

# ADVANCED DATABASE SYSTEMS

“Backup and Recovery in Oracle”

เสนอ

รศ.ดร.ศุภมิตร จิตตะยโสธร

จัดทำโดย

นายพัสกร จุลพล รหัสนักศึกษา 54010907

นางสาวสุจิตรา เลิศศศิภากร รหัสนักศึกษา 5401388

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2557

## ORACLE

### 1 Oracle Backup and Recovery Solutions

สำหรับการดำเนินการ Backup และ Recovery บนพื้นฐานของ Physical Backup นั้นมี 2 วิธีที่มีอยู่ คือ

- **Recovery Manager (RMAN)** เป็นเครื่องมือ (มีทั้งแบบ Command-line client และ Enterprise Manager GUI interfaces) ที่ทำงานร่วมกับ Session บน Oracle Server สำหรับการจัดการ Backup และการ Recovery รวมทั้งเก็บข้อมูลประวัติการ Backup ของระบบด้วย
- **user-managed backup and recovery** เป็นการเข้าไปจัดการไฟล์ที่สร้างขึ้นจากฐานข้อมูลของคำสั่งระบบปฏิบัติการกับSQL\*Plus backup และ recovery ที่ทำงานร่วมกัน

ทั้ง 2 วิธีการได้รับการสนับสนุนโดย Oracle Corporation และ ได้รับการบันทึกไว้ว่าเป็น Recovery Manager แต่วิธีการที่ต้องการสำหรับ Backup และ Recovery Database นั้นจะต้องสามารถดำเนินการ Backup และ Recovery ประเภทเดียวกันที่มีอยู่ได้ด้วยวิธี user-managed ที่ง่ายขึ้น มี Interface สำหรับการทำ Backup

### 2 Backup and recovery

#### 2.1 Backup

Backup เป็นการคัดลอกข้อมูลจากฐานข้อมูลเพื่อที่ใช้ในการสร้างข้อมูลใหม่อีกครั้ง โดยการ Backup แบ่งออกดังนี้

##### 2.1.1 physical backup

เป็นการสำรองข้อมูลของ Physical file ที่ใช้ในการบันทึกและการกู้ข้อมูลของฐานข้อมูล เช่น datafile, control files, archived redo logs หรือกล่าวได้อีกอย่างหนึ่งว่า Physical backup เป็นการคัดลอกไฟล์ที่เก็บข้อมูลของฐานข้อมูล โดยการเก็บลง Disk หรือเทปก็ได้

##### 2.1.2 logical backup

ตัว logical backup จะเก็บ logical data ที่นำออกมาจากฐานข้อมูลด้วย oracle export utility และเก็บในรูปของ binary file

## 2.2 recovery

### 2.2.1 restore

เป็นการนำข้อมูลที่ได้จากการ Backup จากที่เก็บไม่ว่าจะเป็น Tape, Disk หรือ Media อื่นๆ กลับมาทำให้สามารถใช้งานได้บน Database Server

### 2.2.2 recovery

เป็นการนำข้อมูลที่ได้จากการ Restore ไปใช้ในการเปลี่ยนแปลง Redo log ของ Database เพื่อทำการกู้คืน Database ทั้งหมดที่จะดำเนินการ Recovery ในแต่ละ Datafile

#### Physical Database Structures Used in Recovering Data

##### 1. Datafiles and Data Blocks

Oracle Database นั้นจะประกอบด้วย logical storage units หรือเรียกว่า Tablespace โดยที่แต่ละ Tablespace จะประกอบด้วย File เรียกว่า Datafile สำหรับ Physical file ที่อยู่ภายใต้ Operating system นั้นจะรวมเป็น Data store โดยที่รูปแบบของ Oracle database ที่ง่ายที่สุดคือมี 1 Tablespace ถูกเก็บไว้ต่อ 1 Datafile

