UNIT 05

데이터베이스

하나의 인스턴스에는 독립적인 환경을 가지는 한 개 이상의 데이터베이스를 생성할 수 있습니다. 데이터베이스는 시스템 카탈로그와 로그 파일을 가지며, 테이블을 비롯한 여러 오브젝트들을 생성하여 사용하게 됩니다. CREATE DB 명령어로 생성하고, ACTIVATE DB 명령어로 기동시킵니다.

DB2 9.7 운영자 가이드

Administrator Edition

- 데이터베이스
- 데이터베이스 생성과 제거
- 시스템 카탈로그
- 데이터베이스 구성 파일
- 데이터베이스 기동과 중지
- 데이터베이스 접속과 해제
- 데이터베이스에 접속된 응용프로그램 목록
- 응용프로그램 강제 종료
- 원격 노드 등록
- 지역 노드 등록
- 원격 데이터베이스 등록
- 시스템 데이터베이스 목록
- 지역 데이터베이스 목록

TOPIC 05 01 데이터베이스

Point

한 인스턴스에서 한 개 이상의 데이터베이스를 생성할 수 있습니다. 각 데이터베이스별로 데이터베이스 구성 파일, 기본 테이블스페이스, 시스템 카탈로그 테이블, 데이터베이스 로그 파일 등이 생성됩니다.

Tip

제품을 설치하면 SAMPLE이라는 이름의 기본 데이터베이스가 제공됩니다. db2sampl 이라는 명령어로 생성할 수 있습니다.

1 한 인스턴스에서 한 개 이상의 데이터베이스를 생성할 수 있습니다.

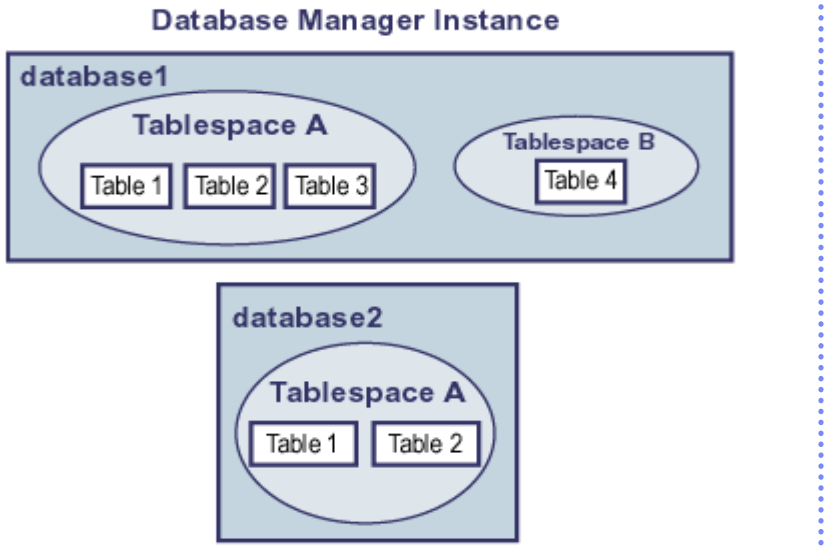

Figure 0501A ··· 동일한 인스턴스에 생성된 두 개의 데이터베이스

2 데이터베이스 구성 파일을 이용하여 개별적인 환경을 구성할 수 있습니다. 데이터베이스 구성 파일에는 다양한 데이터베이스 구성 변수가 저장됩니다.

3 3개의 기본 테이블스페이스인 SYSCATSPACE, TEMPSPACE1, USERSPACE1을 가지고 있으며, 사용자가 다양한 유형의 테이블스페이스를 추가할 수 있습니다.

4 SYSCATSPACE 테이블스페이스에는 데이터베이스의 모든 오브젝트에 대한 정보를 저장하고 있는 메타 테이블인 시스템 카탈로그 테이블이 생성됩니다.

5 데이터베이스의 데이터에 대한 변경 사항은 데이터베이스 트랜잭션 로그 파일에 기록됩니다. 초기에는 3개의 기본 로그 파일과 2개의 보조 로그 파일이 생성되며, 데이터베이스 구성 파일을 이용하여 로그 파일의 개수와 크기를 조절할 수 있습니다.

6 테이블스페이스, 테이블, 인덱스 등의 다양한 오브젝트를 생성하고, 사용자의 데이터를 테이블에 저장하여 SQL문으로 액세스합니다.

Tip

이종DBMS(오라클,MS-SQL)의 테이블과 JOIN 하려면 IFS라는 제품을 추가적으로 설치해야 합니다.

7 서로 다른 데이터베이스에 속한 테이블들은 한 개의 SQL문으로 JOIN 하려면 FEDERATED 기능을 이용합니다. DB2 제품군에 대한 FEDERATED 기능은 기본적으로 제공됩니다.

TOPIC 05 02 데이터베이스 생성과 제거

Point

인스턴스 사용자가 CREATE DB 명령어로 데이터베이스를 생성합니다. ON 옵션으로 데이터베이스와 관련된 파일들이 저장되는 디렉토리를 지정하며, CODESET 옵션으로 코드 페이지를 결정합니다. DROP DB 명령어로 제거합니다.

1 create db 명령어를 이용하여 데이터베이스를 생성합니다.

```
$ login <인스턴스 사용자명>
$ db2 CREATE DB <데이터베이스명>
$ ls –lia <인스턴스 사용자의 홈디렉토리>/$DB2INSTANCE/NODE*
```

Tip

Windows 에서는 ON 옵션에서 디렉토리명 대신에 드라이브명을 지정합니다.

2 기본 디렉토리가 아닌 다른 디렉토리에 데이터베이스 파일을 생성하려면 create db 명령어에서 ON 옵션을 사용합니다.

```
$ db2 CREATE DB <데이터베이스명> ON <디렉토리명>
$ ls –lia <디렉토리명>/$DB2INSTANCE/NODE*
```

