UNIT 13

백업과 복구

백업 방법으로 데이터베이스 백업과 온라인, 테이블스페이스, 인크리멘탈, 델타 백업 등을 소개합니다. 복구방법으로 크래쉬, 버전, 롤포워드 복구 방법이 소개되며, 백업 이미지 파일과 아카이브 로그 파일을 보관한다면, 손상 직전 시점까지 데이터베이스를 복구할 수 있습니다.

DB2 9.7 운영자 가이드

Administrator Edition

- 데이터베이스 로깅
- 데이터베이스 로그를 위한 구성 변수
- 순환 로깅
- 아카이브 로깅
- USER EXIT
- 복구 기록 파일
- LIST HISTORY 명령어
- PRUNE HISTORY 명령어
- 백업의 종류
- BACKUP DB 명령어
- 백업 이미지 파일
- FULL 백업
- INCREMENTAL 백업
- DELTA 백업
- 테이블스페이스 백업
- 복구의 종류
- RESTART DB 명령어
- RESTORE DB 명령어
- ROLLFORWARD DB 명령어
- 크래쉬 복구
- 버전 복구
- 경로 재지정 복구
- 롤포워드 복구
- INCREMENTAL 복구
- DELTA 복구
- 테이블스페이스의 상태
- 테이블스페이스 복구

TOPIC 1301 데이터베이스 로깅

Point

INSERT, UPDATE, DELETE문이 실행되면, 버퍼풀의 데이터가 변경됩니다. 변경 이전의 값과 변경 이후의 값은 로그 버퍼를 통해서 로그 파일에 기록되어 트랜잭션의 롤백 작업에 사용됩니다. 보관된 로그 파일은 데이터베이스 복구에 사용됩니다.

Tip

데이터베이스 단위로 로그 파일을 운영합니다. 한 데이터베이스에 대해 실행되는 모든 트랜잭션은 공통으로 사용되는 로그 파일에 로깅을 합니다.

Tip

새로운 SQL문이 요청되는 시점부터 COMMIT 또는 ROLLBACK문이 요청되는 시점까지를 한 개의 트랜잭션 또는 UOW(Unit Of Work)라고 합니다.

Tip

변경 이전의 값을 '사전(Before 또는 Old) 이미지' 라고 하며, 변경 이후의 값을 '사후(After 또는 New) 이미지' 라고 합니다.

Tip

버퍼풀에서 이미 변경되었으나, 디스크에 반영되지 않은 데이터가 포함된 페이지를 '더티(Dirty) 페이지'라고 합니다.

Tip

변경된 데이터가 디스크에 반영될 때는 반드시 로그 파일에 먼저 반영되는데, 이것을 '로그 먼저 쓰기(Log Ahead Write)' 라고 합니다.

1 데이터베이스가 활성화되면 고정된 개수의 로그 파일들이 로그 디렉토리에 새롭게 생성됩니다. SQL문의 처리에 필요한 데이터는 테이블스페이스 컨테이너에서 버퍼풀로 로드됩니다. 버퍼풀의 데이터를 변경하면 변경 내용은 로그 버퍼에 기록되어 로그 버퍼가 가득 차거나, 트랜잭션이 COMMIT 되면 로그 파일로 반영됩니다. 버퍼풀의 변경된 페이지가 지정한 비율을 초과하면 변경 내용은 로그 파일에 먼저 반영되고, 디스크의 컨테이너에 반영됩니다. 기본적으로 로그 파일 한 개 분량의 로깅이 완료되면 버퍼풀의 변경 내용이 컨테이너로 반영됩니다.

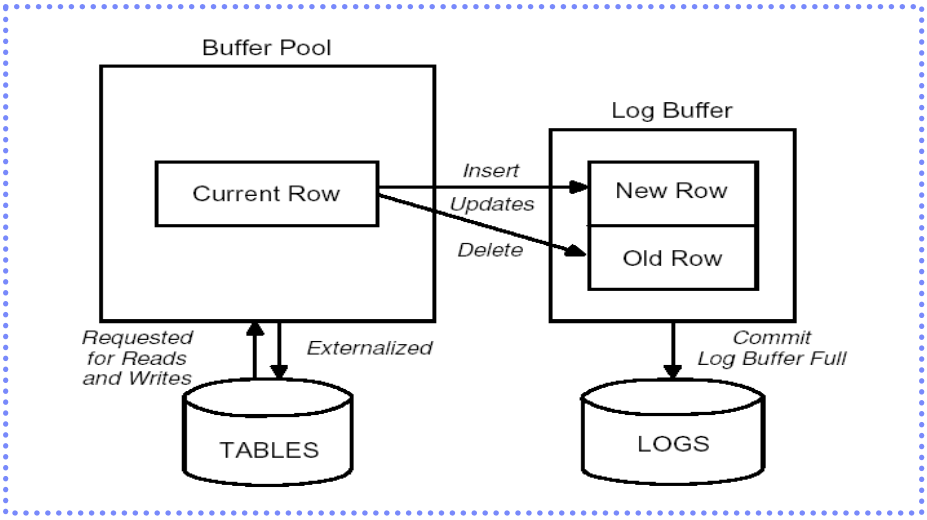


Figure 1301A ••• 버퍼풀과 데이터베이스 로깅

2 SQL문의 종류에 따라 로그 파일에 기록되는 내용이 다릅니다.

SQL문	설 명
SELECT 문	기록되지 않습니다.
INSERT 문	추가된 데이터에 대한 사후 이미지가 기록됩니다.
UPDATE 문	변경된 데이터에 대한 사전, 사후 이미지가 기록됩니다.
DELETE 문	삭제된 데이터에 대한 사전 이미지가 기록됩니다.

3 로그 파일은 용도와 상태에 따라 4가지로 구분됩니다.

로 그	설 명
활성	종료되지 않은 트랜잭션 정보를 포함하고 있는 로그
비활성 로그	종료된 트랜잭션 정보를 포함하고 있는 로그
온라인 아카이브	재사용되지 않고 현재의 활성 로그 디렉토리에 보관된 비활성 로그
오프라인 아카이브	재사용되지 않고 다른 저장 위치에 보관된 비활성 로그

4 로그 파일을 운영하는 방법에 따라 2가지의 로깅 방법으로 구분됩니다.

로 깅 방 법	설 명
순환 로깅	비활성 로그를 재사용하는 방식입니다.
아카이브 로깅	비활성 로그를 아카이브 로그로 보관하는 방식입니다.

TOPIC 13-02 데이터베이스 로그를 위한 구성 변수

Point

로깅과 관련된 환경은 LOGPRIMARY, LOGSECOND, LOGFILSIZ, LOGBUFSZ, MINCOMMIT, NEWLOGPATH, LOGRETAIN, USEREXIT, SOFTMAX 등의 데이터베이스 구성 변수를 이용하여 제어합니다.

Tip
기본 로그 디렉토리는 데이터베이스가 생성된 디렉토리의 SQLOGDIR 서브 디렉토리입니다.

Tip
NEWLOGPATH 에서 지정하는 경로명은 미리 생성되어야 합니다.변경 후에 데이터베이스가 재활성화되면 NEWLOGPATH에 지정했던 값이 <로그 파일에 대한 경로>에 표시되고, NEWLOGPATH 에는 표시되지 않습니다.

Tip
LOGRETAIN과 USEREXIT, LOGARCHMETH1 구성 변수가 모두 OFF인 경우는 순환 로깅이 사용됩니다.

Tip
LOGRETAIN 변수를 ON으로 설정하면 LOGARCHMETH1 변수가 LOGRETAIN 으로 자동적으로 설정됩니다.

Tip
USEREXIT 변수를 ON으로 설정하면 LOGARCHMETH1 변수가 USEREXIT 으로 자동적으로 설정됩니다.

Tip
SOFTMAX는 로그 파일의 크기를 이용한 백분율로 표시됩니다.

Tip
MINCOMMIT 변수의 기본값은 1 이므로 COMMIT 이 요청될 때마다 로그 버퍼의 내용이 로그 파일로 반영됩니다.

Tip
logsecond 변수의 값이 -1 이면, logprimary <= 256 입니다.
logsecond 의 값이 -1이 아니면, (logprimary + logsecond) <= 256입니다

1 데이터베이스 로그와 관련된 주요 데이터베이스 구성 변수는 다음과 같습니다.

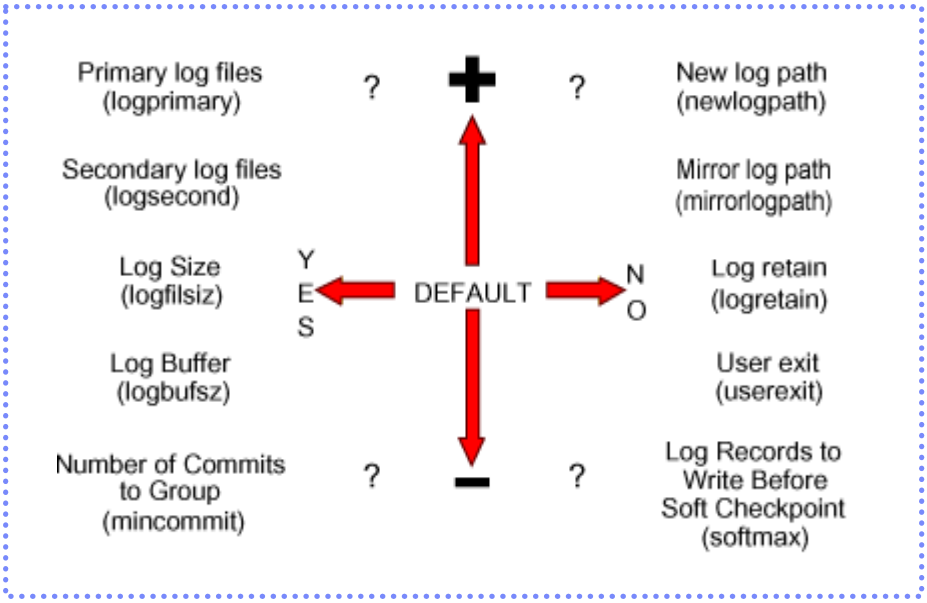


Figure 1302A • • • 로깅과 연관된 데이터베이스 구성 변수

2 UPDATA DB CFG 명령어로 변경하며, 데이터베이스가 재활성화가 필요할 수 있습니다.

구성변수	설명
LOGPRIMARY	1차 로그(Primary Log)의 개수입니다. 데이터베이스의 변경 내용을 기록하며, 데이터베이스가 활성화될 때 미리 할당됩니다.
LOGSECOND	2차 로그(Secondary Log)의 개수입니다. 1차 로그가 모두 사용되면 필요에 따라 한 개씩 할당됩니다.
LOGFILSIZ	로그 파일 한 개의 크기로 4K 페이지 단위로 설정합니다.
LOGBUFSZ	로그 파일을 위한 버퍼의 크기로 4K 페이지 단위로 설정합니다.
NEWLOGPATH	활성 로그 파일을 저장할 경로명을 지정합니다.
LOGRETAIN	비활성 로그를 아카이브 로그로 보관합니다. ON으로 설정하면, 아카이브 로깅 방식이 사용됩니다.
USEREXIT	사용자가 제공한 프로그램을 이용하여 아카이브 로그를 이동시킵니다. ON으로 설정하면, 아카이브 로깅 방식이 사용됩니다.
LOGARCHMETH1, LOGARCHMETH2	비활성 로그를 현재의 로그 디렉토리가 아닌 위치에 아카이브하도록 합니다. 기본값은 OFF이며, LOGRETAIN, USEREXIT, DISK, TSM, VENDER 중에서 설정합니다. DISK를 이용한 아카이브 경로명은 <DISK:절대경로명> 의 형태로 설정합니다.
MINCOMMIT	지정한 개수의 COMMIT 요청을 수행할 때까지 로그 파일에 기록하는 것을 지연합니다. 최대 지연 시간은 1초입니다.
SOFTMAX	응급 복구시에 사용되는 활성 로그의 양을 지정합니다.
CHNGPGS_THRESH	버퍼풀에서 디스크로 반영되지 않은 변경된 페이지의 최대 비율을 설정합니다.

TOPIC 13 03 순환 로깅

Point

고정된 개수의 로그 파일을 순환하며 재사용하는 방식입니다. 기본적으로 적용되는 로깅 방식으로 운영하는 로그 파일의 최대 개수는 LOGPRIMARY 와 LOGSECOND 데이터베이스 구성 변수에 의해 결정됩니다.

Tip

로그 파일명은 S0000000.LOG 부터 시작합니다.

Tip

활성 로그는 응급 복구시에 사용됩니다. 비활성 로그는 재사용되므로 데이터베이스 복구에 이용될 수 없습니다.

1 데이터베이스가 활성화되면 LOGPRIMARY 데이터베이스 구성 변수에 설정된 개수의 로그 파일들이 로그 디렉토리에 새롭게 생성됩니다. 생성된 로그 파일의 번호는 0 번부터 시작합니다.

```
$ db2 get db cfg for <DB명> | grep LOGPRIMARY
$ db2 get db cfg for <DB명> | grep LOGSECOND
```

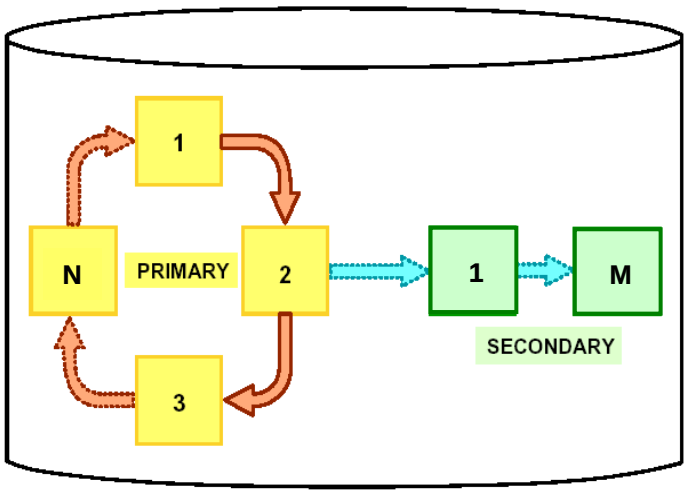


Figure 1303A 비활성 로그를 재사용하는 순환 로깅

Tip

LOGRETAIN과 USEREXIT 변수를 모두 OFF로 설정해도 순환 로깅 모드로 전환됩니다. 이전 버전과의 호환을 위해 유지하는 구성 변수이므로 사용하지 않을 것을 권장합니다.