Database ที่จัดการพื้นที่ในการจัดเก็บใน Datafile ของ Database จะถูกเรียกว่า Data block โดยที่ Data block จะเป็นหน่วยที่เล็กที่สุดที่สามารถใช้งานหรือจองได้

##### 2. Redo Logs

Redo log จะทำหน้าที่ในการเก็บความเปลี่ยนแปลงที่ทำกับ Datafile โดยที่การเปลี่ยนแปลงนั้นจะเก็บไว้ใน Online redo log ก่อนที่จะไปเปลี่ยนแปลงที่ Datafiles Oracle database นั้นต้องใช้ Online redo log อย่างน้อย 2 กลุ่มในการทำงาน ในช่วงของ Database ที่มีการเปลี่ยน Online redo log group การจัดเก็บนั้นจะถูกจัดเก็บไว้ใน current online redo log การเก็บ Redo log นั้นจะเป็นส่วนหลักของการวางแผนในการ Backup โดยที่การเก็บ Redo log นั้นจะผ่านการ Archiving ซึ่ง Database สามารถคัดลอก Online redo log group ที่ไม่ได้มีการใช้งาน Archive location บน Disk ที่เก็บรวมไว้โดยจะเรียกว่า Archived redo log ในแต่ละ File จะถูกกล่าวถึงเป็น Archived redo log files หลังจาก that Redo log file นั้นถูกเก็บก็จะสามารถถูก Backup ไปยังที่อื่นบน Disk หรือ Tape สำหรับเก็บในระยะยาวและใช้ในการ Recovery ในอนาคตได้ หาก Database ไม่มี Archived redo log จะทำให้มีตัวเลือกในการ Backup และ Recovery ที่จำกัด คือต้องทำการ Offline ก่อนที่จะทำการ Backup และถ้าต้องการ Restore Database จาก Backup นั้น ข้อมูลที่ Database ได้จะได้อะเฉพาะข้อมูลที่มีอยู่ขณะที่ทำการ Backup ไว้เท่านั้น

### 3. Undo Segments

โดยปกติแล้ว เมื่อข้อมูลใน datafile ที่มีการอัปเดตแล้ว before image ของข้อมูลจะถูกเขียนลงไปใน undo segments ถ้า transaction นั้นถูก roll back ข้อมูล undo นี้ที่เก็บไว้สามารถใช้ในการบันทึกค่าตั้งต้นของ datafile ในการทำ recovery ข้อมูลการ undo จะใช้ในการ undo กับ transaction ที่ยังไม่ commit Datafile จะมีการเปลี่ยนทั้งหมดจากการยืนยันการเปลี่ยนแปลง redo log ไปยัง datafiles ซึ่งความจริงแล้วฐานข้อมูลจะเปิดก่อนยืนยันการทำ Undo

### 4. Control Files

Control file จะเก็บระเบียบของโครงสร้างกายภาพของฐานข้อมูลและสถานะ ข้อมูลที่เก็บมีหลากหลายประเภทที่เก็บบันทึกอยู่ใน control file ที่เกี่ยวข้องกับการสำรองข้อมูลและกู้ข้อมูลของฐานข้อมูล

- Database information (RESETLOGS SCN and time stamp)
- Tablespace and datafile records
- ข้อมูลเกี่ยวกับ redo threads (current online redo log)
- Log records (log sequence numbers, SCN range in each log)
- Record ของ RMAN backup เก่า
- ข้อมูลของ datafile blocks ที่ผิด

## 3 Failure

### 3.1 Database Failure ที่อาจเกิดขึ้นได้

#### 3.1.1 Statement Failure สาเหตุเกิดจาก

- 1) syntax ที่ใช้ไม่ถูกต้อง
- 2) application มีการทำงานผิดพลาด
- 3) User ไม่มีสิทธิ์ส่งคำสั่งนั้น

ความผิดพลาดประเภทนี้ที่ไม่ต้องมี backup และไม่ต้องทำ recovery เพราะไม่มีผลกระทบอะไรกับ database Oracle สามารถจัดการกับปัญหาที่เกิดขึ้นทั้งหมดได้

