เมกะไบต์
2. ระบบปฏิบัติการโซลาริส เวอร์ชัน 2.7	2. ระบบปฏิบัติการโซลาริส เวอร์ชัน 2.7
3. ระบบจัดการฐานข้อมูลออรากเคิล เวอร์ชัน 8.1.7	3. ระบบจัดการฐานข้อมูลออรากเคิล เวอร์ชัน 8.1.7

ในการทดสอบการทำงานของเครื่องแบ่งเป็น 3 ประเภทดังตารางที่ 5.2 ดังนี้

ตารางที่ 5.2 ประเภทของการทดสอบเครื่องมือ

ประเภทการทดสอบ	ผลการทดสอบ	เวลาที่ใช้ (นาที)
1. การส่งผ่านเพิ่มบันทึกถาวร และการปรับปรุงข้อมูลบนฐานข้อมูลสำรอง	ข้อมูลที่มีการเปลี่ยนแปลงบนฐานข้อมูลหลักสามารถส่งผ่านและปรับปรุงข้อมูลบนฐานข้อมูลได้อย่างถูกต้อง ( ขั้นตอนการทดสอบอยู่ในหัวข้อที่ 5.1 )	5
2. การทดสอบการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลหลัก	ฐานข้อมูลสำรองสามารถเปลี่ยนแปลงโครงสร้างทางกายภาพให้เหมือนฐานข้อมูลหลักได้ ( ขั้นตอนการทดสอบอยู่ในหัวข้อที่ 5.2 )	5

ตารางที่ 5.2 ประเภทของการทดสอบเครื่องมือ(ต่อ)

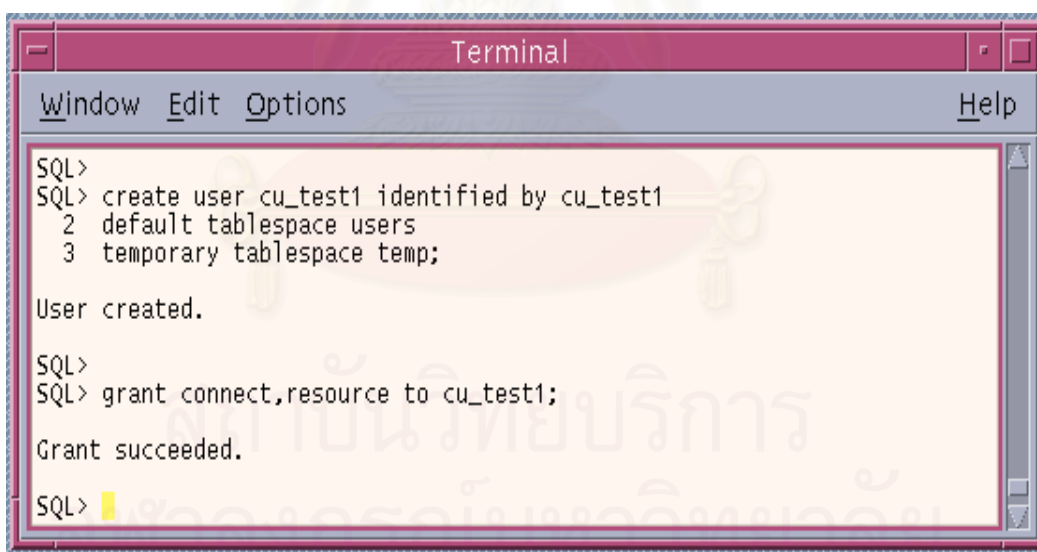
ประเภทการทดสอบ	ผลการทดสอบ	เวลาที่ใช้ (นาที)
3. การเปลี่ยนแปลงสถานะการทำงานของฐานข้อมูลสำรองไปเป็นฐานข้อมูลหลัก	สามารถเปลี่ยนสถานะการทำงานของฐานข้อมูลสำรองเพื่อทำงานทดแทนฐานข้อมูลหลักได้รวดเร็วและถูกต้อง ( ขั้นตอนการทดสอบอยู่ในหัวข้อที่ 5.3 )	3

### ขั้นตอนของการทดสอบในทดสอบในแต่ละประเภทมีดังนี้

#### 5.1 การส่งผ่านแฟ้มบันทึกถาวร และการปรับปรุงข้อมูลบนฐานข้อมูลสำรอง

ในการทดสอบการส่งผ่านแฟ้มบันทึกถาวร และการปรับปรุงข้อมูลบนฐานข้อมูลสำรอง โดยการจำลองการทำงานโดยทำให้มีรายการเปลี่ยนแปลงเกิดขึ้นกับฐานข้อมูลหลักหลังจากนั้นทำการตรวจสอบความถูกต้องบนฐานข้อมูลสำรอง โดยมีขั้นตอนการทดสอบดังนี้

5.1.1 ทำการสร้างผู้ใช้ใหม่ บนฐานข้อมูลหลักใช้ชื่อว่า cu\_test1 และมีรหัสผ่านเป็น cu\_test1 แล้วทำการกำหนดสิทธิที่จำเป็นต่างๆ ให้ผู้ใช้ ตามรูปที่ 5.1



```

Terminal
Window Edit Options Help
SQL>
SQL> create user cu_test1 identified by cu_test1
  2 default tablespace users
  3 temporary tablespace temp;

User created.

SQL>
SQL> grant connect,resource to cu_test1;

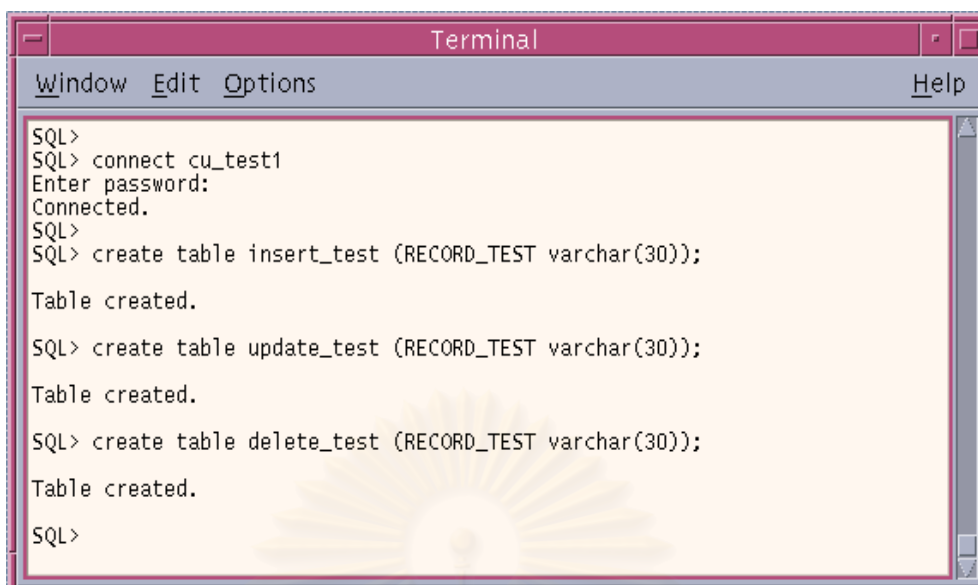
Grant succeeded.

SQL>

```

รูปที่ 5.1 แสดงการสร้างผู้ใช้ใหม่และทำการกำหนดสิทธิ

5.1.2 ทำการสร้างตารางขึ้นมาใหม่ 3 ตารางเพื่อรองรับรายการเปลี่ยนแปลงที่จำลองขึ้น ซึ่งวิธีการสร้างและชื่อตารางที่จะแสดงในรูปที่ 5.2



```

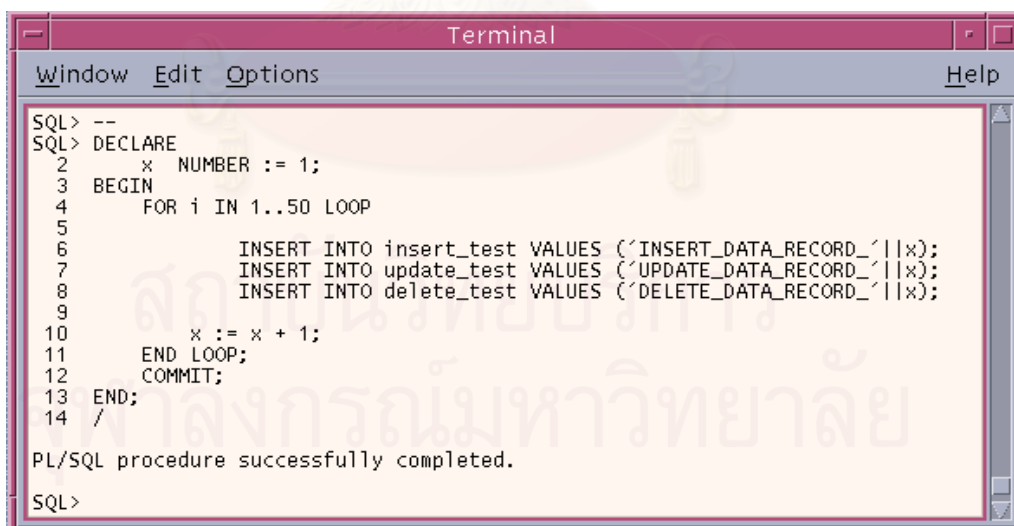
Terminal
Window Edit Options Help
SQL>
SQL> connect cu_test1
Enter password:
Connected.
SQL>
SQL> create table insert_test (RECORD_TEST varchar(30));
Table created.
SQL> create table update_test (RECORD_TEST varchar(30));
Table created.
SQL> create table delete_test (RECORD_TEST varchar(30));
Table created.
SQL>

```

รูปที่ 5.2 แสดงการสร้างตารางเพื่อรองรับการเปลี่ยนแปลงที่จำลองขึ้น

สร้างตารางชื่อ insert\_test1 เพื่อรองรับการเพิ่มข้อมูล , สร้างตารางชื่อ update\_test1 เพื่อรองรับการปรับปรุงข้อมูล, สร้างตารางชื่อ delete\_test1 เพื่อรองรับการลบข้อมูล

5.1.3 ทำการจำลองการทำงานให้มีรายการเปลี่ยนแปลงเกิดขึ้น โดยการเพิ่มข้อมูลจำนวน 50 รายการในตารางทั้งสาม ดังรูปที่ 5.3 หลังจากนั้นทำการยืนยันการเปลี่ยนแปลงรายการ



```

Terminal
Window Edit Options Help
SQL> --
SQL> DECLARE
2   x NUMBER := 1;
3 BEGIN
4   FOR i IN 1..50 LOOP
5
6       INSERT INTO insert_test VALUES ('INSERT_DATA_RECORD_'||x);
7       INSERT INTO update_test VALUES ('UPDATE_DATA_RECORD_'||x);
8       INSERT INTO delete_test VALUES ('DELETE_DATA_RECORD_'||x);
9
10      x := x + 1;
11  END LOOP;
12  COMMIT;
13 END;
14 /
PL/SQL procedure successfully completed.
SQL>

```

รูปที่ 5.3 แสดงการเพิ่มข้อมูลในตารางบนฐานข้อมูลหลัก

5.1.4 ทำการทดสอบปรับปรุงข้อมูลที่อยู่ในตาราง update\_test1 และ ยืนยันการเปลี่ยนแปลงข้อมูล ดังรูป 5.4

```
Terminal
Window Edit Options Help
SQL> update update_test
 2 set record_test='NEW_UPDATE'
 3 where rownum < 10 ;

9 rows updated.

SQL> select * from update_test where rownum < 14;

NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
UPDATE_DATA_RECORD_10
UPDATE_DATA_RECORD_11
UPDATE_DATA_RECORD_12
UPDATE_DATA_RECORD_13

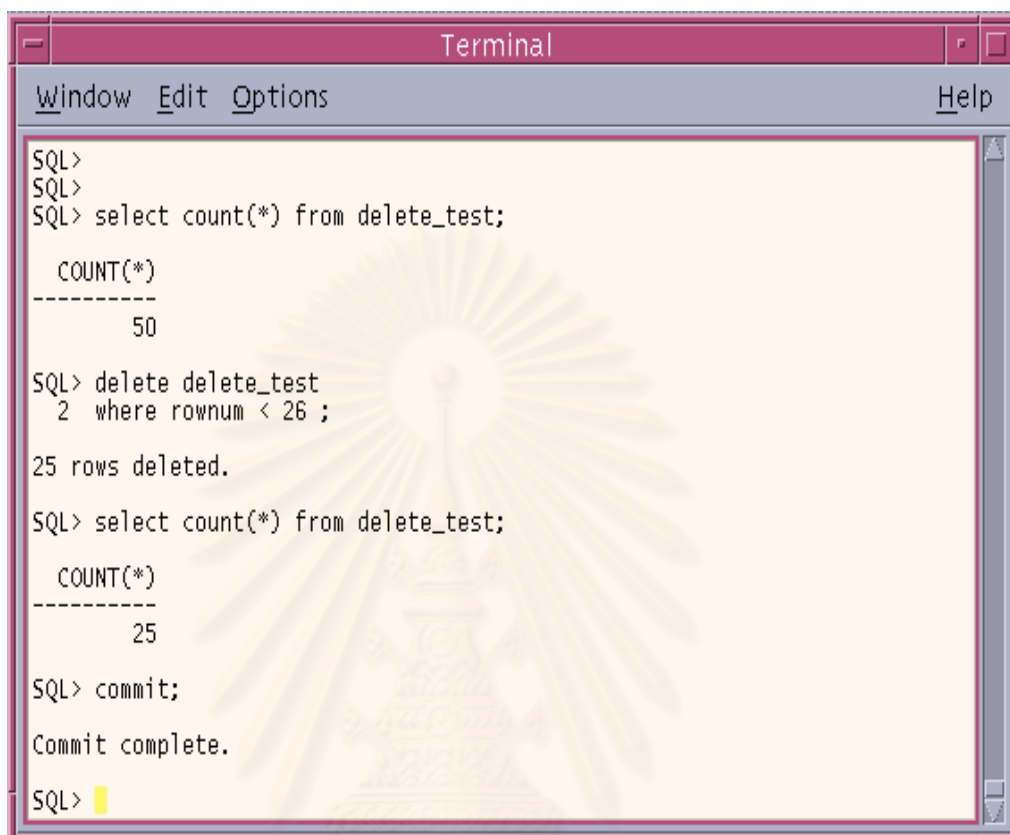
13 rows selected.

SQL>
SQL> commit;

Commit complete.
```

รูปที่ 5.4 แสดงการปรับปรุงข้อมูลในตารางบนฐานข้อมูลหลัก

5.1.5 ทำการทดสอบลบข้อมูลที่อยู่ในตาราง delete\_test1 และ ยืนยันการเปลี่ยนแปลงข้อมูล ดังรูป 5.5



```
Terminal
Window Edit Options Help
SQL>
SQL>
SQL> select count(*) from delete_test;
COUNT(*)
-----
          50
SQL> delete delete_test
2 where rownum < 26 ;
25 rows deleted.
SQL> select count(*) from delete_test;
COUNT(*)
-----
          25
SQL> commit;
Commit complete.
SQL> █
```

รูปที่ 5.5 แสดงการปรับปรุงข้อมูลในตารางบนฐานข้อมูลหลัก

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



5.1.6 หลังจากการจำลองรายการเปลี่ยนแปลงบนฐานข้อมูลหลักเสร็จเรียบร้อยแล้ว ภายใน 5 นาที ข้อมูลที่มีการเปลี่ยนแปลงจะถูกส่งไปปรับปรุงบนฐานข้อมูลสำรองโดยส่งผ่านแฟ้มบันทึกถาวรซึ่งสามารถตรวจสอบการเปลี่ยนแปลงต่างๆ นี้ได้โดยการเปิดฐานข้อมูลสำรองให้อยู่ในสถานะอ่านอย่างเดียว และทำการสอบถามข้อมูลบนตารางที่มีการเปลี่ยนแปลงดังรูปที่ 5.6

```

Terminal
Window Edit Options Help

SQL> select count(*) from insert_test;
COUNT(*)
-----
50

SQL> select count(*) from delete_test;
COUNT(*)
-----
25

SQL> select * from update_test where rownum < 14;
RECORD_TEST
-----
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
NEW_UPDATE
UPDATE_DATA_RECORD_10
UPDATE_DATA_RECORD_11

RECORD_TEST
-----
UPDATE_DATA_RECORD_12
UPDATE_DATA_RECORD_13

13 rows selected.

SQL>

```

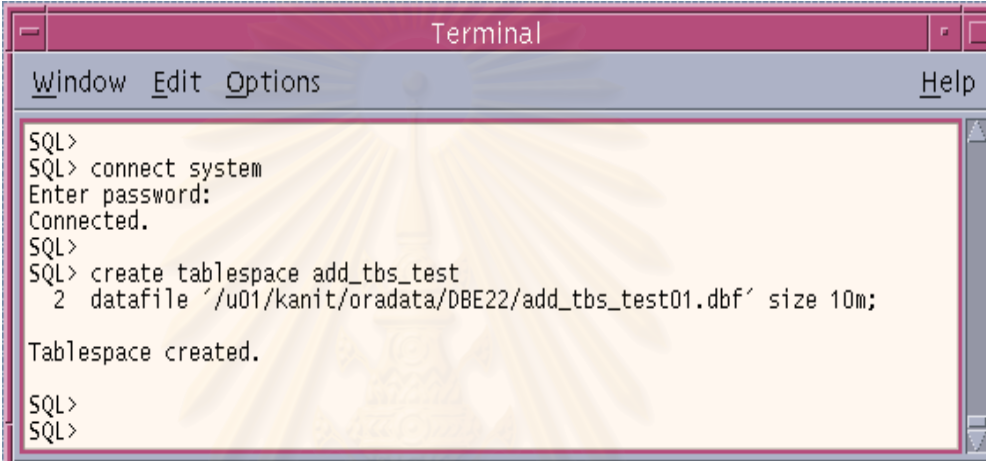
รูปที่ 5.6 แสดงการสอบถามข้อมูลในตารางบนฐานข้อมูลสำรอง

จากการตรวจสอบพบว่าข้อมูลมีการปรับปรุงบนฐานข้อมูลสำรองอย่างถูกต้อง และครบถ้วน เหมือนกับฐานข้อมูลหลัก แล้วหลังจากนั้นจึงทำการเปลี่ยนสถานะของฐานข้อมูลสำรองเป็นรอการปรับปรุงข้อมูลต่อไป

## 5.2 การทดสอบการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลหลัก

ในการทดสอบการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลหลักโดยการเพิ่มเทเบิลสเปซและตรวจสอบความถูกต้องของโครงสร้างบนฐานข้อมูลสำรอง โดยมีขั้นตอนการทดสอบดังนี้

5.2.1 ทำการเพิ่มเทเบิลสเปซบนฐานข้อมูลหลักชื่อ add\_tbs\_test โดยมีการระบุเพิ่มทางกายภาพชื่อ '/u01/kanit/oradata/DBE22/add\_tbs\_test' มีขนาด 10 เมกะไบต์ ดังรูปที่ 5.7



```

Terminal
Window Edit Options Help
SQL>
SQL> connect system
Enter password:
Connected.
SQL>
SQL> create tablespace add_tbs_test
      2 datafile '/u01/kanit/oradata/DBE22/add_tbs_test01.dbf' size 10m;

Tablespace created.

SQL>
SQL>
  
```

รูปที่ 5.7 แสดงการเพิ่มเทเบิลสเปซบนฐานข้อมูลหลัก

5.2.2 หลังจากนั้นสามารถดูการเปลี่ยนแปลงโครงสร้างที่เกิดขึ้นโดยใช้เครื่องมือในการตรวจสอบสถานะการทำงานของฐานข้อมูลหลักและฐานข้อมูลสำรอง จะแสดงดังรูปที่ 5.8 จำนวนเทเบิลสเปซ และ จำนวนแฟ้มข้อมูลของฐานข้อมูลหลักจะมีจำนวนมากกว่าจำนวนเทเบิลสเปซ และ จำนวนแฟ้มข้อมูลของฐานข้อมูลสำรอง เมื่อเวลาผ่านไป 5 นาที ฐานข้อมูลสำรองจะทำการปรับปรุงโครงสร้างให้เหมือนฐานข้อมูลหลัก ดังแสดงในรูปที่ 5.9

จุฬาลงกรณ์มหาวิทยาลัย

```

---ONLINE MONITORING---
Window Edit Options Help
<> MANAGING THE STANDBY DATABASE <>
----- ONLINE STATUS -----
DATE: 25/08/02 TIME: 11:35
-----
*** Primary Database ***
-----
Primary Listener Online
Primary Server: DB2
Database Name: DBE22
Database Mode: READ WRITE
Tablespace: # 6
Datafiles: # 6
Max History Number : 3006
-----
*** Standby Database ***
-----
Standby Listener Online
Standby Server: sserver
Database Name: DBE22
Database Mode: MOUNTED
Tablespace: # 5
Datafiles: # 5
Max History Number : 2999
-----
Archived Transfer From DB2 --> sserver : 3005
-----