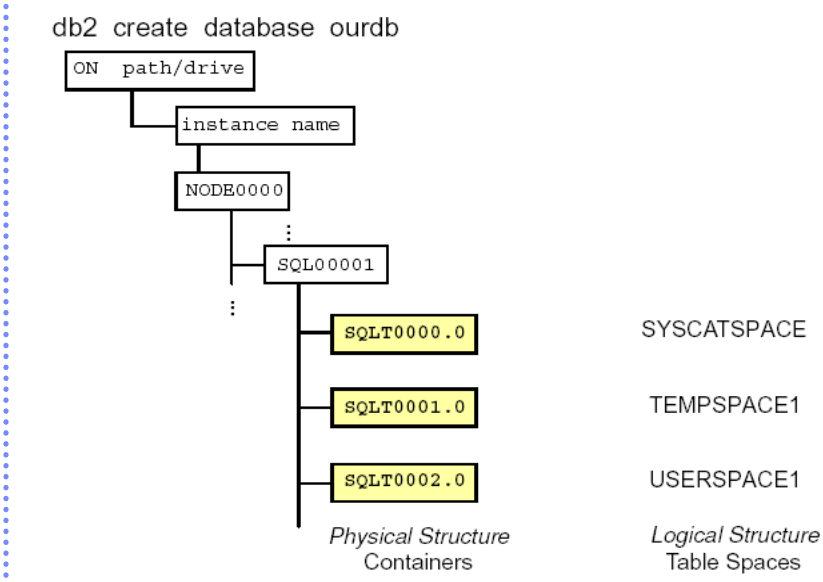

Figure 0502A • • • 데이터베이스와 관련된 서브디렉토리

Tip

데이터베이스와 세션의 코드 페이지가 호환되지 않으면, SQL0332N 오류가 반환되어 데이터베이스에 접속할 수 없습니다. 세션의 코드 페이지를 변경하거나, DB2 레지스터리 변수인 DB2CODEPAGE 를 이용합니다.

3 데이터베이스는 생성시에 사용하는 코드 페이지가 결정되고 변경할 수 없습니다. CODESET과 TERRITORY 옵션을 이용하여 데이터베이스를 위한 코드페이지를 지정할 수 있습니다. 기본값은 현재 세션에 적용된 코드 페이지로 결정됩니다.

```
$ db2 CREATE DB <데이터베이스명> USING CODESET <코드셋>
   TERRITORY <국가코드>
$ db2 GET DB CFG FOR <데이터베이스명> | grep <국가코드>
```

4 데이터베이스가 중지된 상태에서 drop db 명령어를 이용하여 제거합니다.

```
$ db2 drop db <데이터베이스명>
```

TOPIC 05 03 시스템 카탈로그

Point

SYSCATSPACE 테이블스페이스에 기본적인 시스템 카탈로그 테이블이 생성되어 데이터베이스에 대한 메타데이터를 저장합니다. 카탈로그 테이블은 SYSIBM, SYSIBMADM, SYSPUBLIC, SYSCAT, SYSSTAT 라는 스키마명을 가집니다.

Tip

카탈로그에 저장된 데이터베이스 오브젝트인 테이블, 인덱스 등의 이름은 대문자로 저장됩니다.

1. <DB2 사용자>로 로그인합니다.

```
$ login <DB2 사용자명>
```

2. 데이터베이스에 접속합니다.

```
$ db2 CONNECT TO <데이터베이스명>
```

3. 시스템 카탈로그 테이블의 목록을 확인합니다. 카탈로그는 SYSIBM, SYSCAT, SYSSTAT 라는 스키마명을 가집니다.

```
$ db2 LIST TABLES FOR SYSTEM
```

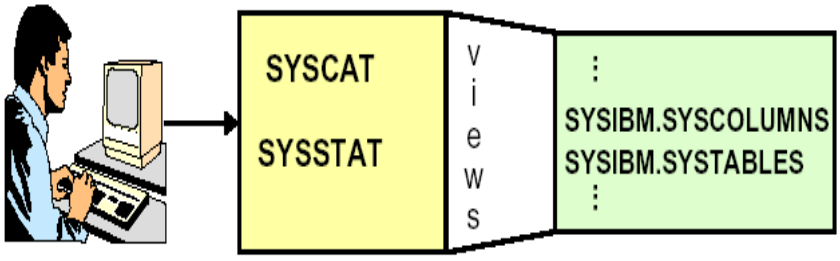

Figure 0503A ••• 시스템 카탈로그

Tip

카탈로그에 저장된 데이터베이스 오브젝트인 테이블, 인덱스 등의 이름은 대문자로 저장됩니다.

4. 시스템 카탈로그 테이블의 정보를 확인합니다.

```
$ db2 “select * from <시스템 카탈로그 테이블>”
```

```
$ db2 “select * from sysibm.systables where name = 'EMPLOYEE' ”
$ db2 “SELECT * FROM SYSIBM.SYSTABLES WHERE NAME = 'ORG' ”
```

Figure 0503B ••• 시스템 카탈로그 테이블의 정보 조회

Tip

시스템 카탈로그 테이블보다 시스템 카탈로그 뷰를 액세스하는 것이 보다 편리합니다.

Tip

시스템 카탈로그 테이블과 뷰는 기본적으로 조회만 가능하고, SYSSTAT 스키마를 가진 시스템 카탈로그 뷰는 UPDATE도 허용됩니다. (SQL튜닝 목적)

5. 시스템 카탈로그 뷰의 정보를 확인합니다.

```
$ db2 “select * from <시스템 카탈로그 뷰>”
```

```
$ db2 “select * from syscat.tables where tanname = 'EMPLOYEE' ”
$ db2 “SELECT * FROM SYSCAT.TABLES WHERE TABNAME = 'ORG' ”
```

Figure 0503C ••• 시스템 카탈로그 뷰의 정보 조회

TOPIC 05 04 데이터베이스 구성 파일

Point

데이터베이스를 생성하면 데이터베이스 구성 파일이 생성됩니다. 다양한 데이터베이스 구성 변수를 이용하여 고유한 환경을 구성할 수 있으며, 인스턴스 사용자가 GET DB CFG, UPDATE DB CFG, RESET DB CFG 명령어를 이용하여 관리합니다.

Tip

get db cfg 명령어의 show detail 옵션을 사용하려면 connect 명령어를 이용하여 데이터베이스에 접속해야 합니다.

1 <인스턴스 사용자>는 데이터베이스에 접속하여 get db cfg 명령어로 데이터베이스 구성 변수의 설정값을 확인합니다.

```
$ login <인스턴스 사용자명>
$ db2 connect to <데이터베이스명>
$ db2 get db cfg for <데이터베이스명> show detail
$ db2 connect reset
```