#### 3.1.2 User Process Failure สาเหตุเกิดจาก

User ที่ disconnect database แบบผิดปกติ เช่น user ทำ transaction ต่างๆอยู่แล้วยังไม่ commit เกิดข้อผิดพลาดขึ้นใน application ทำให้ application ต้องหยุดการทำงานไปทำให้ user ออกจาก database แบบผิดปกติซึ่งถ้าเกิดความผิดปกติประเภทนี้ เราจะได้รู้ได้ทันทีจาก background

process Process Monitor (PMON) โดยสิ่งที่ Oracle จะแก้ไขปัญหานี้คือ Oracle จะ rollback transaction ที่ทำค้างของ User คนนั้น และปลดล็อคคืนให้กับระบบ failure ประเภทนี้ไม่ต้องมี backup และไม่ต้องทำ recovery

### 3.1.3 User Error สาเหตุเกิดจาก

ความผิดพลาดจาก user เอง เช่นการลบ table ผิดซึ่งพบได้บ่อยมาก ใน Oracle Database 10g จึงได้เพิ่ม feature ใหม่เข้ามาคือ flashback ซึ่งสามารถเอา table ที่ถูกลบไปแล้วกลับมาได้ด้วย user คนนั่นเอง แต่ก็ต้องดูด้วยว่าข้อมูลที่จะ flashback กลับมานั้นยังอยู่ใน undo table หรือไม่

### 3.1.4 Media Failure

อาจเกิดขึ้นกับ disk ที่ไม่ว่ากรณีใดๆ เมื่อเกิดขึ้นแล้วต้องมี backup และต้องทำ recovery ให้กับ database

### 3.1.5 Instance Failure สาเหตุอาจเกิดจาก

- ไฟดับ
- Hardware failure
- Background processes Failure
- Emergency shutdown เช่นการส่งคำสั่ง shutdown ABORT และ start FORCE

Failure ประเภทนี้ต้องมี backup เพื่อเอาไว้ recovery Failure ประเภทนี้ Oracle สามารถตรวจเจอได้เอง การที่ Instance Failure มีโอกาสสูงที่จะทำให้ข้อมูลไม่ synchronize กันเพราะเมื่อมีการส่งคำสั่ง update เข้ามามีผลทำให้ข้อมูลมีการเปลี่ยนแปลง ข้อมูลที่เปลี่ยนแปลงจะถูกเก็บลงใน redo log buffer แล้วเมื่อ user ส่ง commit เข้ามาข้อมูลที่เปลี่ยนแปลงจึงถูกเขียนลงใน redo log file แต่ยังไม่เขียนข้อมูลที่เปลี่ยนแปลงลงใน data file มันจะรอจนถึงกระบวนการ checkpoint ให้ background process Database Writer (DBWR) เขียนข้อมูลลง data file ให้เป็นข้อมูลเดียวกันระหว่าง redo log file กับ data file ข้อมูลทั้งสอง file จึง synchronize กัน แต่ถ้าระหว่างนี้ เกิดเหตุการณ์ที่ทำให้เกิด Instance Failure ทำให้ข้อมูลที่ถูกเปลี่ยนแปลงล่าสุดจริงๆอยู่ที่ redo log file แต่ Oracle สามารถตรวจจับความผิดปกติตรงนี้ได้และแก้ไขให้เราแต่ต้องอาศัยให้ dba startup database ให้

## 4 Backup and Recovery with RMAN

การใช้ RMAN สามารถมีหลายวิธีในการ Backup และ Recovery ข้อมูล และบาง Feature ไม่สามารถใช้ได้กับการ Backup และ Recovery แบบ User-managed