2 update db cfg 명령어로 LOGARCHMETH1 구성 변수의 값을 OFF로 설정합니다.

```
$ db2 update db cfg for <DB명> using LOGARCHMETH1 OFF
```

Tip

LOGSECOND의 값을 -1로 설정하면 2차 로그가 무제한으로 할당됩니다. 트랜잭션의 개수나 크기에는 제한이 없어지지만, 응급 복구시에 많은 시간이 소모될 수 있으므로 주의가 필요합니다.

3 UOW가 시작되면 0번 로그는 활성 로그가 되어 기록을 시작하고, 0번을 모두 채우면 1번이 활성 로그가 되어 0번 로그에 이어서 기록을 시작합니다.

Tip

임시로 사용되었던 2차 로그는 비활성 로그가 되면 엔진에 의해 비동기적으로 제거됩니다.

4 0번 로그에 포함된 트랜잭션들이 모두 종료되었다면, 0번 로그는 비활성 로그가 되어서 다음 번에 재사용됩니다. 0번 로그에 아직 종료되지 않은 트랜잭션이 포함되어 있다면, 0번 로그와 1번 로그는 모두 활성 로그가 됩니다.

Tip

활성 로그가 LOGPRIMARY 와 LOGSECOND 의 합을 초과하면 'LOG FULL' 상태라고 합니다.

5 유사한 방법으로 현재 N번 로그가 활성 로그이고, 더 이상 기록할 공간이 없으면 다음 번호의 로그 파일을 요청합니다. 이 시점에서 0번 로그가 비활성 로그였다면, N번 로그에 뒤를 이어 재사용됩니다. 0번 로그가 활성 로그였다면, LOGSECOND에서 지정한 로그가 사용되어 전체 활성 로그의 수는 N+1 이 됩니다.

6 유사한 방법으로 LOGSECOND 가 지정한 M개의 로그가 모두 활성 로그가 된 상태에서 더 이상 기록할 공간이 없으면, 로그 공간 부족 현상이 발생하면서, 진행 중이던 트랜잭션은 모두 롤백됩니다. 즉, 활성 로그의 최대 개수는 LOGPRIMARY 구성 변수와 LOGSECOMD 구성 변수의 합을 초과할 수 없습니다.

7 LOGSECOND로 생성된 M개의 로그들이 비활성 로그가 되면, 비동기식으로 엔진에 의해 제거되고, 다시 LOGPRIMARY 의 값인 N개의 로그가 남게 됩니다.

TOPIC 1304 아카이브 로깅

Point

한 개의 로그 파일이 비활성화되면, 새로운 로그 파일을 생성하여 항상 고정된 개수의 로그 파일을 운영하는 방식입니다. 데이터베이스 구성 변수인 LOGARCHMETH1 의 값을 LOGRETAIN 으로 설정하면 됩니다.

Tip

보관된 로그 파일은 롤포워드 복구시에 사용됩니다. 보관된 로그 파일 중에서 특정 시점 이전의 불필요한 로그 파일은 제거해도 됩니다.

Tip

아카이브 로깅에서 순환 로깅으로 전환하면 로그 파일의 번호는 0 부터 다시 시작됩니다.

Tip

순환 로깅에서 아카이브 로깅으로 전환되면 복구시에 사용될 최초의 데이터베이스 백업이 필요하게 되므로, 데이터베이스는 백업 보류 'Backup Pending' 상태가 됩니다. backup db 명령어로 데이터베이스 백업을 실행하면 됩니다.

Tip

LOGRETAIN 또는 USEREXIT 변수를 ON으로 설정해도 아카이브 로깅 모드로 전환됩니다. 이전 버전과의 호환을 위해 유지하는 구성 변수이므로 사용하지 않을 것을 권장합니다.

Tip

비활성 로그는 기본적으로 활성 로그와 동일한 로그 디렉토리에 보관되는데, 이러한 로그를 '온라인 보관 로그' 라고 합니다.

Tip

사용자의 수작업 또는 user exit 프로그램에 의해 다른 저장 공간으로 이동된 비활성 로그 '오프라인 보관 로그' 라고 합니다.

Tip

활성 로그가 LOGPRIMARY 와 LOGSECOND 의 합을 초과하면 'LOG FULL' 상태라고 합니다.

1. 데이터베이스가 활성화되면 LOGPRIMARY 데이터베이스 구성 변수에 설정된 개수의 로그 파일들이 로그 디렉토리에 새롭게 생성됩니다. 생성된 로그 파일의 번호는 최근에 사용했던 로그 파일의 다음 번호입니다.

```
$ db2 get db cfg for <DB명> | grep "처음에 사용되는 로그 파일"
```

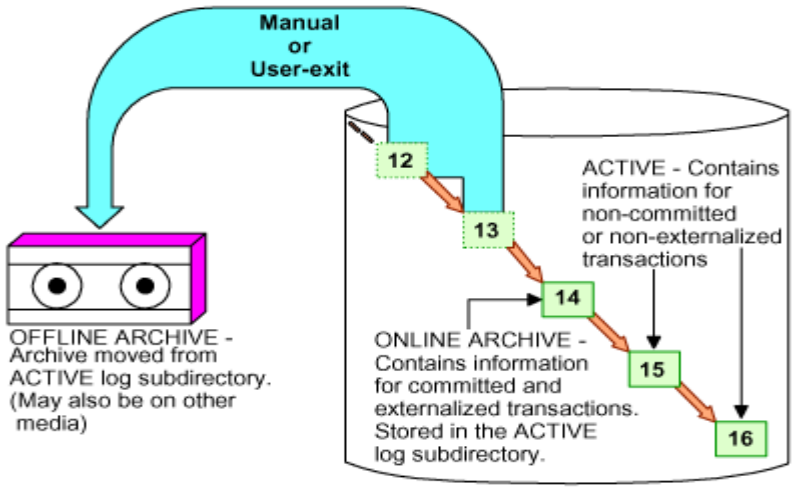


Figure 1304A ··· 온라인 아카이브 로그와 오프라인 아카이브 로그

2. update db cfg 명령어로 LOGARCHMETH1 구성 변수의 값을 LOGRETAIN 으로 설정합니다.

```
$ db2 update db cfg for <DB명> using LOGARCHMETH1 LOGRETAIN
```

3. UOW가 시작되면 0번 로그는 활성 로그가 되어 기록을 시작하고, 0번을 모두 채우면 1번이 활성 로그가 되어 0번 로그에 이어서 기록을 시작합니다.

4. 0번 로그에 포함된 트랜잭션들이 모두 종료되었다면, 0번 로그는 비활성 로그가 되어 더 이상 사용되지 않고 보관됩니다. 새로운 N+1 번의 로그 파일을 생성하여 전체 로그 파일의 개수는 N개를 유지하도록 합니다. 0번 로그에 아직 종료되지 않은 트랜잭션이 포함되어 있다면, 0번 로그와 1번 로그는 모두 활성 로그가 됩고, N+1 번 로그는 생성되지 않습니다.

5. 현재 N개의 로그가 활성 로그이고, 더 이상 기록할 공간이 없으면 LOGSECOND에서 지정한 로그가 사용되어 전체 활성 로그의 수는 N+1 이 됩니다.

6. 유사한 방법으로 LOGSECOND 가 지정한 M개의 로그가 모두 활성 로그가 된 상태에서 더 이상 기록할 공간이 없으면, 로그 공간 부족 현상이 발생하면서, 진행 중이던 트랜잭션은 모두 롤백됩니다. 즉, 활성 로그의 최대 개수는 LOGPRIMARY 구성 변수와 LOGSECOMD 구성 변수의 합을 초과할 수 없습니다.

7. LOGSECOND로 생성된 M개의 로그들이 비활성 로그가 되면, 제거되지 않고 그대로 보관됩니다. 다시 LOGPRIMARY 의 값인 N개의 로그가 남게 됩니다.

TOPIC 13 05 USER EXIT

Point

아카이브 로깅 방식을 운영할 때, 비활성 로그는 기본적으로 활성 로그와 동일한 디렉토리에 보관됩니다. LOGARCHMETH1 구성 변수를 설정하여 비활성 로그를 지정한 저장 영역으로 이동시킬 수 있습니다.

Tip

아카이브 디렉토리는 미리 생성되어 있지 않으면, SQL5099N 오류가 반환됩니다.

Tip

LOGARCHMETH1 변수는 TSM 서버와 VENDER API를 이용한 USEREXIT도 지원합니다.

Tip

deactivate db 명령어를 실행하기 전에 데이터베이스에 접속된 모든 응용프로그램은 종료되어야 합니다.

1. update db cfg 명령어로 LOGARCHMETH1 구성 변수의 값을 DISK:<경로명> 으로 설정합니다. <경로명>은 절대 경로명으로 표현합니다.

```
$ db2 update db cfg for <DB명> using LOGARCHMETH1 DISK:<경로명>
```

2. get db cfg 명령어의 show detail 옵션으로 변경값이 반영되었는지 확인합니다.

```
$ db2 connect to <DB명>
$ db2 get db cfg for <DB명> SHOW DETAIL | grep  LOGARCHMETH1
   첫 번째 로그 아카이브 메소드            (LOGARCHMETH1) = OFF
```

3. update db cfg 명령어를 실행한 후에 SQL1363W 경고를 받았다면, 변경값이 동적으로 반영되지 않았으므로 데이터베이스를 재활성화합니다.

```
$ db2 terminate
$ db2 deactivate db <DB명>
```

4. activate db 명령어로 데이터베이스를 재활성화하면 SQL1116N 오류가 반환됩니다.

```
$ db2 get db cfg for <DB명>  | grep  LOGARCHMETH1
   첫 번째 로그 아카이브 메소드            (LOGARCHMETH1) = DISK:<경로명>
$ db2 activate db <DB명>
   SQL1116N  BACKUP PENDING 때문에 데이터베이스 "<DB명>"에 연결하거나 활
   성화할 수 없습니다.  SQLSTATE=57019
```

5. LOGARCHMETH1 의 값이 OFF 가 아닌 경우는 아카이브로깅 모드로 설정되므로 최초의 오프라인 모드의 데이터베이스 FULL 백업이 필요합니다. backup db 명령어를 실행하여 데이터베이스의 '백업 보류' 상태를 해제하고, 다시 데이터베이스를 재활성화합니다.

```
$ db2 backup db <DB명>
$ db2 activate db <DB명>
```

6. 다양한 트랜잭션을 실행시키면 로그 파일에 변경 내용이 기록되고, 비활성 로그가 되면 자동적으로 LOGARCHMETH1 변수가 지정한 <경로명>로 이동됩니다.

```
$ ls –R <경로명>
S0000000.LOG  S0000001.LOG  S0000002.LOG
```

7. 활성 로그 파일의 시작 번호를 확인하면 마지막 아카이브 로그의 번호를 알 수 있습니다.

```
$ db2 get db cfg for <DB명> | grep "처음에 사용되는 로그 파일"
   처음에 사용되는 로그 파일                    = S0000002.LOG
```

TOPIC 1306 복구 기록 파일

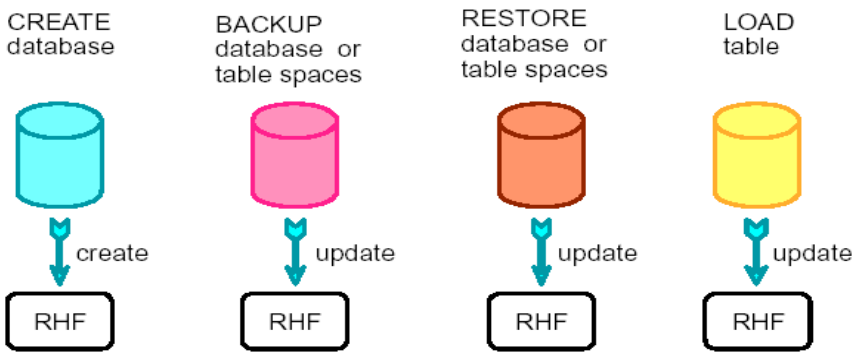
Point

데이터베이스를 생성하면 복구 실행 기록 파일이 생성되어, 데이터베이스에 대한 BACKUP, RESTORE, ROLLFORWARD, LOAD 등의 작업에 대한 정보가 관리됩니다.

Tip

복구 기록 파일이 손상되면, restore db 명령어로 복구할 수 있습니다.

1 create db 명령어로 데이터베이스를 생성하면 복구 기록 파일이 생성됩니다.



- The part of the database that was copied and how
- The time the copy was made
- The location of the copy
- The last time a restore was done

→ `db2 LIST HISTORY ALL FOR MUSICDB`

Figure 1306A ••• 복구 기록 파일의 생성과 갱신

2 복구 기록 파일은 다음과 같은 명령어가 실행될 때 갱신됩니다.

대 상	명 령 어
데이터베이스	BACKUP DATABASE RESTORE DATABASE ROLLFORWARD DATABASE
테이블스페이스	BACKUP DATABASE RESTORE DATABASE ROLLFORWARD DATABASE CREATE TABLESPACE ALTER TABLESPACE QUIESCE TABLESPACE RENAME TABLESPACE DROP TABLESPACE
테이블	LOAD REORG TABLE REORG INDEXES DROP TABLE
로그	ARCHIVE LOG 로그 파일이 아카이브 로그로 저장될 때

TOPIC 1307 LIST HISTORY 명령어

Point

LIST HISTORY 명령어로 복구 실행 기록 파일에서 작업 유형별로 기록을 확인할 수 있습니다. SINCE 옵션을 이용하면 특정 시점 이후의 기록을 확인할 수 있으며, CONTAINING 옵션을 이용하면 지정한 오브젝트와 관련된 기록만 추출할 수 있습니다.

Tip

restore db 명령어를 실행한 기록을 BACKUP 옵션으로 backup db 명령어를 실행한 기록과 함께 확인합니다.

Tip

drop tablespace 문을 실행한 기록을 CREATE TABLESPACE 옵션으로 create tablespace문을 실행한 기록과 함께 확인합니다.