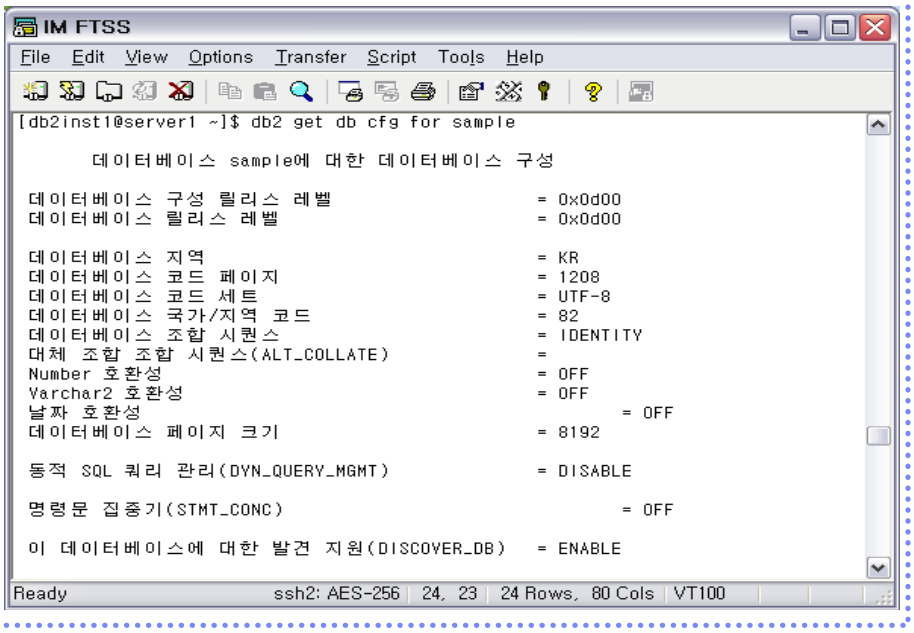

Figure 0504A • • • GET DB CFG 명령어

Tip

값을 변경하는 즉시 반영되는 데이터베이스 구성 변수를 '온라인 구성 가능한 데이터베이스 구성 변수' 라고 합니다.

2 update db cfg 명령어로 데이터베이스 구성 변수를 변경합니다.

```
$ db2 connect to <데이터베이스명>
$ db2 update db cfg for <데이터베이스명> using <구성변수명> <값>
$ db2 connect reset
```

3 reset db cfg 명령어는 데이터베이스의 모든 구성 변수를 초기화 상태로 변경합니다. 일부 데이터베이스 구성 변수는 해당 데이터베이스에 접속된 모든 응용프로그램을 종료한 후에 데이터베이스를 재기동해야 반영됩니다.

```
$ db2 reset db cfg for <데이터베이스명>
$ db2 force applications all
$ db2stop force
$ db2start
$ db2 connect to <데이터베이스명>
$ db2 get db cfg for <데이터베이스명> show detail
```

TOPIC 05-05 데이터베이스 기동과 중지

Point

ACTIVATE DB 명령어를 이용하여 데이터베이스를 기동하면, 데이터베이스와 관련된 프로세스들이 생성되고, 버퍼풀을 포함한 공유 메모리가 할당됩니다. DEACTIVATE DB 명령어로 데이터베이스를 중지한 후에는 데이터베이스를 액세스할 수 없습니다.

Tip

데이터베이스의 활성화는 인스턴스가 시작된 상태에서 가능합니다.

Tip

활성화된 데이터베이스를 deactivate db 명령어로 비활성화하고, activate db 명령어로 다시 활성화시키는 것을 데이터베이스의 재활성화라고 합니다.

Tip

데이터베이스가 활성화되지 않은 상태에서 activate db 명령어 대신에 connect to 명령어를 실행하면 자동으로 데이터베이스가 활성화되고 첫 번째 접속이 완료됩니다.

Tip

activate database 명령어를 실행하지 않은 상태로 데이터베이스가 기동된 경우에는 마지막 응용프로그램이 접속을 해제하는 순간 자동으로 데이터베이스가 중지됩니다.

1 <인스턴스 사용자>로 로그인하여 데이터베이스를 활성화시킵니다.

```
$ login <인스턴스 사용자명>
$ db2 activate db <데이터베이스명>
```

2 연관된 프로세스의 목록과 메모리를 확인합니다.

```
$ ps –ef | grep <인스턴스명>
$ db2mtrk -d
메모리 추적 설정: 21:53:53에서 2009/07/29
데이터베이스용 메모리: SAMPLE

  utilh      pckcacheh   other      catcacheh   bph (1)     bph (S32K)
  64.0K      192.0K      128.0K     64.0K       8.2M        832.0K

  bph (S16K)  bph (S8K)   bph (S4K)   shsorth     lockh       dbh
  576.0K      448.0K      384.0K      0           16.7M       20.9M
```

3 데이터베이스를 기동하기 위해 activate database 명령어를 사용했다면, deactivate database 명령어를 사용하여 중지시킵니다.

```
$ db2 deactivate db <데이터베이스명>
$ db2mtrk -d
활성화된 데이터베이스가 없음
```

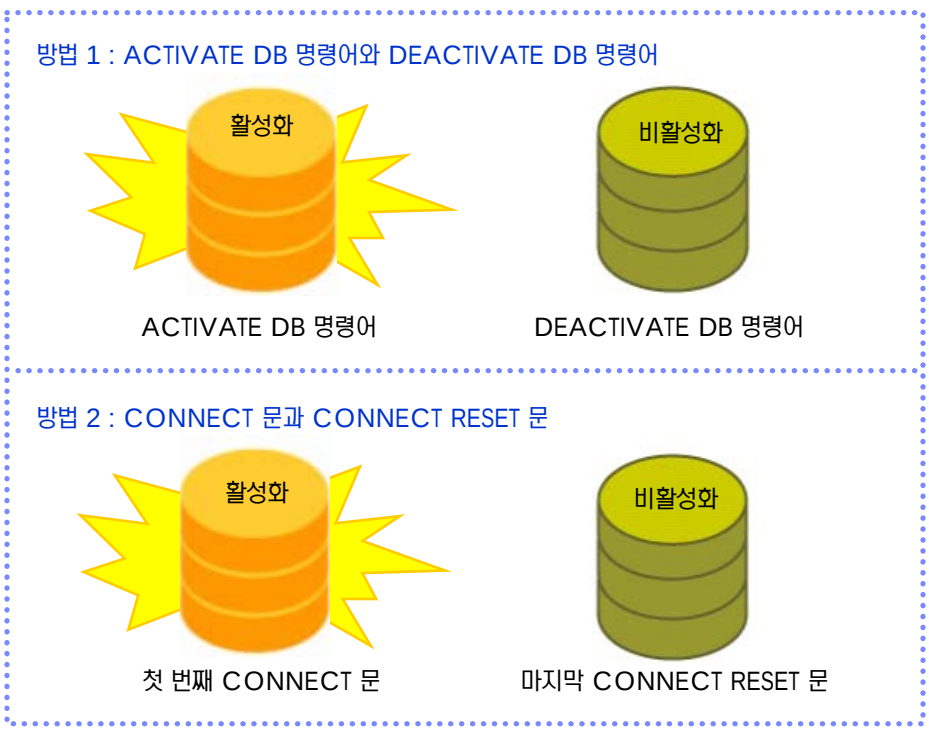

Figure 0505A... 데이터베이스의 활성화와 비활성화

TOPIC 05 06 데이터베이스 접속과 해제

Point

DB2 명령어와 SQL문을 이용하여 데이터베이스에 대한 액세스를 시작하려면 활성화된 데이터베이스에 대한 접속이 필요합니다. CONNECT 명령어를 이용하여 데이터베이스에 접속하고, CONNECT RESET 명령어를 이용하여 접속을 해제합니다.