- Incremental backups ทำให้ขนาดของข้อมูลที่ Backup มีขนาดเล็ก และ Recovery datafile ได้รวดเร็ว
- Block media recovery ทำให้ Datafile ที่มี Data blocks บางส่วนที่มีปัญหาสามารถทำการซ่อมแซมได้โดยไม่ต้องทำการ offline หรือถูก Restore จากการ Backup
- Unused block compression สามารถข้ามบาง Datafile blocks ที่ไม่มีการใช้งานขณะทำการ Backup ได้
- Binary compression ใช้การ Compression ใน Oracle database server เพื่อทำการลดขนาดของการ Backups
- Encrypted backups ใช้การ Encryption สำหรับเข้ารหัสข้อมูลที่ทำ Backup โดยจัดเก็บในรูปแบบการเข้ารหัส

#### 4.1 Files That RMAN Can Back Up

RMAN สามารถ Backup ทุกไฟล์ Database ที่ต้องการใช้ในการ Recovery เมื่อเกิด Failure ขึ้น โดยที่ชนิดของไฟล์ที่ RMAN สนับสนุนในการ Backup มีดังนี้

- Datafiles และ image copies ของ datafiles
- Control files, และ image copies ของ control files
- Archived redo logs
- The current server parameter file
- Backup pieces, ส่วนอื่นๆ ของ Backups ที่ถูกสร้างโดย RMAN

#### 4.2 รูปแบบของการ backup database

##### 4.2.1 Image copies

รูปแบบนี้จะเหมือนกับการ copy physical file ระดับ OS เช่น data file ต้นฉบับมีขนาดเท่าไร backup ก็จะมีขนาดเท่ากัน ข้อเสียของการ backup ในรูปแบบนี้คือเปลืองเนื้อที่

##### 4.2.2 Backup sets

รูปแบบนี้เป็น format ของ Oracle โดยเฉพาะ Oracle เลือก backup set รูปแบบนี้เป็น default backup set แตกต่างกับ image copies ตรงที่จะ backup เฉพาะ block ที่มีข้อมูลจัดเก็บอยู่เท่านั้น โดย backup set นั้นเราสามารถกำหนดระดับ set ได้ และกำหนดได้อีกว่าจะเก็บลง media ชนิดไหน disk หรือ tape

#### 4.3 วิธีการ backup database

#### 4.3.1 Full backup

คือทุกครั้งที่ทำการ backup จะสร้าง backup ด้วยข้อมูลใหม่ทั้งหมด

#### 4.3.2 Incremental backup

เป็นการ backup เฉพาะข้อมูลส่วนที่ต่าง ส่วนที่มีการเปลี่ยนแปลง แต่แบบนี้จะต้องมี backup ตัวเก่าอยู่ก่อนด้วยเพื่อเอาไว้เปรียบเทียบกับมันต่างกันตรงไหน

##### อัลกอริทึม incremental backup

แต่ละบล็อกข้อมูลใน datafile จะเก็บค่า system change number(SCN) ซึ่ง SCN นั้นจะเป็นค่าที่เปลี่ยนแปลงล่าสุดของบล็อกนั้น เนื่องจากการทำ incremental backup ตัว RMAN จะไปอ่านค่า SCN แต่ละบล็อกข้อมูลในไฟล์ input และเปรียบเทียบกับจุดก่อนหน้าของ incremental backup ถ้า SCN ในบล็อกข้อมูล input มากกว่าหรือเท่ากับจุดก่อนหน้าของ SCN แล้ว RMAN จะคัดลอกบล็อกใหม่