Tip

조작 종류 (Op)는 다음과 같습니다.

```
A - Create table space
B - Backup
C - Load copy
D - Dropped table
F - Roll forward
G - Reorganize table
L - Load
N - Rename table space
O - Drop table space
Q - Quiesce
R - Restore
T - Alter table space
U - Unload
X - Archive log
```

Tip

조작 대상 (Obj)은 다음과 같습니다.

```
D - Database
P - Tablespace
T - Table
I - Index
```

Tip

조작 유형은 다음과 같습니다.

```
Backup types:
  F - Offline
  N - Online
  I - Incremental offline
  O - Incremental online
  D - Delta offline
  E - Delta online

Rollforward types:
  E - End of logs
  P - Point in time

Load types:
  I - Insert
  R - Replace

Alter table space types:
  C - Add containers
  R - Rebalance

Quiesce types:
  S - Quiesce share
  U - Quiesce update
  X - Quiesce exclusive
  Z - Quiesce reset
```

1 list history 명령어를 이용하여 복구 기록 파일의 내용을 조회할 수 있습니다.

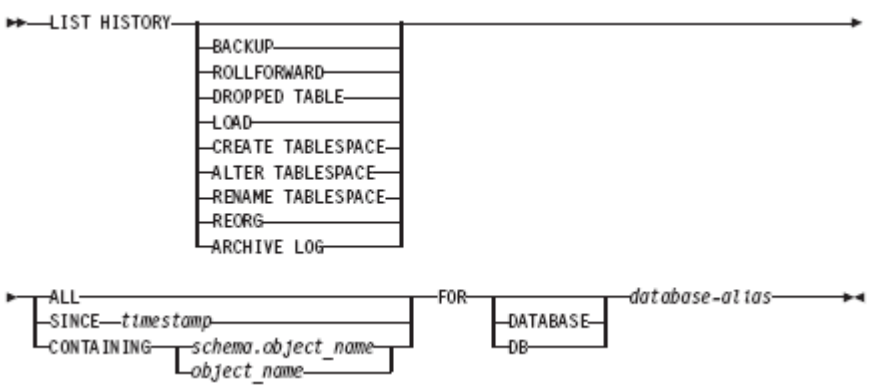


Figure 1307A ··· LIST HISTORY 명령어

2 list history 명령어를 이용하여 복구 기록 파일의 내용을 조회할 수 있습니다.

```
$ db2 list history ALL for <DB명>
$ db2 list history BACKUP all for <DB명>
$ db2 list history ROLLFORWARD all for <DB명>
$ db2 list history backup SINCE <시간소인>  for <DB명>
$ db2 list history rollforward CONTAINING <스키마명>.<테이블명> for <DB명>
$ db2 list history REORG all for sample
```

3 복구 기록 파일이 저장하는 정보는 다음과 같습니다.

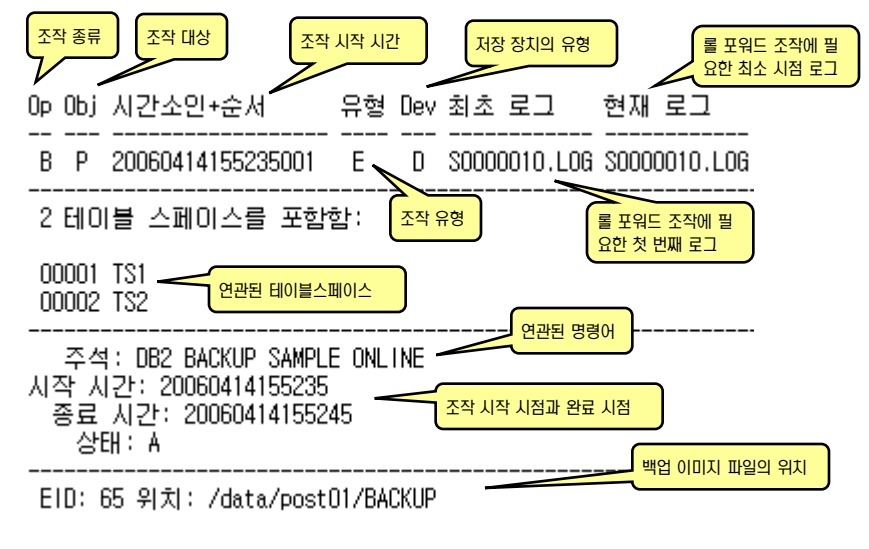


Figure 1307B ··· backup db 명령어의 실행에 관한 기록 정보

TOPIC 1308 PRUNE HISTORY 명령어

Point

PRUNE HISTORY 명령어로 복구 실행 기록 파일에서 특정 <시간 소인> 이전의 기록을 제거할 수 있습니다. 복구 실행 기록 파일의 기본 보관 주기는 366일이며, REC_HIS_RETENTN 데이터베이스 구성 변수로 조절할 수 있습니다.

1 prune history 명령을 이용하여 복구 기록 파일의 내용 중에서 지정한 시점 이전의 기록을 삭제할 수 있습니다.

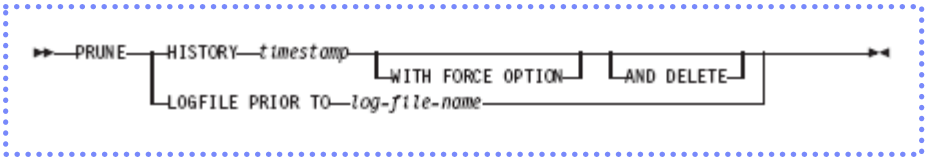


Figure 1308A • • • PRUNE HISTORY 명령어

2 prune history 명령을 이용하여 복구 기록 파일의 내용 중에서 지정한 시점 이전의 기록을 삭제할 수 있습니다. <시간소인>은 <yyyymmddhhmmss> 형식으로 표시합니다. <yyyy> 이상을 명시할 수 있습니다.

```
$ db2 prune history <yyyy>
$ db2 prune history <yyyymmdd>
$ db2 prune history <yyyymmddhhmmss>
```

3 WITH FORCE 옵션을 이용하면 복구 기록 파일의 모든 항목을 완전히 제거합니다. 옵션을 지정하지 않으면, 최근의 FULL 백업 정보와 연관되는 복구 작업의 기록은 유지됩니다.

```
$ db2 prune history <yyyymmddhhmmss> WITH FORCE OPTION
```

4 AND DELETE 옵션을 이용하면 복구 기록 파일에서 항목이 삭제되고, 해당 항목과 관련된 아카이브 로그 파일도 물리적으로 제거됩니다. USEREXIT 프로그램을 사용하여 아카이브 로그 파일을 관리하는 경우에는 지원되지 않습니다.

```
$ db2 prune history <yyyymmddhhmmss> AND DELETE
```

5 prune logfile prior to 명령어로 지정한 번호보다 작은 번호의 로그 파일을 모드 제거합니다. LOGRETAIN 변수가 ON으로 설정되어야 합니다. 로그 파일명은 S0000100.LOG 와 같은 형식으로 표시합니다.

```
$ get db cfg for <DB명> | grep LOGARCHMETH1
첫 번째 로그 아카이브 메소드            (LOGARCHMETH1) = LOGRETAIN
$ db2 prune logfile prior to <로그파일명>
```

6 update dbm cfg 명령어를 이용하여 REC_HIS_RETENTN 데이터베이스 구성 변수를 변경합니다. <기간>은 1일 단위로 지정합니다. 기본값은 366일입니다.

```
$ db2 get db cfg for <DB명> | grep REC_HIS_RETENTN
  복구 실행기록 보유(일)              (REC_HIS_RETENTN) = 366
$ db2 update db cfg for <DB명> using REC_HIS_RETENTN <기간>
```

Tip

update db cfg 명령어를 실행한 후에 SQL1363W 경고를 받았다면, 변경값이 동적으로 반영되지 않았으므로 데이터베이스를 재활성화합니다.

TOPIC 1309 백업의 종류

Point

BACKUP DB 명령어를 이용하여 특정한 데이터베이스의 모든 데이터와 제어 정보를 저장한 이미지 파일을 생성할 수 있습니다. ONLINE/OFFLINE 백업, FULL/TABLESPACE 백업, FULL/INCREMENTAL/DELTA 백업으로 분류됩니다.

Tip

백업 이미지는 한 개 이상의 파일로 생성되어 디스크, 테이프, TSM 등으로 저장됩니다.

Tip

백업 이미지에는 해당 데이터베이스에 대한 모든 정보와 데이터가 저장되므로, 새로운 데이터베이스를 생성하는데 이용되기도 합니다.

1. SYSADM, SYSCTRL, SYSMAINT 권한의 사용자가 원격 또는 지역 데이터베이스에 대해 백업을 실행하면 백업 이미지 파일이 생성됩니다.

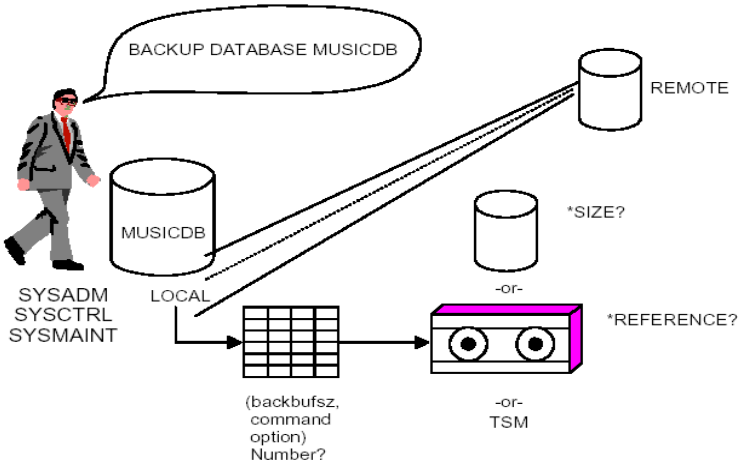


Figure 1309A ••• 지역 또는 원격 데이터베이스의 백업

2. 백업의 모드는 2가지로 분류됩니다.

모 드	설 명
OFFLINE	데이터베이스에 접속된 사용자가 없는 상태에서 백업하는 모드입니다. 백업이 진행되는 동안 데이터의 변경은 발생하지 않으므로, 백업 이미지 파일만 보관하면 복구에 사용할 수 있습니다.
ONLINE	사용자가 데이터베이스에 접속하고 있는 상태에서 백업하는 모드입니다. 백업이 진행되는 동안 데이터가 변경 중일 수도 있으므로, 백업 이미지 파일과 백업이 진행되는 동안 변경된 데이터에 대한 로그 파일을 함께 보관해야 복구에 사용할 수 있습니다. 아카이브 로깅에서만 지원됩니다.

3. 백업의 수준은 2가지로 분류됩니다.

대 상	설 명
DATABASE	데이터베이스의 모든 테이블스페이스를 백업합니다.
TABLESPACE	지정한 테이블스페이스만 백업합니다. 아카이브 로깅에서만 지원됩니다.

4. 백업의 범위는 3가지로 분류됩니다.

범 위	설 명
FULL	데이터베이스의 모든 데이터와 제어 정보를 백업합니다.
INCREMENTAL	최근의 FULL 백업 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다.
DELTA	최근의 FULL 또는 INCREMENTAL 백업 이후에 변경된 부분만 백업합니다. 아카이브 로깅에서만 지원됩니다.

TOPIC 13-10 BACKUP DB 명령어

Point

다양한 방식의 데이터베이스 백업은 모두 BACKUP DB 명령어를 이용해서 실행합니다. SYSADM, SYSCTRL, SYSMAINT 권한을 가진 사용자가 실행하며, 백업 이미지 파일을 생성합니다.

1 BACKUP DB 명령어의 형식은 다음과 같습니다.

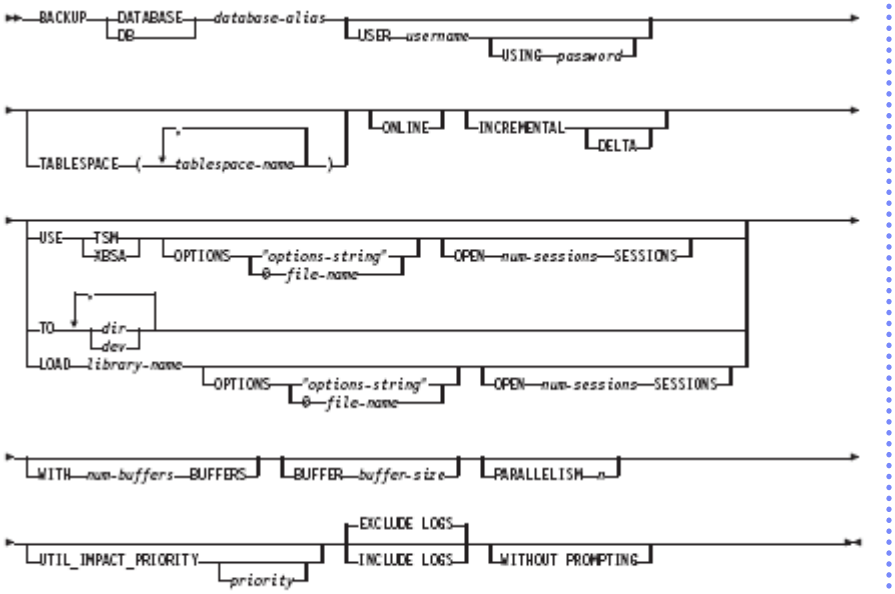


Figure 1310A ••• BACKUP DB 명령어

2 자주 사용되는 옵션은 다음과 같습니다.

옵 션	설 명
DB명	지역 DB명 또는 catalog db 명령어로 등록된 원격 DB명을 지정합니다.
USER, USING	백업을 실행하는 사용자명과 암호명을 지정합니다. 원격 데이터베이스의 백업에는 반드시 USER와 USING 옵션으로 사용자명과 암호명을 지정해야 합니다.
TABLESPACE	괄호안에 백업을 원하는 테이블스페이스의 이름들만 나열합니다.
ONLINE	온라인 모드의 백업을 실행합니다. 아카이브 로깅에서만 지원됩니다.
INCREMENTAL	INCREMENTAL 백업을 실행합니다. 아카이브 로깅에서만 지원됩니다.
DELTA	DELTA 백업을 실행합니다. INCREMENTAL 옵션과 함께 사용합니다. 아카이브 로깅에서만 지원됩니다.
TO	백업 이미지 파일이 생성되는 디렉토리를 지정할 수 있습니다. 한 개 이상의 경로명을 지정하면, 백업 이미지 파일은 여러 개의 파일로 분할되어 생성됩니다. 옵션을 지정하지 않으면, 현재 디렉토리에 한 개의 파일로 생성됩니다.
WITH ~ BUFFERES	백업 작업을 위해 데이터베이스 공유 메모리에서 할당하는 버퍼의 개수를 지정합니다.
BUFFER	백업 작업을 위해 할당하는 버퍼의 크기를 지정합니다.
INCLUDE LOGS	복구에 필요한 최소한의 로그 파일을 백업 이미지에 함께 저장하는 옵션으로 오프라인 백업에는 적용되지 않습니다.