Tip

activate database 를 사용하여 데이터베이스를 활성화한 후에 connect 명령어로 데이터베이스에 접속하는 것을 권장합니다.

Tip

데이터베이스에 접속할 때는 <데이터베이스 별명>을 이용합니다.

1 connect 문을 이용하여 데이터베이스에 접속한 후에 액세스가 가능합니다.

```
$ login <DB2 사용자명>
$ db2 connect to <데이터베이스명>
$ db2 “select * from syscat.tables”
```

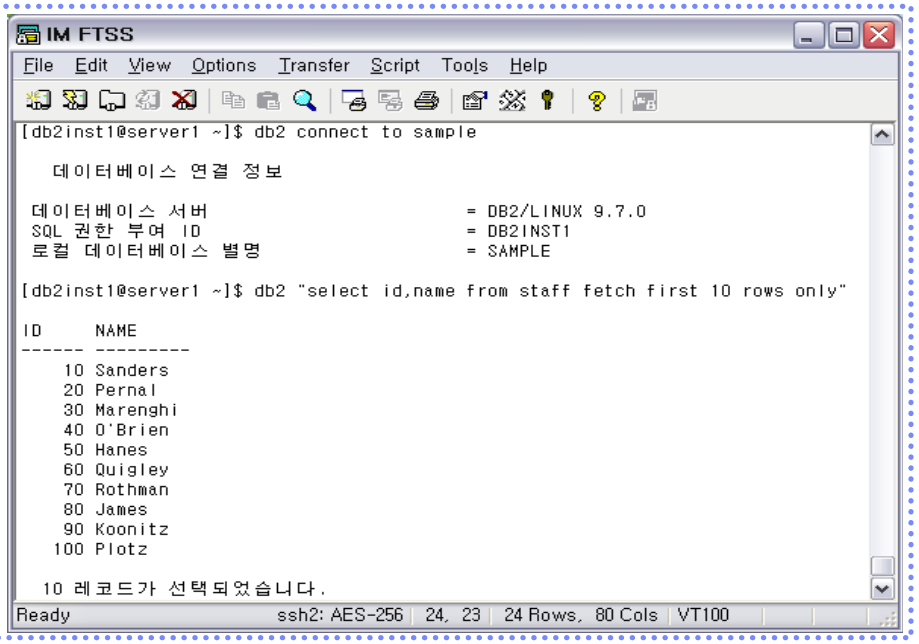

Figure 0506A ••• CONNECT 명명어

Tip

terminate 명령어를 사용하면 현재의 데이터베이스 접속을 해제하고, CLP사용자의 DB2 세션을 완전히 종료합니다.

2 데이터베이스에 대한 접속을 해제하려면, connect reset 문을 이용합니다.

```
$ db2 connect reset
```

3 원격 데이터베이스에 접속하려면, 반드시 사용자명과 암호를 지정해야 합니다.

```
$ db2 connect to <데이터베이스명> user <사용자명> using <암호>
$ db2 “select * from syscat.tables”
$ db2 connect reset
```

Tip

DFT 옵션을 사용하는 경우에는 DB2DBDFT 레지스터리 변수를 설정합니다.

Tip

DB2DBDFT 레지스터리 변수는 현재의 DB2 세션에는 적용되지 않으므로 terminate 가 필요합니다.

4 DB2DBDFT 라는 레지스터리 변수를 이용하면, connect 문을 실행하지 않고 SQL문을 요청했을 때, 자동으로 데이터베이스에 접속됩니다.

```
$ db2set DB2DBDFT=<데이터베이스명>
$ db2 terminate
$ db2 “select * from syscat.tables”
```

TOPIC 05 07 데이터베이스 접속된 응용프로그램 목록

Point

데이터베이스에 접속을 요청하면 중계자 프로세스인 db2agent 가 생성되어 활동에 필요한 개별 메모리를 할당받고, 접속된 응용프로그램은 고유한 핸들 번호로 엔진에 의해 관리됩니다. LIST APPLICATIONS 명령어로 확인합니다.

Tip

db2 명령어를 실행하면 DB2 세션이 시작되어 db2bp 프로세스가 생성되고, connect 명령을 실행하면 db2agent 프로세스가 생성됩니다.

Tip

Show detail로 내용에서 코디네이터 pid 항목은 데이터베이스에 접속을 요청한 응용프로그램의 pid가 아니라, db2agent 프로세스의 pid 입니다.

Tip

db2agent 프로세스를 위한 개별 메모리는 초기에는 최소량으로 할당되고, SQL문이 처리되는 동안 필요한 메모리가 추가적으로 할당됩니다.

1 <DB2 사용자>로 로그인하여 데이터베이스에 접속합니다.

```
$ login <DB2 사용자명>
$ db2 connect to <데이터베이스명>
$ db2 "select * from syscat.tables"
```

2 접속된 응용프로그램의 정보를 확인합니다. show detail 옵션이 제공됩니다.

```
$ db2 list applications
$ db2 list applications show detail
```

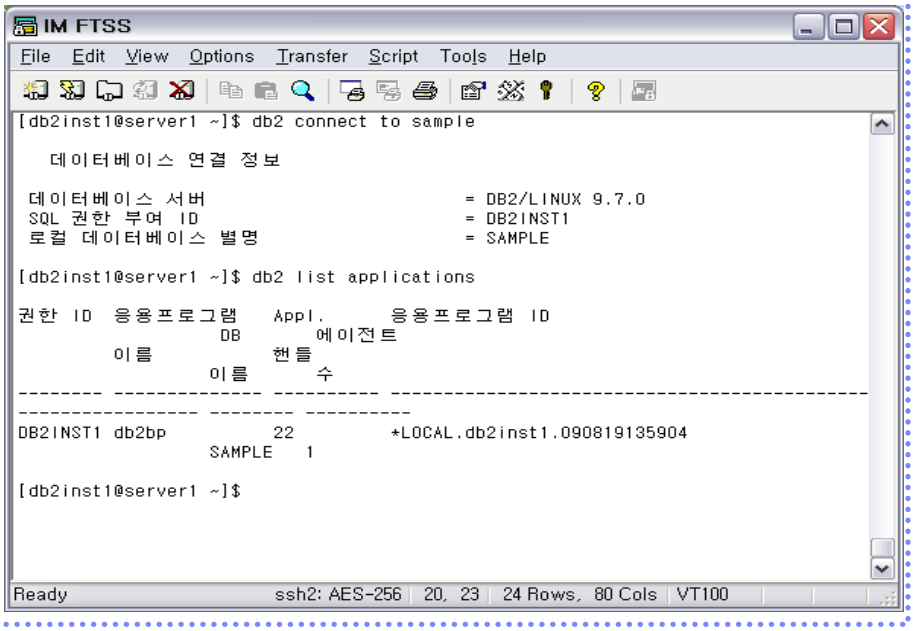

Figure 0507A • • • LIST APPLICATIONS 명령어

3 응용프로그램이 데이터베이스에 접속을 요청하면 db2agent 가 생성됩니다. db2agent 프로세스는 개별 메모리를 할당받습니다.

```
$ ps –ef | grep <인스턴스명> | grep db2agent
$ db2mtrk -p
에이전트 5284006용 메모리
  other    apph    appctlh
  16.0K    128.0K  16.0K
```