##### Level 0 and level 1 incremental backup

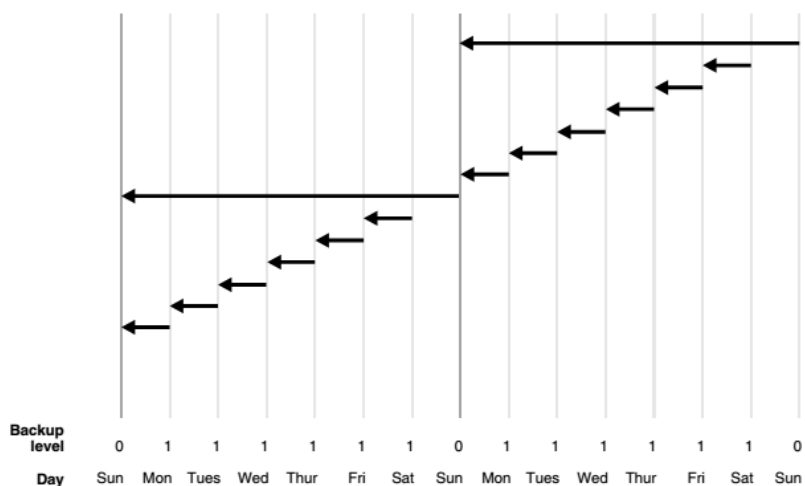
- **Level 0 incremental backup** เป็นการคัดลอกบล็อกที่เก็บข้อมูลทั้งหมด ทำการสำรอง datafile ไปยัง backup set เหมือน full backup ไว้ใช้เป็นตัวเริ่ม recovery ให้กับ level 1

- **Level 1 incremental backup** สามารถทำการสำรองข้อมูลได้ในแต่ละประเภท ดังนี้

##### **Differential backup**

การทำ differential backup จะทำการ backup ส่วนที่เพิ่มต่อ/เปลี่ยนแปลง หลังจาก incremental backup ล่าสุด ที่ level เท่ากับ หรือ ต่ำกว่ามัน (ในที่นี้คือ ต่อจาก level 0 และ level 1 ของวันก่อนหน้า ) สมมติว่า เรามีการใส่ข้อมูลลงใน Block เพิ่มขึ้น วันละ 1 block เมื่อสิ้นวัน โดยคำสั่งของ RMAN ที่ใช้ในการทำ differential backup เป็นดังนี้

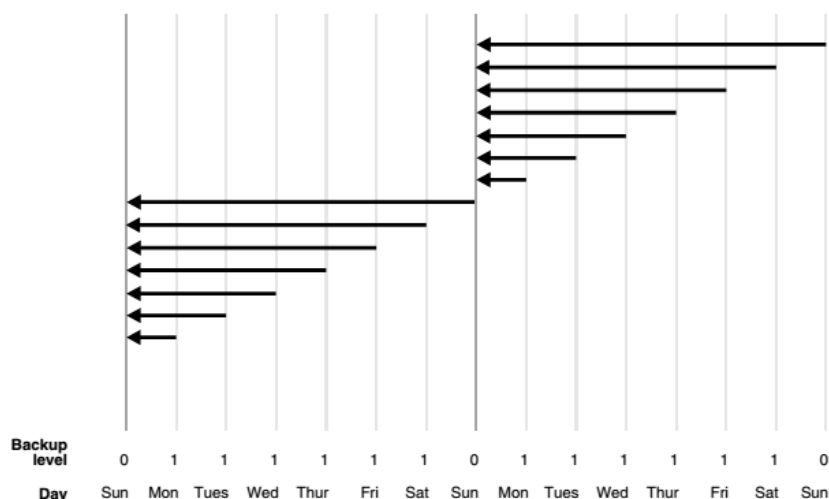
```
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE;
```



### Cumulative backup

ในการ Backup ระดับที่ 1 RMAN จะทำการ Backup ทุกๆ Block ที่มีการใช้งานตั้งแต่การ Incremental backup ครั้งล่าสุดที่ระดับที่ 0 Cumulative incremental backups นั้นจะช่วยลดภาระเมื่อต้องทำการ Restore โดยมั่นใจว่าต้องการแค่ 1 ระดับจาก incremental backup Cumulative backup จะต้องการพื้นที่และเวลามากกว่าการ Backup แบบอื่นๆ เนื่องจากมันจะต้องทำซ้ำจากการ Backup ก่อนหน้าที่ระดับเดียวกัน คำสั่งต่อไปนี้จะเป็นการทำ Incremental backup ระดับที่ 1 ของ Database:

```
BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE; # blocks changed since level
```



