

2장. 오브젝트 작성

2.1 Instance 작성

2.2 Database 작성

2.3 Table Space 작성

2.3.1 Table Space 작성

2.3.2 Table Space 확장

2.4 Table 및 Index 작성

2.5 권한 부여 및 관리

2.5.1 Database 권한

2.5.2 Index 권한

2.5.3 Package 권한

2.5.4 Table/View 권한

2.6 오브젝트 작성 및 관리를 위한 SQL

2.6.1 Table/Table Space 정보 조회

2.6.2 Table/Table Space 특성 변경

2.6.3 권한 관리를 위한 SQL

2.6.4 기타 오브젝트 관리를 위한 SQL

2.1 DB2 Instance 작성 (AIX)

AIX (또는 기타 UNIX) 환경에서 Instance를 작성하는 방법에는 다음과 같이 두 가지가 있다.

- 1) db2setup 유틸리티를 사용하는 방법
- 2) db2icrt 명령어를 사용하는 방법

▶ db2setup 유틸리티를 이용한 Instance 정의

- ▷ db2setup 유틸리티는 일반적으로 제품을 새로 설치하는 경우 이외에도 Instance만을 새로 또는 추가로 작성하는 경우에 사용할 수 있다.
- ▷ db2setup 유틸리티는 DB2 UDB V7 CD-ROM 및 DB2가 이미 설치되어 있는 경우 /usr/lpp/db2_07_01/install 디렉토리에서 실행시킬 수 있다.
- ▷ root user로 login 한다.

```
# login root
```

- ▷ DB2 UDB V7 CD를 cdrom drive에 넣고 mount한 후에 /cdrom 디렉토리에서 db2setup 명령어를 실행한다.

```
# mount /cdrom
# cd /cdrom
# ./db2setup
```

- ▷ DB2가 이미 설치되어 있으면 /usr/lpp/db2_07_01/install 디렉토리로 이동하여 db2setup을 실행해도 된다.

```
# cd /usr/lpp/db2_07_01/install
# ./db2setup
```


▷ 여기서는 Instance ID는 'db2inst2', 그룹 이름은 'db2iadm1'으로서 첫 번째 Instance ID의 그룹을 함께 사용하였다. 미리 'db2inst2'라는 ID가 만들어져 있는 경우에는 해당 ID를 명시하고, 그렇지 않은 경우 '기본 UID 사용' 필드를 체크하여 새로 user ID가 만들어 질 수 있도록 한다.

```

+----- DB2 서비스 작성 -----+
|+--- DB2 인스턴스 -----+
|
|인증:
|  DB2 인스턴스에 사용할 사용자 ID, 그룹 ID, 홈 디렉토리 및 암호를
|  입력하십시오.
|
| 사용자 이름      [db2inst2]
| 사용자 ID       [563]           [ ] 기본 UID 사용
| 그룹 이름       [db2iadm1]
| 그룹 ID        [346]           [ ] 기본 GID 사용
| 홈 디렉토리     [/home/db2inst2 ]
| 암호           [*****]
| 암호 확인      [*****]
|
| 추가 옵션을 보거나 변경하려면 '등록 정보'를           [ 등록 정보... ]
| 선택하십시오.
|
| 기본 설정 모두를 복원하려면 '기본값'을 선택           [ 기본값 ]
| 하십시오.
|
|  확인 [ 취소 ] [ 도움말 ]
+-----+

```

▷ fenced user는 저장 프로시저 및 사용자 정의 함수를 작성하는데 필요한 ID로서 기존의 ID를 공유하여 사용하거나 또는 추가로 만들어 줄 수 있다.

```

+----- 분리(Fenced) 사용자 -----+
|
| 분리(Fenced) 사용자 정의 함수(UDF) 및 저장 프로시저는 이 사용
| 자와 그룹 하에서 실행됩니다.
|
|인증:
|  분리(Fenced) 사용자에 사용될 사용자 ID, 그룹 ID, 홈 디렉토리 및
|  암호를 입력하십시오.
|
| 사용자 이름      [db2fenc2]
| 사용자 ID       [562]           [ ] 기본 UID 사용
| 그룹 이름       [db2fadm1]
| 그룹 ID        [345]           [ ] 기본 GID 사용
| 홈 디렉토리     [/home/db2fenc2 ]
| 암호           [ ]
| 암호 확인      [ ]
|
| 기본 설정 모두를 복원하려면 '기본값'을 선택           [ 기본값 ]
| 하십시오.
|
| [ 확인 ] [ 취소 ] [ 도움말 ]
+-----+