4 데이터베이스에 대한 접속을 해제하면, db2agent는 제거 되거나 idle 상태로 바뀌게 됩니다.

```
$ db2 connect reset
$ db2pd -d sample -edu | grep -i db2agent
```

TOPIC 05 08 응용프로그램 강제 종료

Point

데이터베이스에 접속했던 응용프로그램은 CONNECT RESET 명령어로 접속을 종료해야 합니다. FORCE APPLICATION 명령어는 응용프로그램의 핸들 번호를 이용하여 응용프로그램을 강제 종료합니다. 강제로 종료된 응용프로그램은 롤백됩니다.

Tip

force application 명령어는 비동기 모드로 처리되므로, 응용프로그램이 즉시 종료되지 않을 수 있습니다.

Tip

강제로 종료된 응용프로그램은 마지막으로 실행 중이던 트랜잭션의 롤백을 완료해야 하므로 종료되는데 상당한 시간이 걸릴 수도 있습니다.

Tip

SQL문이 실행 중인 경우에는, force application 명령을 실행하고, list application 명령어로 결과를 확인하도록 합니다. 긴 트랜잭션이 있었다면, Rollback으로 인해 정지하는데 시간이 소요될 수 있습니다.

1 <인스턴스 사용자>로 로그인하여 데이터베이스에 접속합니다.

```
$ login <DB2 사용자명>
$ db2 connect to <데이터베이스명>
$ db2 “select * from syscat.tables”
```

2 접속된 응용프로그램의 이름과 그 핸들 번호를 확인합니다. 핸들 번호는 응용프로그램에게 할당되는 OS의 프로세스 id가 아니라, DB2 엔진이 부여한 고유 번호입니다.

```
$ db2 list applications
```

3 force applications 명령어에서 핸들 번호를 이용하여 특정한 응용프로그램을 강제로 종료할 수 있습니다. 여러 개의 응용프로그램을 강제 종료하려면 ,(컴마) 를 이용하여 핸들 번호를 나열합니다.

```
$ db2 “force application (<핸들 번호>)”
$ db2 “force application (<핸들 번호 1>, <핸들 번호 2>)”
$ db2 list applications
```

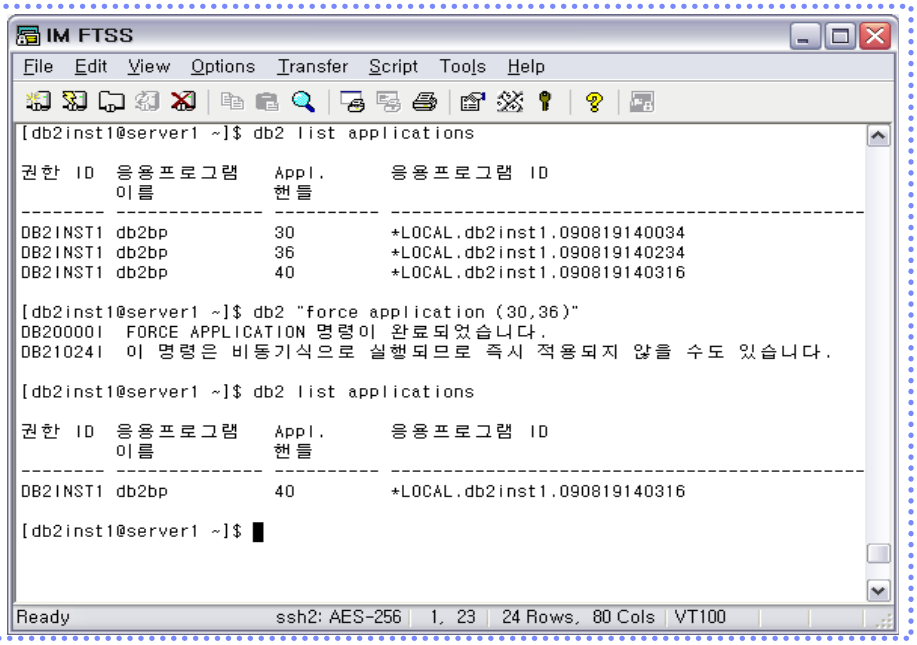

Figure 0508A • • • FORCE APPLICATION 명령어

4 접속된 모든 응용프로그램을 강제로 종료하려면 all 옵션을 사용합니다.

```
$ db2 force applications all
$ db2 list applications
```

TOPIC 05 09 원격 노드 등록

Point

원격 서버의 인스턴스에 존재하는 데이터베이스를 액세스하려면 <원격 서버의 IP 주소>와 <원격 서버 인스턴스의 TCP/IP 서비스 포트 번호>를 이용하여 원격 노드를 등록해야 합니다. CATALOG TCPIP NODE 명령어를 이용합니다.

Tip

Windows인 경우에는 C:\WINDOWS\system32\drivers\etc 디렉토리의 hosts 파일과 services 파일을 이용합니다.

1 서버에서 인스턴스 사용자로 로그인하여 <IP 주소>를 확인합니다. 이 값이 <원격 서버의 IP 주소> 입니다.

```
$ login <서버의 인스턴스 사용자명>
$ cat /etc/hosts
```

2 서버의 인스턴스 구성 변수인 SVCENAME 의 값이 서비스명이면/etc/services 파일을 확인합니다. 이 값이 < 원격 인스턴스의 TCP/IP 서비스 포트 번호>입니다.

```
$ db2 get dbm cfg | grep SVCENAME
$ cat /etc/services
```

3 클라이언트에서 인스턴스 사용자로 로그인하여 catalog tcpip node 명령어로 원격 서버의 인스턴스를 등록합니다. <노드명>은 임의로 정할 수 있으며, 서버에서 확인한 <원격 서버의 IP 주소>와 <원격 인스턴스의 TCP/IP 서비스 포트 번호>를 이용합니다.