## 4.4 การสำรองไฟล์ฐานข้อมูลและ Archived Logs ด้วย RMAN

### 4.4.1 การทำ consistent และ inconsistent

**Consistent backup** เป็นการ backup database เมื่อฐานข้อมูลนั้นอยู่ในสถานะที่ไม่มีการเปลี่ยนแปลง ซึ่งจะได้หลังจากที่มีการปิดฐานข้อมูลนั้นก่อน และการเปลี่ยนทั้งหมดใน redo log ลง datafile

**Inconsistent backup** เป็นการ backup database โดยไม่ต้องทำการ shutdown ฐานข้อมูล ก่อนทำการ backup database เมื่อฐานข้อมูลต้องบันทึกใหม่จาก inconsistent backup ตัว oracle จะต้องดำเนินการกู้ข้อมูลของ media recovery ก่อนจึงจะสามารถเปิดฐานข้อมูลและปรับเปลี่ยนค่าที่มีการเปลี่ยนแปลงจาก redo log ถ้าหากฐานข้อมูลทำงานในโหมด ARCHIVELOG และได้สำรอง



ไฟล์ archived redo log ที่เก็บไว้ใน datafiles ซึ่ง inconsistent backup นั้นเป็นส่วนสำคัญของการสำรองข้อมูลสำหรับฐานข้อมูล

#### 4.4.2 การ backup database ทั้งหมดด้วย RMAN

สามารถดำเนินการ backup database ทั้งหมดด้วยการ mounted หรือเปิดฐานข้อมูล ซึ่งการทำการสำรองข้อมูลทั้งหมดได้จากคำสั่ง BACKUP DATABASE โดยไม่ต้องมีพารามิเตอร์ ดังนี้

```
RMAN> BACKUP DATABASE;
```

#### 4.4.3 การสำรอง Datafile แต่ละตัวและคัดลอก Datafile ด้วย RMAN

คุณสามารถทำการ Backup และคัดลอก Datafile เมื่อ Database มีการ mount หรือเปิดการสำรอง

Datafile โดยที่เมื่อ RMAN ทำการเชื่อมต่อกับ Database ที่ต้องการใช้คำสั่ง BACKUP DATAFILE เพื่อทำการ Backup แต่ละ Datafile โดยที่สามารถทำการระบุ Datafile โดยใช้ชื่อหรือหมายเลขได้ ตัวอย่างนี้ใช้ sbt channel ในการ Backup datafile 1 ถึง 4 และคัดลอก Datafile ไปเก็บไว้ยัง tmp/system01.dbf ไปยัง tape:

```
BACKUP DEVICE TYPE sbt
DATAFILE 1,2,3,4
DATAFILECOPY '/tmp/system01.dbf';
```

ถ้า CONFIGURE CONTROLFILE AUTOBACKUP นั้น On อยู่ RMAN จะเขียน Control file ปัจจุบัน และ SPFILE เพื่อทำการแยกส่วนการ autobackup ไม่อย่างนั้น file นี้จะถูกเพิ่มโดยอัตโนมัติไปใน Backup set ที่ประกอบด้วย Datafile 1

การสำรองข้อมูลสำเนา Datafile :ใช้คำสั่ง BACKUP DATAFILECOPY เพื่อ Backup สำเนาของ Datafile สำเนาของ Datafile นั้นจะอยู่เฉพาะบน Disk เท่านั้นเพื่อทำการ Backup สำเนาของ Datafile :เมื่อทำการเชื่อมต่อกับ Database ที่ต้องการแล้วทำการ run คำสั่ง BACKUP DATAFILECOPY ที่ RMAN prompt ตัวอย่างนี้จะทำการสำรอง Datafile /tmp/system01.dbf ไปยัง tape:

```
BACKUP DEVICE TYPE sbt DATAFILECOPY '/tmp/system01.dbf';
```