```

รูปที่ 5.8 เครื่องมือในการตรวจสอบสถานะหลังจากเพิ่มจำนวนเทเบิลสเปซบนฐานข้อมูลหลัก

```

---ONLINE MONITORING---
Window Edit Options Help
<> MANAGING THE STANDBY DATABASE <>
----- ONLINE STATUS -----
DATE: 25/08/02 TIME: 11:37
-----
*** Primary Database ***
-----
Primary Listener Online
Primary Server: DB2
Database Name: DBE22
Database Mode: READ WRITE
Tablespace: # 6
Datafiles: # 6
Max History Number : 3006
-----
*** Standby Database ***
-----
Standby Listener Online
Standby Server: sserver
Database Name: DBE22
Database Mode: MOUNTED
Tablespace: # 6
Datafiles: # 6
Max History Number : 3004
-----
Archived Transfer From DB2 --> sserver : 3005
-----

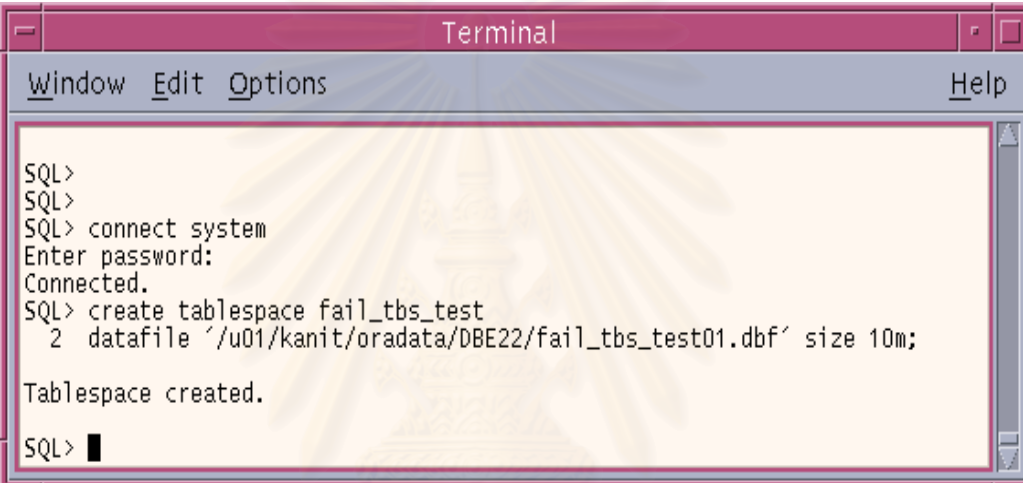
```

รูปที่ 5.9 เครื่องมือในการตรวจสอบสถานะหลังปรับปรุงโครงสร้างบนฐานข้อมูลสำรองแล้ว

### 5.3 การเปลี่ยนแปลงสถานะการทำงานของฐานข้อมูลสำรองไปเป็นฐานข้อมูลหลัก

ในการทดสอบการเปลี่ยนแปลงสถานะการทำงานของฐานข้อมูลสำรองไปเป็นฐานข้อมูลหลัก โดยการปรับปรุงข้อมูลจากแฟ้มบันทึกทำซ้ำบนเครื่องแม่ข่ายสำรอง ในกรณีที่ฐานข้อมูลหลักไม่สามารถทำงานได้มีขั้นตอนการทดสอบดังนี้

5.3.1 ทำการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลหลักโดยการเพิ่มเทเบิลสเปซชื่อว่า fail\_tbs\_test โดยมีการระบุเพิ่มทางกายภาพชื่อ '/u01/kanit/oradata/DBE22/fail\_tbs\_test' มีขนาด 10 เมกะไบต์ ดังรูปที่ 5.10



```

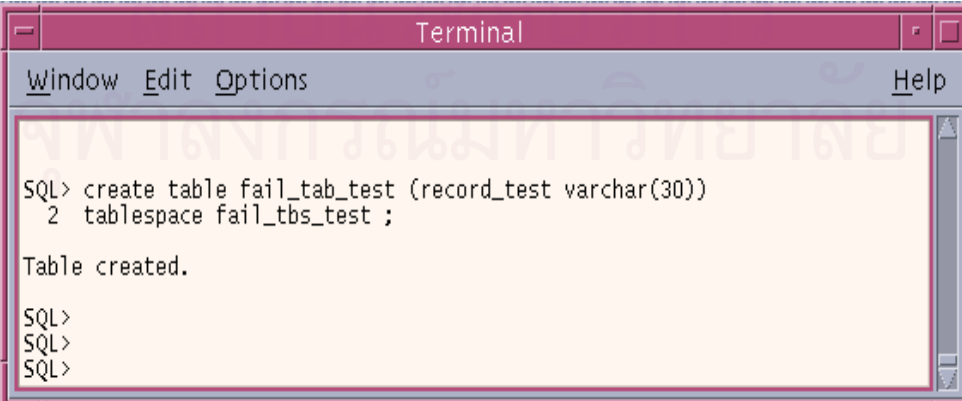
Terminal
Window Edit Options Help
SQL>
SQL>
SQL> connect system
Enter password:
Connected.
SQL> create tablespace fail_tbs_test
  2 datafile '/u01/kanit/oradata/DBE22/fail_tbs_test01.dbf' size 10m;

Tablespace created.

SQL> █
  
```

รูปที่ 5.10 แสดงการเพิ่มเทเบิลสเปซข้อมูลในตารางบนฐานข้อมูลหลัก

5.3.2 ทำการสร้างเทเบิล ชื่อ fail\_tab\_test บนเทเบิลสเปซ fail\_tbs\_test แสดงดังรูปที่ 5.10 และทำการเพิ่มข้อมูลเข้าไป 50 รายการ จากนั้นยืนยันการเปลี่ยนแปลง ดังรูปที่ 5.11



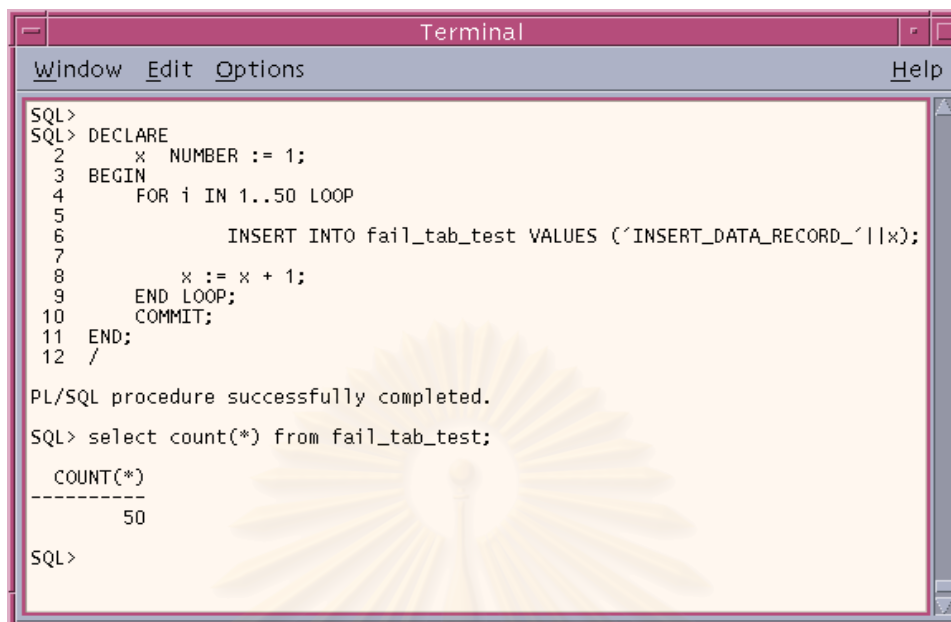
```

Terminal
Window Edit Options Help
SQL> create table fail_tab_test (record_test varchar(30))
  2 tablespace fail_tbs_test ;

Table created.

SQL>
SQL>
SQL>
  
```

รูปที่ 5.11 แสดงการสร้างเทเบิลบนเทเบิลสเปซใหม่ที่สร้างขึ้น



```

Terminal
Window Edit Options Help
SQL>
SQL> DECLARE
2  x NUMBER := 1;
3  BEGIN
4    FOR i IN 1..50 LOOP
5
6        INSERT INTO fail_tab_test VALUES ('INSERT_DATA_RECORD_'||x);
7
8        x := x + 1;
9    END LOOP;
10   COMMIT;
11 END;
12 /

PL/SQL procedure successfully completed.
SQL> select count(*) from fail_tab_test;

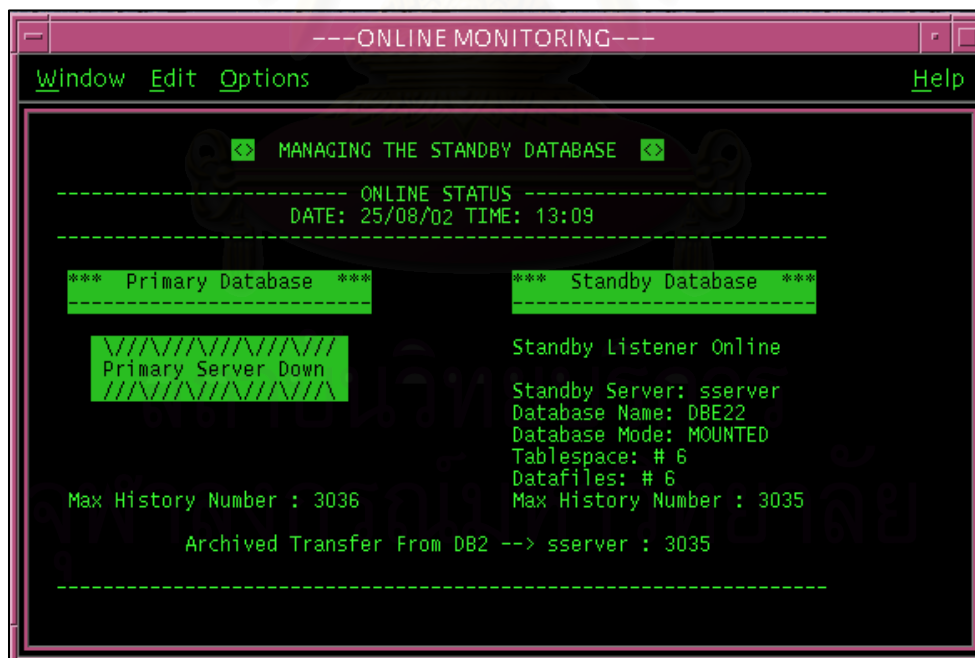
  COUNT(*)
-----
         50

SQL>

```

รูปที่ 5.12 แสดงการเพิ่มข้อมูลในตารางบนฐานข้อมูลหลัก

5.3.3 ทำการจำลองเหตุการณ์ร้ายแรง โดยการปิดไฟ (Power Off) บนเครื่องแม่ข่ายหลัก จะพบว่าเครื่องมือตรวจสอบสถานะจะแจ้งว่าฐานข้อมูลหลักไม่สามารถทำงานได้ปกติดังรูปที่ 5.13



```

---ONLINE MONITORING---
Window Edit Options Help

<> MANAGING THE STANDBY DATABASE <>

----- ONLINE STATUS -----
DATE: 25/08/02 TIME: 13:09
-----

*** Primary Database ***          *** Standby Database ***

////////////////
Primary Server Down
////////////////

Max History Number : 3036

Archived Transfer From DB2 --> sserver : 3035

Standby Listener Online
Standby Server: sserver
Database Name: DBE22
Database Mode: MOUNTED
Tablespace: # 6
Datafiles: # 6
Max History Number : 3035

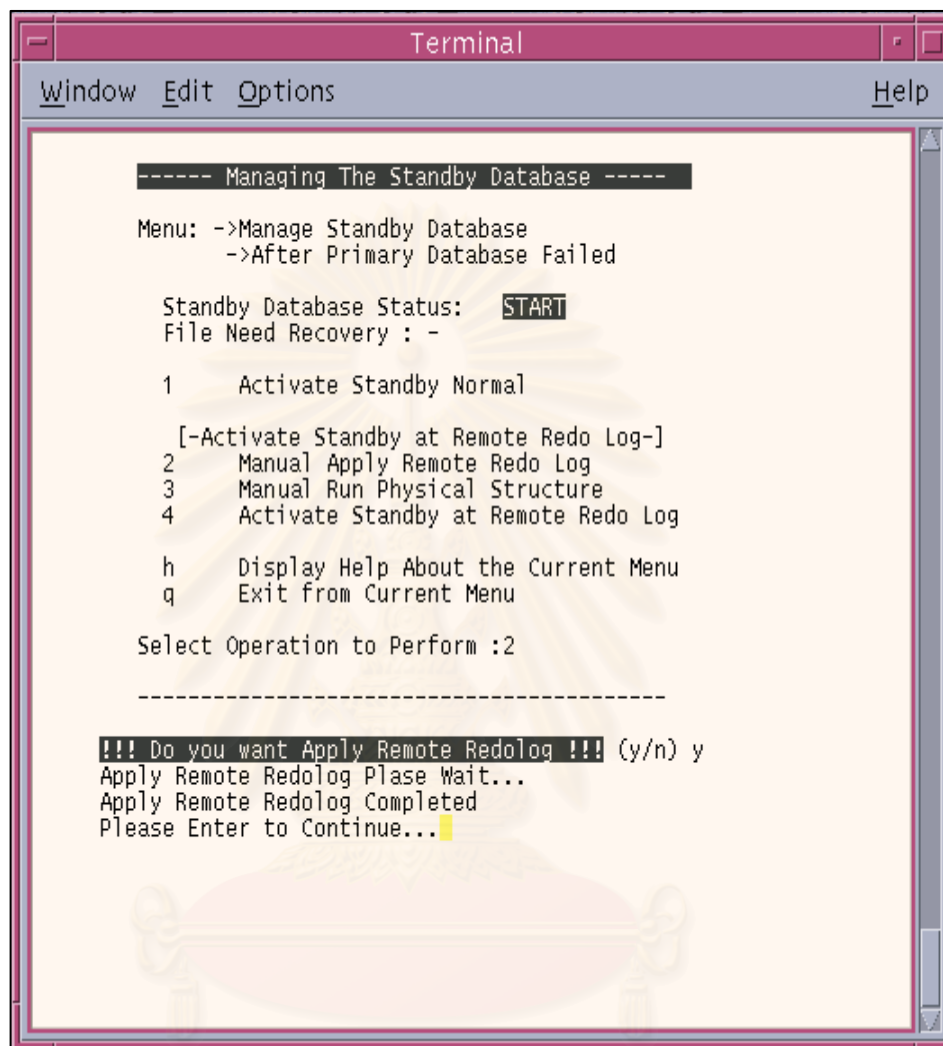
```

รูปที่ 5.13 แสดงสถานะในกรณีที่ฐานข้อมูลหลักไม่สามารถใช้งานได้ตามปกติ

5.3.4 ทำการเปลี่ยนสถานะจากฐานข้อมูลสำรองไปเป็นฐานข้อมูลหลักมีขั้นตอนดังนี้

5.3.4.1 ทำการปรับปรุงข้อมูลจากแฟ้มบันทึกทำซ้ำโดยเลือกเมนูรายการ ดังรูปที่

5.14



```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----
Menu: ->Manage Standby Database
      ->After Primary Database Failed

Standby Database Status:  START
File Need Recovery : -

1   Activate Standby Normal

[-Activate Standby at Remote Redo Log-]
2   Manual Apply Remote Redo Log
3   Manual Run Physical Structure
4   Activate Standby at Remote Redo Log

h   Display Help About the Current Menu
q   Exit from Current Menu

Select Operation to Perform :2

-----

!!! Do you want Apply Remote Redolog !!! (y/n) y
Apply Remote Redolog Plase Wait...
Apply Remote Redolog Completed
Please Enter to Continue...
  
```

รูปที่ 5.14 แสดงการปรับปรุงข้อมูลจากแฟ้มบันทึกทำซ้ำ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

5.3.4.2 หลังจากนั้นจะพบว่ามีการเพิ่มข้อมูลที่ต้องการการกู้คืนจึงเลือกทำรายการปรับปรุงโครงสร้างทางกายภาพของฐานข้อมูลสำรอง ดังรูปที่ 5.15

```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----
Menu: ->Manage Standby Database
      ->After Primary Database Failed

Standby Database Status: START
File Recover: '/u01/kanit/oradata/DBE22/fail_tbs_test01.dbf'

1   Activate Standby Normal

[-Activate Standby at Remote Redo Log-]
2   Manual Apply Remote Redo Log
3   Manual Run Physical Structure
4   Activate Standby at Remote Redo Log

h   Display Help About the Current Menu
q   Exit from Current Menu

Select Operation to Perform :3

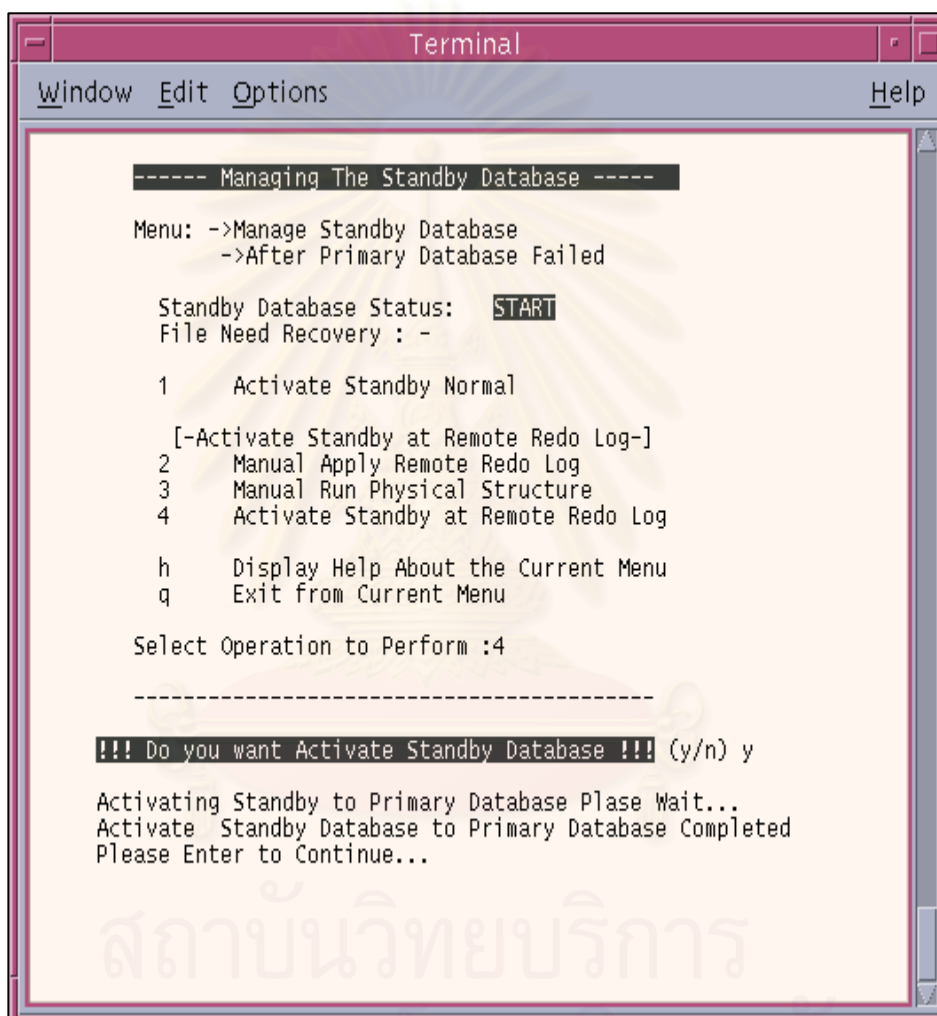
-----

Run Verify the Standby Physical Structure..

```

รูปที่ 5.15 แสดงการปรับปรุงโครงสร้างทางกายภาพ

5.3.4.3 ทำการเปลี่ยนสถานะฐานข้อมูลสำรองเป็นฐานข้อมูลหลักเพื่อให้บริการแทนฐานข้อมูลหลักโดยเลือกรายการดังรูปที่ 5.16 ซึ่งขั้นตอนในการเปลี่ยนสถานะจะใช้เวลาประมาณ 3 นาที และหลังจากนั้นจะพบว่าสถานะของฐานข้อมูลสำรองเปลี่ยนเป็นอ่านและเขียน ดังรูปที่ 5.17 หลังจากนั้นทำการตรวจสอบข้อมูลโดยการสอบถามไปที่ฐานข้อมูลสำรองดังรูปที่ 5.18 ปกติการปรับปรุบบนฐานข้อมูลสำรองอย่างถูกต้อง และครบถ้วน เหมือนกับฐานข้อมูลหลักก่อนเกิดเหตุการณ์ร้ายแรง



```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----

Menu: ->Manage Standby Database
      ->After Primary Database Failed

Standby Database Status:  START
File Need Recovery : -

1  Activate Standby Normal

[-Activate Standby at Remote Redo Log-]
2  Manual Apply Remote Redo Log
3  Manual Run Physical Structure
4  Activate Standby at Remote Redo Log

h  Display Help About the Current Menu
q  Exit from Current Menu

Select Operation to Perform :4

-----

!!! Do you want Activate Standby Database !!! (y/n) y

Activating Standby to Primary Database Plase Wait...
Activate Standby Database to Primary Database Completed
Please Enter to Continue...