Tip

원격 서버를 나타내는 <원격서버의 IP 주소> 대신에 클라이언트의 /etc/hosts 파일을 이용한 <호스트명>을 사용할 수 있습니다.

```
$ login <클라이언트의 인스턴스 사용자명>
$  db2 catalog tcpip node <노드명> remote <원격 서버의 IP 주소>  server <원격 인스턴스의 TCP/IP 서비스 포트 번호>
$ db2 list node directory
```

Tip

원격 인스턴스의 <원격 인스턴스의 TCP/IP 서비스 포트 번호> 대신에 클라이언트의 /etc/services 파일을 이용한 <서비스명>을 사용할 수 있습니다.

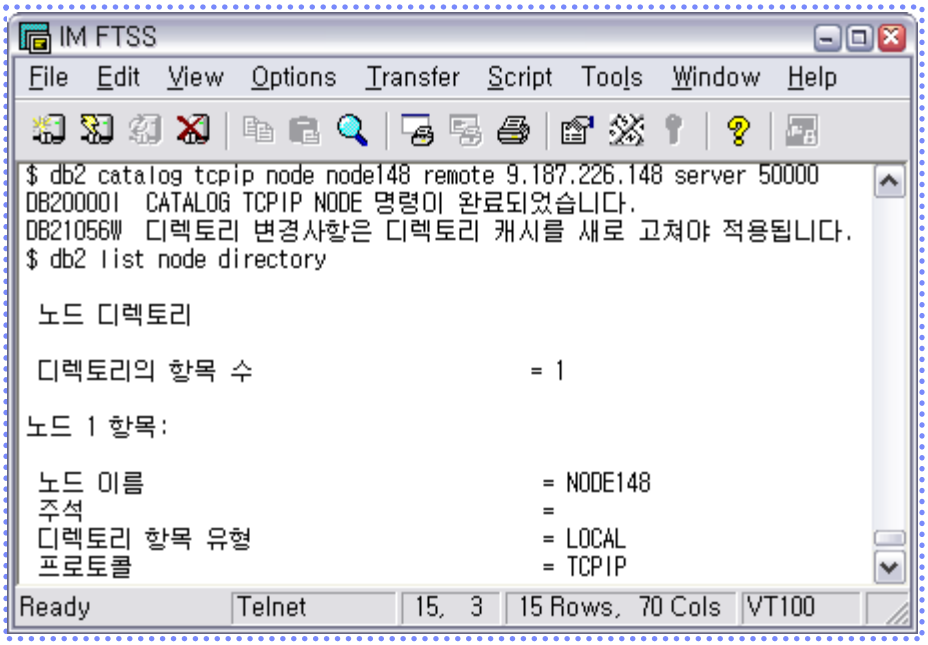

Figure 0509A ••• CATALOG TCPIP NODE 명령어

4 등록된 원격 노드는 uncatalog node 명령어로 제거합니다.

```
$ db2 uncatalog node <노드명>
```

TOPIC 05-10 지역 노드 등록

Point

동일한 서버에 존재하는 다른 인스턴스를 지역 노드라고 합니다. CATALOG LOCAL NODE 명령어를 이용하여 지역 노드를 등록하면, 등록된 지역 노드에 존재하는 데이터베이스를 액세스할 수 있습니다.

Tip

인스턴스명만 지정하면, TCP/IP 통신에 필요한 IP주소와 서비스 포트 번호를 내부적으로 인식합니다.

1. 인스턴스 사용자로 로그인합니다.

```
$ login <인스턴스 사용자명>
```

2. catalog local node 명령어로 다른 인스턴스를 등록합니다. <노드명>은 임의로 정할 수 있습니다. 인스턴스명을 이용합니다.

```
$ db2 catalog local node <노드명> instance <다른 인스턴스명>
```

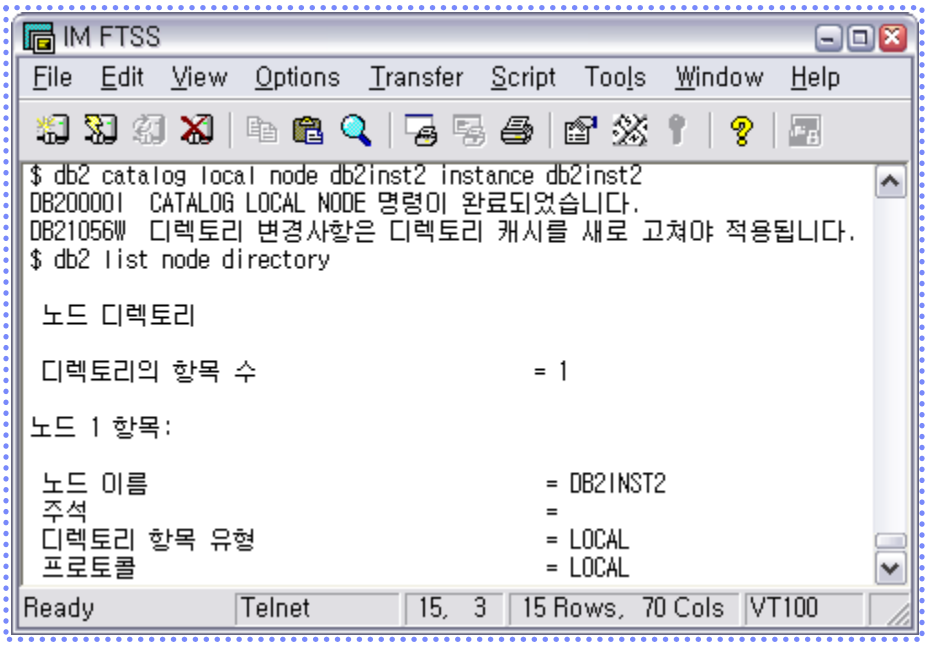

Figure 0510A CATALOG LOCAL NODE 명령어

3. catalog tcpip node 명령어로 원격 서버의 인스턴스와 동일한 방법으로 등록할 수도 있습니다. <노드명>은 임의로 정할 수 있으며, <현재 서버의 IP 주소>와 <지역 인스턴스의 TCP/IP 서비스 포트 번호>를 이용합니다.

```
$ login <인스턴스 사용자명>
$ db2 catalog tcpip node <노드명> remote <현재 서버의 IP 주소> server <지역 인스턴스의 TCP/IP 서비스 포트 번호>
```

4. list node directory 명령어로 등록한 지역 노드의 목록을 확인합니다.

```
$ db2 list node directory
```

5. 등록한 지역 노드는 uncatalog node 명령어로 제거합니다.

```
$ db2 uncatalog node <노드명>
```

TOPIC 05-11 원격 데이터베이스 등록

Point

원격 노드 또는 지역 노드에 존재하는 원격 데이터베이스는 CATALOG DB 명령어를 이용하여 원하는 데이터베이스 별명으로 등록하여 액세스합니다.