#### 4.4.4 การสำรอง Control file ด้วย RMAN

เราสามารถ backup control file ได้เมื่อฐานข้อมูลนั้นถูก mount หรือถูกเปิดไว้ RMAN ใช้ snapshot control file ใช้แน่ใจว่าได้อ่าน consistent version ถ้า CONFIGURE CONTROLFILE

AUTOBACKUP เป็น on (โดยปกติจะเป็น off) ตัว RMAN จะ backup control file และไฟล์พารามิเตอร์เซฟเวอร์ ถ้าไม่ได้ตั้งค่าการ backup อัตโนมัติจะต้อง backup control file เอง ดังนี้

- ใช้คำสั่ง BACKUP CURRENT CONTROLFILE
- เรียกใช้ backup ของ control file ของ backup ตัวอื่น โดยใช้ INCLUDE CURRENT CONTROLFILE ซึ่งเป็นส่วนเสริมของคำสั่ง BACKUP

ในการ backup control file เองนั้นจะไม่เหมือน control file แบบอัตโนมัติ ในการ backup ข้อมูลการ backup ใน RMAN repository ที่อยู่ภายใน RMAN session ปัจจุบันเท่านั้นที่จะมาเป็น control file backup ซึ่งการ backup เองจะไม่สามารถใช้กับการ restore แบบอัตโนมัติได้

#### 4.4.5 การสำรองไฟล์พารามิเตอร์เซฟเวอร์ด้วย RMAN

RMAN จะทำการสำรองไฟล์พารามิเตอร์เซฟเวอร์ล่าสุดแบบอัตโนมัติในบางกรณี ด้วยการใช้คำสั่งดังนี้

BACKUP DEVICE TYPE sbt SPFILE;

SPFILE นั้นเป็นการสำรองข้อมูลที่ instance ใช้ในขณะนั้น ถ้า instance ถูกเริ่มโดยฝั่งผู้ใช้งานเริ่มไฟล์parameter ก่อน RMAN จะไม่ทำการสำรองข้อมูลใดๆ เมื่อใช้คำสั่งดังกล่าว

#### 4.4.6 การสำรอง Archived redo log ด้วย RMAN

Archived redo logs เป็น key ที่ทำให้การ recovery สำเร็จ ในการ backup ปกตินั้น สามารถ backup โดยการใช้คำสั่ง BACKUP ARCHIVELOG หรือ backup log พร้อมกับ datafile และ control file ด้วยคำสั่ง BACKUP ... PLUS ARCHIVELOG. หรือหากต้องการ backup archived log ทั้งหมดได้สามารถทำได้ตามคำสั่งนี้ BACKUP ARCHIVELOG ALL;

การตัดสินใจในการทำงานโหมด ARCHIVELOG และ NOARCHIVELOG

Redo logs ของ database จะทำให้การเก็บการเปลี่ยนแปลงของ Datafile ของ Database นั้นสมบูรณ์ Database สามารถทำงานได้ 2 โหมด คือ ARCHIVELOG หรือ NOARCHIVELOG ในโหมด ARCHIVELOG นั้น Online redo log group ที่ถูกนำมาใช้นั้นจะต้องถูกคัดลอกเพื่อเก็บในหนึ่งหรือหลายที่ก่อนที่จะนำมาใช้งานใหม่ การเก็บ Redo log จะช่วยเก็บทุก Transaction ไว้ใน Log ดังนั้นจะสามารถใช้ในการ Recovery ในภายหลังได้ ในโหมด NOARCHIVELOG นั้น Online redo log group เป็นเพียงการเขียนทับซ้ำเมื่อนำกลับมาใช้งานใหม่ ข้อมูลเกี่ยวกับ Transaction ทั้งหมดที่ถูกเก็บอยู่ใน redo log group จะสูญหายไป

#### ผลกระทบของการทำงานในโหมด NOARCHIVELOG

การทำงาน Database ในโหมด NOARCHIVELOG จะทำให้มีข้อจำกัดในการ Backup และ การวางแผนในการ Recovery