```

รูปที่ 5.16 แสดงการเปลี่ยนสถานะฐานข้อมูลสำรองเป็นฐานข้อมูลหลัก



```

---ONLINE MONITORING---
Window Edit Options Help

<> MANAGING THE STANDBY DATABASE <>

----- ONLINE STATUS -----
DATE: 25/08/20 TIME: 13:19
-----

*** Primary Database ***          *** Standby Database ***

////////////////
Primary Server Down
////////////////

Max History Number : 3036

Archived Transfer From DB2 --> sserver : 3035

Standby Listener Online
Standby Server: sserver
Database Name: DBE22
Database Mode: READ WRITE
Tablespace: # 7
Datafiles: # 7
Max History Number : 2

```

รูปที่ 5.17 แสดงสถานะของฐานข้อมูลสำรองที่สามารถทำงานแทนฐานข้อมูลหลัก

```

Terminal
Window Edit Options Help

SQL> connect system
Enter password:
Connected.
SQL> select count(*) from fail_tab_test;

COUNT(*)
-----
          50

SQL>

```

รูปที่ 5.18 แสดงการสอบถามข้อมูลบนฐานข้อมูลสำรองหลังเกิดเหตุการณ์ร้ายแรง

## บทที่ 6

### สรุปการวิจัยและข้อเสนอแนะ

#### 6.1 สรุปการวิจัย

ผลการวิจัยครั้งนี้ทำให้ได้เครื่องมือในการสร้างและจัดการฐานข้อมูลสำรอง ซึ่งพัฒนาขึ้นโดยใช้คอร์นเซลล์ โปรแกรมอรรถประโยชน์บนระบบยูนิกซ์ (UNIX) และคำสั่งในการจัดการฐานข้อมูลออลาเคิล ซึ่งมีส่วนติดต่อผู้ใช้เป็นลักษณะเมนูให้เลือกทำรายการ

จากการทดสอบพบว่า มีความสะดวกในการสร้างและจัดการฐานข้อมูลสำรอง และลดการทำงานที่ซ้ำซ้อนเพื่อลดความผิดพลาดที่อาจจะเกิดขึ้นได้ อีกทั้งช่วยให้เจ้าหน้าที่ดูแลระบบฐานข้อมูลสามารถกู้คืนฐานข้อมูลให้กลับคืนสู่สภาพเดิมได้อย่างรวดเร็ว ลดปัญหาการสูญหายของข้อมูลได้ ซึ่งมีผลทำให้ข้อมูลมีความถูกต้องและมีความน่าเชื่อถือด้วย

#### 6.2 เงื่อนไขในการใช้งาน

6.2.1 เครื่องแม่ข่ายหลักและเครื่องแม่ข่ายสำรองต้องมีเนื้อที่ของฮาร์ดดิสสำหรับเก็บข้อมูลเท่ากันทั้งสองเครื่อง เพื่อป้องกันกรณีของการเพิ่มแฟ้มข้อมูลบนฐานข้อมูลหลักได้แล้วแต่ไม่สามารถเพิ่มแฟ้มข้อมูลบนฐานข้อมูลสำรองตามได้ เนื่องจากเนื้อที่ของฮาร์ดดิสบนเครื่องแม่ข่ายสำรองมีไม่พอ

6.2.2 เครื่องแม่ข่ายหลักและเครื่องแม่ข่ายสำรองต้องใช้ฮาร์ดแวร์ชั้นเดียวกัน

#### 6.3 ปัญหาที่พบ

6.3.1 ในการใช้งานเครื่องจะต้องใช้ คอนโซล (Console) หรือ เทอร์มินอล (terminal) ที่สนับสนุนการทำงานที่มีวินโดว์แบบ X-window เนื่องจากมีโปรแกรมบางตัวทำงานผูกติดกับวินโดว์ลักษณะนี้ ทำให้ผู้ใช้งานต้องมีการเตรียมในส่วนนี้ด้วย

6.3.2 ในระหว่างการวิจัยถ้ามีการเปลี่ยนเครื่องที่ใช้การพัฒนาจะต้องทำการกำหนดในส่วนของระบบปฏิบัติการที่จำเป็นให้เหมือนเดิมด้วยดังที่กล่าวถึงการกำหนดในบทที่ 5

## 6.4 ข้อเสนอแนะ

6.4.1 การกำหนดให้มีการสร้างสมาชิกของแฟ้มบันทึกทำซ้ำเพิ่มบนเครื่องแม่ข่ายสำรองผ่านระบบเครื่องข่าย อาจมีผลต่อประสิทธิภาพของฐานข้อมูลหลัก ให้ทำงานช้าลงจึงควรออกแบบให้เครื่องแม่ข่ายหลัก และเครื่องแม่ข่ายสำรองมีเครื่องข่ายส่วนตัว (Private Network) เพื่อรองรับการทำงานในส่วนนี้

6.4.2 สามารถนำเครื่องมือที่พัฒนาขึ้นเป็นต้นแบบไปพัฒนาในรูปแบบของเว็บเบส (Webbase) เพื่อเพิ่มประสิทธิภาพในการจัดการฐานข้อมูลในระยะทางที่ห่างไกลออกไปรวมถึงการแสดงผลเป็นภาษาไทยโดยไม่ขึ้นอยู่กับวิธีการสนับสนุนการทำงานบนวินโดว์แบบ X-window



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## รายการอ้างอิง

### หนังสือ

1. Alice Y.H. Tsai . DATABASE SYSTEMS Management and use. Prentice-Hall Canada Inc.,Scarborough,Ontario,1990.
2. Burleson,Donald K. Managing Distributed Database, A Wiley/QED,1994.
3. C.J. Date. An Introduction To Database Systems. USA: Addison-Wesley Publishing Company,Inc., 1986.
4. Connie Dialeris,Joyce Free,Lance Ashdown . Oracle8i Backup and Recovery Guide. Oracle Corporation, 1999.
5. Lance Ashdown,Anna Logan. Oracle8i Standby Database Concept. Oracle Corporation, 1999.
6. Lefty Leverenz,Diana Rehfield,Cathy Baird. Oracle8i Concepts. Oracle Corporation, 1999.
7. Ruth Baylis,Joyce Fee. Oracle8i AdminGuide. Oracle Corporation, 1999.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก ก

### การใช้เครื่องมือ

ก่อนการใช้งานเครื่องมือที่พัฒนาขึ้นจะต้องมีการกำหนดในส่วนจากระบบปฏิบัติการ (Operating System) ทั้งเครื่องแม่ข่ายหลัก และเครื่องแม่ข่ายสำรองก่อนเพื่อให้สามารถรองรับการทำงานของเครื่องมือที่ทำการทดสอบได้ ซึ่งในการวิจัยใช้เครื่องชั้น ที่มีโซลาริส เป็นระบบปฏิบัติการ โดยที่การกำหนดจะกำหนดทั้งสองเครื่องดังนี้

#### การกำหนดเครื่องแม่ข่ายสำรอง

- กำหนดชื่อเครื่องแม่ข่ายหลัก และเครื่องแม่ข่ายสำรองในแฟ้ม /etc/hosts เพื่อให้เครื่องแม่ข่ายสำรองสามารถติดต่อผ่านเครื่องแม่ข่ายหลัก โดยใช้ชื่อได้ โครงสร้างดังนี้

```
# Ip-address    hostname    alias
202.20.105.243  sserver    standby
202.20.105.103  DB2        primary
```

- กำหนด Network File System (NFS) ในแฟ้ม /etc/dfs/dfstab ซึ่งจะเป็นการแบ่งเนื้อที่บางส่วนจากระบบดิสก์ให้เครื่องแม่ข่ายหลักสามารถทำการบันทึกข้อมูลลงไปได้ จะมีโครงสร้างดังนี้

```
# share network file system
share -F nfs -o rw /u01/kanit/oradata/REMOTER_REDO
```

- กำหนดชื่อผู้ใช้ (User) ให้เหมือนกับชื่อผู้ใช้ที่ทำหน้าที่ดูแลระบบฐานข้อมูลบนเครื่องแม่ข่ายหลัก ในแฟ้ม /etc/passwd จะมีโครงสร้างดังนี้

```
kanit:x:445:400:Standby-DB-Admin:/u01/kanit:/bin/ksh
```

ในแฟ้ม /etc/passwd แต่ละเขต (field) จะแยกกันด้วยเครื่องหมาย colon(:)

- กำหนดชื่อกลุ่มผู้ใช้ (Group) ให้กับชื่อกลุ่มผู้ใช้ที่ทำหน้าที่ดูแลระบบฐานข้อมูลบนเครื่องแม่ข่ายหลัก ในแฟ้ม /etc/group จะมีโครงสร้างดังนี้

```
dba::400:
```

- กำหนดสภาพแวดล้อมของผู้ใช้ให้เหมือนกับสภาพแวดล้อมของผู้ใช้ที่ทำหน้าที่ ดูแลระบบฐานข้อมูลบนเครื่องแม่ข่ายหลัก ในแฟ้ม .profile ภายใต้ไดเรกทอรีบ้าน (home directory) ของผู้ใช้

```
#
#@(#)local.profile 1.6 98/02/06 SMI
#
stty istrip
umask 022
stty erase ^H
export CONFIG=/u01/kanit/project

# Oracle Environment
export ORACLE_BASE=/u01/kanit/app/oracle8
export ORACLE_HOME=$ORACLE_BASE/product/8.1.7
export ORACLE_SID=DBE22
export MANPATH=/usr/share/man:/usr/dt/man
export PATH=$CONFIG/bin:/usr/bin:/usr/sbin:/sbin:$ORACLE_HOME/bin:.
set -o vi
alias cdh="cd $ORACLE_HOME"
alias cdb="cd $ORACLE_BASE/admin/DBE22/bdump"
alias cdp="cd $CONFIG"
```

- หลังจากที่มีการกำหนดค่าต่างๆ ที่จำเป็นแล้วบนเครื่องแม่ข่ายสำรองจะต้องมีการติดตั้ง Oracle Software ที่เส้นทางเดียวกันกับที่ติดตั้งบนเครื่องแม่ข่ายหลักด้วย

### การกำหนดเครื่องแม่ข่ายหลัก

- กำหนดในส่วนของการนำเนื้อที่บางส่วนของฮาร์ดดิสก์ที่เครื่องแม่ข่ายสำรองแบ่งไว้ผ่านระบบเครื่องข่ายมาใช้งาน (mount Network File System) โดยการเพิ่มบรรทัดนี้ในแฟ้ม /etc/vfstab จะมีโครงสร้างดังนี้

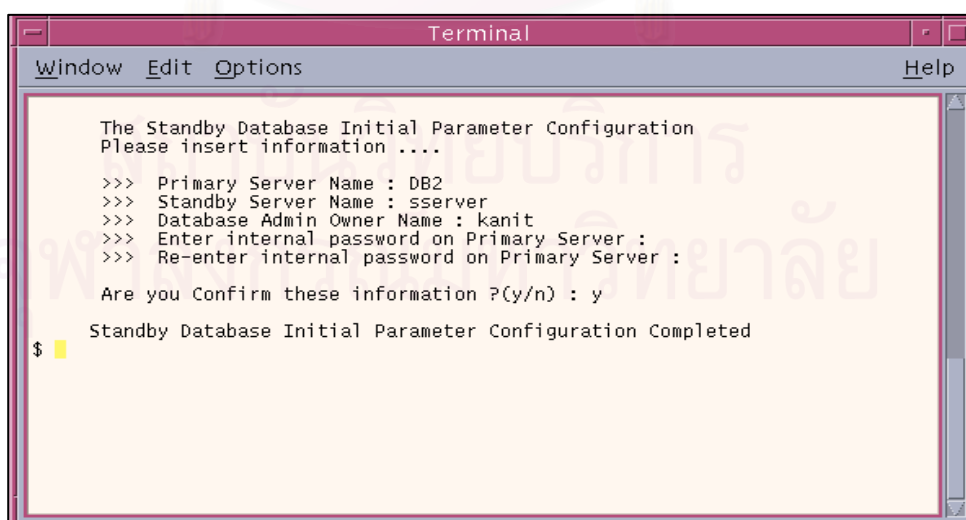
```
sserver:/u01/kanit/oradata/REMOTE_REDO - /remote_redo nfs 2 yes bg
```

- กำหนดในส่วนขงสิทธิต่างๆที่จะยอมให้เครื่องแม่ข่ายสำรองสามารถทำการติดต่อเข้ามาเพื่อใช้คำสั่งต่างๆบนเครื่องแม่ข่ายหลักได้ ในแฟ้ม .rhosts ภายใต้ ไดเรกทอรีบ้านของผู้ใช้ จะมีโครงสร้างดังนี้

```
sserver      kanit
```

### การกำหนดพารามิเตอร์เริ่มต้น

การใช้งานเครื่องมือที่พัฒนาขึ้นจะเริ่มจากออกคำสั่งให้ โปรแกรมชื่อ config.sh ทำงาน เพื่อให้มีการกำหนดค่าพารามิเตอร์เริ่มต้นต่างๆ ที่ต้องการแสดงในรูปแบบที่ 1 ข้อมูลที่กำหนดมีดังนี้ ชื่อเครื่องแม่ข่ายหลัก, ชื่อเครื่องแม่ข่ายสำรอง, ชื่อผู้ใช้ที่ทำหน้าที่ดูแลระบบฐานข้อมูล, รหัสผ่านภายใน (Internal Password)



```
Terminal
Window Edit Options Help

The Standby Database Initial Parameter Configuration
Please insert information ....

>>> Primary Server Name : DB2
>>> Standby Server Name : sserver
>>> Database Admin Owner Name : kanit
>>> Enter internal password on Primary Server :
>>> Re-enter internal password on Primary Server :

Are you Confirm these information ?(y/n) : y

Standby Database Initial Parameter Configuration Completed
$
```

รูปที่ 1 แสดงการกำหนดค่าพารามิเตอร์เริ่มต้น



## การทำงานของเครื่องมือ

หลังจากที่ขั้นตอนการกำหนดค่าพารามิเตอร์เริ่มต้นเรียบร้อยแล้วก็จะเข้ามาในส่วนของรายการหลัก(Main Menu) ดังแสดงในรูป 2

```

Terminal
Window Edit Options Help
----- Managing The Standby Database -----
Menu : Main Menu Standby Database

Primary Server: DB2      [202.20.105.103]
Standby Server: sserver [202.20.105.243]

1   Primary Database Information
2   Create/Drop Standby Database
3   Manage Standby Database
4   Standby Database Listener

h   Display Help About the Current Menu
q   Exit from Manager Standby Database

Select Operation to Perform :
-----

```

รูปที่ 2 แสดงรายการหลัก (Main Menu)

ในส่วนของรายการหลักจะมีทั้งหมด 4 รายการดังนี้

เมนูหลักรายการที่ 1. เป็นการแสดงข้อมูลของฐานข้อมูลหลัก

ซึ่งในรายการนี้จะมีรายการย่อยที่สามารถตรวจสอบข้อมูลของฐานข้อมูลหลัก ได้แก่ โครงสร้างทางด้านกายภาพ , ขนาดของแฟ้มข้อมูล , ข้อมูลการใช้เนื้อที่และเนื้อที่ว่างของ เทเบิลสเปซ รวมถึงการจัดการในส่วนของการเพิ่มและลบสมาชิกของแฟ้มลงบันทึกบนระบบฐานข้อมูลหลักด้วย ดังแสดงในรูปที่ 3

```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----

Menu: Primary Database Information

Primary Listener Status:  START
Primary Database Status:  START

1 Primary Database Information
2 Primary Database Space

-- Remote Redo Logfile --
3 Primary Remote Redo Logfile
4 Add Remote Redo Logfile
5 Drop Remote Redo Logfile

h Display Help About the Current Menu
q Exit from Current Menu

Select Operation to Perform :
-----

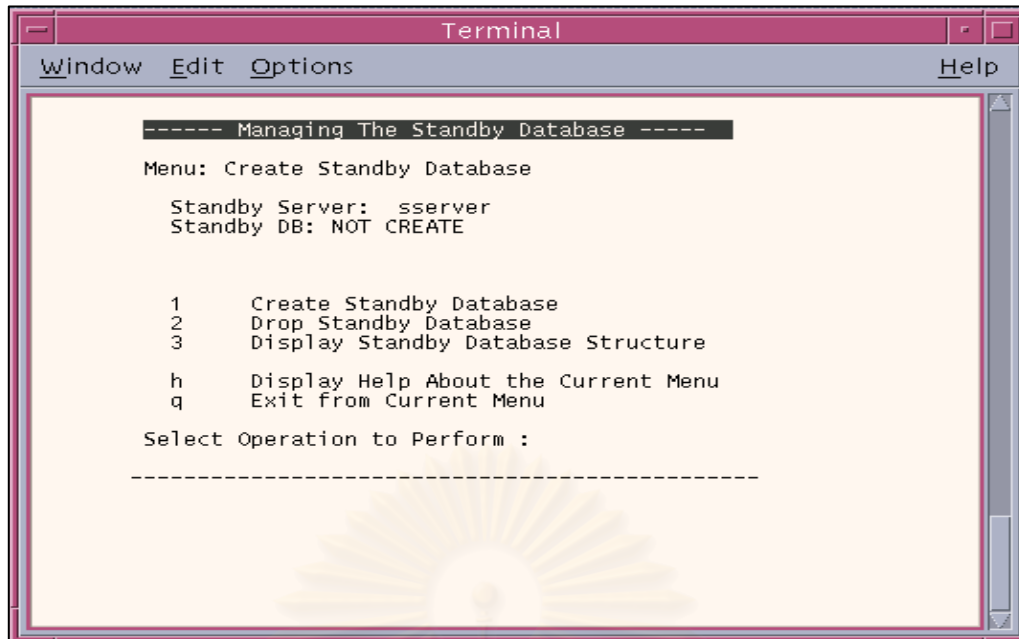
```

รูปที่ 3 เมนูย่อยของการแสดงข้อมูลของฐานข้อมูลหลัก

เมนูหลักรายการที่ 2 การสร้างและการลบฐานข้อมูลสำรอง

ทำหน้าที่ในการสร้างฐานข้อมูลสำรอง ลบฐานข้อมูลสำรอง และแสดงโครงสร้างฐานข้อมูลสำรอง ในรายการนี้เมื่อเริ่มใช้งานครั้งแรก สถานะการสร้างฐานข้อมูลสำรองจะขึ้นว่ายังไม่มีการสร้าง ดังแสดงในรูปที่ 4

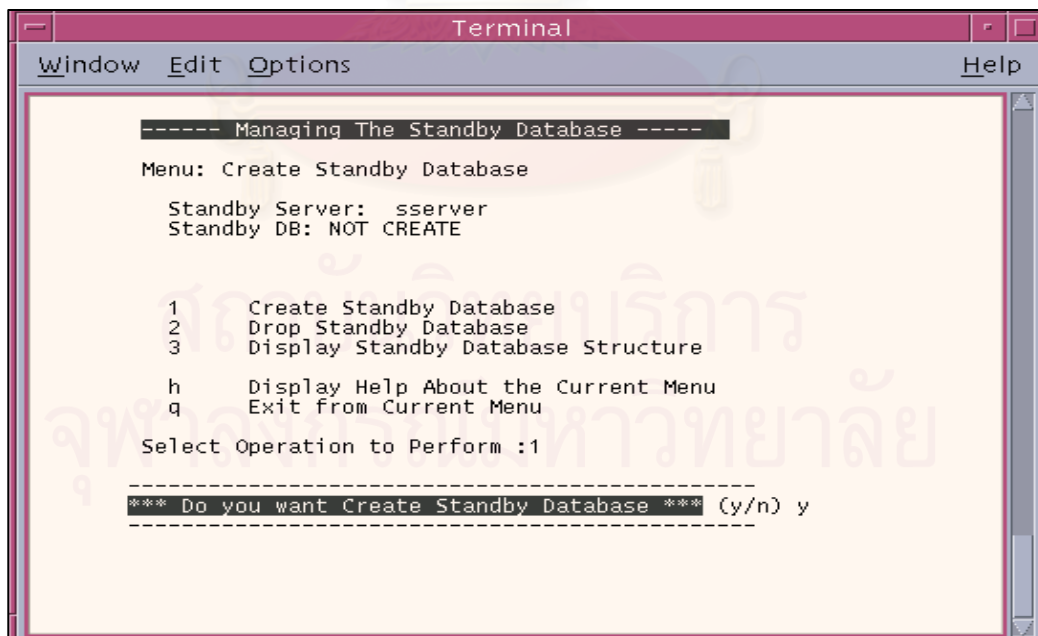
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4 การสร้างและการลบฐานข้อมูลสำรอง

ในการสร้างฐานข้อมูลสำรองจะมีการยืนยันการสร้างดังรูปที่ 5 ถ้ามีการยืนยันการสร้างเรียบร้อยแล้วจะมีการแสดงขั้นตอนในการสร้างแต่ละขั้นตอนดังรูปที่ 6

การสร้างฐานข้อมูลสำรอง



รูปที่ 5 แสดงการยืนยันการสร้างฐานข้อมูลสำรอง

```

Terminal
Window Edit Options Help

Creating Standby Database Network ./ Please wait...
Creating Standby Database DBE22 at sserver./ Please wait...
Tablespace = 5 Datafile = 5
Make Database Directory Completed..
Create Remote Redolog file member Please wait..
Create Standby Datafiles Please wait..

        datafile /u01/kanit/oradata/DBE22/rbs01.dbf

>>> Transfer tablespace RBS completed

        datafile /u01/kanit/oradata/DBE22/system01.dbf

>>> Transfer tablespace SYSTEM completed

        datafile /u01/kanit/oradata/DBE22/temp01.dbf

>>> Transfer tablespace TEMP completed

        datafile /u01/kanit/oradata/DBE22/tools01.dbf

>>> Transfer tablespace TOOLS completed

        datafile /u01/kanit/oradata/DBE22/users01.dbf

>>> Transfer tablespace USERS completed

Create Standby Datafiles Completed..
Create Standby Control files Completed..
Transfer Archive log files Completed..
Startup Standby Listener Completed..
Apply Archive log Completed..
Standby Database Startup Mount Completed..
Set Time to Automatic Transfer Archive log Completed..
Set Time to Automatic Change the Physical Database Completed..

Create Standby Database DBE22 at sserver are Completed

Please Enter to Continue...