Tip

오프라인 모드의 데이터베이스 백업을 받으려면, 접속된 모든 응용프로그램을 종료해야 합니다.

Tip

INCREMENTAL 백업, DELTA 백업은 아카이브 로깅 모드에서 TRACKMOD 구성 변수가 YES로 설정되어야 가능합니다.

Tip

원격 데이터베이스의 백업 이미지는 서버 머신에 생성되므로 TO 옵션에는 원격 서버의 경로명을 지정합니다.

Tip

백업이 완료되면, 사용되던 마지막 활성 로그는 truncate 되고, 새로운 로그 파일이 사용됩니다.,

TOPIC 13.11 백업 이미지 파일

Point

백업 이미지 파일은 기본적으로 한 개입니다. UNIX 에서 백업 이미지 파일의 이름에는 DB명, 백업의 유형, 인스턴스명, 데이터베이스 파티션 번호, 백업 완료 시간 소인, 파일 일련 번호 등의 정보가 반영됩니다.

Tip

데이프 장치에 생성된 백업 이미지는 db2ckbkp 유틸리티를 이용하여 헤더 파일에 저장된 백업 정보를 확인할 수 있습니다.

Tip

UNIX 에서는 복구시에 임의의 디렉토리에 이미지 파일을 복사하면 됩니다. Windows 인 경우에는 하위 경로명이 백업 이미지를 생성했을 때와 동일해야 합니다.

Tip

백업 이미지 파일의 이름에 온라인, 오프라인 유형 정보는 반영되지 않습니다. 상세한 list history 명령어로 확인할 수 있습니다.

Tip

백업 이미지 파일의 크기가 OS에서 지정한 한계를 초과하거나 backup db 명령어의 TO 옵션으로 강제로 분할해서 생성한 경우에는 두 개 이상의 파일로 분할됩니다.

1. UNIX에서 백업 이미지 파일명이 백업에 관한 정보를 반영하고 있습니다..

Figure 1311A ••• UNIX 에서 생성된 백업 이미지 파일명

2. Windows 에서는 서브경로명과 백업 이미지 파일명이 함께 백업 정보를 반영합니다.

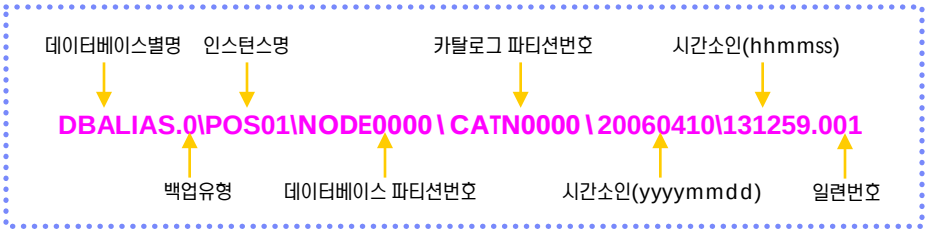


Figure 1311B ••• Windows 에서 생성된 백업 이미지 파일명

3. 백업 이미지 파일명의 각 항목은 다음과 같습니다.

옵션	설명
DB명	지역 DB명 또는 catalog db 명령어로 등록된 원격 DB명을 표시합니다.
백업 유형	0 은 데이터베이스 백업 이미지, 3은 테이블스페이스 백업 이미지, 4 는 LOAD 유틸리티의 COPY YES 옵션으로 생성된 백업 이미지를 나타냅니다.
인스턴스명	데이터베이스가 존재하는 서버의 인스턴스 이름입니다. 원격 데이터베이스인 경우에는 원격 서버 머신에 백업 이미지 파일이 생성되고, 원격 인스턴스명이 표시됩니다.
데이터베이스 파티션 번호	백업된 데이터베이스 파티션의 번호입니다. DPF 환경에서는 파티션별로 백업 이미지가 생성되며, 단일 파티션인 경우에는 항상 NODE0000 이 됩니다.
카탈로그 파티션 번호	카탈로그 테이블이 생성된 데이터베이스 파티션의 번호입니다. 단일 파티션인 경우에는 항상 CATN0000 이 됩니다.
시간 소인	백업 작업이 완료된 시간 소인입니다. UNIX 머신에서는 <yyyymmddhhmmss>의 형태로 표시되고, Windows 에서는 <yyyymmdd>과 <hhmmss>로 분리됩니다.
일련 번호	백업 이미지 파일이 두 개 이상의 파일로 분할된 경우에는 동일한 이름을 가지며, 일련 번호가 다른 여러 개의 백업 이미지 파일이 생성됩니다.

TOPIC 13-12 FULL 백업

Point

BACKUP DB 명령어로 데이터베이스 전체에 대한 백업 이미지 파일을 생성합니다. 순환 로깅에서는 오프라인 백업만 지원되며, 아카이브 로깅 모드에서는 오프라인 백업과 온라인 백업을 모두 지원합니다. 원격 데이터베이스도 백업할 수 있습니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

Tip

아카이브 로깅 모드를 사용하면 백업을 실행하는 시간동안 연관되었던 로그 파일들도 함께 보관해야 복구를 완료할 수 있습니다. 순환 로깅에서 생성된 백업 이미지는 로그 파일과 연관되지 않습니다.

Tip

backup db 명령어를 실행하는 세션은 내부적으로 데이터베이스에 접속하여 백업 이미지 파일을 생성하고, 작업이 완료되면 데이터베이스에 대한 접속을 종료합니다. 온라인 백업을 받은 세션도 마찬가지로 종료됩니다.

Tip

원격 데이터베이스의 백업 이미지 파일은 원격 데이터베이스 서버 머신에 생성됩니다.

Tip

원격 데이터베이스의 백업 경로명은 원격 데이터베이스 서버 머신의 유효한 경로명으로 지정합니다.

Tip

FULL 백업은 <Obj> 항목에 'D'라고 표시되며, <유형> 항목에 오프라인 백업이면 'F', 온라인 백업이면 'N'이라고 표시됩니다.

1 주기적으로 FULL 백업을 실행합니다. 마지막으로 생성한 FULL 백업의 이미지는 복구시에 사용되므로 안전하게 보관합니다. 온라인 백업이라면 백업 작업 동안 사용된 로그 파일을 함께 보관해야 합니다.

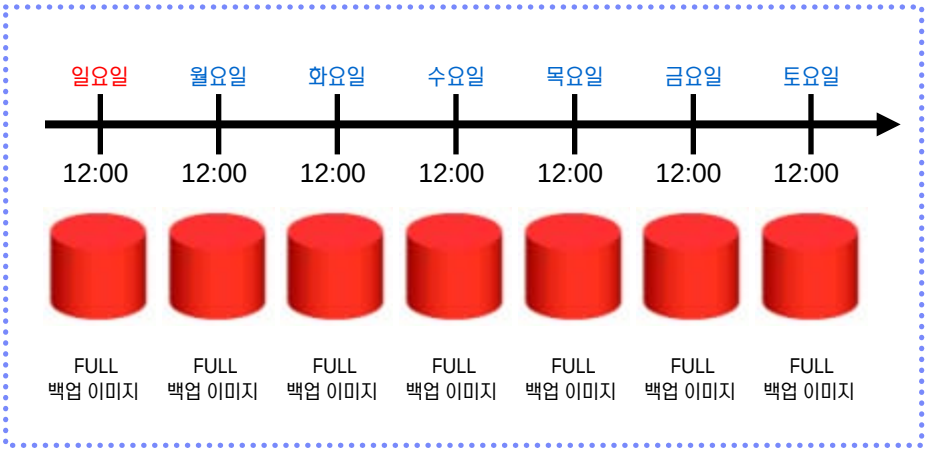


Figure 1312A FULL 백업

2 순환 로깅 모드인 경우에는 오프라인 모드의 FULL 백업만 가능합니다.

```
$ get db cfg for <DB명> | grep LOGARCHMETH1
첫 번째 로그 아카이브 메소드          (LOGARCHMETH1) = OFF
$ db2 backup db <DB명>
```

3 아카이브 로깅 모드인 경우에만 오프라인 백업과 온라인 백업이 모두 가능합니다.

```
$ get db cfg for <DB명> | grep LOGARCHMETH1
첫 번째 로그 아카이브 메소드          (LOGARCHMETH1) = LOGRETAIN
$ db2 backup db <DB명>
$ db2 backup db <DB명> ONLINE
```

4 원격 데이터베이스의 백업을 실행할 때는 반드시 <사용자명>과 <암호명>를 입력해야 합니다.

```
$ db2 backup db <DB명> user <사용자명> using <암호명>
```

5 TO 옵션을 이용하여 백업 이미지를 독립적인 디렉토리에 따로 생성하는 것이 좋습니다.

```
$ db2 backup db <DB명> TO <경로명>
$ db2 backup db <DB명> TO <경로명1>, <경로명2>
```

6 list history 명령어로 해당 INCREMENTAL 백업과 연관된 로그 파일의 번호를 확인합니다.

```
$ db2 list history backup for <DB명>
```

TOPIC 13.13 INCREMENTAL 백업

Point

BACKUP DB 명령어의 INCREMENTAL 옵션을 이용하여 최근의 FULL 백업 이후에 변경된 부분만 백업 이미지 파일에 저장합니다. 아카이브 로깅 모드이고, TRACKMOD 구성 변수가 ON으로 설정되어 있어야 합니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

Tip

증분식(cumulative) 백업이라고도 합니다.

1 주기적으로 FULL 백업을 실행하고, 다음 FULL 백업이 되기 전까지는 INCREMENTAL 백업을 여러 번 실행합니다. 기준이 되는 FULL 백업 이미지와 마지막으로 생성한 INCREMENTAL 백업의 이미지, 백업 작업 동안 사용된 로그 파일은 복구를 대비하여 안전하게 보관해야 합니다.

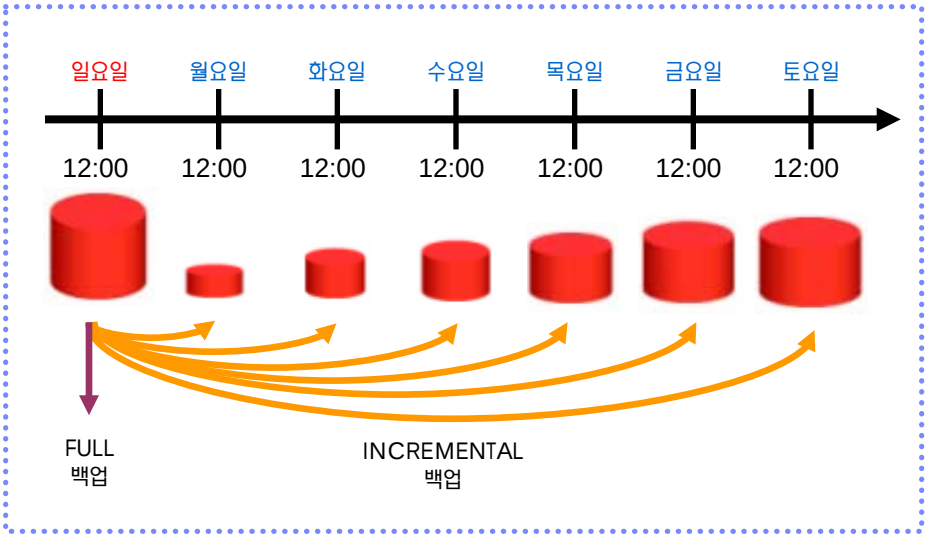


Figure 1313A FULL, INCREMENTAL 백업

Tip

아카이브 로깅 모드를 사용하므로 백업을 실행하는 시간동안 연관되었던 로그 파일들도 함께 보관해야 복구를 완료할 수 있습니다.

2 아카이브 로깅 모드에서만 지원되므로 LOGARCHMETH1 데이터베이스 구성 변수의 값이 OFF 가 아닌 것을 확인합니다.

```
$ get db cfg for <DB명> | grep LOGARCHMETH1
첫 번째 로그 아카이브 메소드          (LOGARCHMETH1) = LOGRETAIN
```

3 TRACKMOD 데이터베이스 구성 변수의 값이 ON 인 것을 확인합니다.

```
$ db2 get db cfg for <DB명> | grep TRACKMOD
 트랙 수정 페이지                    (TRACKMOD) = ON
```

4 backup database 명령에서 INCREMENTAL 옵션을 이용하여 최근의 FULL 백업 이후에 변경된 부분에 대해서만 백업 이미지를 생성합니다.

```
$ db2 backup db <DB명> INCREMENTAL
$ db2 backup db <DB명> ONLINE incremental
$ db2 backup db <DB명> online incremental  TO <디렉토리명>
$ db2 backup db <DB명> online incremental  to <디렉토리명>  INCLUDE
    LOGS
```

Tip

INCREMENTAL 백업은 <유형> 항목에 오프라인 백업이면 'I', 온라인 백업이면 'O' 라고 표시됩니다.

5 list history 명령어로 해당 INCREMENTAL 백업과 연관된 로그 파일의 번호를 확인합니다.

```
$ db2 list history backup for <DB명>
```

TOPIC 13 14 DELTA 백업

Point

BACKUP DB 명령어의 INCREMENTAL DELTA 옵션을 이용하여 최근의 FULL 또는 INCREMENTAL 백업 이후에 변경된 부분만 백업 이미지 파일에 저장합니다. 아카이브 로깅 모드이고, TRACKMOD 구성 변수가 ON으로 설정되어 있어야 합니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

1 주기적으로 FULL 백업을 실행하고, 다음 FULL 백업이 되기 전까지는 INCREMENTAL 백업 또는 DELTA 백업을 여러 번 실행합니다. 기준이 되는 FULL 백업 이미지와 마지막으로 생성한 INCREMENTAL 백업 이미지, 최근까지의 DELTA 백업 이미지, 백업 작업 동안 사용된 로그 파일은 복구시에 사용되므로 안전하게 보관해야 합니다.