```

▷ DB2 웨어하우스 제어 데이터베이스 및 DB2 Distributed Join기능은 필요한 경우, 설치를 선택하면 인스턴스와 함께 작성된다. 필요없는 경우는 설치하지 않는다.

----- DB2 웨어하우스 제어 데이터베이스 -----

DB2 웨어하우스 제어 데이터베이스의 설치 여부를 선택하십시오.

DB2 웨어하우스 제어 데이터베이스 설치
 DB2 웨어하우스 제어 데이터베이스 설치 안함

[확인] [취소] [도움말]

----- DB2 데이터 소스용 DB2 Distributed Join -----

DB2 데이터 소스용 DB2 Distributed Join의 설정 여부를 선택하십시오.

DB2 데이터 소스용 DB2 Distributed Join 설정
 DB2 데이터 소스용 DB2 Distributed Join 설정 안함

[확인] [취소] [도움말]

▷ 모든 설정이 완료되면 커서를 [확인]에 위치시키고 Enter

----- DB2 서비스 작성 -----

작성하려는 항목을 선택하고 완료되면 '확인'을 선택하십시오.

DB2 인스턴스는 사용자가 데이터를 저장하고 응용프로그램을 수행하는 환경입니다. 하나의 인스턴스에 여러 개의 데이터베이스가 있을 수 있습니다.

(*) **DB2** 인스턴스 작성 [사용자 정의...]
() 기존 **DB2** 인스턴스 설정 : 사용자 정의... :
() **DB2** 인스턴스 작성 안함

관리 서버는 **DB2** 데이터베이스에 대한 연결 구성을 자동화하는 클라이언트 도구를 지원하는 서비스를 제공합니다.

() 관리 서버 작성 : 사용자 정의... :
(*) 관리 서버 작성 안함

[**확인**] [취소] [도움말]

- ▷ 다음과 같이 관리 서버가 작성되지 않았다는 경고 메시지는 무시하고 확인을 누른다.
- ▷ 관리 서버는 최초 DB2 설치 및 Instance 작성시 만들어졌으면 추가로 작성할 필요가 없다.
즉 관리 서버는 한 시스템에 1개 이상 있을 필요는 없다.
- ▷ 아직 작성된 관리 서버가 없어 최초의 관리 서버 작성이 필요한 경우에는 1.1.2장을 참조한다.

----- DB2 서비스 작성 -----

+--- 경고 -----+

DBI1755W 관리 서버가 작성되지 않았습니다.

설명: 관리 서버를 작성하지 않았습니다. 관리 서버가 없으면 **DB2** 데이터베이스와의 연결 구성을 자동화하는 클라이언트 도구를 지원하는 서비스를 제공할 수 없고, 제어 센터를 사용하여 사용할 시스템이나 원격 클라이언트에서 **DB2**를 관리할 수 없습니다.

사용자 조치:

- o 계속하려면 [확인]을 누르고,
- o 되돌아가려면 [취소]를 누르십시오.

[**확인**] [취소]

▷ 설치에 들어가기에 앞서 요약보고서에서 입력사항을 확인한다.

```
----- DB2 설치 유틸리티 -----
+-- 요약 보고서 -----+
|
| DB2 서비스 작성
|
| 분리(fenced) 사용자
|
|   새 그룹 이름 작성:
|   그룹 이름                db2fadm2
|   그룹 ID                   104
|   새 사용자 이름 작성:
|   사용자 이름              db2fenc2
|   홈 디렉토리              /home/db2fenc2
|
| DB2 인스턴스
|
|                                     [ 계속... ]
+-----+
|
|                                     [ 계속 ]
+-----+
```