```

รูปที่ 6 แสดงขั้นตอนการสร้างฐานข้อมูลสำรอง

ในการสร้างฐานข้อมูลสำรองจะทำงานตามขั้นตอนดังนี้

- ทำการสร้างการกำหนดการให้บริการด้านเครือข่ายบนฐานข้อมูลสำรองเพื่อให้สามารถรองรับการร้องขอจากเครื่องลูกข่ายได้
- เข้าไปทำการตรวจสอบจำนวนเทเบิลสเปซและเพิ่มข้อมูลบนฐานข้อมูลหลัก
- ทำการสร้างไดเรกทอรีสำหรับเป็นที่เก็บฐานข้อมูลสำรอง
- เพิ่มสมาชิกเพิ่มบันทึกทำซ้ำตามจำนวนกลุ่มที่มี

- ส่งผ่านเพิ่มข้อมูลเรียงทีละเทเบิลสเปซ
- ทำการสร้าง และส่งผ่านเพิ่มควบคุมสำหรับฐานข้อมูลสำรอง
- ทำการส่งผ่านเพิ่มบันทึกทำซ้ำถาวร
- ทำให้ฐานข้อมูลสำรองอยู่ในสถานะรอการกู้คืนเพื่อรอปรับปรุงข้อมูลจากเพิ่มบันทึกทำซ้ำถาวร
- ทำการกำหนดให้โปรแกรมที่ใช้ในการส่งผ่านเพิ่มบันทึกทำซ้ำถาวรและโปรแกรมตรวจสอบและปรับปรุงโครงสร้าง ตั้งเวลาทำงานแบบอัตโนมัติทุกๆ 5 นาที

หลังจากที่ฐานข้อมูลสำรองเสร็จเรียบร้อยแล้ว จะกลับมาสู่เมนูรายการเดิม และจะมีการแสดงข้อมูลวันที่ทำการสร้าง และสถานะของฐานข้อมูลสำรองในขณะนั้นตามที่แสดงดังรูปที่ 7

```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----

Menu: Create Standby Database

Standby Server: sserver
Standby DB: CREATED   Create Date: 07/08/02
DB Name   : DBE22     Status   : MOUNTED

1   Create Standby Database
2   Drop Standby Database
3   Display Standby Database Structure

h   Display Help About the Current Menu
q   Exit from Current Menu

Select Operation to Perform :

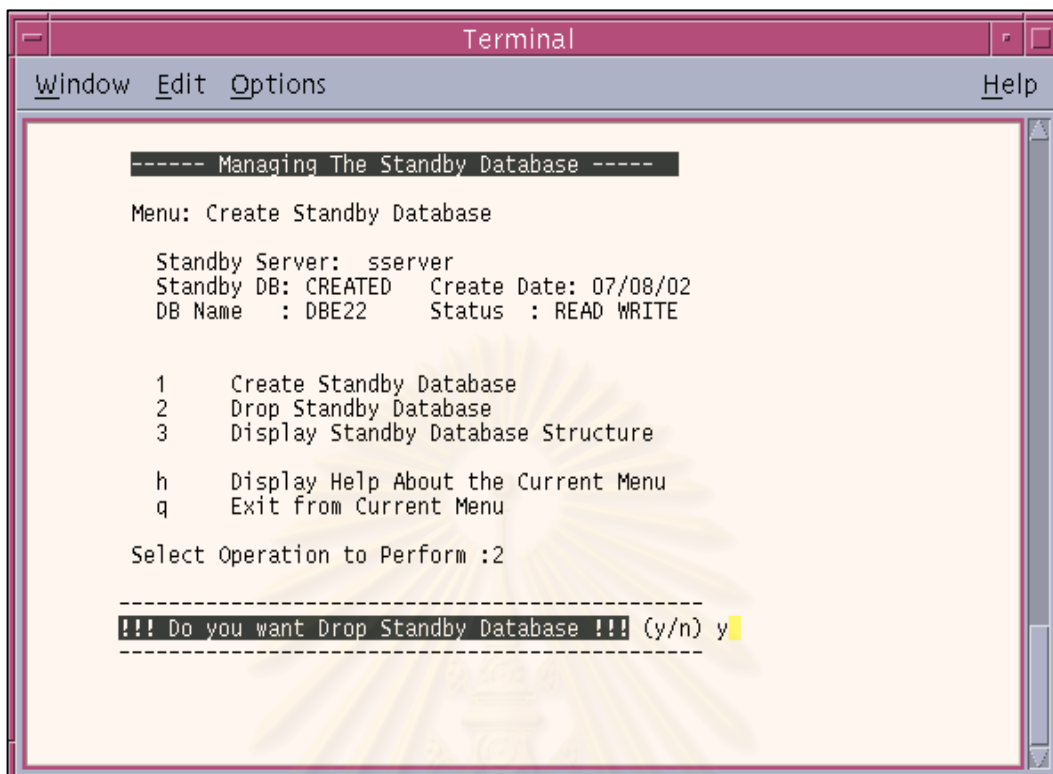
-----

```

รูปที่ 7 แสดงสถานะฐานข้อมูลสำรองหลังจากสร้างเรียบร้อยแล้ว

#### การลบฐานข้อมูลสำรอง

ในกรณีที่ฐานข้อมูลสำรองมีความผิดพลาดเกิดขึ้นจนไม่สามารถทำงานตามปกติได้ทำให้ต้องทำการสร้างฐานข้อมูลสำรองใหม่อีกครั้งจึงจำเป็นต้องทำการลบฐานข้อมูลสำรองตัวเดิมก่อน ซึ่งมีขั้นตอนการทำงานดังรูปที่ 8



```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----

Menu: Create Standby Database

Standby Server: sserver
Standby DB: CREATED Create Date: 07/08/02
DB Name : DBE22 Status : READ WRITE

1 Create Standby Database
2 Drop Standby Database
3 Display Standby Database Structure

h Display Help About the Current Menu
q Exit from Current Menu

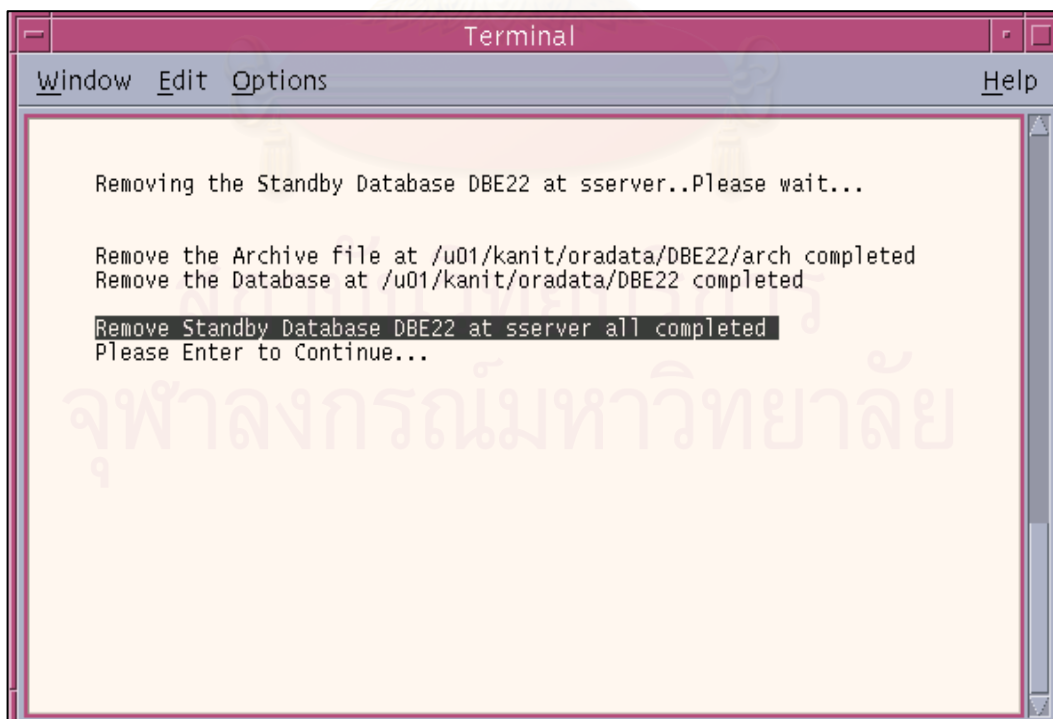
Select Operation to Perform :2

-----
!!! Do you want Drop Standby Database !!! (y/n) y
-----

```

รูปที่ 8 แสดงการยืนยันการลบฐานข้อมูลสำรอง

หลังจากที่มีการยืนยันการลบฐานข้อมูลสำรองแล้วจะเป็นขั้นตอนการลบดังที่แสดงในรูปที่ 9



```

Terminal
Window Edit Options Help

Removing the Standby Database DBE22 at sserver..Please wait...

Remove the Archive file at /u01/kanit/oradata/DBE22/arch completed
Remove the Database at /u01/kanit/oradata/DBE22 completed

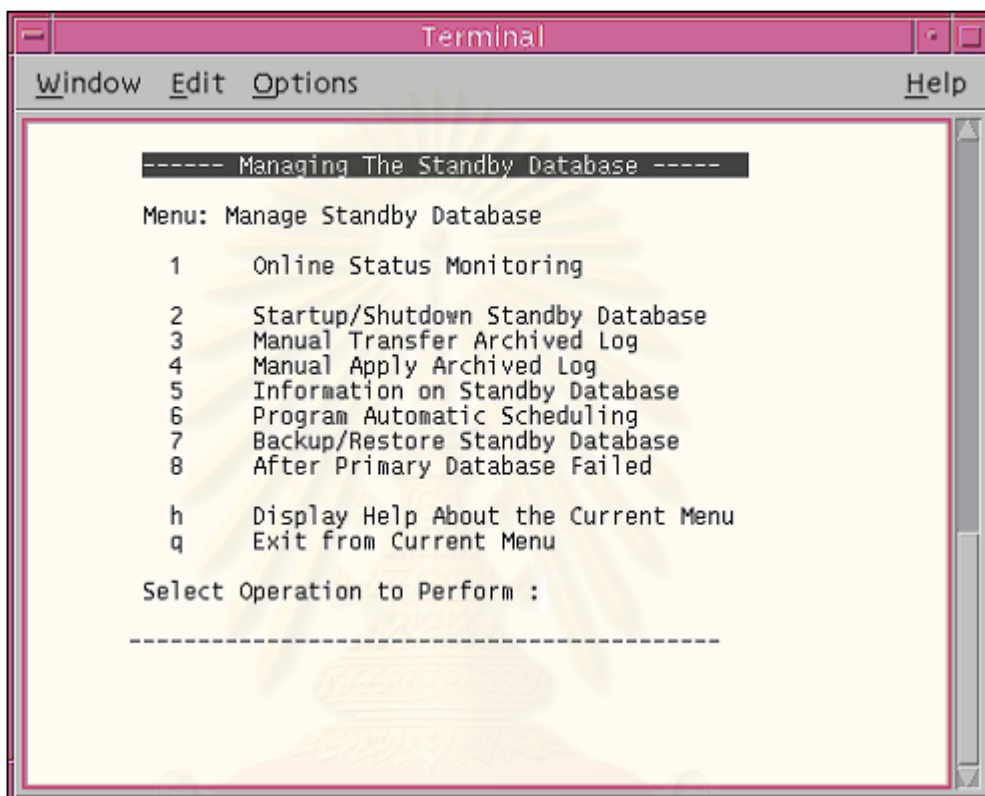
Remove Standby Database DBE22 at sserver all completed
Please Enter to Continue...

```

รูปที่ 9 แสดงขั้นตอนการลบฐานข้อมูลสำรอง

เมนูหลักรายการที่ 3 การจัดการฐานข้อมูลสำรอง

ในรายการนี้จะมีรายการย่อยที่ทำหน้าที่ในการจัดการฐานข้อมูลสำรอง ซึ่งเมนูหลัก รายการที่ 3 นี้จะมีรายการย่อยที่ทำหน้าที่ต่างๆ ดังแสดงในรูปที่ 10



รูปที่ 10 แสดงรายการจัดการฐานข้อมูลสำรอง

#### รายการแสดงการตรวจสอบสถานะของฐานข้อมูลทั้งสอง

ในรายการนี้จะแสดงสถานะของฐานข้อมูลหลักและฐานข้อมูลสำรอง ดังนี้

- แสดงสถานะการทำงานของกรให้บริการด้านเครือข่ายบนฐานข้อมูลทั้งสอง
- แสดงชื่อเครื่องแม่ข่ายทั้งสอง ในที่นี้ชื่อเครื่องแม่ข่ายหลักคือ DB2 และชื่อเครื่องแม่ข่ายสำรองคือ sserver
- แสดงชื่อฐานข้อมูล ในที่นี้คือชื่อ DBE22

- แสดงสถานะของฐานข้อมูลหลักเป็นสถานะ READ WRITE และฐานข้อมูลสำรองเป็นสถานะ MOUNT คือ สถานะรอการปรับปรุงข้อมูลจากแฟ้มบันทึกถาวร
- แสดงจำนวนเทเบิลสเปซ และ แฟ้มข้อมูล
- แสดงหมายเลขแฟ้มบันทึกถาวรที่ทำการปรับปรุงเรียบร้อยแล้ว
- แสดงหมายเลขแฟ้มบันทึกถาวรที่ทำการส่งจากเครื่องแม่ข่ายหลักมาเครื่องแม่ข่ายสำรองเรียบร้อยแล้ว

```

---ONLINE MONITORING---
Window Edit Options Help
<> MANAGING THE STANDBY DATABASE <>
----- ONLINE STATUS -----
DATE: 06/08/02 TIME: 09:18
-----
*** Primary Database ***
-----
Primary Listener Online
Primary Server: DB2
Database Name: DBE22
Standby Status: READ WRITE
Tablespace: # 5
Datafiles: # 5
Max History Number : 1758

*** Standby Database ***
-----
Standby Listener Online
Standby Server: sserver
Database Name: DBE22
Standby Status: MOUNTED
Tablespace: # 5
Datafiles: # 5
Max History Number : 1756

Archived Transfer From DB2 --> sserver : 1756
-----

```

รูปที่ 11 แสดงการตรวจสอบสถานะของฐานข้อมูลทั้งสอง

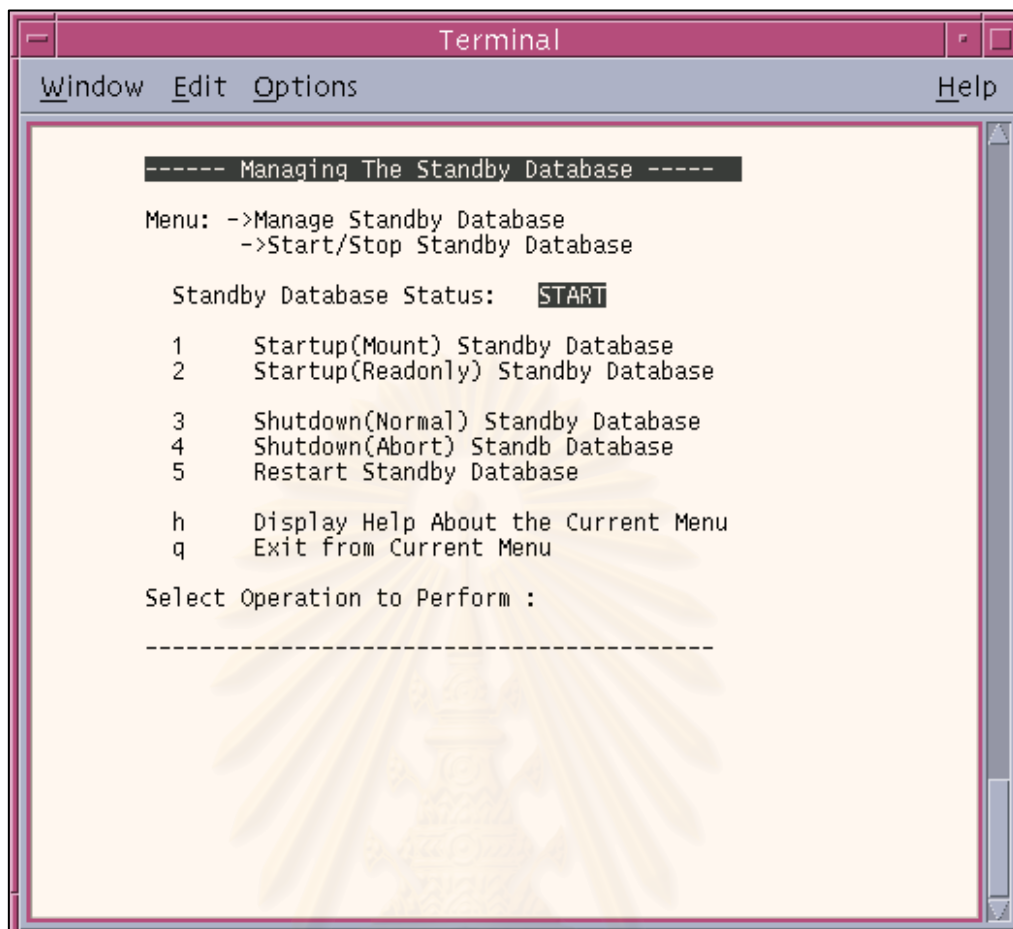
### เมนูรายการย่อสำหรับการเริ่มการทำงานและหยุดการทำงาน

ฐานข้อมูลสำรอง

ในเมนูรายการย่อนี้ทำหน้าที่ในการเริ่มการทำงานและหยุดการทำงานฐานข้อมูลสำรอง

ซึ่งมีการแสดงรายการดังรูปที่ 12





รูปที่ 12 เมนูรายการย่อยสำหรับการเริ่มการทำงานและหยุดการทำงานฐานข้อมูลสำรอง

#### การกำหนดการส่งผ่านแฟ้มบันทึกถาวรมาฐานข้อมูลสำรองด้วยตัวเอง

เมนูรายการนี้กำหนดเพื่อให้เจ้าหน้าที่ดูแลระบบฐานข้อมูลสำรองสามารถทำการส่งผ่านแฟ้มบันทึกถาวรเอง ในกรณีที่มีการส่งผ่านแฟ้มบันทึกถาวรเกิดความผิดพลาดจากสาเหตุใดๆ ก็ตาม โดยที่การใช้งานจะต้องระบุ หมายเลขของแฟ้มบันทึกถาวร ดังรูปที่ 13

จุฬาลงกรณ์มหาวิทยาลัย

```

Terminal
----- Managing The Standby Database -----
Menu: Manage Standby Database

1   Online Status Monitoring
2   Startup/Shutdown Standby Database
3   Manual Transfer Archived Log
4   Manual Apply Archived Log
5   Information on Standby Database
6   Program Automatic Scheduling
7   Backup/Restore Standby Database
8   After Primary Database Failed

h   Display Help About the Current Menu
q   Exit from Current Menu

Select Operation to Perform :3

-----

Please Enter Number of Archive Log to Transfer:4116
>>> /u01/kanit/oradata/DBE22/arch/1_4116.dbf
Transfer completed /Please enter to continue..

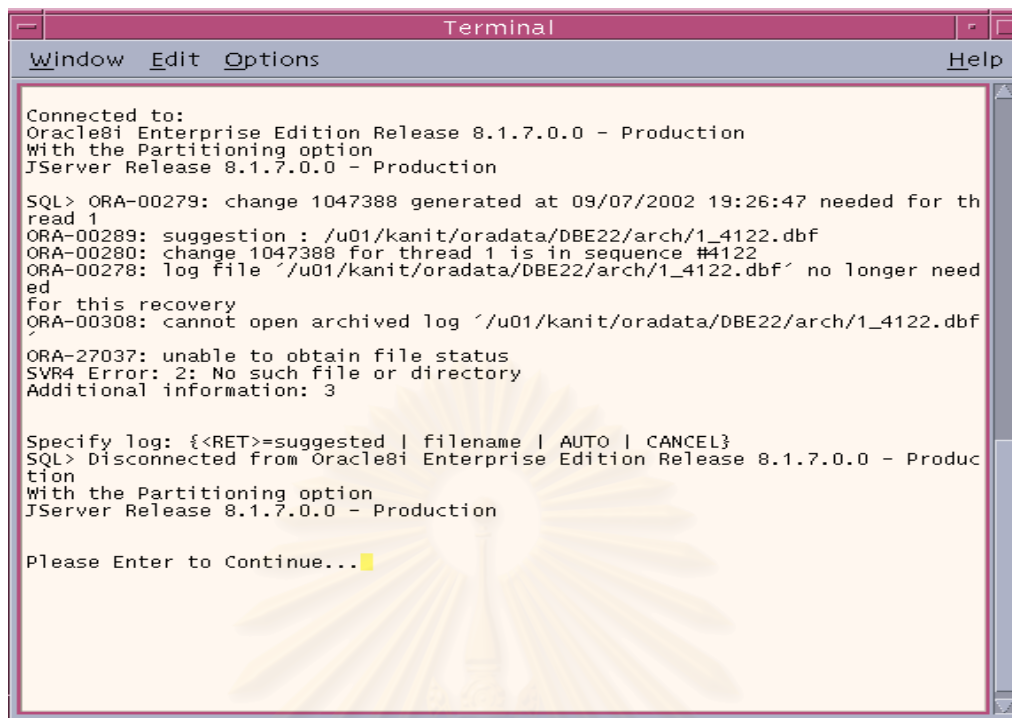
```