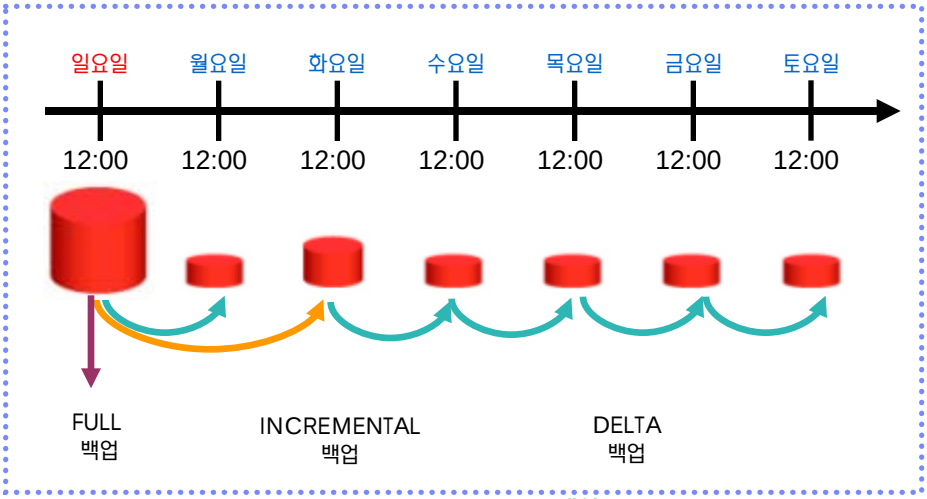


Figure 1314A • • • FULL, INCREMENTAL, DELTA 백업

Tip

아카이브 로깅 모드를 사용하므로 백업을 실행하는 시간동안 연관되었던 로그 파일들도 함께 보관해야 복구를 완료할 수 있습니다.

2 아카이브 로깅 모드에서만 지원되므로 LOGARCHMETH1 데이터베이스 구성 변수의 값이 OFF 가 아닌 것을 확인합니다.

```
$ get db cfg for <DB명> | grep LOGARCHMETH1
첫 번째 로그 아카이브 메소드          (LOGARCHMETH1) = LOGRETAIN
```

3 TRACKMOD 데이터베이스 구성 변수의 값이 ON 인 것을 확인합니다.

```
$ db2 get db cfg for <DB명> | grep TRACKMOD
 트랙 수정 페이지                    (TRACKMOD) = ON
```

4 backup database 명령에서 INCREMENTAL DELTA 옵션을 이용하여 최근의 FULL 또는 INCREMENTAL 백업 이후에 변경된 부분에 대해서만 백업 이미지를 생성합니다.

```
$ db2 backup db <DB명> INCREMENTAL DELTA
$ db2 backup db <DB명> ONLINE incremental delta
$ db2 backup db <DB명> online incremental  delta TO <디렉토리명>
$ db2 backup db <DB명> online incremental  delta to <디렉토리명>
    INCLUDE LOGS
```

Tip

DELTA 백업은 <유형> 항목에 오프라인 백업이면 'D', 온라인 백업이면 'E' 라고 표시됩니다.

5 list history 명령어로 해당 INCREMENTAL 백업과 연관된 로그 파일의 번호를 확인합니다.

```
$ db2 list history backup for <DB명>
```

TOPIC 13.15 테이블스페이스 백업

Point

데이터베이스 전체를 백업하는 것이 아니라, 중요한 테이블스페이스만 백업합니다. BACKUP DB 명령어에서 TABLESPACE 옵션을 이용합니다. 한 개 이상의 테이블스페이스를 함께 백업할 수 있으며, 아카이브 로깅 모드에서 지원됩니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

1. 카탈로그 테이블스페이스 또는 중요한 테이블스페이스만 별도로 백업합니다. 테이블스페이스에 대한 백업 이미지 파일과 백업 작업 동안 사용된 로그 파일을 안전하게 보관해야 합니다.

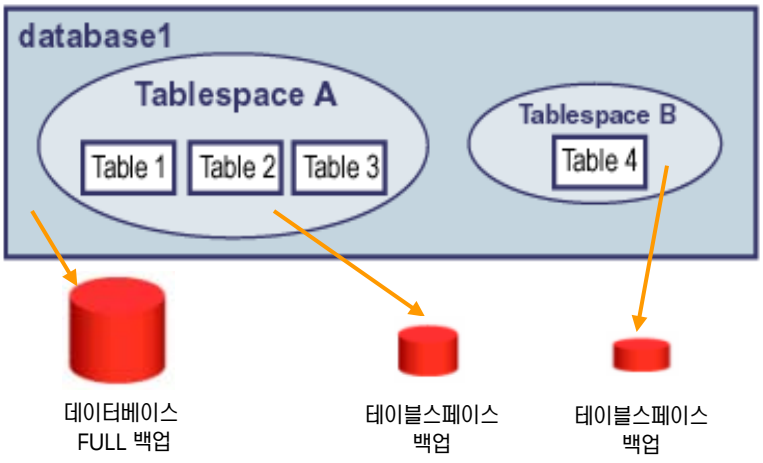


Figure 1315A 테이블스페이스 백업

2. 아카이브 로깅 모드에서만 지원되므로 LOGARCHMETH1 데이터베이스 구성 변수의 값이 OFF 가 아닌 것을 확인합니다.

```
$ get db cfg for <DB명> | grep LOGARCHMETH1
첫 번째 로그 아카이브 메소드            (LOGARCHMETH1) = LOGRETAIN
```

3. backup database 명령에서 TABLESPACE 옵션을 이용하여 한 개 이상의 테이블스페이스에 대한 백업 이미지를 생성합니다. 연관된 테이블스페이스는 함께 백업합니다.

```
$ db2 "backup db <DB명> TABLESPACE (<TS명>)"
$ db2 "backup db <DB명> TABLESPACE (<TS명 1>, <TS명 2>)"
```

4. ONLINE, TO, INCREMENTAL, DELTA 등의 옵션과 함께 사용될 수 있습니다.

```
$ db2 "backup db <DB명> tablespace (<TS명>) ONLINE "
$ db2 "backup db <DB명> tablespace (<TS명>) online TO <디렉토리명>"
$ db2 "backup db <DB명> tablespace (<TS명>) online INCREMENTAL"
$ db2 "backup db <DB명> tablespace (<TS명>) online INCREMENTAL
    DELTA"
```

Tip

TABLESPACE 백업은 <Obj> 항목에 'P'라고 표시됩니다.

5. list history 명령어로 해당 INCREMENTAL 백업과 연관된 로그 파일의 번호를 확인합니다.

```
$ db2 list history backup for <DB명>
```

TOPIC 13-16 복구의 종류

Point

RESTORE DB, ROLLFORWARD DB 명령어에서 백업 이미지와 아카이브 로그로 원하는 시점으로 복구합니다. CRASH/VERSION/ROLLFORWARD, DATABASE/TABLESPACE, FULL/INCREMENTAL/DELTA 복구로 분류됩니다.

Tip

백업 이미지는 한 개 이상의 파일로 생성되어 디스크, 테이프, TSM 등으로 저장됩니다.

Tip

롤포워드 복구, 테이블스페이스 복구, INCREMENTAL 복구, DELTA 복구는 아카이브 로깅에서만 지원되며, 백업 이미지와 연관된 최소한의 아카이브 로그가 반드시 존재해야 합니다.

1 데이터베이스를 사용할 수 없는 상태가 되면 SYSADM, SYSCTRL, SYSMAINT 권한의 사용자가 원격 또는 지역 데이터베이스의 백업 이미지 파일, 아카이브 로그 파일, 활성 로그 파일 등을 이용하여 데이터베이스를 복구합니다.

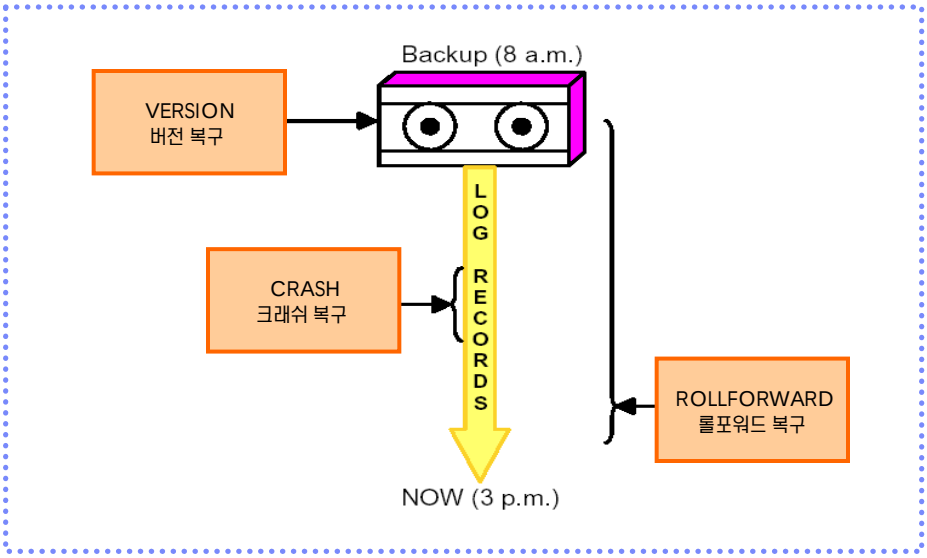


Figure 1316A 다양한 복구 방법

2 복구의 유형은 세 가지로 분류됩니다.

방 법	설 명
크래쉬 복구	정전 등의 비상시에 활성 로그 파일을 이용하여 자동 실행됩니다.
버전 복구	과거 시점에 생성했던 오프라인 모드의 데이터베이스 백업 이미지를 이용하여 과거의 시점과 동일한 상태로 데이터베이스를 복구합니다.
롤포워드 복구	버전 복구를 완료한 후에 아카이브 로그 파일을 이용하여 원하는 시점까지의 로그 파일을 적용하는 복구입니다.

3 복구의 수준은 2가지로 분류됩니다.

대 상	설 명
DATABASE	데이터베이스의 모든 테이블스페이스를 복원합니다.
TABLESPACE	지정한 테이블스페이스만 복원합니다.

4 복구의 범위는 3가지로 분류됩니다.

범 위	설 명
FULL	FULL 백업 이미지를 이용하여 복구합니다.
INCREMENTAL	최근의 INCREMENTAL 백업 이미지와 기준이 되는 FULL 백업 이미지를 이용하여 복구합니다.
DELTA	최근의 DELTA 백업 이미지들과 기준이 되는 INCREMENTAL 백업 이미지, 기준이 되는 FULL 백업 이미지를 이용하여 복구합니다.

TOPIC 13.17 RESTART DB 명령어

Point

시스템 정전 등의 비상시에 실행되는 크래쉬 복구를 실행하는 명령어입니다. 인스턴스가 비정상적으로 종료되는 경우에도 크래쉬 복구가 필요합니다. 크래쉬 복구의 자동 실행 여부는 AUTORESTART 데이터베이스 구성 변수로 조절합니다.

Tip

restart db 명령어를 실행하는 세션은 내부적으로 데이터베이스에 접속합니다. 작업을 완료한 후에 데이터베이스에 대한 접속은 유지됩니다.

1 RESTART DB 명령어의 형식은 다음과 같습니다.

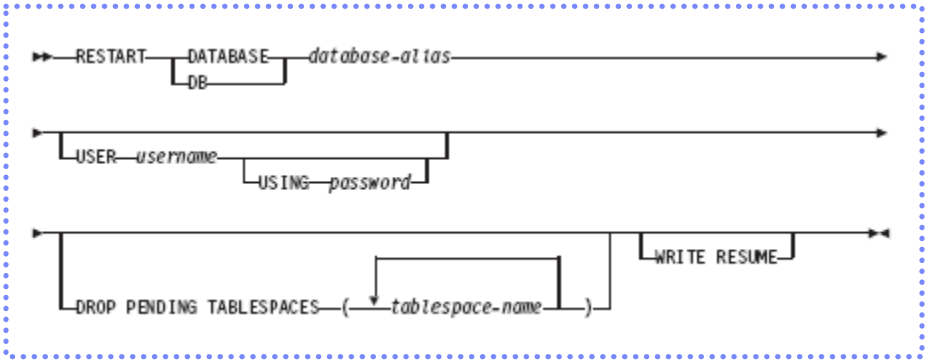


Figure 1317A RESTART DB 명령어

2 사용되는 옵션은 다음과 같습니다.

옵션	설명
DB명	지역 DB명 또는 catalog db 명령어로 등록된 원격 DB명을 지정합니다.
USER, USING	크래쉬 복구를 실행하는 사용자명과 암호명을 지정합니다. 원격 데이터베이스의 복구에는 반드시 USER와 USING 옵션으로 사용자명과 암호명을 지정해야 합니다.
DROP PENDING TABLESPACES	테이블스페이스의 컨테이너가 손상되면 해당 테이블스페이스는 사용할 수 없게 됩니다. 이러한 테이블스페이스가 발견되면, 크래쉬 복구를 완료할 수 없습니다. 괄호안에 DROP을 원하는 테이블스페이스의 이름들을 나열하면 해당 테이블스페이스는 'DROP 보류' 상태가 됩니다. 크래쉬 복구가 완료된 후에 데이터베이스에 접속하여 테이블스페이스를 재생성해야 합니다.

TOPIC 13-18 RESTORE DB 명령어

Point

BACKUP DB 명령어로 생성된 백업 이미지 파일을 이용하여 데이터베이스를 과거의 시점과 동일한 버전으로 복구합니다. SYSADM, SYSCTRL, SYSMAINT 권한을 가진 사용자가 실행하며, 백업 이미지 파일이 필요합니다.