▷ 설치가 완료되면 다음과 같은 상태보고서에서 성공여부를 확인하고 [로그보기]를 선택하여 로그를 조회할 수 있다.

```
----- DB2 설치 유틸리티 -----
+-- 상태 보고서 -----+
|
| 로그 파일은 /tmp/db2setup.log에 있습니다.
|
| DB2 서비스 작성
|-----+
|
| 분리(fenced) 사용자
|
|   새 사용자 이름 작성                성공
|   새 사용자에게 대한 암호 설정      성공
|
| DB2 인스턴스
|
| DB2 인스턴스 작성                    성공
|
|                                     [ 계속... ]
+-----+
|
| [ 로그 보기 ]                                     [ 확인 ]
+-----+
```

▶ db2icrt 명령어를 이용한 Instance 정의

a. DB2 Instance 작성

- ▷ db2icrt 명령어는 DB2에서 인스턴스를 작성하는 명령어이다.
db2icrt를 사용하여 인스턴스를 생성하고자 하는 경우에는 사용할 User-ID 및 Group-ID를 미리 만들어 놓아야 한다.

- ▷ db2icrt 명령문

```
# login root
# cd /usr/lpp/db2_07_01/instance
# db2icrt -u fenced_user_id InstName
```

☞ 여기서 *InstName*은 인스턴스 소유자(Instance owner)의 User-ID를 입력한다.

b. 관리 서버 작성

- ▷ 관리서버 인스턴스를 생성하기 위한 명령어는 dasicrt 이다.
- ▷ db2setup 유틸리티를 이용하는 경우와 마찬가지로, 시스템에 작성된 관리서버가 없는 경우에만 아래 명령문을 이용하여 관리서버를 작성한다.

- ▷ dasicrt 명령문

☞ *DasName*은 관리서버 소유자가 될 User-ID를 입력한다.

```
# cd /usr/lpp/db2_07_01/instance
# dasicrt DasName
```

2.2 DataBase 작성

▶ Database 작성 구문

▷ Syntax

```
Create Database database-name [On path] [Alias alias-name]
    [Using codeset(970)] [Numsegs 10] [DFT_Extente_Size 32]
    [Catalog Tablespace tblspace-defn]
```

▷ Syntax 내용에 대하여

```
[On path] => Systm Tablespace등을 작성하기 위한 Directory
[Using codeset(970)] => Language code set
[DFT_Extente_Size 32] => Directory extent size
[Catalog Tablespace tblspace-defn] => Tablespace 관련 정보
```

▷ Sample Script

```
-- data base 생성
create database db7

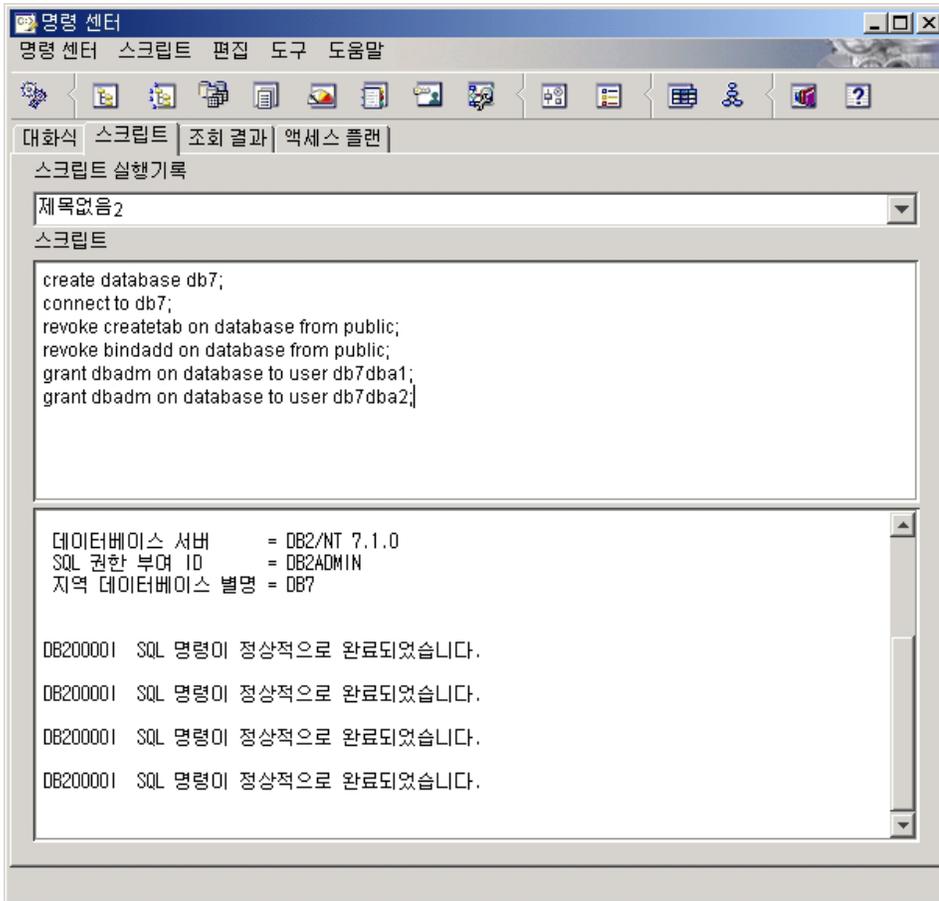
-- connect database
connect to db7

-- dssba1에 대한 권한 부여
revoke createtab on database from public
revoke bindadd on database from public
grant dbadm on database to user db7dba1
grant dbadm on database to user db7dba2

<< 권한 부여 내용 >>
db2inst1 (Database 작성자) => DBADM 권한
lnidzp1, lnidzp2           => DBADM 권한
public                    => create table 및 bind application 권한 박탈
                           connect 권한만 자동 부여됨
```