รูปที่ 13 แสดงการส่งผ่านแฟ้มบันทึกถาวรมาฐานข้อมูลสำรอง

### การปรับปรุงข้อมูลบนแฟ้มบันทึกถาวรด้วยตัวเอง

เมนูรายการนี้กำหนดเพื่อให้เจ้าหน้าที่ดูแลระบบฐานข้อมูลสามารถ ทำการปรับปรุงข้อมูลบนแฟ้มบันทึกถาวรเอง ในกรณีที่ยังไม่ถึงเวลาการทำงานของโปรแกรมตั้งเวลาทำงานอัตโนมัติ หรือ ใช้การปรับปรุงข้อมูลหลังจากที่มีการการนำฐานข้อมูลสำรองกลับคืน เพื่อให้ฐานข้อมูลสำรองมีข้อมูลอยู่ในสถานะใกล้เคียงกับฐานข้อมูลหลักมากที่สุด ดังแสดงในรูปที่ 14

จุฬาลงกรณ์มหาวิทยาลัย



```

Terminal
Window Edit Options Help
Connected to:
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

SQL> ORA-00279: change 1047388 generated at 09/07/2002 19:26:47 needed for th
read 1
ORA-00289: suggestion : /u01/kanit/oradata/DBE22/arch/1_4122.dbf
ORA-00280: change 1047388 for thread 1 is in sequence #4122
ORA-00278: log file '/u01/kanit/oradata/DBE22/arch/1_4122.dbf' no longer need
ed
for this recovery
ORA-00308: cannot open archived log '/u01/kanit/oradata/DBE22/arch/1_4122.dbf'
ORA-27037: unable to obtain file status
SVR4 Error: 2: No such file or directory
Additional information: 3

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
SQL> Disconnected from Oracle8i Enterprise Edition Release 8.1.7.0.0 - Produc
tion
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

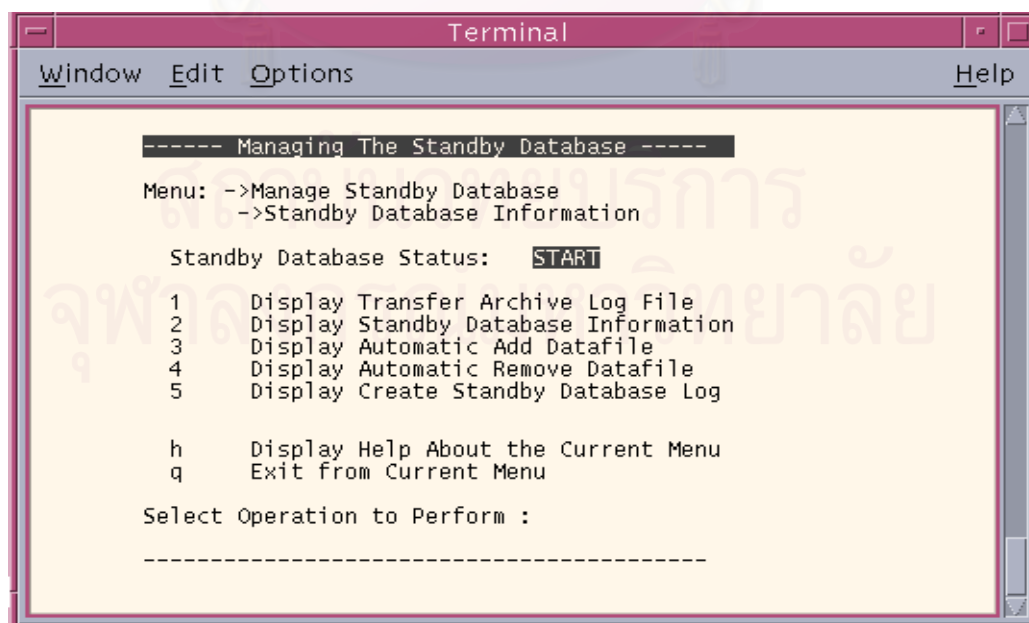
Please Enter to Continue...

```

รูปที่ 14 แสดงปรับปรุงข้อมูลบนแฟ้มบันทึกถาวรเอง

#### รายการแสดงรายละเอียดของฐานข้อมูลสำรอง

ซึ่งในรายการนี้จะมีรายการย่อยนี้แสดงข้อมูลของฐานข้อมูลสำรองเกี่ยวกับโครงสร้างของฐานข้อมูลสำรอง รวมถึงประวัติการการส่งผ่านแฟ้มบันทึกถาวร ดังแสดงในรูปที่ 15



```

Terminal
Window Edit Options Help
----- Managing The Standby Database -----
Menu: ->Manage Standby Database
      ->Standby Database Information

Standby Database Status:  START

1  Display Transfer Archive Log File
2  Display Standby Database Information
3  Display Automatic Add Datafile
4  Display Automatic Remove Datafile
5  Display Create Standby Database Log

h   Display Help About the Current Menu
q   Exit from Current Menu

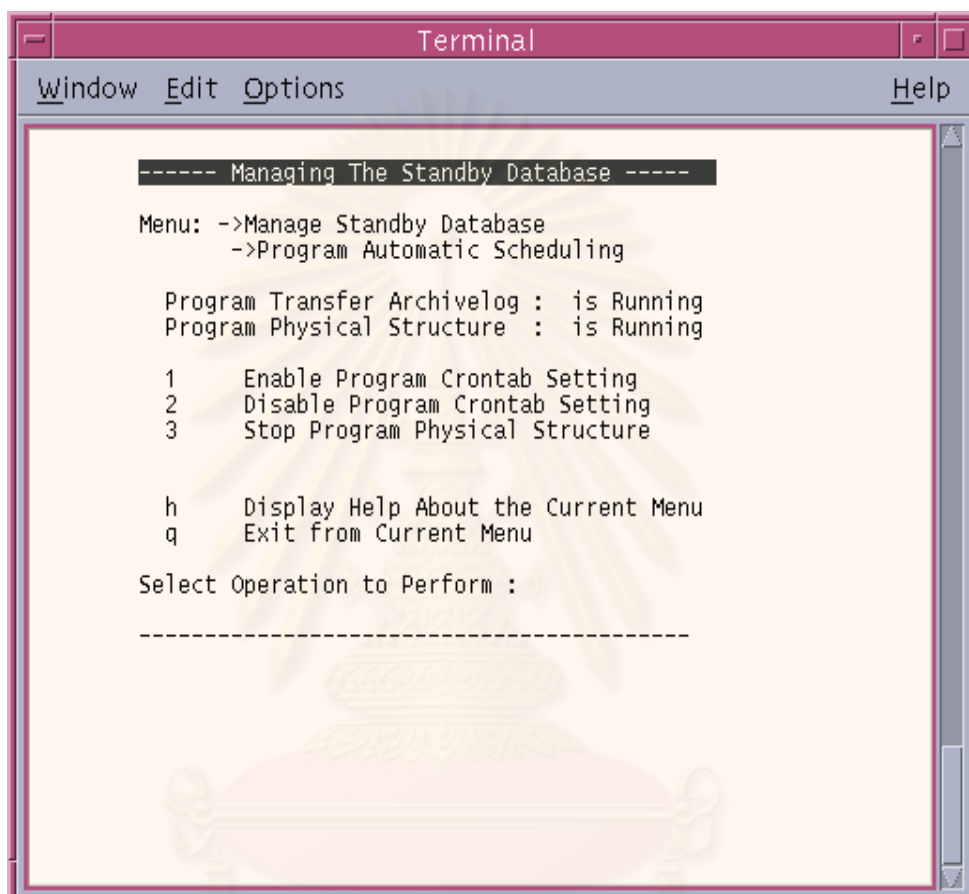
Select Operation to Perform :
-----

```

รูปที่ 15 รายการแสดงรายละเอียดของฐานข้อมูลสำรอง

### เมนูรายการย่อยการทำงานของโปรแกรมตั้งเวลาทำงานอัตโนมัติ

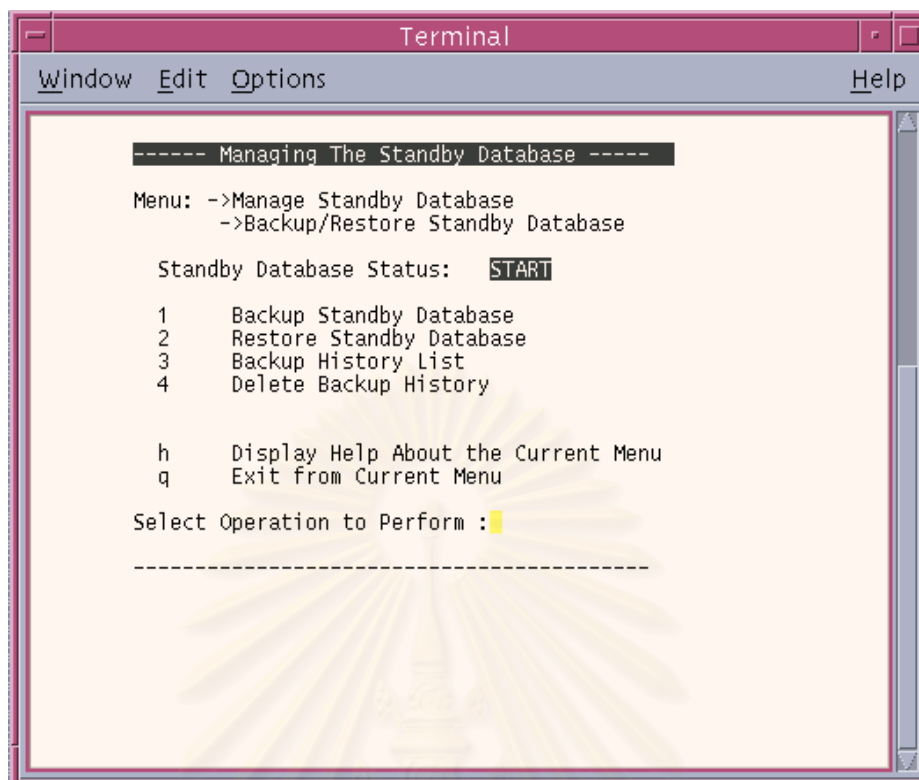
ในเมนูรายการย่อยนี้สามารถสั่งให้โปรแกรมในการส่งผ่านและปรับปรุงแฟ้มบันทึกถาวร รวมถึงการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลอัตโนมัติที่ตั้งเวลาทำงานอัตโนมัติเริ่มต้น หรือหยุดการทำงานได้ ดังแสดงดังรูปที่ 16



รูปที่ 16 แสดงรายการย่อยการทำงานของโปรแกรมตั้งเวลาทำงานอัตโนมัติ

### เมนูรายการย่อยการทำสำรอง และการนำฐานข้อมูลสำรองกลับคืน

ในเมนูรายการนี้จะทำหน้าที่ในการจัดการเกี่ยวกับสำรองข้อมูล การนำฐานข้อมูลสำรองกลับคืน และสามารถเก็บประวัติการทำสำรองข้อมูล รวมถึงการจัดการลบประวัติการทำสำรองข้อมูลครั้งที่ผ่านมดังรูปที่ 17



รูปที่ 17 แสดงเมนูรายการทำสำรอง และการนำฐานข้อมูลสำรองกลับคืน

ในการสำรองข้อมูลบนฐานข้อมูลสำรองจะทำการเก็บลงบนฮาร์ดดิสก์บนเครื่องแม่ข่ายสำรองเพื่อให้สามารถทำการนำฐานข้อมูลสำรองกลับคืนได้ หลังจากที่เราเปลี่ยนสถานะจากฐานข้อมูลสำรองเป็นสถานะเขียนอ่านแล้ว เพื่อให้ฐานข้อมูลสำรองกลับมาอยู่ในสถานะรอการกู้คืนได้จะมีขั้นตอนดังรูปที่ 18

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

```

Terminal
Window Edit Options Help

Backup The Standby Database
-----

Starting backup tablespace: RBS ..
>>>> backup /u01/kanit/oradata/DBE22/rbs01.dbf
End of backup tablespace RBS
*****

Starting backup tablespace: SYSTEM ..
>>>> backup /u01/kanit/oradata/DBE22/system01.dbf
End of backup tablespace SYSTEM
*****

Starting backup tablespace: TEMP ..
>>>> backup /u01/kanit/oradata/DBE22/temp01.dbf
End of backup tablespace TEMP
*****

Starting backup tablespace: TOOLS ..
>>>> backup /u01/kanit/oradata/DBE22/tools01.dbf
End of backup tablespace TOOLS
*****

Starting backup tablespace: USERS ..
>>>> backup /u01/kanit/oradata/DBE22/users01.dbf
end of backup tablespace USERS
*****

-----

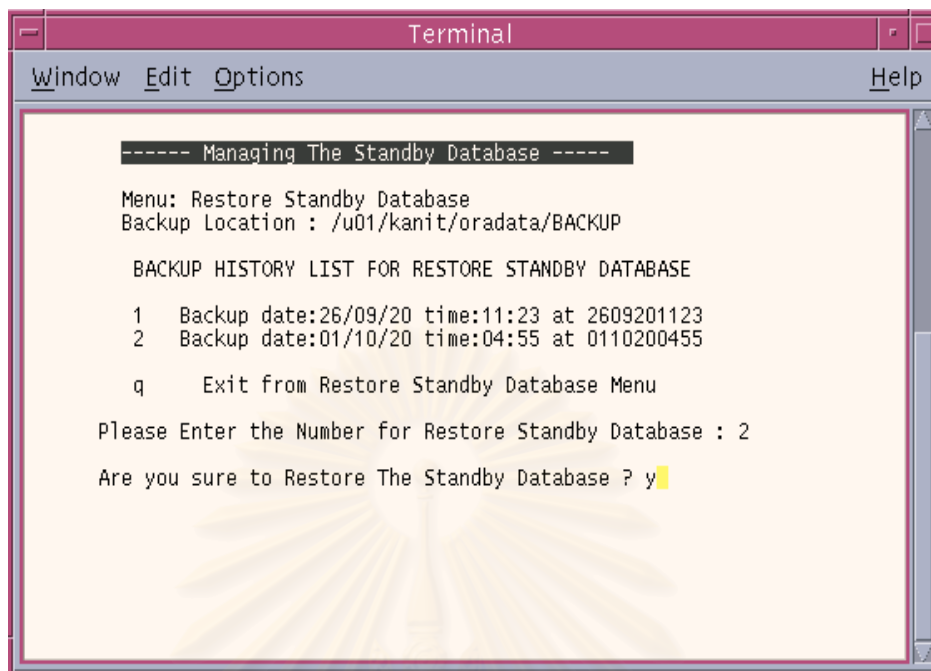
Backup The Standby Database Completed

Please Enter to continue and wait a minute..

```

รูปที่ 18 แสดงการสำรองข้อมูลบนฐานข้อมูลสำรอง

หลังจากที่ทำการสำรองข้อมูลบนฐานข้อมูลสำรองเสร็จเรียบร้อยแล้ว ยังสามารถทำการนำฐานข้อมูลสำรองกลับคืนโดยการเลือกจากประวัติการทำสำรองที่ผ่านมาดังรูปที่ 19 โดยที่ขั้นตอนการนำฐานข้อมูลสำรองกลับคืนจะแสดงดังรูปที่ 20 เมื่อการนำฐานข้อมูลสำรองกลับคืนเสร็จเรียบร้อยแล้วจะต้องทำการปรับปรุงฐานข้อมูลสำรองให้เป็นปัจจุบันมากที่สุด โดยเริ่มจากสั่งให้ฐานข้อมูลสำรองเริ่มทำงานใหม่และให้อยู่ในสถานะรอการปรับปรุงข้อมูล หลังจากนั้นจะมีการปรับปรุงข้อมูลจากแพมบันทึกถาวรชุดเก่าที่มาพร้อมกับการนำฐานข้อมูลสำรองกลับคืนให้อัตโนมัติ หลังจากนั้นฐานข้อมูลสำรองจะอยู่ในสถานะรอการกู้คืน เพื่อรอการปรับปรุงข้อมูลจากแพมบันทึกถาวรที่จะส่งมาจากฐานข้อมูลหลักต่อไป



```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----

Menu: Restore Standby Database
Backup Location : /u01/kanit/oradata/BACKUP

BACKUP HISTORY LIST FOR RESTORE STANDBY DATABASE

1 Backup date:26/09/20 time:11:23 at 2609201123
2 Backup date:01/10/20 time:04:55 at 0110200455

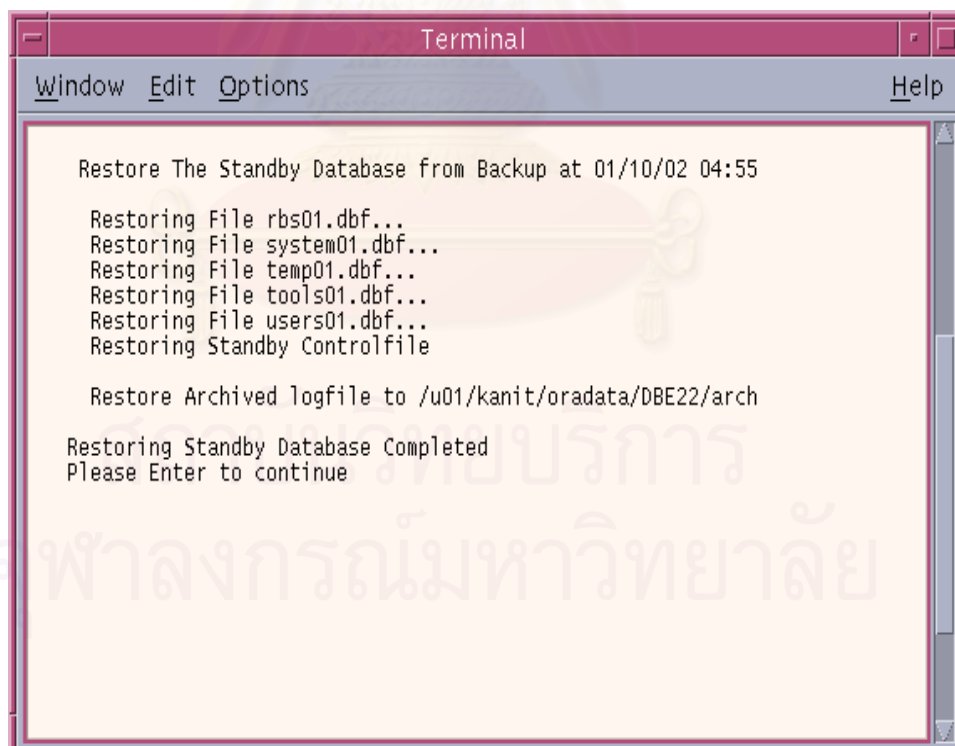
q Exit from Restore Standby Database Menu

Please Enter the Number for Restore Standby Database : 2

Are you sure to Restore The Standby Database ? y