1 RESTORE DB 명령어의 형식은 다음과 같습니다.

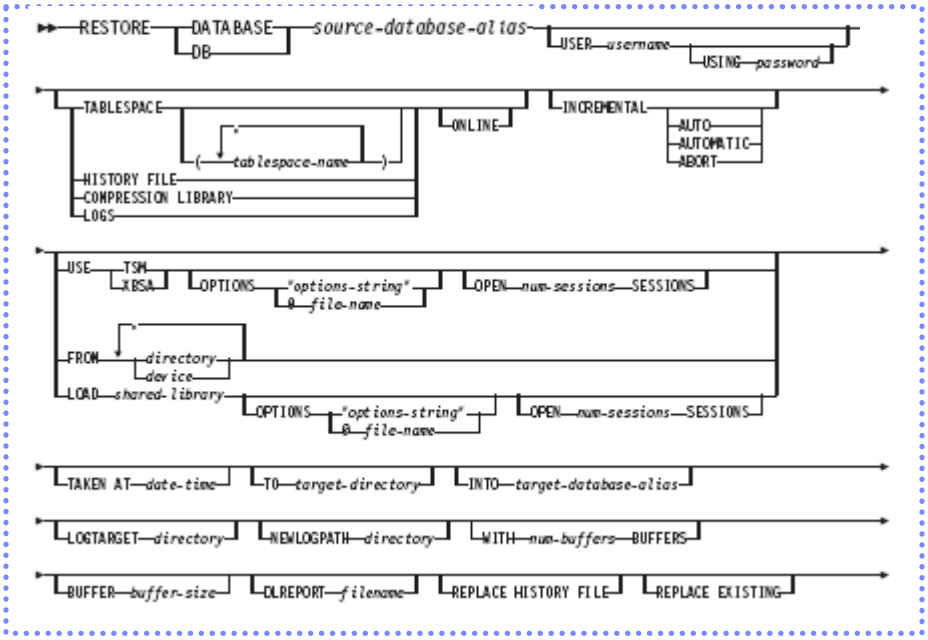


Figure 1318A ··· RESTORE DB 명령어

2 자주 사용되는 옵션은 다음과 같습니다.

옵션	설명
DB명	지역 DB명 또는 등록된 원격 DB의 별명을 지정합니다.
USER, USING	복구를 실행하는 사용자명과 암호명을 지정합니다. 원격 데이터베이스에는 반드시 사용자명과 암호명을 지정해야 합니다.
TABLESPACE	괄호안에 복구를 원하는 테이블스페이스의 이름들만 나열합니다.
HISTORY FILE	복구 실행 기록 파일만 복구합니다.
ONLINE	온라인 모드로 복구합니다. TABLESPACE 복구에 지원됩니다.
INCREMENTAL	INCREMENTAL 또는 DELTA 백업 이미지의 복구를 실행합니다.
FROM	백업 이미지 파일이 존재하는 디렉토리를 지정합니다. 백업 이미지 파일이 여러 개의 파일로 분할되어 생성되었으면, 한 개 이상의 경로명을 지정해야 합니다. 옵션을 지정하지 않으면, 현재 디렉토리에 한 개의 백업 이미지 파일을 이용합니다.
TAKEN AT	지정한 경로명에 동일한 DB에 대한 백업 이미지가 여러 개 있는 경우에는 백업 이미지의 생성 시간 소인을 명시하여 구별합니다.
TO	새로운 데이터베이스가 생성될 경로명을 지정합니다.
INTO	새로 생성될 데이터베이스명을 지정합니다.
WITH ~ BUFFERES	복구 작업을 위해 데이터베이스 공유 메모리에서 할당하는 버퍼의 개수를 지정합니다.
BUFFER	복구 작업을 위해 할당하는 버퍼의 크기를 지정합니다.

Tip

지정한 디렉토리에 동일한 데이터베이스에 대한 백업이 한 개만 있으면 TAKEN AT 옵션이 필요없습니다.

Tip

INCREMENTAL 복구, DELTA 복구는 아카이브 로깅 모드에서 TRACKMOD 구성 변수가 YES로 설정되어야 가능합니다.

Tip

FULL 데이터베이스 복구시에는 데이터베이스가 비활성화되어야 하므로 접속된 모든 응용프로그램을 종료해야 합니다.

Tip

원격 데이터베이스의 백업 이미지는 서버 머신에 존재하므로 FROM 옵션에는 원격 서버의 경로명을 지정합니다.

TOPIC 13-19 ROLLFORWARD DB 명령어

Point

RESTORE DB 명령어로 데이터베이스가 과거의 시점으로 버전 복구가 완료된 후에 아카이브 로그를 차례로 읽어들여 로그에 기록된 변경 내용을 데이터베이스에 반영합니다. 버전 복구에 사용된 백업 이미지가 생성된 시점 이후의 아카이브 로그가 필요합니다.

1 ROLLFORWARD DB 명령어의 형식은 다음과 같습니다.

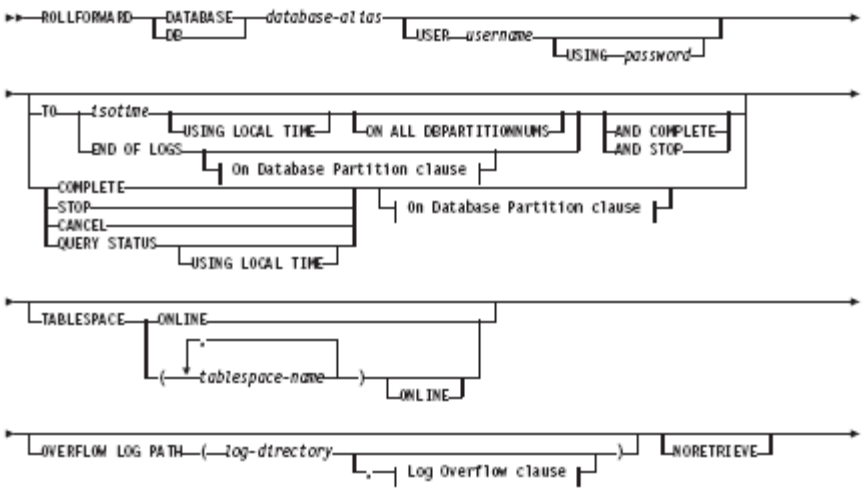


Figure 1319A ··· ROLLFORWARD DB 명령어

2 자주 사용되는 옵션은 다음과 같습니다.

옵션	설명
DB명	롤포워드의 대상이 되는 데이터베이스의 별명을 지정합니다.
USER, USING	롤포워드를 실행하는 사용자명과 암호명을 지정합니다. 원격 데이터베이스는 반드시 사용자명과 암호명을 지정해야 합니다.
TO isotime	<표준시간> 으로 표시한 시점까지의 로그 파일을 적용합니다. OS의 TIMEZONE 값을 이용한 환산이 필요합니다. <표준시간>은 <yyyy-mm-dd-hh.mm.ss.nnnnnn>의 형태로 표현합니다.
USING LOCAL TIME	TO isotime 옵션과 함께 사용합니다. <표준시간> 대신에 현 시스템의 시간대에서 사용하는 <지역 시간>을 지정합니다. <지역 시간>도 <yyyy-mm-dd-hh.mm.ss.nnnnnn>의 형태로 표현합니다.
TO END OF LOGS	로그 디렉토리에 존재하는 온라인 아카이브 로그 파일의 최대 번호까지 롤포워드를 실행합니다.
STOP	롤포워드를 종료하고, '롤포워드 보류' 상태를 해제합니다.
CANCEL	롤포워드를 중단하고, 데이터베이스를 다시 '리스토어 보류' 상태가 되게 합니다. RESTORE DB 명령어로 다시 복구 작업을 실행해야 합니다.
QUERY STATUS	롤포워드 작업의 진행 정보를 알려줍니다.
TABLESPACE	테이블스페이스 수준의 롤포워드 작업시에 사용합니다.
ONLINE	테이블스페이스 수준의 롤포워드를 온라인 모드로 실행합니다. 다른 테이블스페이스는 액세스할 수 있습니다.

Tip

로그 파일의 내용은 표준 시간으로 기록됩니다. 한국에서 사용하는 데이터베이스 서버는 대부분 TIMEZONE의 값이 +9 이므로, 원하는 시점에서 -9를 하면 표준 시간이 산출됩니다. USING LOCAL TIME 옵션을 사용할 것을 권장합니다.

Tip

END OF LOGS 는 최근의 로그까지 적용한다는 의미가 아니라, 로그 디렉토리에 존재하는 모든 아카이브 로그를 적용한다는 의미이므로 구별해야 합니다.

Tip

ROLLFORWARD 명령은 여러 번 반복적으로 실행할 수 있습니다. 적용할 아카이브 로그 파일이 너무 많아서 로그 디렉토리로 한꺼번에 복사할 수 없는 경우에는 일정 개수씩 분할하여 여러 번 반복해서 END OF LOGS 옵션으로 적용합니다. 마지막 적용이 끝나면 STOP 옵션으로 롤포워드를 종료합니다.

TOPIC 13-20 크래쉬 복구

Point

데이터베이스가 비정상적으로 종료되면, 데이터의 일관성을 보장하기 위해 restart db 명령어로 활성 로그 파일을 읽어서 실행 중이던 트랜잭션의 COMMIT과 ROLLBACK을 완료합니다. AUTORESTART 변수가 ON 이므로 자동적으로 실행됩니다.

Tip

특별한 권한이 필요하지 않습니다.

Tip

크래쉬 복구는 로깅 모드와 상관이 없습니다. 크래쉬 복구에 필요한 것은 손상 직전의 활성 로그입니다.

1. 응용프로그램이 연결되어 있을 때 데이터베이스가 비정상 종료하는 경우에는 크래쉬 복구가 필요합니다. 데이터베이스의 비정상 종료는 전원 공급 중단이나 시스템 소프트웨어 장애로 인해 발생할 수 있습니다. 이러한 종료는 실패시 데이터베이스 버퍼 풀에 있으나 디스크에 기록되지 않은 커미트 트랜잭션에 적용됩니다. 또한 디스크에 기록된 미확약 트랜잭션을 롤백시킵니다.

2. get dbm cfg 명령어로 데이터베이스 구성 변수인 AUTORESTART의 기본값이 ON 인 것을 확인합니다. ON 은 필요시에 크래쉬 복구가 자동적으로 실행되는 것을 의미합니다.

```
$ login <인스턴스 사용자>
$ db2 get db cfg for <DB명> | grep AUTORESTART
자동 재시작 사용                    (AUTORESTART) = ON
```

3. AUTORESTART 값을 OFF로 설정하면, 필요시에 크래쉬 복구가 자동적으로 실행되지 않습니다. activate db 명령어로 데이터베이스를 활성화시키거나, connect 명령어로 최초의 데이터베이스 접속을 요청하면, SQL1015N 오류가 반환됩니다. restart db 명령어를 사용하여 크래쉬 복구를 실행합니다. 크래쉬 복구가 완료되면 데이터베이스를 사용할 수 있습니다.

```
$ db2 RESTART DATABASE <DB명>
$ db2 ACTIVATE DATABASE <DB명>
$ db2 CONNECT TO <DB명>
```

4. 크래쉬 복구에 대한 정보는 오류 진단 파일인 db2diag.log 파일에서 확인할 수 있습니다.

```
$ vi  $HOME/sqllib/db2diag.log
```

```
2006-03-08-16.32.39.217725+540 I990834A425        LEVEL: Warning
PID     : 4190438              TID  : 1           PROC : db2agent (SAMPLE) 0
INSTANCE: db2inst1               NODE : 000         DB   : SAMPLE
APPHDL  : 0-7                  APPID: 11210612.C3E6.06ADB8073211
FUNCTION: DB2 UDB, recovery manager, sqlpresr, probe:410
MESSAGE : Crash recovery started. LowtranLSN 00000297C9853A02 MinbuffLSN
          00000297C8275DB1

2006-03-08-16.32.39.271097+540 I991260A409        LEVEL: Warning
PID     : 4190438              TID  : 1           PROC : db2agent (SAMPLE) 0
INSTANCE: db2inst1               NODE : 000         DB   : SAMPLE
APPHDL  : 0-7                  APPID: 11210612.C3E6.06ADB8073211
FUNCTION: DB2 UDB, recovery manager, sqlprecm, probe:2000
MESSAGE : Using parallel recovery with 13 agents 52 QSets 624 queues and 256 chunks

2006-03-08-16.32.44.217305+540 I991670A306        LEVEL: Warning
PID     : 1863868              TID  : 1           PROC : db2lfr (SAMPLE) 0
INSTANCE: db2inst1               NODE : 000
FUNCTION: DB2 UDB, recovery manager, sqlplfrVerifyLogPages, probe:1750
MESSAGE : Saving partial page, pageLsn is 0297C9854854

2006-03-08-16.32.46.835129+540 I991977A435        LEVEL: Warning
PID     : 4190438              TID  : 1           PROC : db2agent (SAMPLE) 0
INSTANCE: db2inst1               NODE : 000         DB   : SAMPLE
APPHDL  : 0-7                  APPID: 11210612.C3E6.06ADB8073211
FUNCTION: DB2 UDB, recovery manager, sqlprecm, probe:4000
MESSAGE : DIA2051W Forward phase of crash recovery has completed.  Next LSN is
          "00000297C9854855".

2006-03-08-16.32.46.857746+540 I992413A390        LEVEL: Warning
PID     : 4190438              TID  : 1           PROC : db2agent (SAMPLE) 0
INSTANCE: db2inst1               NODE : 000         DB   : SAMPLE
APPHDL  : 0-7                  APPID: 11210612.C3E6.06ADB8073211
FUNCTION: DB2 UDB, recovery manager, sqlpresr, probe:3170
MESSAGE : Crash recovery completed. Next LSN is 00000297C9854855
```

Figure 1320A • • • 크래쉬 복구에 관한 메시지

TOPIC 13.21 버전 복구

Point

RESTORE DB 명령어로 원격 또는 지역 데이터베이스를 이전 버전으로 복구하는 유틸리티입니다. 오프라인 모드의 FULL 백업 이미지 파일이 필요합니다. RESTBUFSZ 옵션으로 복구 작업을 위한 버퍼를 할당할 수 있습니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

Tip

백업 이미지는 한 개 이상의 파일로서 디스크, 테이프, TSM 등에 미리 저장되어 있어야 됩니다.

1 데이터베이스가 손상된 경우에는 최근에 생성했던 FULL 백업 이미지 파일만 있으면, restore db 명령으로 데이터베이스를 이전 버전으로 복구할 수 있습니다. 복구에 사용된 백업 이미지가 생성되었던 시점 이후부터 현재까지의 데이터베이스에 대한 모든 변경 내용은 유실됩니다.