▶ Windows 환경

▷ 명령센터에서 스크립트를 입력하여 실행한다



▷ 스크립트가 완료된 후 제어 센터에 보면 새로운 DataBase가 생성 된 것을 확인할 수 있다.

▶ AIX 환경

▷ Script를 file로 작성하여 Command Line에서 실행

```
$ db2 -svtf Script File명
```

2.3 Table Space 작성

2.3.1 Table Space 작성

▶ Table Space 작성 구문

-SMS : SMS(System Managed Space) 테이블 공간은 운영 체제 파일에 데이터를 저장한다. 일반적으로 작은 테이블이 많을 때 적합하고 공간 할당은 추가 공간에 대한 요구가 있을 때 수행된다. 또한 관리에는 용이하나, DB2가 제공하는 여러가지 옵션을 사용할 수 없다.

-DMS : DMS(Database Managed Space)테이블 공간은 데이터베이스 관리 프로그램이 저장 공간을 제어한다. SMS에 비해 지속적인 관리가 필요하지만, long 데이터, index등을 따로 저장할 수 있고, 하나의 테이블을 여러 테이블 공간에 저장할 수 있는 등 성능면에서 이점이 있다

▷ Syntax

```
-. Windows 환경
  Create tablespace TS-name managed by database
    using ('x:\...\...' 1000, file 'x:\...\...' 2000)
    extentsize 10
-. AIX 환경
  Create tablespace TS-name managed by database
    using (device '/dev/raaa' 1000, device '/dev/rbbb' 2000)
    extentsize 10
```

▷ Syntax 내용에 대하여

```
-. 2개의 TableSpace로 구성된 DMS TableSpace
-. 1000, 2000 : TableSpace에서 사용할 Page(4K단위) 개수로 Logical Volume보다
  크면 Error 발생
-. extentsize : 복수 컨테이너 사용시 컨테이너간 이동 간격
  (예, 컨테이너 1에서 10개의 content에 data를 적재했을 경우
  다음부터는 컨테이너 2에 적재, 컨테이너 1과 2가 반복하여 data 저장)
```

↳ Tablespace에 할당되는 페이지수와 물리적인 Size

Table Space에 할당되는 페이지 개수는 4K(4, 8, 16, 31k 가능) 단위이며, Logical Volume 작성시 부여한 LV의 Size보다 크게 지정할 경우 Error가 발생하며 적게 지정할 경우 적은