```

รูปที่ 19 แสดงการนำฐานข้อมูลสำรองกลับคืนจากประวัติการทำสำรอง



```

Terminal
Window Edit Options Help

Restore The Standby Database from Backup at 01/10/02 04:55

Restoring File rbs01.dbf...
Restoring File system01.dbf...
Restoring File temp01.dbf...
Restoring File tools01.dbf...
Restoring File users01.dbf...
Restoring Standby Controlfile

Restore Archived logfile to /u01/kanit/oradata/DBE22/arch

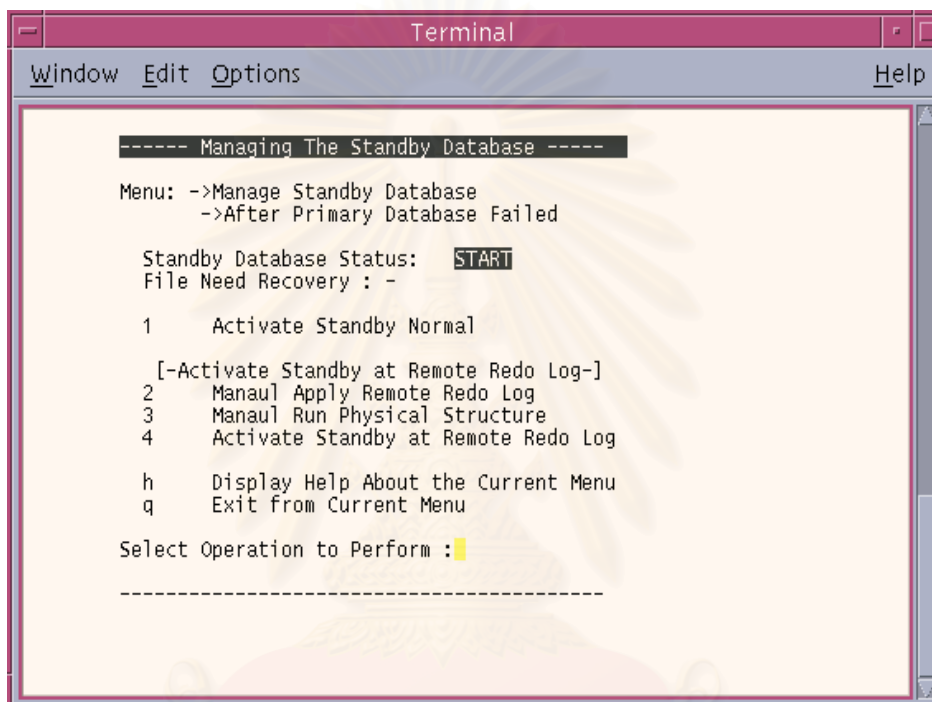
Restoring Standby Database Completed
Please Enter to continue

```

รูปที่ 20 แสดงการนำฐานข้อมูลสำรองกลับคืน

### การจัดการฐานข้อมูลสำรองหลังจากฐานข้อมูลหลักเสียหาย

ในเมนูรายการนี้จะทำหน้าที่ในการจัดการเกี่ยวกับฐานข้อมูลสำรองหลังจากเครื่องแม่ข่ายเสียหายจากเหตุการณ์ต่างๆ มีผลให้ฐานข้อมูลหลักไม่สามารถทำงานได้ตามปกติ ซึ่งเจ้าหน้าที่ดูแลระบบฐานข้อมูลจำเป็นต้องทำการเปลี่ยนสถานะของฐานข้อมูลสำรองบนเครื่องแม่ข่ายสำรองให้เป็นฐานข้อมูลหลักเพื่อทำงานทดแทนฐานข้อมูลหลักจริงที่ไม่สามารถทำงานได้ ดังแสดงเมนูรายการในรูปที่ 21



รูปที่ 21 แสดงเมนูรายการจัดการฐานข้อมูลสำรองหลังจากเครื่องแม่ข่ายเสียหาย

จากรูปที่ 21 รายการที่แสดงจะสามารถเลือกการทำงานได้สองลักษณะดังนี้ คือ สามารถเลือกการเปลี่ยนสถานะของฐานข้อมูลสำรองแบบปกติ ซึ่งจะไม่มีการปรับปรุงข้อมูลบนแฟ้มบันทึกทำซ้ำ ซึ่งอาจทำให้ข้อมูลสูญหายได้เนื่องจากข้อมูลบางส่วนที่ยังอยู่บนแฟ้มบันทึกทำซ้ำ หรือเลือกการเปลี่ยนสถานะของฐานข้อมูลสำรองโดยจะมีการปรับปรุงข้อมูลบนแฟ้มบันทึกทำซ้ำ ซึ่งจะได้ข้อมูลที่ครบถ้วนเหมือนฐานข้อมูลหลัก ซึ่งจะต้องมีขั้นตอนเพิ่มขึ้นในส่วนของการใช้งานเมื่อเครื่องแม่ข่ายหลักไม่สามารถทำงานได้เครื่องมือในการตรวจสอบสถานะของฐานข้อมูลทั้งสอง จะแจ้งสถานะการทำงานของเครื่องแม่ข่ายหลักเสียหายดังแสดงในรูปที่ 22 ขั้นตอนต่อไปเจ้าหน้าที่ดูแลระบบฐานข้อมูลจะต้องทำการเลือกทำการปรับปรุงข้อมูลจากแฟ้มบันทึกทำซ้ำดังแสดงในรูปที่ 23



```

---ONLINE MONITORING---
Window Edit Options Help
<> MANAGING THE STANDBY DATABASE <>
----- ONLINE STATUS -----
DATE: 09/10/02 TIME: 15:03
-----
*** Primary Database ***
-----
////////////////
Primary Server Down
////////////////
Max History Number : 13822

*** Standby Database ***
-----
Standby Listener Online
Standby Server: sserver
Database Name: DBE22
Database Mode: READ WRITE
Tablespace: # 5
Datafiles: # 5
Max History Number : 2

Archived Transfer From DB2 --> sserver : 13821
-----

```

รูปที่ 22 แสดงสถานะของเครื่องแม่ข่ายหลักไม่สามารถใช้งานได้

```

Terminal
Window Edit Options Help
----- Managing The Standby Database -----
Menu: ->Manage Standby Database
      ->After Primary Database Failed

Standby Database Status: START
File Need Recovery : -

1 Activate Standby Normal

[-Activate Standby at Remote Redo Log-]
2 Manual Apply Remote Redo Log
3 Manual Run Physical Structure
4 Activate Standby at Remote Redo Log

h Display Help About the Current Menu
q Exit from Current Menu

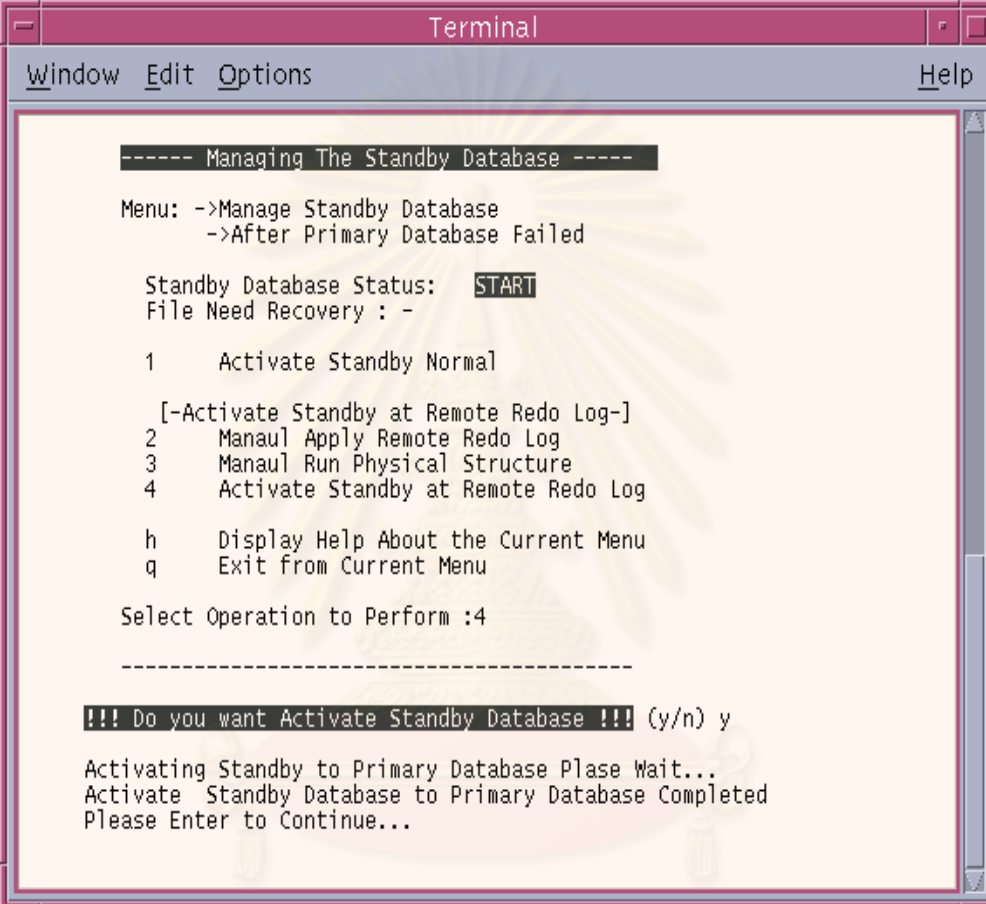
Select Operation to Perform :2

-----
!!! Do you want Apply Remote Redolog !!! (y/n) y
Apply Remote Redolog Plase Wait...
Apply Remote Redolog Completed
Please Enter to Continue...

```

รูปที่ 23 การปรับปรุงข้อมูลจากเพ็บบันทึกทำซ้ำ

หลังจากนั้นจึงทำการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลสำรองเองในกรณีที่มีการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลหลักและมีการบันทึกบนแฟ้มบันทึกทำซ้ำแล้ว แต่ยังไม่มีการบันทึกลงบนแฟ้มบันทึกถาวร ซึ่งจะมีผลทำให้ไม่สามารถปรับปรุงข้อมูลต่อไปได้ ซึ่งจะแสดงขั้นตอนทำงานใน บทที่ 6 หลังจากนั้นจึงทำการเปลี่ยนสถานะของฐานข้อมูลสำรองบนเครื่องแม่ข่ายสำรองให้เป็นฐานข้อมูลหลักเพื่อทำงานทดแทนฐานข้อมูลหลักต่อไป ดังแสดงในรูปที่ 24



```

Terminal
Window Edit Options Help

----- Managing The Standby Database -----

Menu: ->Manage Standby Database
      ->After Primary Database Failed

Standby Database Status:  START
File Need Recovery : -

1   Activate Standby Normal

[-Activate Standby at Remote Redo Log-]
2   Manual Apply Remote Redo Log
3   Manual Run Physical Structure
4   Activate Standby at Remote Redo Log

h   Display Help About the Current Menu
q   Exit from Current Menu

Select Operation to Perform :4

-----

!!! Do you want Activate Standby Database !!! (y/n) y

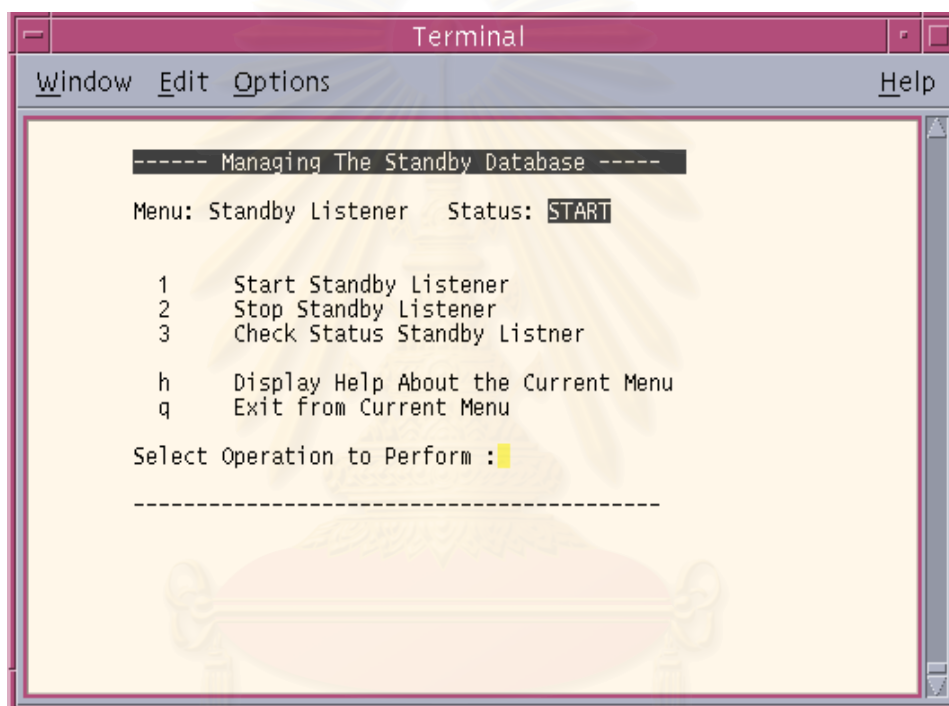
Activating Standby to Primary Database Plase Wait...
Activate Standby Database to Primary Database Completed
Please Enter to Continue...

```

รูปที่ 24 การทำการเปลี่ยนสถานะของฐานข้อมูลสำรองเป็นฐานข้อมูลหลัก

เมนูหลักรายการที่ 4 การจัดการด้านเครือข่ายบนฐานข้อมูลสำรอง

บนฐานข้อมูลของจำเป็นต้องมีการจัดการด้านเครือข่าย สำหรับของเครื่องลูกข่าย(client) ที่มีการร้องขอเข้ามาเพื่อติดต่อกับฐานข้อมูล โดยมีการกำหนดช่องทางบริการ(service port) ขึ้นเพื่อรองรับการร้องขอ ซึ่งในเมนูรายการนี้ช่วยให้สามารถเริ่มและหยุดการให้บริการในส่วนช่องทางบริการบนฐานข้อมูลสำรองได้ดังรูป 25



รูปที่ 25 การจัดการด้านเครือข่ายบนฐานข้อมูลสำรอง

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก ข

### ตัวอย่างโปรแกรม

ตัวอย่างโปรแกรมชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลหลัก

โปรแกรม base\_command\_primary.sh

```
#!/bin/ksh
#####
#       Program   : base_command_primary.sh
#       Description : Keep the Function Module with connect to Primary Server
#                   and Primary database for get some information
#####
#       Function Name : create_standby_controlfile
#       Description  : create standby controlfile on Primary Database
#####
create_standby_controlfile() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF > /dev/null
    alter database create standby controlfile as '$STDBY_CONTROLFILE' reuse;
    exit
EOF
}
#####
#       Function Name : rcp_control_file_sqlplus
#       Description  : Get location of primary controlfile
#                   and generate command for remote copy the standby controlfile
#                   to standby server in log file $LOG_DIR/rcp_control_file.log
#####
rcp_control_file_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    @$SQL_DIR/rcp_control_file.sql
EOF
}
#####
#       Function Name : redo_group_count
#       Description  : count the redo log group for create remote redo log
#                   on standby database
```

## โปรแกรม base\_command\_primary.sh (ต่อ)

```

redo_group_count() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    set head off
    set feedback off
        spool $LOG_DIR/redo_group_count.log
        select group# from v$log;
EOF
}

#####
#      Function Name : add_remote_redo_group
#      Description   : create redo log group for create remote redo log
#                      on standby server
#####

add_redo_remote_group() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    alter database add logfile member
'PRIMARY_REMOTE_REDO/$ORACLE_SID/remote_redo_0$REDO_GROUP.log' reuse to group $REDO_GROUP;
EOF
}

#####
#      Function Name : primary_find_remote_redolog (1)
#      Description   : Function call for find remote_redog on standby database
#####

primary_find_remote_redolog() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    set head off
    set feedback off
        spool $LOG_DIR/primary_find_remote_redolog.log
        select member from v$logfile;
EOF
}

#####
#      Function Name : primary_find_current_redolog (2)
#      Description   : Function call for find current remote_redog on standby database
#####

```

## โปรแกรม base\_command\_primary.sh (ต่อ)

```

primary_find_current_redolog() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    set head off
    set feedback off
    spool $LOG_DIR/primary_find_current_redolog.log
    select member from v$log a, v$logfile b
    where a.status='CURRENT'
        and a.group#=b.group#;
EOF
}
#####
#      Function Name : primary_find_current_redolog (3)
#      Description  : Function call for drop all remote_redog on standby database
#                    after activate standby database to primary database
#####

primary_drop_remote_redolog() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
spool $LOG_DIR/primary_drop_remote_redolog.log
    alter database drop logfile member '$DROP_PRIMARY_REDO';
EOF
}
#####
#      Function Name : primary_redo_infor_sqlplus
#      Description  : Function call for get redolog information on primary database
#####

primary_redo_infor_sqlplus() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF > /dev/null
    @$SQL_DIR/primary_redo_infor.sql
EOF
}
#####
#      Function Name : get_info_tbs_dbf_sqlplus
#      Description  : Function call for get number of tablespace and datafile on
#                    primary database
#####

```

## โปรแกรม base\_command\_primary.sh (ต่อ)

```

get_info_tbs_dbf_sqlplus() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF > /dev/null

    set trims off
    set head off
    set heads off
    set term off
    set heading off
    set echo off
    set feedback off

    spool $ONLINE_DIR/log/primary_tbs.log
    select count(*) from v$tablespace;
    spool off

    spool $ONLINE_DIR/log/primary_dbf.log
    select count(*) from v$datafile;
    spool off

EOF
}

#####
#      Function Name : get_info_primary_sqlplus
#      Description   : Get primary database structure information
#####

get_info_primary_sqlplus() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF > /dev/null
    @$SQL_DIR/primary_information.sql
EOF
}

#####
#      Function Name : get_chk_space_db_sqlplus
#      Description   : check free space on primary database
#####

get_chk_space_db_sqlplus() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF > /dev/null
    @$SQL_DIR/chk_space_db.sql
EOF
}

```

## โปรแกรม base\_command\_primary.sh (ต่อ)

```
#####
#      Function Name : get_datafiles
#      Description  : get datafiles on primary database
#####

get_datafiles () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
select FILE_NAME from dba_data_files;
exit
EOF }

#####
#      Function Name : archivelog_current
#      Description  : flush data on current redolog write to archive log
#####

archivelog_current () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    alter system archive log current;
    exit
EOF
}

#####
#      Function Name : switchlog
#      Description  : force database switch redolog file
#####

switchlog () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    alter system switch logfile;
    exit
EOF
}

#####
#      Function Name : gen_hot_backup_sqlplus
#      Description  : generate command for online backup in $SQL_DIR/hotbackup.sql
#####

gen_hot_backup_sqlplus () {
```



## โปรแกรม base\_command\_primary.sh (ต่อ)

```

$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    @$SQL_DIR/gen_hotbackup.sql
EOF
}
#####
#      Function Name : run_hot_backup_sqlplus
#      Description  : run script $SQL_DIR/hotbackup.sql to rcp datafile
#                   from primary server to standby server
#####
run_hot_backup_sqlplus () {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$PRIMARY_LISTENER <<EOF
    @$SQL_DIR/hotbackup.sql
EOF
}
#####
#      Function Name : max_new_log
#      Description  : find the new archivelog on primary server
#####
max_new_log () {
export MAX_NEW_LOG=`rsh $PRIMARY_SERVER ls -lt $PRIMARY_ARCHIVE | grep dbf | head -1 | awk '{print $9}' | cut -
c3-7 | cut -d . -f1`
#####
#      Function Name : rcp_archive_log
#      Description  : remote copy archive log file from primary database to standby database
#####
rcp_archive_log () {
FLAG_OLD_LOG=`tail -1 $LOG_DIR/FLAG_OLD_LOG.log`
echo $FLAG_OLD_LOG
for FILE in `rsh $PRIMARY_SERVER ls $PRIMARY_ARCHIVE | grep dbf`
do
    SEQ_NUM=`echo $FILE | cut -c3-7 | cut -d . -f1`
    if [ $SEQ_NUM -lt $MAX_NEW_LOG ] && [ $SEQ_NUM -ge $FLAG_OLD_LOG ]
    then
        echo $SEQ_NUM
    fi
done
}
#####

```

## โปรแกรม base\_command\_primary.sh (ต่อ)

```

rcp $PRIMARY_SERVER:$PRIMARY_ARCHIVE/ $FILE $STDBY_ARCHIVE
if [ $? -ne 0 ]
then
    # err_msg "$0: rcp of $FILE to $STDBY_SERVER failed"
    echo "$0: rcp of $FILE to $STDBY_SERVER failed" >> $LOG_DIR/rcp_archivelog.log
    exit 1
else
    echo "`date` rcp of $FILE to $STDBY_SERVER complete" >> $LOG_DIR/rcp_archivelog.log
    # echo "$SEQ_NUM" >> $LOG_DIR/rcp_archivelog.log
fi

fi

done

if [ -z $MAX_NEW_LOG ]          # check null value when primary database
then                            # not available
    # when primary server down
    echo "NULL" > /dev/null
else
    echo $MAX_NEW_LOG >> $LOG_DIR/FLAG_OLD_LOG.log
fi
}

#####
#      Function Name : mail_primary_fault
#      Description  : send mail to database administrator when primary database fault
#
#####

mail_primary_fault(){
    rmail $OWNER_DBA@$STDBY_SERVER << EOF
    Subject : Primary Database Server Not Connect

    Primary Server Cannot Connect .....
EOF
}

```

ตัวอย่างโปรแกรมชุดฟังก์ชันที่ทำการติดต่อกับฐานข้อมูลสำรอง

โปรแกรม base\_command\_standby.sh

```
#!/bin/ksh
#
#####
#
#      Program   : base_command_standby.sh
#      Description : Keep the Function Module with connect to Standby database
#
#####
#####
#      Function Name : find_remote_redolog
#      Description   : for find remote_redog on standby database
#####
find_remote_redolog() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    set head off
    set feedback off
    spool $LOG_DIR/find_remote_redolog.log
    select member from v$logfile;
EOF
}
#####
#      Function Name : find_current_redolog
#      Description   : Function call for find current remote_redog on standby database
#####
find_current_redolog() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    set head off
    set feedback off
    spool $LOG_DIR/find_current_redolog.log
```