- ไม่สามารถดำเนินการ Online Backup ของ Database ได้ จะต้องทำการปิดการทำงานของ Database ก่อนที่จะสามารถทำการ Backup ในโหมด NOARCHIVELOG ได้
- ไม่สามารถใช้งานเทคนิคการ Recovery ที่ต้องการการทำงานแบบ Archived redo log ได้ รวมถึงการ Recovery ที่สมบูรณ์ และ Point-in-time recovery

หากกำลังทำงานในโหมด NOARCHIVELOG และต้องการที่จะ Recover datafile ที่ได้รับความเสียหายจากการเกิด Disk failure นั้นจะมีทางเลือกหลักสำหรับการ Recovery อยู่ 2 วิธีคือ

- ดรอปทุก Object ที่มีขอบเขตอยู่ใน File ที่ถูกผลกระทบ แล้วทำการดรอป File นั้น ข้อมูลที่เหลืออยู่ของ Database นั้นจะเป็นเหมือนเดิม แต่ทุก Data ที่ได้รับผลกระทบจะหายไป
- Restore ข้อมูลของ Database ทั้งหมดจากการ Backup ล่าสุด และข้อมูลที่มีการเปลี่ยนแปลงไปยัง Database ทั้งหมดตั้งแต่มีการ Backup จะหายไป (การ Recovery ข้อมูลที่มีเปลี่ยนแปลงตั้งแต่มีการ Backup นั้นต้องใช้การดำเนินการของ Media recovery ซึ่งใช้ Archived redo log ในการทำงาน)

#### ผลกระทบของการทำงานในโหมด ARCHIVELOG

สำหรับ Application ส่วนใหญ่จะทำงานในโหมด ARCHIVELOG เป็นที่นิยมกว่าการทำงานในโหมด NOARCHIVELOG เนื่องจาก มีความยืดหยุ่นในทางเลือกของการ Recovery เมื่อเกิด Data loss เช่น การ recovery แบบ Point-in-time ของ database หรือ บาง tablespaces แต่จะมีค่าใช้จ่ายในการทำงานในโหมด ARCHIVELOG ดังนี้

- จะต้องมีส่วนที่สำหรับการจัดเก็บบน Disk ที่จะถูกใช้สำหรับจัดเก็บ Archived redo logs ซึ่งค่อนข้างจะมีขนาดใหญ่ใน Database หากการ Update มีจำนวนมาก
- การจัดเก็บ Archived redo logs จะต้องมีการจัดการ เพื่อจำกัดพื้นที่บน Disk ที่ถูกใช้โดย Archived redo logs โดย Archived redo logs นั้นสามารถถูกย้ายไปจัดเก็บบน Tape เพื่อการจัดเก็บในระยะยาว และ Logs ที่เก่ากว่าที่นานเกินกว่าที่จะนำมาใช้ในการ Recovery นั้นก็จะถูกลบทิ้ง (RMAN ส่วนมากสามารถทำงานโดยอัตโนมัติในการจัดการ Archived redo logs โดยการเก็บตำแหน่งที่เก็บและข้อมูลทั้งหมดของ archived redo logs ซึ่งทำให้ง่ายต่อการเคลื่อนย้าย Archived logs ไปเก็บยัง Tape รวมถึงการระบุ และการลบ redo logs ที่ไม่ต้องการใช้ในการ Recovery แล้ว)
- บางครั้งจะมีเรื่องของ Overhead ในการดำเนินการเข้ามาเกี่ยวข้องด้วย เมื่อมีการ Process ตั้งแต่ ARC0 จนถึง ARCn ซึ่งทำการคัดลอก Online redo log ทั้งหมดไปยังที่จัดเก็บข้อมูล เมื่อต้องการประสิทธิภาพในการทำงานสูงหรือ Disk space มีจำกัด ก็อาจจะดีกว่าหากทำงานในโหมด NOARCHIVELOG ถึงแม้ว่าจะมีข้อจำกัดในการเลือกรูปแบบของการ Recovery ก็ตาม