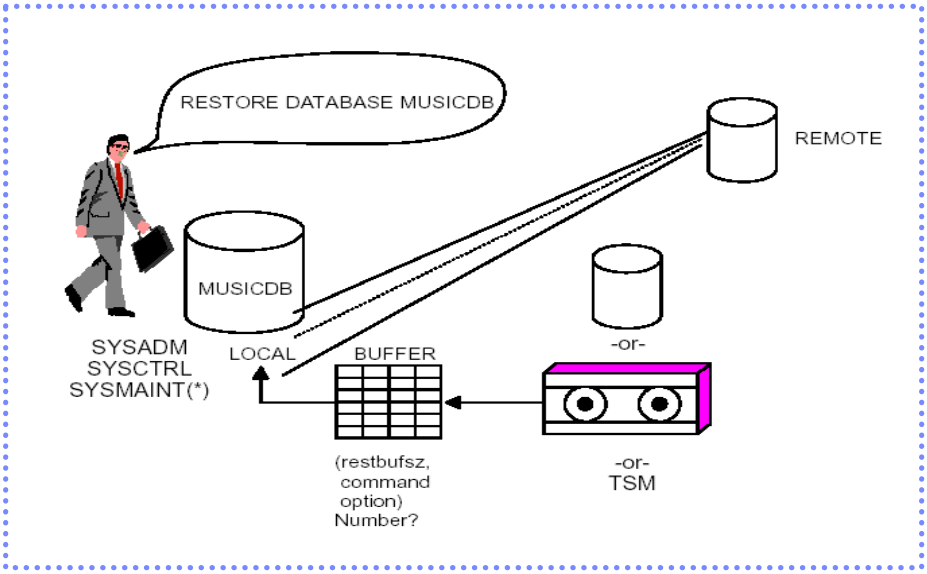


Figure 1321A RESTORE DB 명령어를 이용한 버전 복구

2 restore database 명령을 이용하여 데이터베이스의 백업 이미지로부터 기존 데이터베이스로 복구합니다. 접속된 모든 응용프로그램은 종료되어야 합니다.

```
$ db2 force applications all
$ db2 restore db <DB명>
```

Tip

신규 데이터베이스로 복구하는 경우에는 SYSADM 또는 SYSCTRL 권한이 필요합니다.

3 INTO 옵션을 사용하면 신규 데이터베이스로 복구할 수 있습니다.

```
$ db2 restore db <DB명> INTO <신규 DB명>
```

4 백업 이미지가 다른 디렉토리에 있는 경우에는 restore 명령어의 FROM 옵션을 이용합니다.

```
$ db2 restore db <DB명> FROM <디렉토리명>
```

Tip

<백업시간소인> 은 백업 이미지 파일명의 <yyyymmddhhmmss> 형태로 지정합니다.

5 백업 이미지가 여러 개 있으면 restore 명령어의 TAKEN AT 옵션을 이용합니다.

```
$ db2 restore db <DB명> taken at <백업시간소인>
```

Tip

restore db 명령어는 <Op> 항목에 'R' 이라고 표시됩니다.

6 list history 명령어로 해당 버전 복구와 연관된 기록을 확인합니다.

```
$ db2 list history backup for <DB명>
```

TOPIC 13-22 경로 재지정 복구

Point

백업 이미지 파일에 저장된 테이블스페이스의 컨테이너 구성을 사용자가 변경하여 복구하는 방식입니다. 테이블 스페이스 컨테이너가 손상된 경우의 복구에 유용합니다. RESTORE DB 명령어의 REDIRECT 옵션을 이용합니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

Tip

restore db 명령어와 set tablespace containers 명령어는 동일한 세션에서 실행합니다.

Tip

원본 데이터베이스의 테이블스페이스의 ID와 컨테이너에 대한 정보를 미리 알고 있어야 합니다. 원본 데이터베이스에서 list tablespace containers 명령어 또는 db2look 명령어를 이용합니다.

1. 백업 이미지 파일에는 테이블스페이스의 컨테이너에 대한 정보가 함께 저장되어 있으므로 복구가 시작되면, 백업 이미지 파일에 설정된 정보에 의해 테이블스페이스가 생성됩니다. 지정한 테이블스페이스 컨테이너를 액세스할 수 없는 상태가 되면 복구 작업은 진행될 수 없습니다. RESTORE DB 명령어의 REDIRECT 옵션과 SET TABLESPACE CONTAINERS 문을 이용하여 액세스 불가능한 컨테이너에 대한 경로를 재지정하면 새로운 컨테이너로 복구됩니다.

2. 백업 이미지는 기본적으로 기존의 데이터베이스를 복구할 때 사용됩니다. 백업 이미지에는 데이터베이스에 대한 모든 정보가 저장되어 있으므로 새로운 데이터베이스로 복구하는 것도 가능합니다. 동일한 데이터베이스 서버에서 새로운 데이터베이스로 복구하게 되면, 테이블스페이스의 컨테이너가 중복되므로 경로 재지정 복구 방법이 사용됩니다.

3. RESTORE DB 명령어에서 REDIRECT 옵션을 사용하면 경로 재지정 복구를 지정됩니다.

```
$ RESTORE DB <백업이미지의 DB명> INTO <새로운 DB명> REDIRECT
```

4. SET TABLESPACE CONTAINERS 명령어에서 컨테이너에 대한 경로를 재지정합니다. FOR 옵션에서 사용되는 번호는 테이블스페이스 ID입니다. 기존의 테이블스페이스의 컨테이너를 제거 또는 변경할 수 있으며, 새로운 컨테이너를 추가할 수도 있습니다.

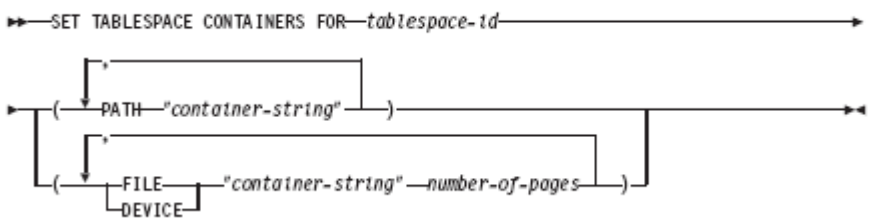


Figure 1322A ••• SET TABLESPACE CONTAINERS 명령어

```
$ SET TABLESPACE CONTAINERS FOR 0 USING (FILE  "<파일명>"  <크기>)
$ SET TABLESPACE CONTAINERS FOR 1 USING (PATH "<경로명>")
```

5. RESTORE DB 명령어에서 ABORT 옵션을 사용하면 경로 재지정 복구가 취소됩니다.

```
$ RESTORE DB <백업이미지의 DB명> INTO <새로운 DB명> ABORT
```

6. RESTORE DB 명령어에서 CONTINUE 옵션을 사용하면 경로 재지정 복구가 시작됩니다.

```
$ RESTORE DB <백업이미지의 DB명> INTO <새로운 DB명> CONTINUE
```

7. ROLLFORWARD DB 명령어로 아카이브 로그를 적용합니다.

```
$ ROLLFORWARD DB <새로운 DB명> TO END OF LOGS AND STOP
```

TOPIC 13.23 롤포워드 복구

Point

RESTORE DB 명령어로 버전 복구를 완료하고, ROLLFORWARD DB 명령어로 로그 디렉토리에 제공된 아카이브 로그 파일들을 차례로 이용하여 과거의 UOW에 의한 변경 내역을 복구합니다. 아카이브 로깅 방식에서만 지원됩니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

Tip

롤포워드 복구는 restore db 명령어를 이용한 데이터베이스 버전 복구와 rollforward db 명령어를 이용한 로그 재적용 과정을 합친 복구 방법입니다. rollforward db 명령어는 아카이브 로그 파일을 데이터베이스에 재적용하는 조작이므로 롤포워드 복구와 구별해야 합니다.

1 데이터베이스가 손상된 경우에는 최근에 생성했던 FULL 백업 이미지 파일과 아카이브 로그가 있으면, restore db 명령어와 rollforward db 명령어로 데이터베이스를 원하는 시점으로 복구할 수 있습니다. 복구에 사용된 마지막 아카이브 로그에 저장된 트랜잭션 이후부터 현재까지의 데이터베이스에 대한 변경 내용은 유실됩니다.

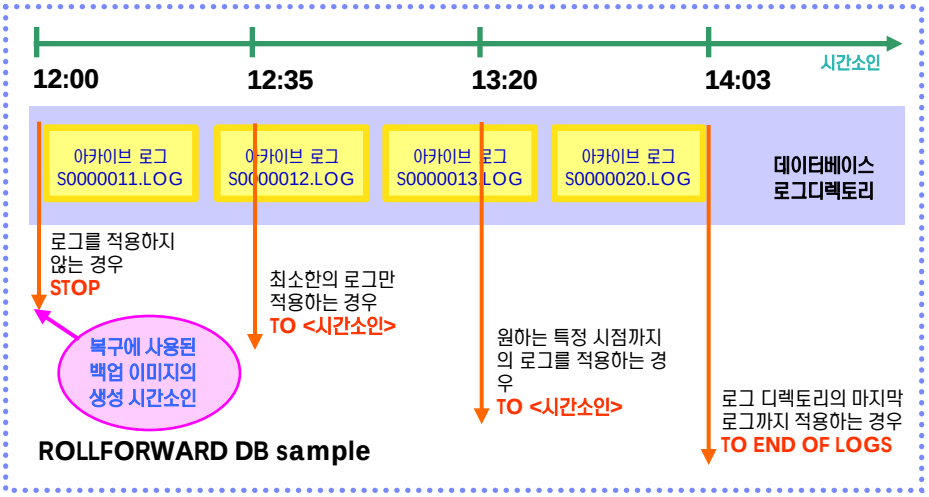


Figure 1323A ··· ROLLFORWARD DB 명령어를 이용한 로그 재적용의 범위

Tip

ROLLFORWARD 명령은 여러 번 반복적으로 실행할 수 있습니다.

Tip

END OF LOGS 옵션은 로그 디렉토리에 존재하는 로그 파일을 모드 적용시키는 옵션입니다. 최근의 로그 파일을 의미하는 것이 아닙니다.

2 최초의 rollforward db 명령어는 버전 복구를 완료된 이후에 시작할 수 있습니다.

```
$ db2 restore db <DB명>
$ db2 rollforward db <DB명> to end of logs
```

3 적용할 로그 파일의 크기가 크고, 개수가 많은 경우에는 로그 디렉토리에 로그 파일의 일부만 옮겨 두고, 여러 번에 걸쳐 rollforward db 명령어를 실행합니다.

```
$ db2 rollforward db <DB명> to end of logs
```

4 마지막 rollforward db 명령을 입력할 때, TO 옵션을 이용하여 원하는 시점까지의 복구가 가능합니다. <시간소인>은 USING LOCAL TIME 옵션으로 지정합니다.

```
$ db2 rollforward db <DB명> to <시간소인> using local time
```

Tip

USING LOCAL TIME 옵션이 없으면, GMT 시간을 명시해야 합니다.

Tip

<시간 소인>은 yyyy-mm-dd-hh:mm:ss 형태로 지정합니다.

5 STOP 옵션을 이용하면, 로그 재적용 작업은 중지되고, 마지막으로 적용된 로그 파일의 다음 번호부터 데이터베이스 로깅이 다시 시작됩니다.

```
$ db2 rollforward db <DB명> stop
```

Tip

rollforward db 명령어는 <Op> 항목에 'F' 라고 표시됩니다. END OF LOGS 옵션은 'E', <시간소인> 옵션은 'P' 로 표시됩니다.

6 list history 명령어로 해당 롤포워드 복구와 연관된 기록을 확인합니다.

```
$ db2 list history rollforward for <DB명>
```

TOPIC 13-24 INCREMENTAL 복구

Point
최근의 FULL 백업 이미지 파일과 FULL 백업 이후에 생성된 최근의 INCREMENTAL 백업 이미지 파일, INCREMENTAL 백업 시점 이후의 아카이브 로그 파일이 보관되어 있으면, INCREMENTAL 복구가 가능합니다.

Tip
SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

1 restore db 명령어로 최근의 FULL 백업 이미지를 이용하여 버전 복구를 합니다. restore db 명령어로 FULL 백업 시점 이후의 INCREMENTAL 백업 이미지 중에서 최근 것으로 FULL 백업과 INCREMENTAL 백업 시점 구간에 있는 변경 내용을 복구합니다. 마지막으로 INCREMENTAL 백업 이후 시점부터 현재까지의 아카이브 로그를 차례로 적용하면 데이터베이스는 손상 직전의 상태로 복구됩니다.

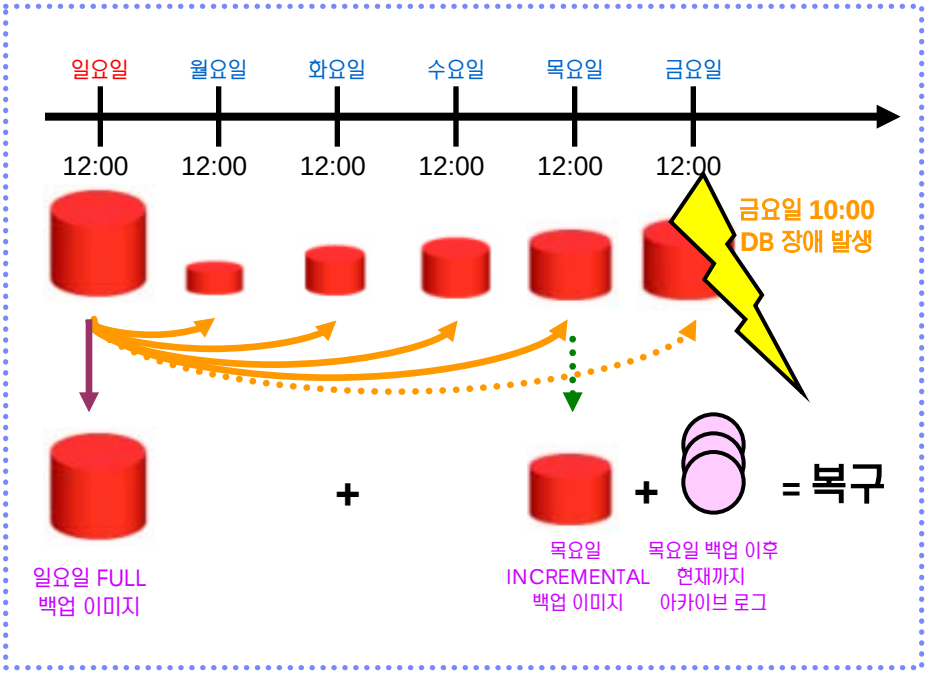


Figure 1324A FULL, INCREMENTAL 이미지, 아카이브 로그를 이용한 복구

Tip
INCREMENTAL AUTOMATIC 옵션을 이용하면, 복구 기록 파일을 이용하여 필요한 INCREMENTAL 백업 이미지를 자동적으로 찾아서 복구를 실행합니다.