Size만큼은 사용이 불가능 하기 때문에 정확하게 계산하여 지정해야 함

▶ Windows 환경

▷ Sample Script

```
-- connect to database
connect to DB01

-- Table Space 생성
create tablespace TS01 managed by database using
    (file 'd:\container\cont1' 2500)
create tablespace TS02 managed by database using
    (file 'd:\container\cont2' 2500)
create tablespace TS03 managed by database using
    (file 'd:\container\cont3' 2500)
create tablespace TS04 managed by database using
    (file 'd:\container\cont4' 2500)
```

- managed by database : DMS 방식으로 테이블 공간을 관리.
- file : container로 file을 사용.
- d:\container\cont1 : file이 위치할 경로를 지정.
- 2500 : 사용할 file의 크기를 page단위로 지정

▶ AIX 환경

▷ Sample Script

```
-- connect to database
connect to DB01

-- Table Space 생성
create tablespace TS01 managed by database using
    (device '/dev/rd001' 6144);
create tablespace TS02 managed by database using
    (device '/dev/rd002' 13312);
create tablespace TS03 managed by database using
    (device '/dev/rd003' 25600);
create tablespace TS04 managed by database using
    (device '/dev/rd004' 13312);
```

- device : container로 raw device를 사용.

2.3.2 Table Space 확장

▶ Table Space Size 확장 방법

▷ TableSpace에 공간 부족시 발생하는 Error

- > SQL3306N : Table에 행을 삽입하는 동안 sql 오류발생 "-289"
- > SQL0289N : TableSpace에 새로운 페이지를 작성할수 없습니다.

▷ TableSpace의 공간 확인

```
- connect to DB01  
- list tablespaces show detail | more
```

- > '상태' 필드가 0x0000으로 나오는지 확인
- > '가용 페이지 수' 가 충분한지 확인
- > 만약 가용페이지 수가 충분하지 못하면 Table Space의 크기를 확장

☞ Table Space가 Full일 경우 Logical Volumn(L/V) Size를 확대하여

그 L/V에 작성된 Table Space Size를 확대하는 방법은 없다.

따라서 별도의 L/V(컨테이너)를 작성하여 해당TableSpace에 추가하는 방법으로 해결한다.

이럴 경우 Table Space가 복수개의 컨테이너로 구성된다.

▷ 처리내용

1) Alter Tablespace 기능을 이용한 Size 확장작업

(가) logical volumn을 작성한다.

(나) logical volumn의 owner, group을 변경한다.

(Owner = db2inst1, Group = db2grp)

(다) alter tablespace tsblespace명

add (device '/dev/new_dev' 1000)

-> /dev/new_dev : tablespace에 새로 추가하고자 하는 Raw Device명

-> 10000 : 추가하고자 하는 size (기존 tablespace의 page 크기를 따라감)

ex) TS01의 공간이 부족할 경우

```
# su - root
smit에서 datavg1에 Logical Volume 64M 짜리(lv002)를 하나 만듦
# cd /dev
# chown db2inst1:db2iadm1 lv002
$ su - db2inst1
$ db2 connect to DB01
$ db2 "alter tablespace TS01
add (device '/dev/lv002' 16000)"
```

2.4 Table 및 Index 작성

▶ Table 작성 구문

▷ Syntax

```
-- create table table-name
      (column .... ,
       column .... ,
       column .... )
      in table-space-name;

-- create unique index index-name on table-name(column, column);
create index index-name on table-name(column, column) (asc | desc);
```

▷ Syntax 내용에 대하여

- Table 작성시 'in table-space-name'을 부여하지 않을 경우 Userspace에 Table이 작성된다.
- create table문 내에서 primary key를 지정하고 작성할 경우 unique Index가 자동으로 작성(name 확인불가)되기 때문에 별도로 primary key와 동일한 unique index 작성시 Warning Error 발생
따라서 Primary key를 지정하여 처리하거나,
지정하지 않고 Unique index를 작성하여 처리
- Index를 신규로 작성했을 경우에는 Package를 Rebind해야 새로운 Path가 지정되어 처리속도가 향상된다.