1 서버에서 인스턴스 사용자로 로그인하여 <원격 데이터베이스의 별명>을 확인합니다.

```
$ login <서버의 인스턴스 사용자명>
$ db2 list db directory
```

2 클라이언트에서 인스턴스 사용자로 로그인하여 list node 명령어로 원격 데이터베이스가 존재하는 <원격 노드명>을 확인합니다.

```
$ login <클라이언트의 인스턴스 사용자명>
$ db2 list node directory
```

Tip

AS 옵션을 지정하지 않으면 원격 데이터베이스의 별명과 동일한 별명으로 등록됩니다.

3 catalog db 명령어를 이용하여 원격 데이터베이스를 등록합니다. <등록할 데이터베이스 별명> 은 고유한 데이터베이스 별명으로 임의로 정합니다.

```
$ db2 catalog <원격 데이터베이스의 별명> as <등록할 데이터베이스 별
    명> at node <원격 노드명>
```

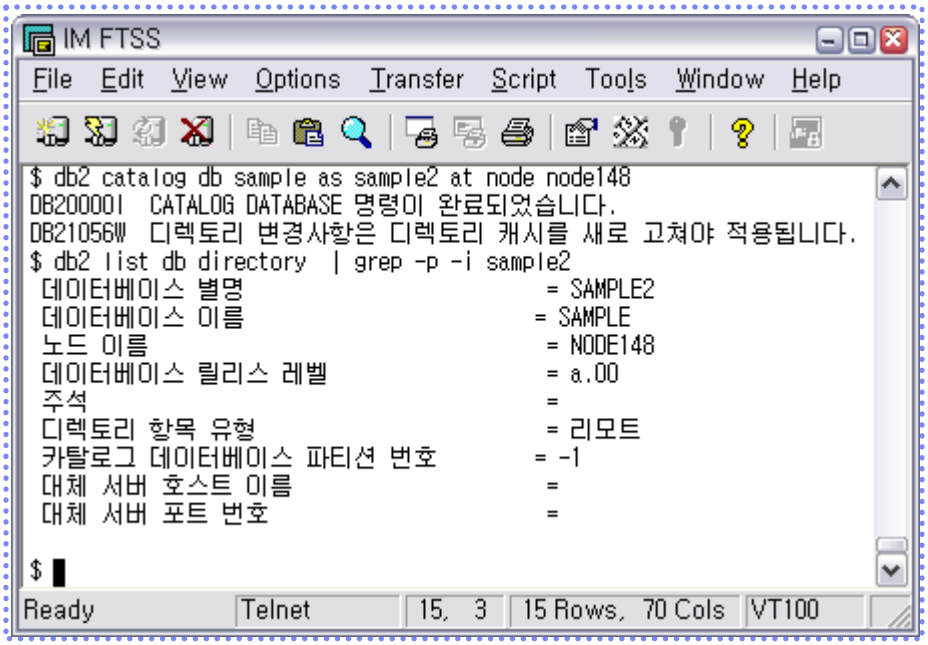

Figure 0511A ··· CATALOG DB 명령어

Tip

원격 데이터베이스에 접속할 때는 반드시 서버의 OS 계정의 <사용자명>과 <암호>를 지정해야 합니다.

4 connect 문을 이용하여 원격 데이터베이스에 접속합니다.

```
$ db2 connect to <등록 데이터베이스 별명>  user <원격 사용자명> using <원격
    사용자의 암호명>
```

Tip

db2ubind.lst 파일의 바인드 작업은 클라이언트 플랫폼 유형별로 한 번만 실행하면 됩니다.

5 원격 데이터베이스에 대해 DB2 유틸리티를 실행하려면 db2ubind.lst 파일을 바인드합니다.

```
$ cd <인스턴스 사용자의 홈디렉토리>/sqllib/bnd
$ db2  bind @db2ubind.lst blocking all grant public
```

TOPIC 05-12 시스템 데이터베이스 목록 확인

Point

현재 인스턴스에 등록된 모든 데이터베이스 목록을 시스템 데이터베이스 목록이라고 합니다. 현재 인스턴스에서 생성된 지역 데이터베이스와 등록된 원격 데이터베이스의 목록을 모두 포함합니다. LIST DB DIRECTORY 명령어를 이용합니다.

Tip

원격 데이터베이스는 '리모트' 유형으로 분류되며, catalog node 명령어로 등록된 원격 노드과 catalog db 명령어로 등록한 데이터베이스 별명이 표시됩니다.

Tip

지역 데이터베이스는 '간접' 유형으로 분류되며, create db 명령어에서 지정된 데이터베이스 생성 디렉토리 명이 표시됩니다.

Tip

create db 명령어로 데이터베이스를 생성하면 데이터베이스명과 동일한 데이터베이스 별명이 자동으로 등록됩니다.