## โปรแกรม base\_command\_standby.sh (ต่อ)

```

        select member from v$log a, v$logfile b
        where a.status='CURRENT'
        and a.group#=b.group#;
EOF
}
#####
#   Function Name : find_current_redolog
#   Description   : for drop all remote_redog on standby database
#                 after activate standby database to primary database
#####
drop_remote_redolog() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    spool $LOG_DIR/drop_remote_redolog.log
    alter database drop logfile member '$DROP_REDO';
EOF
}
#####
#   Function Name : restart_standby_sqlplus
#   Description   : shutdown immediate and start standby database to
#                 mount(recover) mode
#####
restart_standby_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    shutdown immediate;
    startup nomount;
    alter database mount standby database;
EOF
}
#####
#   Function Name : start_mount_standby_sqlplus
#   Description   : start standby database to mount(recover) mode
#####
startup_mount_standby_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    startup nomount;
    alter database mount standby database;

```

## โปรแกรม base\_command\_standby.sh (ต่อ)

```

EOF
}
#####
#   Function Name : startup_readonly_sqlplus
#       Description : open database to read only mode
#####
startup_readonly_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    alter database open read only;
EOF
}
#####
#   Function Name : startup_mount_readonly_standby_sqlplus
#       Description :      1. startup database nomount
#                          2. mount standby database
#                          3. open database read only mode
#####
startup_mount_readonly_standby_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    startup nomount;
    alter database mount standby database;
    alter database open read only;
EOF
}
#####
#   Function Name : shutdown_normal_sqlplus
#       Description : shutdown database normal operation
#####
shutdown_normal_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    shutdown ;
EOF
}
}

```

## โปรแกรม base\_command\_standby.sh (ต่อ)

```

#####
#   Function Name : shutdown_immediate_sqlplus
#       Description : shutdown database immediate operation
#####
shutdown_immediate_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    shutdown immediate;
EOF
}
#####
#   Function Name : shutdown_abort_sqlplus
#       Description : shutdown database abort operation
#####
shutdown_abort_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    shutdown abort;
EOF
}
#####
#   Function Name : recover_standby_sqlplus
#       Description : apply archive log on standby database
#####
recover_standby_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    recover automatic standby database until cancel;
EOF }
#####
#   Function Name : standby_switchlog
#       Description : switch redologfile on standby database
#####
standby_switchlog () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    alter system switch logfile;
    exit
EOF
}

```

## โปรแกรม base\_command\_standby.sh (ต่อ)

```

#####
#   Function Name :      1. recover_remote_redolog_sqlplus
#       2. recover_remote_redolog_sqlplus_loop2
#       Description  : recover database by apply on remote redolog
#####
recover_remote_redolog_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    recover standby database;
    $REMOTE_REDOLOG
    recover standby database cancel;
EOF
}
recover_remote_redolog_sqlplus_loop2 () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    recover standby database;
    $REMOTE_REDOLOG_LOOP2
    recover standby database cancel;
EOF
}
#####
#   Function Name : activate_standby_sqlplus
#       Description  : activate standby database when primary database
#                   not available
#####
activate_standby_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    alter database activate standby database;
    shutdown;
    startup;
EOF
}
#####
#   Function Name : startup_normal_sqlplus
#       Description  : startup database normal
#                   : after standby activate to primary database
#####

```

## โปรแกรม base\_command\_standby.sh (ต่อ)

```

startup_normal_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    startup;
EOF
}

#####
#   Function Name : get_dbfiles_standby_sqlplus
#       Description : get name of datafile on standby database
#####

get_dbfiles_standby_sqlplus() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    set heading off
    set feedback off
    spool $LOG_DIR/standby_datafile.log
    select name from v$datafile;
EOF
}

#####
#   Function Name : get_info_standby_sqlplus
#       Description : get information on standby sqlplus
#                   : for display standby database strcuture
#####

get_info_standby_sqlplus() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF > /dev/null
    @$SQL_DIR/standby_information.sql
EOF
}

#####
#   Function Name : menu_manage_standby_status
#       Description : get name and database open mode on standby database
#####

menu_manage_standby_status() {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    set trims off
    set head off
    set heads off

```



## โปรแกรม base\_command\_standby.sh (ต่อ)

```

set term off
set heading off
set echo off
set feedback off
spool $LOG_DIR/standby_status.log
select name,open_mode from v$database;
exit
EOF
}
#####
#   Function Name : gen_standby_backup_sqlplus
#   Description   : generate script for backup on standby database
#####
gen_standby_backup_sqlplus () {
$ORACLE_HOME/bin/sqlplus internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
@$SQL_DIR/gen_standby_backup.sql
EOF
}
#####
#   Function Name :      1. standby_listener_start
#                       2. standby_listener_stop
#                       3. standby_listener_status
#   Description   :    manage the listener on standby database
#                   startup listener,stop listener, check status of listener
#####
standby_listener_start () {
    $ORACLE_HOME/bin/lsnrctl start $STDBY_LISTENER
}
standby_listener_stop () {
    $ORACLE_HOME/bin/lsnrctl stop $STDBY_LISTENER
}
standby_listener_status (){
    $ORACLE_HOME/bin/lsnrctl status $STDBY_LISTENER
}
#####
#   Function Name : standby_max_log_to_backup

```

## โปรแกรม base\_command\_standby.sh (ต่อ)

```

#      Description   : find maxlog number on v$log_history for backup archive
#####
standby_max_log_to_backup() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    set trims off
    set echo off
    set head off
    spool $LOG_DIR/standby_max_log_to_backup.log
    select max(sequence#) from v$log_history;
    exit
EOF
}
#####
#      Function Name : find_need_recover_sqlplus
#      Description   : find datafile on standby database for need recover
#####
file_need_recover_sqlplus() {
$ORACLE_HOME/bin/sqlplus -S internal/$INTERNAL_PWD@$STDBY_LISTENER <<EOF
    spool $LOG_DIR/file_need_recover_sqlplus.log
    @$SQL_DIR/file_need_recovery.sql
EOF
}

```

## ตัวอย่างของโปรแกรมรายการเมนู

### โปรแกรม Main\_Menu\_STDBY.sh

```
#!/bin/ksh
#####
# Program      :      Main_Menu_STDBY.sh
# Description   :      Display List of MainMenu for select the
#                number to process
#####
# run Set environment
. $CONFIG/ENV/env.cfg
. $SCRIPT_DIR/base_command_primary.sh
. $SCRIPT_DIR/base_command_standby.sh
clear
while [ $LOOP_MENU -eq 1 ]
do
tput smso
tput cup 1 8 ;echo "----- Managing The Standby Database -----"
tput rmso
tput cup 5 10 ;echo "Primary Server: $PRIMARY_SERVER "
tput cup 6 10 ;echo "Standby Server: $STDBY_SERVER"
tput cup 5 34 ;echo "[$PRIMARY_IP]"
tput cup 6 34 ;echo "[$STDBY_IP]"
tput cup 3 8 ;echo "Menu : Main Menu Standby Database      "
tput cup 8 8 ;echo " 1 Primary Database Information      "
tput cup 10 8 ;echo " 2 Create/Drop Standby Databae      "
tput cup 11 8 ;echo " 3 Manage Standby Database          "
tput cup 12 8 ;echo " 4 Standby Database Listener        "
tput cup 15 8 ;echo " q Exit from Manager Standby Database "
tput cup 19 8 ; echo "-----"
tput cup 17 8 ; echo "Select Operation to Perform :c"
read Menu1
case $Menu1 in
1) #-- Call Script Primary Databaes Menu to Process --#
. $MENU_DIR/MenuPrimaryDB.sh
clear
;;
2) #-- Call Script Create/Drop Menu Standby Databaes to Process --#
```

## โปรแกรม Main\_Menu\_STDBY.sh (ต่อ)

```

    . $MENU_DIR/MenuCreateSTDBY.sh
    clear
    ;;
3) #-- Call Script Manage the Standby Database to Process --#
    STDBY_GEN=`tail -1 $LOG_DIR/FLAG_STANDBY_CREATED.log`
    if [ $STDBY_GEN = YES ]
    then
        . $MENU_DIR/MenuManageSTDBY.sh
    else
        tput cup 21 5 ;echo ">>> Standby Database does not Create on $STDBY_SERVER <<<"
        tput cup 22 5 ;echo ">>> Please Create Standby Database First <<<"
        tput cup 24 5 ; echo "Please Enter to Continue...\c"
        read Enter
        tput cup 21 5 ; echo " "
        tput cup 22 5 ; echo " "
        tput cup 24 5 ; echo " "
    fi
    ;;
4) #-- Call Script Manage the Standby Database Listener to Process --#
    STDBY_GEN=`tail -1 $LOG_DIR/FLAG_STANDBY_CREATED.log`
    if [ $STDBY_GEN = YES ]
    then
        . $MENU_DIR/MenuListenerSTDBY.sh
    else
        tput cup 21 5 ;echo ">>> Standby Database does not Create on $STDBY_SERVER
<<<<"
        tput cup 22 5 ;echo ">>> Please Create Standby Database First <<<<"
        tput cup 24 5 ; echo "Please Enter to Continue...\c"
        read Enter
        tput cup 21 5 ; echo " "
        tput cup 22 5 ; echo " "
        tput cup 24 5 ; echo " "
    fi
    ;;
q) clear
    export LOOP_MENU=0

```

## โปรแกรม Main\_Menu\_STDBY.sh (ต่อ)

```
cat $MENU_DIR/cu_image2
sleep 1
exit
;;
*) tput cup 21 8; echo "[Enter Error Number] Please Enter to continue..\c"
    read Enter
    tput cup 21 8; echo "
;;
esac
done
```



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ตัวอย่างของโปรแกรมสร้างฐานข้อมูลสำรอง

### โปรแกรม CREATE\_STANDBY\_DB.sh

```
#!/bin/ksh
#
# Program   : CREATE_STANDBY_DB.sh
# Description : Create Standby Database on Standby Server
#
#
# . $SCRIPT_DIR/base_command_primary.sh
# . $SCRIPT_DIR/base_command_standby.sh
while true
do
  STDBY_GEN=`tail -1 $LOG_DIR/FLAG_STANDBY_CREATED.log`
  if [ $STDBY_GEN = YES ]
  then
    tput cup 20 5 ;echo ">>> Standby Database at $STDBY_SERVER already created <<<"
    tput cup 22 5 ; echo "Please Enter to Continue...\c"
    read Enter
    exit
  else
    tput smso
    tput cup 20 7 ;echo "**** Do you want Create Standby Database ****"
    tput rmso
    tput cup 21 7 ;echo "-----"
    tput cup 20 50;echo "(y/n) \c"
    read CREATEDB
    sleep 1
    case $CREATEDB in
      n)
        exit
      ;;
      y)
        ##### Set FLAG 0 to FLAG_OLD_LOG.log#####
        # reset for used to counter tranfer archivelog #

```

## โปรแกรม CREATE\_STANDBY\_DB.sh (ต่อ)

```

echo "0" > $LOG_DIR/FLAG_OLD_LOG.log
#####
##### Make Directory for Oracle engine #####
#####
mkdir -p $STDBY_BDUMP > /dev/null
mkdir -p $STDBY_CDUMP > /dev/null
mkdir -p $STDBY_UDUMP > /dev/null
mkdir -p $STDBY_REMOTE_REDO > /dev/null
#####
cp /dev/null $LOG_DIR/create_standby_messages.log
touch $STDBY_BDUMP/alert_${ORACLE_SID}.log > /dev/null
rcp $PRIMARY_SERVER:$ORACLE_HOME/dbs/init${ORACLE_SID}.ora
$ORACLE_HOME/dbs/init${ORACLE_SID}.ora > /dev/null
tput cup 0 0 ;echo " Creating Standby Database Network ./ Please wait..." | tee -a
$LOG_DIR/create_standby_messages.log
. $SCRIPT_DIR/create_network.sh > /dev/null
tput cup 1 0 ;echo " Creating Standby Database $ORACLE_SID at $STDBY_SERVER./ Please wait..." | tee -a
$LOG_DIR/create_standby_messages.log

#####
##### Call Functions get number for tablespace and datafile on primary database
#####

get_info_tbs_dbf_sqlplus > /dev/null
export PRI_TBS=`grep -v SQL $ONLINE_DIR/log/primary_tbs.log|awk {'print $1'} |tail -1`
export PRI_DBF=`grep -v SQL $ONLINE_DIR/log/primary_dbf.log|awk {'print $1'} |tail -1`
tput cup 2 0; echo " Tablespace = $PRI_TBS Datafile = $PRI_DBF" | tee -
$LOG_DIR/create_standby_messages.log
mkdir -p $STDBY_ARCHIVE

```

## โปรแกรม CREATE\_STANDBY\_DB.sh (ต่อ)

```

tput cup 3 0 ;echo " Make Database Directory Completed.." | tee -$LOG_DIR/create_standby_messages.log
#####

##### run script create remmote redo logfile #####
#####

tput cup 4 0; echo " Create Remote Redolog file member Please wait.." | tee -a
$LOG_DIR/create_standby_messages.log
. $SCRIPT_DIR/create_remote_redolog.sh > /dev/null
#####

##### create Standby Datafile #####
#####

tput cup 5 0; echo " Create Standby Datafiles Please wait.." | tee -a
$LOG_DIR/create_standby_messages.log

. $SCRIPT_DIR/create_standby_dbfiles.sh # DISPLAY Datafile on SCREEN >>
$LOG_DIR/create_database.log
echo " Create Standby Datafiles Completed.." | tee -a $LOG_DIR/create_standby_messages.log

#####

##### create Standby Controlfile #####
#####

. $SCRIPT_DIR/create_standby_controlfile.sh >> $LOG_DIR/create_database.log
echo " Create Standby Control files Completed.." | tee -a $LOG_DIR/create_standby_messages.log
. $SCRIPT_DIR/transfer_archived.sh >> $LOG_DIR/create_database.log
echo " Transfer Archive log files Completed.." | tee -a $LOG_DIR/create_standby_messages.log
standby_listener_start >> $LOG_DIR/create_database.log
echo " Startup Standby Listener Completed.." | tee -a $LOG_DIR/create_standby_messages.log
recover_standby_sqlplus >> $LOG_DIR/create_database.log

```



## โปรแกรม CREATE\_STANDBY\_DB.sh (ต่อ)

```

echo " Apply Archive log Completed.." | tee -a $LOG_DIR/create_standby_messages.log
startup_mount_standby_sqlplus >> $LOG_DIR/create_database.log
echo " Standby Database Startup Mount Completed.. " | tee -a $LOG_DIR/create_standby_messages.log
##### Set FLAG to FLAG_STANDBY_CREATED.log #####
echo "YES" >> $LOG_DIR/FLAG_STANDBY_CREATED.log
date '+Create Date: %d/%m/%y' >> $LOG_DIR/DATE_STANDBY_CREATED.log
#####
##### Set Crontab #####
echo " Set Time to Automatic Transfer Archive log Completed.. " | tee -a
$LOG_DIR/create_standby_messages.log
echo " Set Time to Automatic Change the Physical Database Completed.." | tee -a
$LOG_DIR/create_standby_messages.log
crontab $CRONTAB_DIR/add_crontab.cmd
echo " "
HL=`tput smso`
echo " $HL Create Standby Database $ORACLE_SID at $STDBY_SERVER are Completed"
tput rmso
echo " "
echo " Please Enter to Continue...\c "
read Enter
exit ;;
*)
exit ;;
esac

fi
done

```

ตัวอย่างโปรแกรมการส่งผ่านเพิ่มบันทึกถาวร

โปรแกรม tranfer\_archived\_control.sh

```
#!/bin/ksh
#####
# Program   : create_standby_control.sh
#####
#
# Call Function : switchlog
# Description  : call Function to switch online redolog on primary database
#               before remote copy archivedlog to standby server
#####
archivelog_current
switchlog
switchlog
#####
#
# Call Function : max_new_log
# Description  : call Function to get new archive log on primary database
#
#####
max_new_log
#####
#
# Call Function : rcp_archive_log
# Description  : Call Function to remote copy new archivedlog file
#               from primary server to standby server
#
#####
rcp_archive_log
```

ตัวอย่างโปรแกรมของการตรวจสอบการเพิ่มและลบเพิ่มข้อมูลบนฐานข้อมูลหลัก

โปรแกรม find\_physical\_structure.sh

```
#!/bin/ksh
#####
#   Program Name : find_physical_structure.sh
#   Description   : verify Primary Structure and Apply Archive logfile on standby database
#####
export CONFIG=/u01/kanit/project
. $CONFIG/ENV/env.cfg
. $SCRIPT_DIR/base_command_standby.sh
. $PHYSICAL_DIR/physical_standby.sh
. $PHYSICAL_DIR/send_mail.sh
grep ORA-01110 $STDBY_BDUMP/alert_${ORACLE_SID}.log |awk '{print $5 }'|uniq > $PHYSICAL_DIR/log/dbfile_ORA-01110.log
get_new_datafiles_standby
while read FILE_ORA
do
    NEW_DBFILE_NAME=`grep $FILE_ORA $PHYSICAL_DIR/log/get_new_dbfiles_standby.log`
    if [ $? -ne 0 ]
    then
        echo "NOT NEW FILE" > /dev/null
    else
        export NEW_DBFILE_NAME
        echo "Runing ADD $NEW_DBFILE_NAME to standby Server..." >
        $PHYSICAL_DIR/log/NEW_DBFILE.LOG
        add_new_datafiles_standby
        keep_add_dbfile
        mail_add_datafile
    fi
    sleep 3
done < $PHYSICAL_DIR/log/dbfile_ORA-01110.log
#####
#### drop datafiles from standby server #####
#####
grep "Recovery deleting file" $STDBY_BDUMP/alert_${ORACLE_SID}.log | awk '{print $4}' |cut -f2 -d' ' >
$PHYSICAL_DIR/log/drop_dbfiles_new.log
```

## โปรแกรม find\_physical\_structure.sh (ต่อ)

```
while read DROP_DBFILE
do
    grep $DROP_DBFILE $PHYSICAL_DIR/log/drop_dbfiles_old.log > /dev/null
    if [ $? -ne 0 ]
    then
        rm $DROP_DBFILE
        echo $DROP_DBFILE >> $PHYSICAL_DIR/log/drop_dbfiles_old.log
        keep_remove_dbfile
        mail_drop_datafile
    fi
done < $PHYSICAL_DIR/log/drop_dbfiles_new.log
#####
# Call Function : recover_standby_sqlplus
# Description : if database is not mount then not apply archiveolog
grep MOUNTED $ONLINE_DIR/log/standby_status.log > /dev/null
if [ $? -ne 0 ]
then
    echo "database not mount" > /dev/null
else
    recover_standby_sqlplus > /dev/null
fi
```

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างโปรแกรมที่ใช้ในการตรวจสอบสถานะของฐานข้อมูลหลักและฐานข้อมูลสำรอง

โปรแกรม online\_monitoring.sh

```
#!/bin/ksh
# Program name : online_monitoring.sh
# Descript      : Check the primary and standby standby database status

clear
export CONFIG=/u01/kanit/project
. $ONLINE_DIR/base_online_check.sh
l=1
j=0
while true
do
tput smso
tput cup 1 17; echo "<>"
tput rmso
tput cup 1 21; echo "MANAGING THE STANDBY DATABASE"
tput smso
tput cup 1 52; echo "<>"
tput rmso
tput cup 2 0; echo " "
tput cup 3 2; echo "----- ONLINE STATUS -----"
tput cup 5 2; echo "-----"
tput smso
tput cup 7 3; echo "**** Primary Database ****"
tput cup 8 3; echo "-----"
tput cup 7 41; echo "**** Standby Database ****"
tput cup 8 41; echo "-----"
tput rmso
tput cup 4 22;date '+DATE: %d/%m/%y TIME: %H:%M'
tput cup 30 0
#----- Check Listener Primary DB-----#

/usr/sbin/ping $PRIMARY_SERVER 3 > /dev/null
```

## โปรแกรม online\_monitoring.sh (ต่อ)

```

if [ $? -ne 0 ]
then
    #--- call mail to DBA Manager
    mail_primary_fault_from_onlinemonitoring
    tput cup 10 1; echo "                "
    tput cup 11 1; echo "                "
    tput cup 12 1; echo "                "
    tput smso
    tput cup 10 5; echo " \\\\/\\\/\\\/\\\/\\\/ "
    tput cup 11 5; echo " Primary Server Down "
    tput cup 12 5; echo " //\\\/\\\/\\\/\\\/\\\/ "
    tput rmso
    tput cup 13 1;echo "                "
    tput cup 14 1;echo "                "
    tput cup 15 1;echo "                "
    tput cup 16 1;echo "                "
    tput cup 30 0
else
    $ORACLE_HOME/bin/tnsping $PRIMARY_LISTENER > /dev/null
    if [ $? -ne 0 ]
    then
        tput cup 10 3; echo "                "
        tput cup 10 3; echo "Primary Listener Offline"
        tput cup 11 1;echo "                "
        tput cup 12 1;echo "                "
        tput cup 13 1;echo "                "
        tput cup 14 1;echo "                "
        tput cup 15 1;echo "                "
        tput cup 16 1;echo "                "
        tput cup 30 0
    else
        tput cup 10 3; echo "                "
        tput cup 10 3; echo "Primary Listener Online"
        tput cup 11 1;echo "                "
        tput cup 30 0
        rsh $PRIMARY_SERVER ps -ef|grep $ORACLE_SID |grep pmon > /dev/null
    
```