▷ Sample Script

```
-- create table db2usr.tab01
(cpagjym dec(7,0) not null with default,
 cpabrno char(3) not null with default,
 cpaspid char(9) not null with default,
 cpagjcd char(6) not null with default,
 cpajs dec(9,0) not null with default,
 cpajanak dec(15,0) not null with default,
 cpagyeak dec(15,0) not null with default,
 cpawpjan dec(15,0) not null with default,
 cpagpjan dec(15,0) not null with default)

in tablespace1;

-- index 작성
create unique index db2admin.idx01 on
                db2admin.tab01(cpagjym, cpabrno, cpaspid);
create index db2admin.idx02 on
                db2admin.tab01(cpagjym, cpabrno, cpagjcd);

-- 권한부여
grant select on db2usr.tab01 to public;
grant insert on db2usr.tab01 to user db2usr;
grant delete on db2usr.tab01 to user db2usr ;
grant update on db2usr.tab01 to user db2usr ;
```

2.5 권한 부여 및 관리

2.5.1 Database 권한

▷ select * from syscat.dbauth

Grantor	Grantee	Granteetype	dbadmauth	createtabauth	bindaddauth	connectaut	create_not_fenced
---------	---------	-------------	-----------	---------------	-------------	------------	-------------------

sysibm	db2inst1	U	Y	Y	Y	Y	Y
sysibm	public	G	N	Y	Y	Y	Y

1. Create database시 Creator user-id에 모든권한이 부여되고,
2. public group에도 상기와 같은 권한이 자동 부여된다.
3. 따라서 create database후 public에 자동 부여된 권한을 revoke한후 user-id 또는 group-id별 정당한 권한을 부여해야함.

▷ 권한 revoke

1. connect to database 명
2. revoke createtab on database from public
3. revoke bindadd on database from public
4. revoke connect on database from public
5. revoke create_not_fenced on database from public

▷ 각 user-id별 정당한 권한 부여

1. connect to database 명
2. grant connect on database to public
(모든 사용자에게 DB에 Connect 권한만 부여하는 예임)

☞ 종류

- Connect 권한 : 사용자가 DB를 Access할 수 있는 기초권한
- Bindadd 권한 : 사용자가 DB를 새로운 Package를 작성할 수 있는 권한
- Createtab권한 : 사용자가 DB에 Table을 작성할 수 있는 권한
- Create_not_fenced : 사용자함수(UDF)를 작성할 수 있는 권한

2.5.2 Index 권한

▷ select * from syscat.indexauth

grantor	grantee	granteetype	indschema	indname	controlauth
sysibm	db2inst1	u	svrba	xsba001u	y

▷ 필요한 user-id에 Control 권한 부여

1. connect to database 명
2. grant control on index to user-id

☞ Create index user-id에만 control 권한이 자동 부여된다.

2.5.3 Package 권한

▷ select * from syscat.packageaut

Grantor	Grantee	Granteetype	pkgschema	pkgname	controlauth	bindauth
ysibm	svrba	U	svrba	brpb001	y	y

- ☞ 다음과 같은 특권을 사용하기 위해 DB에 Connect 권한이 필요함
- Control 권한 : 사용자에게 Package에 대한 제어권을 제공
- Bind 권한 : 기존 Package를 Rebind 할 수 있는 권한
- Execute 권한 : 사용자가 기존 Package를 실행할 수 있는 권한

2.5.4 Table / View 권한

▷ Select * from syscat. tabauth

```
Grantor  Grantee  Granteetype  tabschema  tabname  control  alter  delete_index
insert  select  replace
```

가. create table user-id에만 모든권한(control, alter, delete, index, insert, select, refauth, update)이 자동 부여된다.

나. 따라서, table별로 필요한 user-id별로 권한을 부여하여야 한다.