2 최근의 FULL 백업 이미지 파일을 이용하여 restore database 명령으로 데이터베이스를 이전 시점으로 버전 복구합니다.

```
$ db2 restore db <DB명>
```

3 restore database 명령의 INCREMENTAL 옵션으로 최근의 INCREMENTAL 백업 이미지 파일을 이용하여 FULL 백업 이후의 변경분을 복구합니다.

```
$ db2 restore db <DB명> INCREMENTAL
```

Tip
손상 직전의 시점으로 복구하려면, 마지막 INCREMENTAL 백업 이후부터 손상 직전 시점까지의 아카이브 로그 파일을 로그 디렉토리에 모두 복사하고, TO END OF LOGS AND STOP 옵션을 사용합니다.

4 rollforward db 명령으로 아카이브 로그 파일을 이용하여 최근의 INCREMENTAL 백업 이미지 생성 시점 이후부터 원하는 시점까지의 변경분을 복구합니다.

```
$ db2 rollforward db <DB명> to <시간소인> using local time and stop
```

5 list history 명령어로 해당 롤포워드 복구와 연관된 기록을 확인합니다.

```
$ db2 list history rollforward for <DB명>
```

TOPIC 13.25 DELTA 복구

Point

최근의 FULL 백업 이미지 파일과 INCREMENTAL 백업 또는 DELTA 백업 이미지 파일, 데이터베이스 로그 파일이 보관되어 있으면, INCREMENTAL 복구가 가능합니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

1. 최근의 FULL 백업 이미지 파일, 최근의 INCREMENTAL 백업 이미지, 현 시점까지의 DELTA 백업 이미지 파일, 아카이브 로그 파일을 준비하고, restore db 명령어와 rollforward db 명령어를 실행합니다.

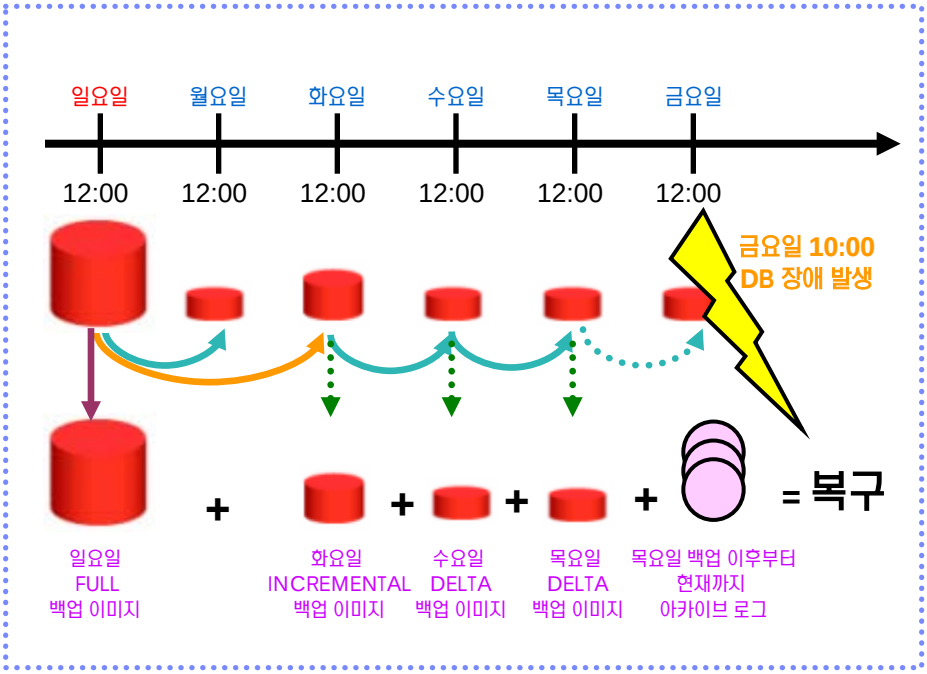


Figure 1325A FULL, INCREMENTAL, DETA 이미지, 아카이브 로그를 이용한 복구

Tip

restore db 명령어에서 사용하는 옵션은 INCREMENTAL 복구와 동일합니다.

2. 최근의 FULL 백업 이미지 파일을 이용하여 Restore database 명령으로 데이터베이스를 과거의 시점으로 복구합니다.

```
$ db2 restore db <DB명>
```

Tip

INCREMENTAL AUTOMATIC 옵션을 이용하면, 복구 기록 파일을 이용하여 필요한 INCREMENTAL 백업 이미지를 자동적으로 찾아서 복구를 실행합니다.

3. 최근의 INCREMENTAL 백업 이미지 파일이 있다면 restore db 명령으로 FULL 백업 이후의 변경분을 복구합니다. restore db 명령을 반복적으로 사용하여 최근까지의 DELTA 백업 이미지 파일들로부터 변경분을 복구합니다.

```
$ db2 restore db <DB명> INCREMENTAL
```

Tip

손상 직전의 시점으로 복구하려면, 마지막 INCREMENTAL 백업 이후부터 손상 직전 시점까지의 아카이브 로그 파일을 로그 디렉토리에 모두 복사하고, TO END OF LOGS AND STOP 옵션을 사용합니다.

4. 데이터베이스 로그 파일을 이용하여 rollforward database 명령으로 최근의 DELTA 백업 이미지 생성 시점 이후의 변경분을 복구합니다.

```
$ db2 rollforward db <DB명> to end of logs
$ db2 rollforward db <DB명> to <시간소인> using local time and stop
```

5. list history 명령어로 해당 롤포워드 복구와 연관된 기록을 확인합니다.

```
$ db2 list history rollforward for <DB명>
```

TOPIC 13-26 테이블스페이스의 상태

Point

테이블 스페이스의 컨테이너 액세스에 문제가 발생하면, 테이블 스페이스의 상태가 OFFLINE이 되어 해당 테이블 스페이스를 액세스할 수 없습니다. 문제가 발생한 테이블스페이스를 제외한 다른 테이블스페이스는 사용 가능합니다.

1 컨테이너에 문제가 발생한 테이블스페이스는 OFFLINE 상태가 되고, 사용할 수 없습니다.

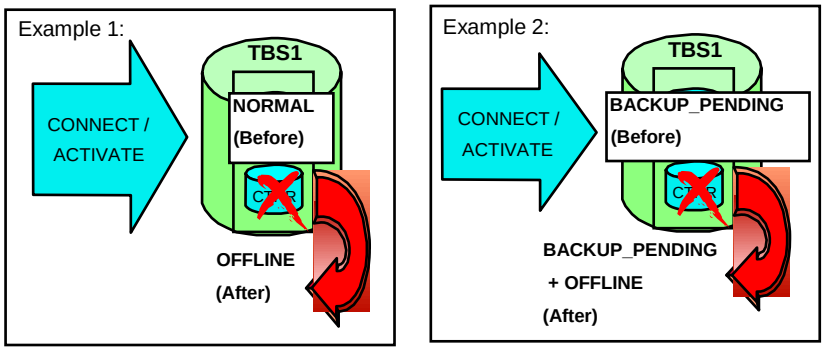


Figure 1326A • • • 테이블스페이스 오프라인

2 테이블 스페이스의 현재 상태를 확인합니다.

```
$ db2 "SELECT * FROM <테이블명>"
  SQL0290N  테이블 스페이스 액세스가 허용되지 않습니다.  SQLSTATE=55039
$ db2 list tablespaces | grep –p –i <테이블스페이스명>
  테이블 스페이스 ID                    = 3
  이름                                  = <테이블스페이스명>
  유형                                  = 데이터베이스 관리 스페이스
  내용                                  = 임의의 데이터
  상태                                  = 0x4000
  세부사항 설명:
    오프라인
$ db2tbst 0x4000
State = Offline and not accessible
```

3 테이블스페이스 컨테이너의 문제점을 해결하고, alter tablespace 문을 이용하여 테이블스페이스의 상태를 ON 으로 변환하고, 데이터를 정상적으로 액세스할 수 있습니다. 컨테이너의 문제점이 해결되지 않으면, SQL0293N 오류 코드와 함께 명령은 실패합니다.

```
$ db2 ALTER TABLESPACE <테이블스페이스명> SWITCH ONLINE
```

4 특정한 테이블스페이스 컨테이너에 문제점이 발생하여 데이터베이스가 활성화될 수 없다면, restart db 명령어를 이용하여 해당 테이블스페이스만 DROP PENDING 상태로 전환하고, 데이터베이스를 활성화할 수 있습니다.

```
$ db2 "RESTART DATABASE <DB명> DROP PENDING (<TS명>)"
$ db2 drop tablespace <TS명>
```

Tip

테이블스페이스의 주요 상태값은 다음과 같습니다.

0x0	Normal
0x1	Quiesced: SHARE
0x2	Quiesced: UPDATE
0x4	Quiesced: EXCLUSIVE
0x8	Load pending
0x10	Delete pending
0x20	Backup pending
0x40	Rollforward in progress
0x80	Rollforward pending
0x100	Restore pending
0x200	Disable pending
0x400	Reorg in progress
0x800	Backup in progress
0x1000	Storage must be defined
0x2000	Restore in progress
0x4000	Offline
0x8000	Drop pending

Tip

DROP PENDING 상태로 전환된 테이블스페이스는 접속 후에 DROP 하고 재생성해야 합니다.

TOPIC 13.27 테이블스페이스 복구

Point

RESTORE DB 명령어로 테이블스페이스 백업 이미지 파일을 이용하여 특정 테이블스페이스만 과거 시점으로 복구하고, ROLLFORWARD DB 명령어로 아카이브 로그를 적용하여 다른 테이블스페이스와의 일관성을 복구합니다.

Tip

SYSADM,SYSCTRL,SYSMAINT 권한이 있는 사용자가 실행합니다.

1 원하는 시점의 TABLESPACE 백업 이미지 파일과 아카이브 로그 파일을 준비하고, restore db 명령어와 rollforward db 명령어를 실행하면, 테이블스페이스를 복구할 수 있습니다.

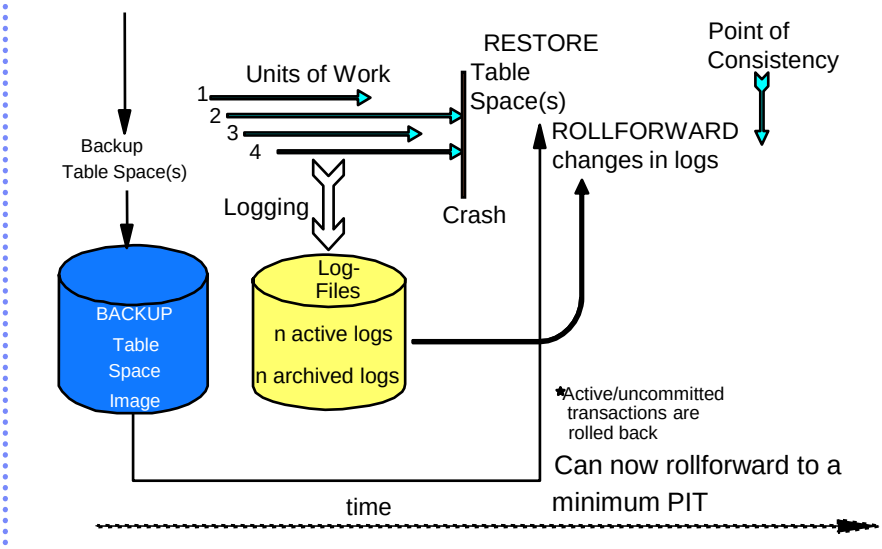


Figure 1327A ••• 테이블스페이스 백업 이미지를 이용한 복구

Tip

일관성을 보장하기 위해 적용되어야 할 최소한의 로그 파일 번호는 LIST HISTORY 명령어를 이용하여 복구 기록 파일에서 확인할 수 있습니다.

2 원하는 TABLESPACE 백업 이미지 파일을 이용하여 restore db 명령으로 데이터베이스의 특정 테이블스페이스를 과거 시점으로 복구합니다. restore db 명령어가 완료되면, 해당 테이블스페이스가 '롤포워드 보류' 상태에 놓이게 됩니다.

```
$ db2 "restore db <DB명> TABLESPACE (<TS명>) ONLINE"
$ db2 connect to <데이터베이스명>
$ db2 "SELECT * FROM <테이블명>"
SQL0290N  테이블 스페이스 액세스가 허용되지 않습니다.  SQLSTATE=55039
$ db2 list tablespaces | grep -p –i <테이블스페이스명> | grep "상태"
 상태                                = 0x0080
```

Tip

RESTORE DB 명령어 또는 ROLLFORWARD DB 명령어를 실행한 세션의 데이터베이스에 대한 접속은 해제됩니다.

3 rollforward db 명령어의 TO 옵션으로 해당 테이블스페이스와 다른 테이블스페이스의 일관성을 보장할 수 있는 시점 이상의 로그 파일을 복구합니다.

```
$ db2 ROLLFORWARD DB <DB명> TO END OF LOGS AND STOP
  TABLESPACE (<TS명>) ONLINE"
SQL4907W  데이터베이스 "<데이터베이스명>"이(가) 복구되었으나, 롤 포워드 조작을
  위해 포함된 테이블 스페이스 목록에서 하나 이상의 테이블이 점검 보류 상태에 있습니다.
```

Tip

복구된 테이블스페이스에 존재하는 테이블이 다른 테이블과 RI 관계에 있는 부모 테이블이라면, 자손 테이블은 '점검 보류' 상태에 놓이게 됩니다. SET INTEGRITY 문을 이용하여 해결합니다.

4 복구된 테이블스페이스에 포함된 테이블의 데이터를 정상적으로 액세스할 수 있습니다.

```
$ db2 "SELECT * FROM <테이블명>"
```

Memo