1 <DB2 사용자>로 로그인합니다.

```
$ login <DB2 사용자명>
```

2 list db directory 명령어로 시스템 데이터베이스의 목록을 확인합니다.

```
$ db2 list db directory
```

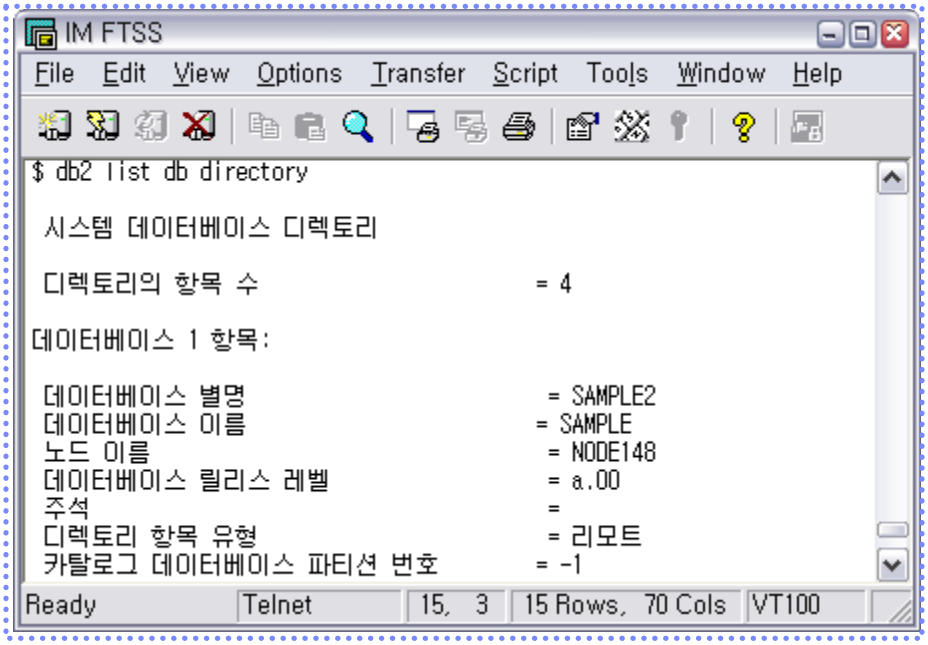

Figure 0512A ••• LIST DB DIRECTORY 명령어로 원격 데이터베이스 정보 확인

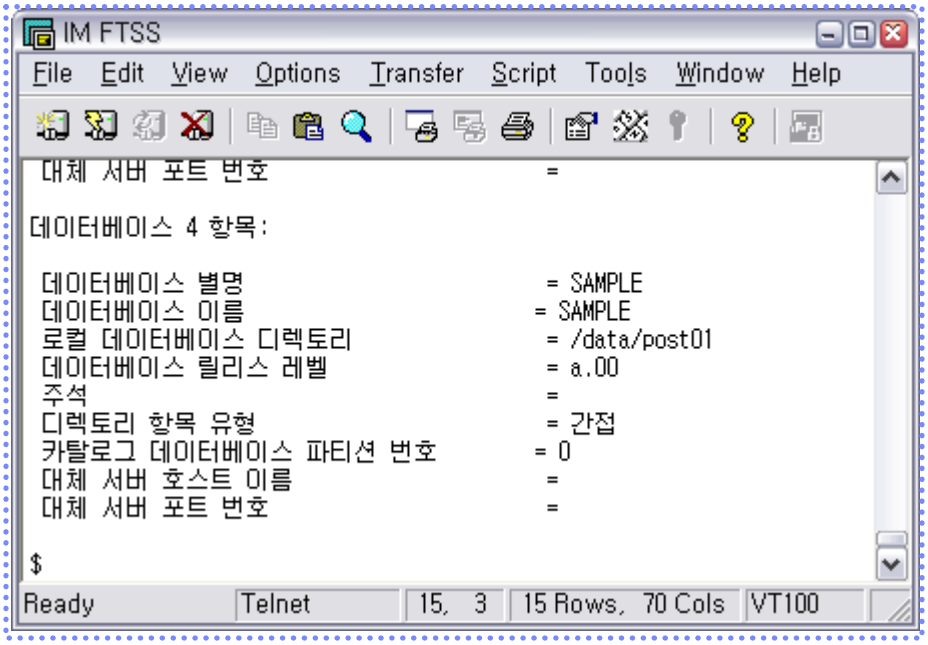

Figure 0512B ••• LIST DB DIRECTORY 명령어로 지역 데이터베이스 정보 확인

TOPIC 05-13 지역 데이터베이스 목록 확인

Point

현재 인스턴스에 실제로 존재하는 데이터베이스 중에서 특정한 디렉토리 또는 드라이브에 실제로 생성되어 있는 데이터베이스의 목록을 지역 데이터베이스 목록이라고 하며, LIST DB DIRECTORY 명령어의 ON 옵션으로 확인합니다.

Tip

'데이터베이스 디렉토리' 항목에 표시된 값은 현재 서버에서 데이터베이스와 관련된 실제 파일들이 생성된 서브 디렉토리명입니다.

1. <DB2 사용자>로 로그인합니다.

```
$ login <DB2 사용자명>
```

2. list db directory 명령어의 ON 옵션을 이용하여 특정한 <디렉토리명>에 실제로 생성되어 있는 <지역 데이터베이스>의 목록을 확인합니다.

```
$ db2 list db directory ON <디렉토리명>
```

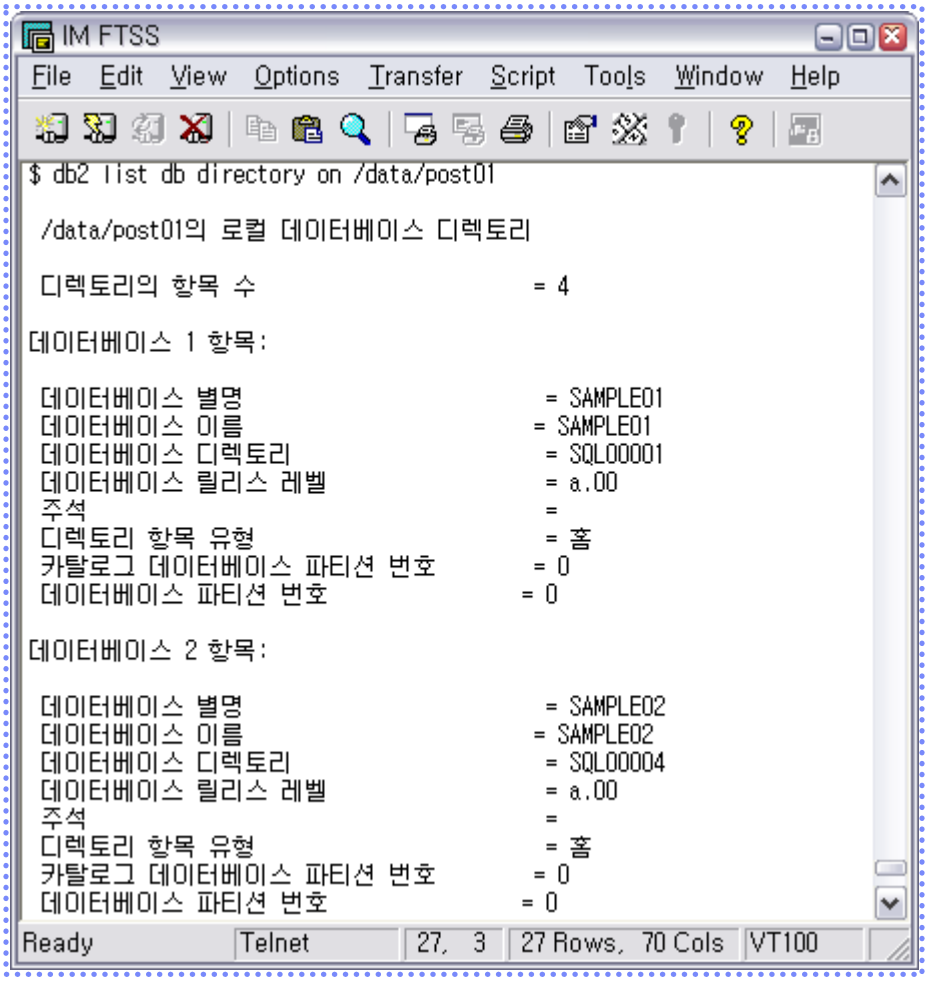

Figure 0513A ••• LIST DB DIRECTORY 명령어의 ON 옵션

3. Windows 에서는 특정한 <디렉토리명> 대신에 <드라이브명>을 이용합니다.

```
시작 → 실행 → db2cmd
C:\> db2 list db directory ON <드라이브명>
```

Memo