☞ 다음과 같은 특권을 사용하기 위해 DB에 Connect 권한이 필요함

1. grant all on table Table명 to user-id로 권한을 부여할 경우
=> Control을 제외한 모든 권한이 부여됨
2. grant control on table Table명 to user-id
3. grant (all | alter | control | delete | index | insert |
reference | select | update) on table (Table명 | View 명)
to (user | group | public) (Userid | Group-id)
=> Alter, index, reference 특권은 View에 적용되지 않음

2.6 오브젝트 작성 및 관리를 위한 SQL

2.6.1 Table/Table Space 정보 조회

(1) Tablespace 정보 조회

```
- connect to DB01  
- list tablespaces | more  
or  
- list tablespace show detail | more
```

(2) Table 목록 조회

```
- connect to DB01  
- list tables for all | more
```

(3) Table의 Column 정보 조회

```
- connect to DB01  
- select tabschema, tablename, colname, typename,  
      length, scale, default, nulls  
      from syscat.columns  
또는 제어센터 이용
```

2.6.2 Table/Table Space 특성 변경

(1) Alter Table 문 : Table의 정의를 변경

가. Table에 칼럼 추가

```
-- alter table TAB01 add COL09 char(20)
```

나. Primary key, foreign key 추가 및 삭제

```
-- alter table TAB01 add constraint RI01  
foreign key(COL001) reference TAB02 on delete set null
```

다. 점검 강제 규정 정의 삭제

```
-- alter table TAB01 drop constraint RI01
```

(2) Alter TableSpace 문 : TableSpace의 정의를 변경

가. 컨테이너를 DMS TableSpace에 추가 (2.3.2 참조)

```
-- alter tablespace TS01 add(device '컨테이명' 10000)
```

나. TableSpace에 대해 Prefetchsize 설정값 수정

다. TableSpace에 대해 Overhead 설정값 수정

```
-- alter tablespace TS01  
prefetchsize 64  
overhead 19.3
```

라. TableSpace에 대해 Transferrate 설정값 수정

2.6.3 권한 관리를 위한 SQL

(1) Grant : 전체 Database에 적용되는 권한을 부여

가. DataBase 권한

```
-- connect to database-name ;  
-- grant (bindadd | connect | createtab | create_not_fenced | dbadm)  
    on database to (user | group | public)  
    권한명(group-id, user-id)
```

나. 색인 특권

```
-- connect to database-name  
-- grant control on index index-name to (user | group | public)  
    권한명(group-id, user-id)
```

다. Package 특권

```
-- connect to database-name  
-- grant (bind | control | excute) on package  
    package-name to (user | group | public)  
    권한명(group-id, user-id)
```

=> alter, index, reference 특권은 View에는 적용되지 않음

=> all 특권을 부여할 경우 alter, delete, index, insert, reference, select, update의 특권이 동시에 부여됨

(2) Revoke : 전체 DataBase에 적용되는 권한을 취소

가. DataBase 권한

```
-- connect to database-name  
-- revoke (bindadd | connect | createtab | create_not_fenced | dbadm) on database  
      from (user | group | public)  
      권한명(group-id, user-id)
```

나. 색인특권

```
-- connect to database-name  
-- revoke control on index index-name  
      from (user | group | public)  
      권한명(group-id, user-id)
```

다. Package 특권

```
-- connect to database-name  
-- revoke (bind | control | excute) on package package-name  
      from (user | group | public)  
      권한명(group-id, user-id)
```

=> run은 excute에 대한 동의어,
package는 Program에 대한 동의어로 사용가능

라. Table/ View 특권

```
-- connect to database-name  
-- revoke (all | alter | control | delete | index | insert | reference | select | update)  
      on table (table명 | view명)  
      from (user | group | public)  
      권한명 (group-id, user-id)
```

2.6.4 기타 오브젝트 관리를 위한 SQL

(1) Create Trigger

특정 Table에 대한 삭제, 삽입 또는 갱신 조작시 수행되는 또는 조작으로 트리거되는 조치세트를 정의

```
-- Create Trigger New_Tirgger명
  After Insert On EMPLOYEE
  For Each Row Mode DB2SQL
  Update COMPANY_STATS set nbemp = nbemp + 1
```

=> EMPLOYEE Table에 새로운 Row가 Insert된 다음에는
COMPANY_STATS Table의 nbemp Column에 1을 자동 Add하는 Trigger임

(2) Create View

```
-- Create View JJ-VIEW
  as select HENGBUN, NAME, JUSO, DEPTNO, SALARY,
  SALARY+BONUS+COMM as TOTAL_PAY
  from INSATBL, PAYTABLE
```

(3) Drop

각 Object(Alias, distinct, event, function, index, package, table, tablespace, trigger, view, database)를 삭제하는 기능