## โปรแกรม online\_monitoring.sh (ต่อ)

```

if [ $? -ne 0 ]
then
    tput cup 11 1; echo "          "
    tput cup 12 1; echo "          "
    tput cup 13 1; echo "          "
    tput cup 14 1; echo "          "
    tput cup 15 1; echo "          "
    tput cup 15 1; echo "          "
    tput cup 16 1; echo "          "

    tput cup 13 1; echo " PRIMARY DATABASE NOT START "
    tput cup 14 1; echo " ----- "

else
    primary_status > /dev/null
    grep -i $ORACLE_SID $ONLINE_DIR/log/primary_status.log |grep -vi sql >
$ONLINE_DIR/log/primary.log
    read PRI_DBNAME PRI_DBMODE < $ONLINE_DIR/log/primary.log
    ### find primary max archive number from v$loghistory
    primary_max_log_history > /dev/null
    export PRI_MAX_HIS=`grep -v SQL $ONLINE_DIR/log/primary_max_log_history.log|awk {'print $1'}
|tail -1`

```

```
export PRI_TBS=`grep -v SQL $ONLINE_DIR/log/primary_tbs.log|awk
```

```
{'print $1'} |tail -1`
```

```

export PRI_DBF=`grep -v SQL $ONLINE_DIR/log/primary_dbf.log|awk {'print $1'} |tail -1`
### get information from file primary.log
tput cup 11 1; echo "          "
tput cup 12 3;echo "Primary Server: $PRIMARY_SERVER "
tput cup 13 3;echo "Database Name: $PRI_DBNAME "
tput cup 14 3;echo "Database Mode: $PRI_DBMODE "
tput cup 15 3;echo "Tablespace: # $PRI_TBS "
tput cup 16 3;echo "Datafiles: # $PRI_DBF "
tput cup 17 3;echo "Max History Number : $PRI_MAX_HIS "
tput cup 30 0
# primary_max_log_history > /dev/null

fi
fi

```

## โปรแกรม online\_monitoring.sh (ต่อ)

```

fi
#----- Check Listener Standby DB-----#
$ORACLE_HOME/bin/tnsping $STDBY_LISTENER > /dev/null
if [ $? -ne 0 ]
then
    tput cup 10 41; echo "          "
    tput cup 10 41; echo "Standby Listener Offline"
    tput cup 11 39; echo "          "
    tput cup 12 39; echo "          "
    tput cup 13 39; echo "          "
    tput cup 14 39; echo "          "
    tput cup 15 39; echo "          "
    tput cup 16 39; echo "          "
    tput cup 30 0
else
    tput cup 10 41; echo "          "
    tput cup 10 41; echo "Standby Listener Online"
    tput cup 30 0
    ps -ef|grep $ORACLE_SID |grep pmon > /dev/null
    if [ $? -ne 0 ]
    then
        tput cup 11 39; echo "          "
        tput cup 12 39; echo "          "
        tput cup 13 39; echo "          "
        tput cup 14 39; echo "          "
        tput cup 15 39; echo "          "
        tput cup 16 39; echo "          "
        tput cup 13 39; echo " STANDBY DATABASE NOT START"
        tput cup 14 39; echo " -----"
    else
        standby_status > /dev/null
        grep -i $ORACLE_SID $ONLINE_DIR/log/standby_status.log |grep -vi sql >
$ONLINE_DIR/log/standby.log
        read STD_DBNAME STD_DBMODE < $ONLINE_DIR/log/standby.log
        ### find standby max archive number from v$loghistory

```



## โปรแกรม online\_monitoring.sh (ต่อ)

```

standby_max_log_history > /dev/null
export STD_MAX_HIS=`grep -v SQL $ONLINE_DIR/log/standby_max_log_history.log|awk '{print $1}'`
|tail -1`

export STD_TBS=`grep -v SQL $ONLINE_DIR/log/standby_tbs.log|awk '{print $1}' |tail -1`
export STD_DBF=`grep -v SQL $ONLINE_DIR/log/standby_dbf.log|awk '{print $1}' |tail -1`

### get information form file standby.log
tput cup 11 39;echo "                "
tput cup 12 41;echo "Standby Server: $STDBY_SERVER      "
tput cup 13 41;echo "Database Name: $STD_DBNAME          "
tput cup 14 41;echo "Database Mode: $STD_DBMODE         "
tput cup 15 41;echo "Tablespace: # $STD_TBS             "
tput cup 16 41;echo "Datafiles: # $STD_DBF              "
tput cup 17 41;echo "Max History Number : $STD_MAX_HIS  "
tput cup 30 0

fi

fi

tput cup 21 2; echo "-----"
export ARCH_TO_STDBY=`tail -1 $LOG_DIR/rcp_archivelog.log | awk '{print $9}' |cut -f2 -d _ |cut -f1 -d.`
tput cup 19 13;echo "Archived Transfer From $PRIMARY_SERVER --> $STDBY_SERVER : $ARCH_TO_STDBY"
"

tput cup 30 0
cp /dev/null $ONLINE_DIR/log/primary.log
cp /dev/null $ONLINE_DIR/log/standby.log
sleep 3
j=$j+0
i=`expr $i + 1`

done

```

ตัวอย่างโปรแกรมที่ใช้ในการทำสำรองข้อมูล

โปรแกรม Backup\_standby.sh

```
#!/bin/ksh
#####
#      Program Name   : Backup_standby.sh
#      Description    : Backup the standby database for restore to old status
#####

export BACKUP_DATE=`date '+%d%m%y%H%M'`
export BACKUP_DATE_PATH=$BACKUP_DIR/$BACKUP_DATE
export BACKUP_DATA_PATH=$BACKUP_DIR/$BACKUP_DATE/data
export BACKUP_ARCH_PATH=$BACKUP_DIR/$BACKUP_DATE/arch
export BACKUP_CTL_PATH=$BACKUP_DIR/$BACKUP_DATE/ctl
#####
## mkdir -p $BACKUP_DIR > /dev/null
#####
mkdir -p $BACKUP_DATE_PATH
mkdir -p $BACKUP_DATA_PATH
mkdir -p $BACKUP_ARCH_PATH
mkdir -p $BACKUP_CTL_PATH
clear
echo "Backup The Standby Database "
echo "-----"
startup_mount_readonly_standby_sqlplus > /dev/null
#####
##### run script gen command for backup datafile
#####
gen_standby_backup_sqlplus > /dev/null
chmod 744 $SCRIPT_DIR/run_standby_backup.sh
#####
##### run run_standby_backup for backup datafile
#####
. $SCRIPT_DIR/run_standby_backup.sh
grep cp $SCRIPT_DIR/run_standby_backup.sh > $BACKUP_DATE_PATH/backup_patch.log
#####
```

## โปรแกรม Backup\_standby.sh (ต่อ)

```
##### Run Functions Create Standby Controlfile
#####
create_standby_controlfile > /dev/null
rcp $PRIMARY_SERVER:$STDBY_CONTROLFILE $BACKUP_CTL_PATH > /dev/null
#####
##### Call Function find current max applied log history
#####
standby_max_log_to_backup > /dev/null
#####
##### Run Script move archive log to backup directory
#####
. $SCRIPT_DIR/backup_standby_arch.sh > /dev/null
#####
##### shutdown and startup standby database
#####
shutdown_immediate_sqlplus > /dev/null
startup_mount_standby_sqlplus > /dev/null
echo "-----"
echo "Backup The Standby Database Completed"
echo ""
echo " Please Enter to continue and wait a minute..c"
read Enter
```

## ตัวอย่างโปรแกรมที่ใช้ในการกู้คืนฐานข้อมูลสำรอง

### โปรแกรม Gen\_restore\_standb.sh

```
#!/bin/ksh
#####
# Program Name : Gen_restore_standby.sh
#####
# call function shutdown abort before restore
shutdown_abort_sqlplus > /dev/null
#
rm $STDBY_BDUMP/alert_${ORACLE_SID}.log > /dev/null 2>&1
touch $STDBY_BDUMP/alert_${ORACLE_SID}.log > /dev/null 2>&1
#####
##### rm all file for restore to backup time
#####
rm $DB_LOCATION/${ORACLE_SID}/* > /dev/null 2>&1
#####
##### rm old restore standby database script file
#####
rm $SCRIPT_DIR/Restore_Standby_Data_files.sh > /dev/null 2>&1
rm $SCRIPT_DIR/Restore_Standby_ctl_files.sh > /dev/null 2>&1
#
# gen restore Datafile to original location
#
echo "#!/bin/ksh " >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
echo "clear " >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
echo "echo \" \" >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
echo "echo \" \" Restore The Standby Database from Backup at $RESTORE_BACKUP_DATE
$RESTORE_BACKUP_TIME" >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
echo "echo \" \" >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
#
# export RESTORE_BACKUP_DATE=`echo "$ADD/$AMM/$AYY`
# export RESTORE_BACKUP_TIME=`echo "$AHR:$ATM`
for RS_FILE_LS in `ls -1 $BACKUP_DIR/$A/data`
do
    # grep $RS_FILE_LS $BACKUP_DIR/$A/backup_patch.log |awk '{print "cp "
$BACKUP_DIR/"$A"/data/"$RS_FILE_LS " $2}'
```

## โปรแกรม Gen\_restore\_standb.sh (ต่อ)

```

echo " " >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
echo "echo \" Restoring File $RS_FILE_LS...\" " >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
echo "cp $BACKUP_DIR/$A/data/$RS_FILE_LS \c" >> $SCRIPT_DIR/Restore_Standby_Data_files.sh
grep $RS_FILE_LS $BACKUP_DIR/$A/backup_patch.log |awk {'print $2'} >>
$SCRIPT_DIR/Restore_Standby_Data_files.sh
done

#####
#### gen restore controlfile script
#####
BACKUP_CTL_FILE=`ls -1 $BACKUP_DIR/$A/ctl`
echo "#!/bin/ksh" >> $SCRIPT_DIR/Restore_Standby_ctl_files.sh
echo "echo \" Restoring Standby Controlfile\" " >> $SCRIPT_DIR/Restore_Standby_ctl_files.sh
for CTR_FILE_LS_ORG in `awk {'print $3'} $SCRIPT_DIR/rcp_control_file.sh`
do
    echo "cp $BACKUP_DIR/$A/ctl/$BACKUP_CTL_FILE $CTR_FILE_LS_ORG" >>
$SCRIPT_DIR/Restore_Standby_ctl_files.sh
done

echo "echo \" \" >> $SCRIPT_DIR/Restore_Standby_ctl_files.sh
#####
### change permission on script for user can execution
#####
chmod u+x $SCRIPT_DIR/Restore_Standby_Data_files.sh
chmod u+x $SCRIPT_DIR/Restore_Standby_ctl_files.sh
sleep 1
clear
#####
### Restore datafile to original location
#####
. $SCRIPT_DIR/Restore_Standby_Data_files.sh
#####
### Restore standby controlfile to original location
#####
. $SCRIPT_DIR/Restore_Standby_ctl_files.sh

```

## โปรแกรม Gen\_restore\_standb.sh (ต่อ)

```
#####  
### Restore Archive log to original location  
#####  
cp $BACKUP_DIR/$A/arch/* $STDBY_ARCHIVE  
echo "   Restore Archived logfile to $STDBY_ARCHIVE "  
echo " "  
echo " Restoring Standby Database Completed "  
echo " Please Enter to continue \c "  
read ENTER  
  
#####  
##### Enable program automatic time setting on Crontab  
#####  
crontab $CRONTAB_DIR/add_crontab.cmd
```



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างโปรแกรมทำหน้าที่ในการปรับปรุงข้อมูลจากแฟ้มลงบนที่กทำซ้ำ

โปรแกรม apply\_remote\_redolog.sh

```
#!/bin/ksh
#####
#
# Program Name : apply_remote_redolog.sh
# Description : This Script used when The Primary Database Failed
#               then Activate the standby database is primary database
#
#####
tput smso
tput cup 23 5 ;echo "!!! Do you want Apply Remote Redolog !!!"
tput rmso
tput cup 23 45;echo " (y/n) \c"
read Shutdown
case $Shutdown in
  n)  tput cup 23 5; echo " "
      tput cup 24 5; echo " "
  ;;
  y)  #--make dir for oracle create redolog when activate standby db
      mkdir -p /remote_redo/$ORACLE_SID
      ps -ef|grep $ORACLE_SID |grep pmon > /dev/null
      if [ $? -ne 0 ]
      then
          tput cup 24 5; echo "Apply Remote Redolog Plase Wait..."
          ##### Disable Crontab
          crontab $CRONTAB_DIR/remove_crontab.cmd
          startup_mount_standby_sqlplus > /dev/null
          recover_standby_sqlplus > /dev/null
          ##### Point in time Standby Recovery
          . $SCRIPT_DIR/recover_remote_redolog.sh > /dev/null
          tput cup 25 5; echo "Apply Remote Redolog Completed "
          sleep 1
          tput cup 26 5; echo "Please Enter to Continue...\c"
          read Enter
      fi
  ;;
  *)
  ;;
esac
```

## โปรแกรม apply\_remote\_redolog.sh (ต่อ)

```

        tput cup 25 5; echo "
            tput cup 26 5; echo "
            tput cup 27 5; echo "
else
    sleep 1
    menu_manage_standby_status > /dev/null
    grep -i $ORACLE_SID $LOG_DIR/standby_status.log |grep -vi sql >
$LOG_DIR/menu_manage_standby_dbmode.log
    read STD_DBNAME STD_DBMODE1 STD_DBMODE2 <
$LOG_DIR/menu_manage_standby_dbmode.log
    #####
    #### Check Arguments between $STD_DBMODE mixed $STD_DBMODE2
    #####
    if [ $STD_DBMODE1$STD_DBMODE2 = MOUNTED ]
    then
    ### Disable Crontab
    crontab $CRONTAB_DIR/remove_crontab.cmd
tput cup 24 5; echo "Apply Remote Redolog Plase Wait..."
    ##### recover until cancel
    recover_standby_sqlplus > /dev/null
    ##### Point in time Standby Recovery
        $SCRIPT_DIR/recover_remote_redolog.sh > /dev/null
        tput cup 25 5; echo "Apply Remote Redolog Completed "
    sleep 1
        tput cup 26 5; echo "Please Enter to Continue...\c"
    read Enter
        tput cup 25 5; echo "
        tput cup 26 5; echo "
        tput cup 27 5; echo "
    else

```



## โปรแกรม apply\_remote\_redolog.sh (ต่อ)

```

tput cup 24 5; echo "Shutdown Standby Database from Read Only Mode"
shutdown_immediate_sqlplus > /dev/null
startup_mount_standby_sqlplus > /dev/null

#### Disable Crontab
crontab $CRONTAB_DIR/remove_crontab.cmd
tput cup 25 5; echo "Apply Remote Redolog Plase Wait..."
sleep 1
#### recover until cancel
recover_standby_sqlplus > /dev/null
#### Point in time Standby Recovery
SCRIPT_DIR/recover_remote_redolog.sh > /dev/null
tput cup 26 5; echo "Apply Remote Redolog Completed "
sleep 1
tput cup 27 5; echo "Please Enter to Continue...\c"
read Enter
tputcup255;echo"
tput cup 26 5; echo "
tput cup 27 5; echo "
tput cup 28 5; echo "
fi
fi
clear
;;
*)
;;
esac

```

## ตัวอย่างโปรแกรมการเปลี่ยนฐานข้อมูลสำรองให้อยู่ในสถานะเขียนอ่าน

### โปรแกรม Activate\_remote\_redo\_STDBY.sh

```
#!/bin/ksh
# Program Name : Activate_remote_redo_STDBY.sh
# Description : This Script used when The Primary Database Failed
#               then Activate the standby database is primary database
#
#
tput smso
tput cup 23 5;echo "!!! Do you want Activate Standby Database !!!"
tput rmso
tput cup 23 50;echo "(y/n) \c"
read Shutdown
case $Shutdown in
n)
    tput cup 23 5; echo " "
    ;;
y)
    #--make dir for oracle create redolog when activate standby db
    mkdir -p /remote_redo/$ORACLE_SID
    ps -ef|grep $ORACLE_SID |grep pmon > /dev/null
    if [ $? -ne 0 ]
    then
        tput cup 25 5; echo "Activating Standby to Primary Database Please Wait..."
        ##### Disable Crontab
        crontab $CRONTAB_DIR/remove_crontab.cmd
        startup_mount_standby_sqlplus > /dev/null
        ##### recover until cancel
        recover_standby_sqlplus > /dev/null
        ##### Point in time Standby Recovery
        . $SCRIPT_DIR/recover_remote_redolog.sh > /dev/null
        ##### Activate Standby Database Replace the Primary Database
        activate_standby_sqlplus > /dev/null
        ##### Run Script to drop remote redolog on standby database
        . $SCRIPT_DIR/drop_remote_redolog.sh > /dev/null
    standby_switchlog > /dev/null
    tput cup 26 5; echo "Activate Standby Database to Primary Database Completed"
```

## โปรแกรม Activate\_remote\_redo\_STDBY.sh (ต่อ)

```

sleep 1
tput cup 27 5; echo "Please Enter to Continue...\c"
read Enter
tput cup 26 5; echo "
tput cup 27 5; echo "
else
sleep 1
menu_manage_standby_status > /dev/null
grep -i $ORACLE_SID $LOG_DIR/standby_status.log |grep -vi sql >
$LOG_DIR/menu_manage_standby_dbmode.log
read STD_DBNAME STD_DBMODE1 STD_DBMODE2 <
$LOG_DIR/menu_manage_standby_dbmode.log
#####
#### Check Arguments between $STD_DBMODE mixed $STD_DBMODE2
#####
if [ $STD_DBMODE1$STD_DBMODE2 = MOUNTED ]
then
#### Disable Crontab
crontab $CRONTAB_DIR/remove_crontab.cmd
# tput cup 23 5; echo "Activate Standby Database[mounted] to Primary
Database Please Wait..."
tput cup 25 5; echo "Activating Standby to Primary Database Plase Wait..."
#### recover until cancel
recover_standby_sqlplus > /dev/null
#### Point in time Standby Recovery
.$SCRIPT_DIR/recover_remote_redolog.sh >
/dev/null
#### Activate Standby Database Replace the Primary Database
activate_standby_sqlplus > /dev/null
#### Run Script to drop remote redolog on
standby database
.$SCRIPT_DIR/drop_remote_redolog.sh >
/dev/null

```

## โปรแกรม Activate\_remote\_redo\_STDBY.sh (ต่อ)

```

standby_switchlog > /dev/null

tput cup 26 5; echo "Activate Standby Database to Primary Database Completed "
sleep 1

tput cup 27 5; echo "Please Enter to Continue...\c"

read Enter

tput cup 26 5; echo " "
tput cup 27 5; echo " "
tput cup 28 5; echo " "

else

tput cup 25 5; echo "Shutdown Standby Database from Read Only Mode"
sleep 1

shutdown_immediate_sqlplus > /dev/null

startup_mount_standby_sqlplus > /dev/null

#### Disable Crontab
crontab $CRONTAB_DIR/remove_crontab.cmd

tput cup 26 5; echo "Activating Standby to Primary Database Plase Wait..."
sleep 1

#### recover until cancel
recover_standby_sqlplus > /dev/null

#### Point in time Standby Recovery
.$SCRIPT_DIR/recover_remote_redolog.sh > /dev/null

#### Activate Standby Database Replace the Primary Database
activate_standby_sqlplus > /dev/null

#### Run Script to drop remote redolog on stanbdy database
.$SCRIPT_DIR/drop_remote_redolog.sh > /dev/null

```

## โปรแกรม Activate\_remote\_redo\_STDBY.sh (ต่อ)

```
standby_switchlog > /dev/null
    tput cup 27 5; echo "Activate Standby Database to Primary Database
Completed "

    tput cup 28 5; echo "Please Enter to Continue...\c"
read Enter
tput cup 25 5; echo "
tput cup 26 5; echo "
    tput cup 27 5; echo "

tput cup 28 5; echo "
tput cup 29 5; echo "
tput cup 28 5; echo "
fi
fi
clear
;;
*)
;;
esac
```

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ประวัติผู้เขียนวิทยานิพนธ์

นายคณิต เลิศศรีบัณฑิต เกิดเมื่อวันที่ 30 มกราคม 2516 ที่กรุงเทพมหานคร สำเร็จ  
การศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต จากภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
มหาวิทยาลัยหอการค้าไทยในปี พ.ศ. 2538 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ที่  
จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2543 ปัจจุบันทำงานในตำแหน่ง วิศวกรฐานข้อมูล  
บริษัทล่อจิก จำกัด



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย