

## ArcSDE 운영자 지침서

### 1 ArcSDE의 구조 및 프로세스

#### 클라이언트/서버 수행 구조

클라이언트는 서버에 feature 를 요청한다.

서버는 클라이언트에서 요구한 레코드를 검색하고 buffer 를 채운다.

서버의 buffer 에서 클라이언트로 보내고 다른 것을 buffer 에 채운다.

클라이언트는 buffer 의 데이터를 처리하고 다른 것을 서버에 요청한다.

서버에서 buffer 가 채워지는 동안 클라이언트는 buffer 를 비운다.

데이터 로딩은 위 순서의 반대로 한다.

#### 클라이언트 connection 과 서버 tasks

각각의 서버는 하나의 GIOMGR task 를 가진다.

각각의 클라이언트 connection 은 하나의 GSRVR task 를 가진다.

각각의 GSRVR task 는 클라이언트 라이선스를 사용한다.

### 1.5 ArcSDE 디렉토리 구조 및 파일

ArcSDE 에는 다음과 같은 서브 디렉토리와 파일이 존재한다.

Sdeexe91 – ArcSDE version 9.1 의 최상위 디렉토리

/bin – ArcSDE version 9.1 의 실행 프로그램

/bin/giomgr – I/O 관리자 프로세서

/bin/gsrvr – ArcSDE 서버의 링크

/bin/gsrvr.shared – ArcSDE 의 Shared 버전

/bin/gsrvr.static – ArcSDE 의 static 버전

/bin/sdeexport – ArcSDE layer 의 export 명령어

/bin/sdeimport – ArcSDE layer 의 import 명령어

/bin/sdelayr – Map layer 관리 도구

/bin/sdelicsserv – 라이선스 관리자와 함께 사용된다.

/bin/sdelog – log 파일 관리

/bin/sdemon – ArcSDE 모니터 유틸리티

/bin/sdetable – business 테이블 관리

/bin/sde2shp – Arcsde 데이터를 shape 데이터로 변환

/bin/shp2sde – shape 데이터를 sde 데이터로 변환

/bin/cov2sde – coverage 데이터를 sde 데이터로 변환

/bin/sde2cov – Arcsde 데이터를 coverage 데이터로 변환

/bin/tbl2sde - 다른 DBMS 테이블 포맷을 ArcSDE 테이블로 변환  
/bin/sde2tbl - ArcSDE 테이블을 다른 DBMS 테이블로 변환  
/bin/sdegroupp - single-part shape 을 Multi-part shape 으로 결합  
/bin/sderealese - 서버 버전의 Release 리스트 또는 업그레이드  
/bin/sdeversion - Versioned 테이블의 관리  
/bin/shpinfo - shape 파일의 정보  
/bin/sdexinfo - Arcsde export 파일의 정보  
/codepage - 국가언어 지원 파일 디렉토리  
/etc - 시스템에 관련된 파일 포함 디렉토리  
/etc/dbinit.sde - DBMS 연결 정보 포함 파일  
/etc/dbtune.sde - 키워드 파라미터 구성 파일  
/etc/geomgr.defs - ArcSDE 와 ORACLE 간의 연결 파라미터 구성 파일  
/etc/geomgr.log - 서버 관리자 프로세스관련 에러 로그파일  
/etc/services.sde - OS 서비스 파일(/etc/services)의 SDE service port 정의  
/etc/sde.errlog - geomgr 에 관련된 모든 에러 내용이 저장되는 로그 파일  
/etc/sdelic.log - 라이선스 매니저로 부터의 에러메세지 포함  
/include - ArcSDE 클라이언트 include 파일 디렉토리  
/include/pe.h - Projection Engine 데이터 구조 및 형태 정의 헤더 파일  
/include/pe\_coordsys\_from\_prj.h - Projection Engine 의 사전 정의 헤더 파일  
/include/pedef.h - Projection Engine 의 사전 정의 헤더 파일  
/include/pef.h - Projection Engine 의 사전정의 오브젝트의 기능  
/include/sdeerno.h - ArcSDE 에러 코드 헤더 파일  
/include/sderaster.h - 래스터 헤더파일  
/include/sdetype.h - ArcSDE 데이터 구조 및 형태 정의 헤더 파일  
/include/sg.h - Shape 파일의 구조 및 형태 정의 헤더 파일  
/include/sderr.h - Shape 에러 헤더 파일  
/lib - 라이브러리 와 보안 데이터베이스를 포함하는 디렉토리  
/lib/jsde80.jar - 자바 클라이언트 아카이브 파일  
/lib/libpe80.a - Projection Engine 을 위한 Static 라이브러리  
/lib/libpe80.sl - Projection Engine 을 위한 shared 라이브러리  
/lib/libras80.a - 래스터에 대한 static 라이브러리  
/lib/libras80.sl - 래스터에 대한 shared 라이브러리  
/lib/libsg80.a - Static shape geometry 라이브러리

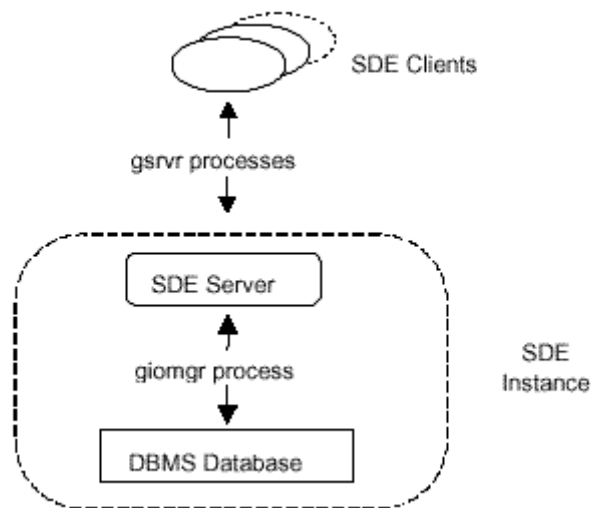
/lib/libsg80.sl – Shared shape geometry 라이브러리  
/lib/libdesrvr.sl – shared run-time 라이브러리  
/lib/lib/sde80.a – static sde 라이브러리  
/lib/lib/sde80.sl – shared sde 라이브러리  
/lib/lib/sde80\_trase.sl – 트레이싱 기능에 대한 shared 라이브러리  
/lib/make.include – 컴파일/링크시 사용하는 파일  
/lib/metadata\_util.spb – 메타데이터 유지관리를 위한 Stored procedure  
/lib/metadata\_util.sps - 메타데이터 유지관리를 위한 Stored procedure  
/lib/layers\_util.spb – 레이어 유지관리를 위한 Stored procedure  
/lib/layers\_util.sps – 레이어 유지관리를 위한 Stored procedure  
/lib/locator\_util.spb – Locator 유지관리를 위한 Stored procedure  
/lib/locator\_util.sps – Locator 유지관리를 위한 Stored procedure  
/lib/registry\_util.spb – 레지스트리 유지관리를 위한 Stored procedure  
/lib/registry\_util.sps – 레지스트리 유지관리를 위한 Stored procedure  
/lib/sde\_util.sps – ArcSDE 유지관리를 위한 Stored procedure  
/lib/sref\_util.sps – Spatial reference 유지관리를 위한 Stored procedure  
/lib/version\_util.spb – Version 유지관리를 위한 Stored procedure  
/lib/version\_util.sps – Version 유지관리를 위한 Stored procedure  
/logs – ArcSDE 로그 파일 저장 디렉토리  
/sysgen - 라이선스 관리자 포함 디렉토리  
/sysgen/ESRI – 라이선스 대몬  
/sysgen/adminlicense – license.dat 파일의 관리 유틸리티  
/sysgen/license.boot – 라이선스 관리자의 자동 시작 스크립트  
/sysgen/license.dat – 라이선스 파일  
/sysgen/lmgrd – ESRI 라이선스 대몬을 실행하는 프로그램  
/sysgen/lmutil – 라이선스 관리자를 관리해주는 FLEXlm 유틸리티  
/sysgen/sample.dat – 샘플 라이선스 파일  
/temp – ArcSDE temp 디렉토리  
/tools – DBMS 모니터링 도구를 포함하고 있는 디렉토리  
/tools/generic – layer 관리 도구를 포함하고 있는 디렉토리  
/tools/oracle – ORACLE 관련 관리도구를 포함하고 있는 디렉토리

## 1.6 서버 관리자(Server manager)

ArcSDE 서버 관리자는 백그라운드로 수행되고 있는 대몬이며, 이 대몬은 `sdemon` 이라는 모니터 프로그램에 의해서 관리 되어 진다. 서버 관리자 프로그램은 각각의 클라이언트 프로그램이 시작 할 때 필요한 런타임 환경을 생성해 주며, LAYERS 메타 데이터 테이블을 관리한다.

## 1.7 인스턴스

ArcSDE 의 인스턴스는 하나의 DBMS 에 하나의 `iomgr` 만 정의 된다. 다중의 `iomgr` 을 실행하기 위해서는 각각의 분리된 DBMS 인스턴스를 사용하여야 한다. 각각의 인스턴스는 메모리 공간 과 실행 가능한 환경 그리고 그 인스턴스가 연결 되어 있는 데이터베이스로 구성된다. `Giomgr` 프로세스, 즉 ArcSDE 인스턴스가 시작이 되면 하나의 데이터베이스에 연결이 되고, 이때 `giomgr` 프로세스를 연결해주는 것은 `sde` 계정에서 가능하다. `Giomgr` 프로세스가 시작된 후에는 `sde` 계정의 `VERSION` 테이블에 `lock` 를 설정하며, 이것은 다른 connection 이 연결하는 것을 제어한다.



## 1.8 ArcSDE 의 시작

ArcSDE 를 시작하려면 우선 사용자는 ArcSDE 를 관리할 수 있는 UNIX 계정 및 ORACLE 계정을 생성 하여야 한다.

ArcSDE 를 관리할 수 있는 사용자로 로그인 한다.

I/O 관리 태스크를 실행 한다.

`$ sdemon -o start -p <패스워드>`라고 실행을 하면 아래와 같이 화면에 디스플레이 된다.

```
-----  
ArcSDE 9.1 for Oracle9i Build 334 Fri Feb 18 07:04:29 PST 2006  
-----
```

```
DBMS Connection established...
```

RDBMS: "Oracle"

Instance Name: "esri\_sde"

ArcSDE Instance esri\_sde started Tue Jul 4 15:13:51 2000

giomgr 이 시작된 후에 에러가 발생하면 그 에러 메시지는 \$SDEHOME/etc 디렉토리에 있는 sdelog 파일에 기록되어 진다.

### 1.9 ArcSDE 의 상태 모니터링

sdemon 유틸리티를 이용하여 ArcSDE 시스템의 현재 상태를 모니터링 할 수 있다. 이 유틸리티는 ArcSDE 서버의 현재 모드, 시스템을 사용중인 클라이언트의 개수, 클라이언트/서버간의 연결정보, 현재 정의 되어 있는 시스템 구성 파라미터 환경등 다양한 종류의 SDE 관리 정보를 모니터링 할 수 있다.

ArcSDE 의 현재 상태 확인 방법

**\$ sdemon -o status**

연결되어진 클라이언트의 개수를 보여준다.

클라이언트/서버 정보 확인 방법

ArcSDE 를 사용하는 모든 클라이언트/서버의 자세한 정보를 보여준다.

**\$sdemon -o info -l users**

아래의 프로세스 정보는 각각의 등록된 ArcSDE 클라이언트/서버의 정보를 보여준다.

PID : 프로세스 식별자

USER : 클라이언트의 사용자 이름

HOST:OS : 호스트 컴퓨터의 이름과 OS 이름

STARTED : 프로세스가 시작된 때의 이름과 시간

Lock 이 되어 있는 테이블의 정보 확인 방법

ArcSDE lock 이 되어있는 테이블에 대한 자세한 정보를 보여준다.

**\$sdemon -o info -l locks**

현재의 ArcSDE 구성 파라미터 값을 보여준다.

**\$sdemon -o info -l config**

### 1.10 ArcSDE 서버의 일시중지, 재시작, 중지(shutdown)

서버 관리자 프로세스는 수행, 일시중지, 중지 세가지 모드를 가지고 있다:

ArcSDE 서버가 수행중일 때에는 클라이언트 응용프로그램은 로그인 및 ArcSDE 명령어 사용이 가능하며 ArcSDE 서버가 일시 중지일 때에는 현재 수행중인 응용프로그램은 계속 되어지나, 추가적인 응용프로그램은 로그인 할 수 없다. 이때 에는 사용자는 완전히 중지(shutdown)를

할 때 까지 작업을 계속 할 수 있도록 보장한다. 서버 관리자가 일시 중지일 때 resume 옵션을 이용하여 수행 상태로 변환할 수 있다. ArcSDE 서버를 완전히 중지하면 로그인 되어있는 클라이언트는 강제적으로 종료하게 되고 새로운 클라이언트 또한 로그인을 할 수 없다.

ArcSDE 서버의 일시 중지

```
$ sdemon -o pause -p <패스워드>
```

이 명령어를 수행하면 시스템은 다음과 같은 메시지를 보여준다.

```
ArcSDE Instance esri_sde on hpjsvr is Paused, no further connections will be allowed.
```

ArcSDE 서버의 재시작

```
$sdemon -o resume -p <패스워드>
```

이 명령어를 수행하면 시스템은 다음과 같은 메시지를 보여준다.

```
ArcSDE Instance esri_sde on hpjsvr is Resuming, new connections will now be allowed.
```

ArcSDE 서버의 중지(shutdown)

```
$sdemon -o shutdown -p <패스워드>
```

이 명령어를 수행하면 시스템은 다음과 같은 메시지를 보여준다.

```
ArcSDE Instance esri_sde on hpjsvr is Shutdown!
```

### 1.11 ArcSDE 서버 프로세스 관리

시스템에 로그인 되어있는 각각의 클라이언트 응용프로그램에 의하여 ArcSDE 는 서버 프로세스를 생성 시킨다. 이 프로세스들은 ArcSDE 서버 관리자에 의하여 관리되어 진다. 각각의 프로세스들은 응용프로그램이 종료되어지거나 프로그램상에서 ArcSDE session 을 종료하는 함수를 호출 하였을 때 종료되어 진다.

오류가 발생한 서버 프로세스의 관리

ArcSDE 서버 관리자는 모든 서버 프로세스들을 관리한다. 사용자가 ArcSDE 서버가 프로세스를 요청하는 동안에 시스템의 에러에 의하여 비 비정상적으로 종료되었을 때에는 프로세스를 강제적으로 종료를 시켜야 한다.

ArcSDE 프로세스를 삭제하는 방법은 아래와 같이 실행 한다.

```
$ ps -ef (프로세스들이 리스트된다.)
```

```
$ sdemon -o kill -t 8292
```

서버 프로세스의 리스트

아래의 명령어를 이용하면 서버 프로세스의 정보를 보여줄 수 있다.

```
$sdemon -o info -l users
```

서버 프로세스정보는 아래와 같은 모습으로 보여진다.

```
Server Instance esri_sde Registered Server Tasks on stout at mon oct
25 12:52:51 1999
```

```
-----
PID User Host:OS Started
-----
```

```
7084 world bopomo:Win32:XDR mon oct 25 15:41:18 1999
```

```
8021 world bopomo:Win32:XDR mon oct 25 15:41:18 1999
```

```
7141 world kkaddeche:Win32:XDR Tue oct 26 16:41:18 1999
```

서버 프로세스 통계 디스플레이

현재 수행중인 각각의 서버 프로세스에 대한 통계 정보를 보여준다. 이러한 프로세스의 통계 정보는 \$SDEHOME/etc/giomgr.log file 에 저장된다.

**\$sdemon -o info -l stats**

PID : 프로세스 인식자

OPS : 클라이언트/서버 오퍼레이션의 개수

READS : 디스크로부터 읽어오는 Feature/identifier 의 개수

WRITES : 디스크에 쓰는 feature 의 개수

BUFFERS : 클라이언트에 보내지는 버퍼의 개수

PARTIAL : 클라이언트에 보내지는 feature 가 버퍼크기 보다 큰 feature 의 개수

F/BUF : 버퍼 당 feature/identifier 의 평균 개수

BUF AVG : 버퍼 크기의 평균

TOT : 클라이언트에 보내지는 데이터의 총 합계

서버 프로세스의 삭제

서버 프로세스가 종료되었을 때 sdemon 에 의하여 구동 된 모든 ArcSDE 프로세스들은 인식자 (PID)를 가지고 있으며, 이 프로세스를 삭제하는 방법은 다음과 같다.

**\$ sdemon -o kill -t <{all | pid}>**

또한 -t all 옵션을 이용하면 모든 서버 프로세스들이 삭제된다.

## 1.12 시스템의 백업

사용자는 DBMS의 데이터 파일, 즉 ArcSDE 시스템 구성 파일 및 ArcSDE 로그 파일을 백업하여야 한다.사용자는 유닉스의 표준 백업 유틸리티(tar 또는 cpio)를 사용하여 ArcSDE 시스템 파일을 백업 할 수 있다. 백업을 하기 전에는 모든 ArcSDE 사용자는 로그인을 하지 말아야 한다. 프로세스가 구동 되고 있는 중에 백업을 하면 부분적인 정보만 백업을 할 수도 있다.

시스템의 디스크 장애에 의한 데이터베이스 손상 시 media 에 의한 복구(recovery)를 수행토록  
수시로 데이터베이스를 백업할 필요성이 있다. 데이터베이스의 백업은 ArcSDE 의 기반이 되  
는 DBMS 에 의존하며,ORACLE 의 경우에는 Full 백업과 Partial 백업이 있다.

ORACLE 에는 NOARCHIVELOG 와 ARCHIVELOG 인 2 개의 ARCHIVE 모드가 있다, 디폴트  
는 NOARCHIVELOG 이며, online redo log file 을 archive 하지 않는다. 따라서 만일 디스크장애  
가 발생하면 최근의 백업한 시점으로 데이터베이스의 복구가 이루어진다. 반면  
에 ARCHIVELOG 모드상태에서는 디스크 장애가 발생하더라도 archive 된 redo log file 로 최근  
백업한 시점까지는 물론 디스크 장애 직전까지 rolling forward 가 이루어 진다. 데이터베이스가  
생성될 때에는 NOARCHIVELOG 모드상태에서 작업하게 된다. 일단 데이터 로드가 끝나면 데이터  
베이스를 ARCHIVELOG 모드로 전환하는 것을 권장한다. 그렇지 않을 경우 데이터 로드 후 추가  
된 작업내용은 디스크 장애에 의해 손실 될 수 있기 때문이다.

Full 백업에 적용되는 것은 ORACLE 데이터베이스,ArcSDE 로그 디렉토리, dbtune.sde,  
dbinit.sde, services.sde 파일 이다. ORACLE 데이터베이스에는 테이블스페이스 데이터 파일,온  
라인 redo log 파일, 컨트롤 파일이 포함된다.

ArcSDE 의 백업 및 복구 체계는 ArcSDE 의 토대가 되는 DBMS 의 백업 및 복구 체계를 정확히  
준수한다.



## 2. ArcSDE 관리기법

ArcSDE의 관리자는 ArcSDE 로의 데이터 로딩, business 테이블, layers, 로그 파일을 관리하며, 이 장에서는 ArcSDE를 관리하는 하는 기법을 소개한다.

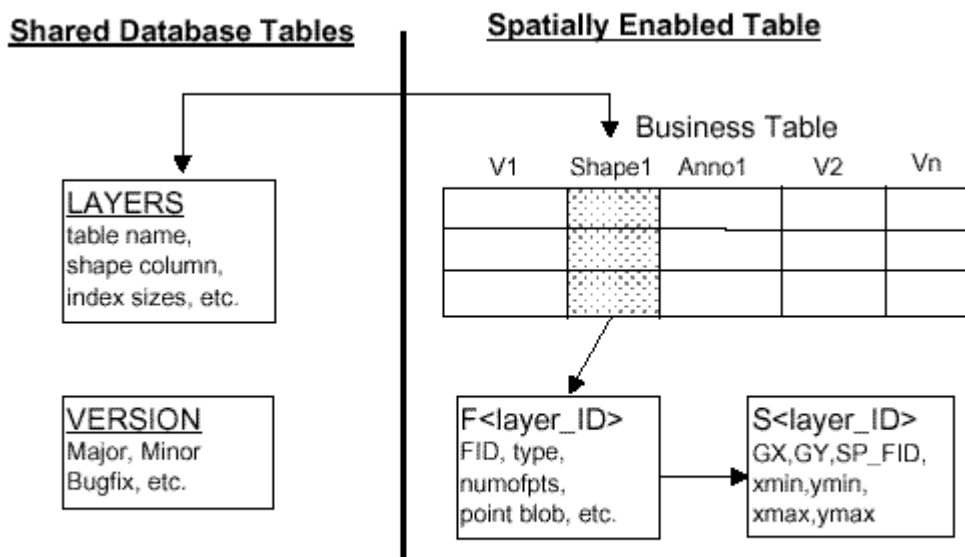
### ArcSDE 데이터베이스 테이블

ArcSDE는 spatial 데이터를 관리하는 ArcSDE 사용자 계정에 의하여 두개의 메타 데이터 테이블을 생성한다.

VERSION : ArcSDE 서버의 version을 관리

LAYERS : layer 당 하나의 레코드인 데이터베이스에 있는 사용 가능한 layer의 리스트를 관리. 여기에는 business 테이블명, layer명, 인덱스 그리드 크기, 데이터 타입, 소유자명과 같은 특성들이 저장된다.

각각의 Spatially enable 테이블에는 DBMS 사용자 계정 스키마에 저장되어지며, 두개의 다른 테이블은 spatially enable 테이블의 shape 컬럼을 통하여 생성, 조인을 한다.



The relationship of DBMS tables in SDE.

F Feature 테이블. Feature의 Shape geometry를 포함하며, 연결된 정보를 가지고 있다. 예를 들어 area, length, type 등의 정보를 가지고 있다.

S Spatial 인덱스 테이블. 인덱스 그리드 셀에 포함되어 있는 각각의 shape의 위치 정보를 가진 레코드들을 포함한다.

### VERSION 테이블

VERSION 테이블은 ArcSDE의 version 정보를 관리하는 테이블이다. 이 테이블에는 version update에 의하여 실행된 ArcSDE의 version 릴리즈 번호가 포함되어 있으며, ArcSDE I/O 관리자는 실행되는 version이 정확한지 VERSION 테이블을 점검한다.

VERSION 테이블의 소유자는 sde RDBMS 계정이며, version 테이블은 새로운 version 의 ArcSDE 가 설치되거나, 데이터베이스 테이블이 update 되면 sdeversion 명령어에 의하여 update 된다. VERSION 테이블의 스키마 구조는 아래와 같다.

#### **VERSION 테이블**

##### **NAME DATA TYPE NULL?**

Major SE\_INTEGER NOT NULL

Minor SE\_INTEGER NOT NULL

Bugfix SE\_INTEGER NOT NULL

Description SE\_INTEGER NOT NULL

Release SE\_INTEGER NOT NULL

#### **LAYERS 테이블**

LAYERS 테이블은 spatially enable 된 business 테이블의 메타 데이터를 관리한다. 이 테이블은 데이터베이스 내의 각각의 shape 컬럼을 위한 하나의 레코드들로 구성되어 있다.

Shape 컬럼의 Layer 메타 데이터는 아래의 정보를 가지고 있다.

shape 컬럼의 소유자, 테이블, 컬럼명,

현재의 shape 데이터를 포함하는 테이블명

Coordinate reference

Spatial 인덱스 그리드 셀의 크기

ArcSDE 내부적으로 할당된 layer ID

Layer 에 대한 설명

Layer 에 들어 있는 Shape 의 통계정보

LAYERS 테이블은 ArcSDE 관리자의 소유이며, I/O 관리 프로세스에 의하여 관리 된다.

LAYERS 테이블의 스키마 구조는 아래와 같이 구성 되어 있다.

#### **LAYERS 테이블**

##### **NAME NULL? DATA TYPE**

LAYER\_ID NOT NULL NUMBER(38)

DESCRIPTION VARCHAR2(65)

DATABASE\_NAME VARCHAR2(32)

OWNER NOT NULL VARCHAR2(32)

TABLE\_NAME NOT NULL VARCHAR2(160)

SPATIAL\_COLUMN NOT NULL VARCHAR2(32)

EFLAGS NOT NULL NUMBER(38)

LAYER\_MASK NOT NULL NUMBER(38)

GSIZE1 NOT NULL NUMBER(64)  
GSIZE2 NOT NULL NUMBER(64)  
GSIZE3 NOT NULL NUMBER(64)  
MINX NOT NULL NUMBER(64)  
MINY NOT NULL NUMBER(64)  
MAXX NOT NULL NUMBER(64)  
MAXY NOT NULL NUMBER(64)  
CDATE NOT NULL NUMBER(38)  
LAYER\_CONFIG VARCHAR2(32)  
OPTIMAL\_ARRAY\_SIZE NUMBER(38)  
STATS\_DATE NUMBER(38)  
MINIMUM\_ID NUMBER(38)  
SRID NOT NULL NUMBER(38)  
BASE\_LAYER\_ID NOT NULL NUMBER(38)

Geometry\_columns 테이블

Geometry\_columns 테이블은 Feature 클래스의 이름, 스토리지 유형 그리고 좌표계에 대한 내용을 가지고 있다.

**Geometry\_column** 테이블

**NAME NULL? DATA TYPE**

F\_TABLE\_CATALOG VARCHAR2(32)  
F\_TABLE\_SCHEMA NOT NULL VARCHAR2(32)  
F\_TABLE\_NAME NOT NULL VARCHAR2(160)  
F\_GEOMETRY\_COLUMN NOT NULL VARCHAR2(32)  
G\_TABLE\_CATALOG VARCHAR2(32)  
G\_TABLE\_SCHEMA NOT NULL VARCHAR2(32)  
G\_TABLE\_NAME NOT NULL VARCHAR2(160)  
STORAGE\_TYPE NUMBER(38)  
GEOMETRY\_TYPE NUMBER(38)  
COORD\_DIMENSION NUMBER(38)  
MAX\_PPR NUMBER(38)  
SRID NOT NULL NUMBER(38)

Business 테이블

Business 테이블은 존재하는 DBMS 테이블에 Shape 컬럼을 추가하여 ArcSDE spatially enable 된 테이블이다. Shape 컬럼은 shape 데이터를 참조하는 유일한 값의 feature 인식자가 포함 되어있는 integer 컬럼 이다.

#### **Spatially enabled Business 테이블**

##### **NAME DATA TYPE NULL?**

<="" span="">컬럼>

<="" span="">컬럼> SE\_INTEGER NOT NULL

<="" span="">컬럼>

#### **Feature 테이블**

Feature 테이블은 각각의 feature 의 geometric shape 가 저장된다. 이 테이블은 F 를 사용하여 shape 컬럼의 layer number 에 의하여 인식된다. Business 테이블과 feature 테이블간의 관계는 FID 를 통해서 관리되며, SDE 에 의해서 관리되는 이 key 는 spatial 컬럼에서 유일하다. Feature 테이블에 저장되는 것은 SDE 질의 프로세싱을 도와주는 shape geometry 와 추가적인 컬럼을 가지고 있으며 그 종류는 다음과 같다.

#### **Feature 테이블**

##### **NAME DATA TYPE NULL?**

fid SE\_INTEGER NOT NULL

numofpts SE\_INTEGER NOT NULL

entity SE\_SMALLINT NOT NULL

eminx SE\_FLOAT(64) NOT NULL

eminy SE\_FLOAT(64) NOT NULL

emaxx SE\_FLOAT(64) NOT NULL

emaxy SE\_FLOAT(64) NOT NULL

eminz SE\_FLOAT(64) NULL

emaxz SE\_FLOAT(64) NULL

min\_measure SE\_FLOAT(64) NULL

max\_measure SE\_FLOAT(64) NULL

area SE\_FLOAT(64) NOT NULL

len SE\_FLOAT(64) NOT NULL

points SE\_BLOB NULL

#### **저장 되는 Geometry**

Point(SE\_BLOB) - shape 의 geometry 에 정의되어 있는 point coordinate 의 바이트 스트림을 포함한다.

## 질의 프로세싱에 사용

Area(SE\_FLOAT) – shape의 영역

Len(SE\_FLOAT) – shape의 길이 또는 파라미터 값

Eminx,eminy,emaxx,emaxy(SE\_FLOAT) – shape의 envelope

Eminz(SE\_FLOAT) – shape의 최소 z 값

Emaxz(SE\_FLOAT) – shape의 최대 z 값

Min\_measure(SE\_FLOAT) – shape의 최소 measure 값

Max\_measure(SE\_FLOAT) – shape의 최대 measure 값

## ArcSDE 내부에서 사용

Fid(SE\_INTEGER) – business 테이블과 feature 테이블 간의 Join 수행시 사용하는 유일한 ID를 포함

Entity(SE\_INTEGER) – shape 컬럼에 저장된 geometric feature의 종류

Numoftps(SE\_INTEGER) – shape에 정의되어있는 point의 개수

## Spatial 인덱스 테이블

Spatial 인덱스 테이블에는 simple 그리드와 regular 그리드에 의한 shape 참조 정보가 저장된다. 이 테이블은 S를 사용하여 spatial 컬럼의 layer number에 의해서 인식된다. Spatial 인덱스는 각각의 shape와 그리드 셀이 조합되어 포함되어진다. 예를 들어 세계의 그리드에 걸쳐있는 그리드 셀은 테이블의 세계의 entry로 저장된다.

Spatial 인덱스 테이블의 스키마는 다음과 같이 구성되어 있다.

### Spatial 인덱스 테이블

#### NAME DATA TYPE NULL?

sp\_fid SE\_INTEGER NOT NULL

gx SE\_INTEGER NOT NULL

gy SE\_INTEGER NOT NULL

eminx SE\_INTEGER NOT NULL

eminy SE\_INTEGER NOT NULL

emaxx SE\_INTEGER NOT NULL

emaxy SE\_INTEGER NOT NULL

### Business 테이블의 관리

사용자는 sdetable 명령어를 이용하여 business 테이블의 생성, 변환, 삭제, truncate 그리고 business 테이블 정의를 할 수 있다.

Business 테이블의 생성

이 명령어는 속성 테이블을 생성한다. 실행 결과는 name 이라는 string 컬럼, tot\_pop 이라는 integer 컬럼 등 두개의 컬럼이 생성된다.

```
$sddtable -o create -t victoria -d "name string(20), tot_pop integer(9)" -k vict -u <사용자명> -p <패스워드> -s ultra -i esri_sde
```

Business 테이블의 삭제

Business 테이블을 삭제할 때에는 sddtable 의 delete 옵션을 사용한다. 사용법의 예는 다음과 같다.

```
$sddtable -o delete -t Victoria -u <사용자명> -p <패스워드> -s ultra -i esri_sde
```

Business 테이블의 truncating

테이블의 Truncating 은 business 테이블의 스키마를 제외한 모든 레코드를 삭제한다.

그 사용법의 예는 아래와 같다.

```
$ sddtable -o delete -t Victoria -u <사용자명> -p <패스워드> -s ultra -i esri_sde
```

Business 테이블의 altering

Business 테이블의 변경은 DBMS 의 Alter SQL 유틸리티를 사용한다.

Describing a business 테이블 정의

Business 테이블의 정의는 describe 옵션을 사용한다. 이것은 컬럼 이름, 데이터 타입, business 테이블의 각각의 컬럼의 길이의 리스트를 보여준다.

```
$sddtable -o describe -t Victoria -u <사용자명> -p <패스워드> -i esri_sde
```

테이블 row 의 리스트

Business 테이블의 row 리스트는 테이블의 컬럼명과 컬럼의 값들이 리스트 된다. 질의시 질의된 내용에 일치하는 값에 대한 테이블의 fetch 와 row 를 디스플레이를 한다.

아래의 예는 business 테이블의 parcel 컬럼의 값이 10 인 row 를 리스트 한다.

```
$sddtable -o list -t Victoria -c parcels -v 10 -u <사용자명> -p <패스워드>
```

인덱스 생성

create\_index 옵션은 Business 테이블 컬럼에 인덱스를 생성해 준다. 아래의 예는 parcel\_no 라는 인덱스 컬럼을 생성하는 예이다.

```
$sddtable -o create_index -t Victoria -c parcel_no,zone_no -n index2 -u <사용자명> -p <패스워드> -i esri_sde
```

인덱스 삭제

attribute 테이블에서 인덱스를 삭제하는 것은 아래의 예와 같다.

```
$ sddtable -o delete_index -n index1 -u <사용자명> -p <패스워드> -i esri_sde
```

### ArcSDE layer 의 관리

ArcSDE 의 layer 를 생성할 때 반드시 포함 되어야 하는 요소.

테이블과 컬럼 이름 : spatially enable 된 테이블과 컬럼의 이름

엔티티 마스크 : layer 생성시 인식하는 shape 종류(예, point,area...)

그리드 크기 : 빠른 Feature 탐색을 위한 spatial 인덱스

사용자의 옵션 요소

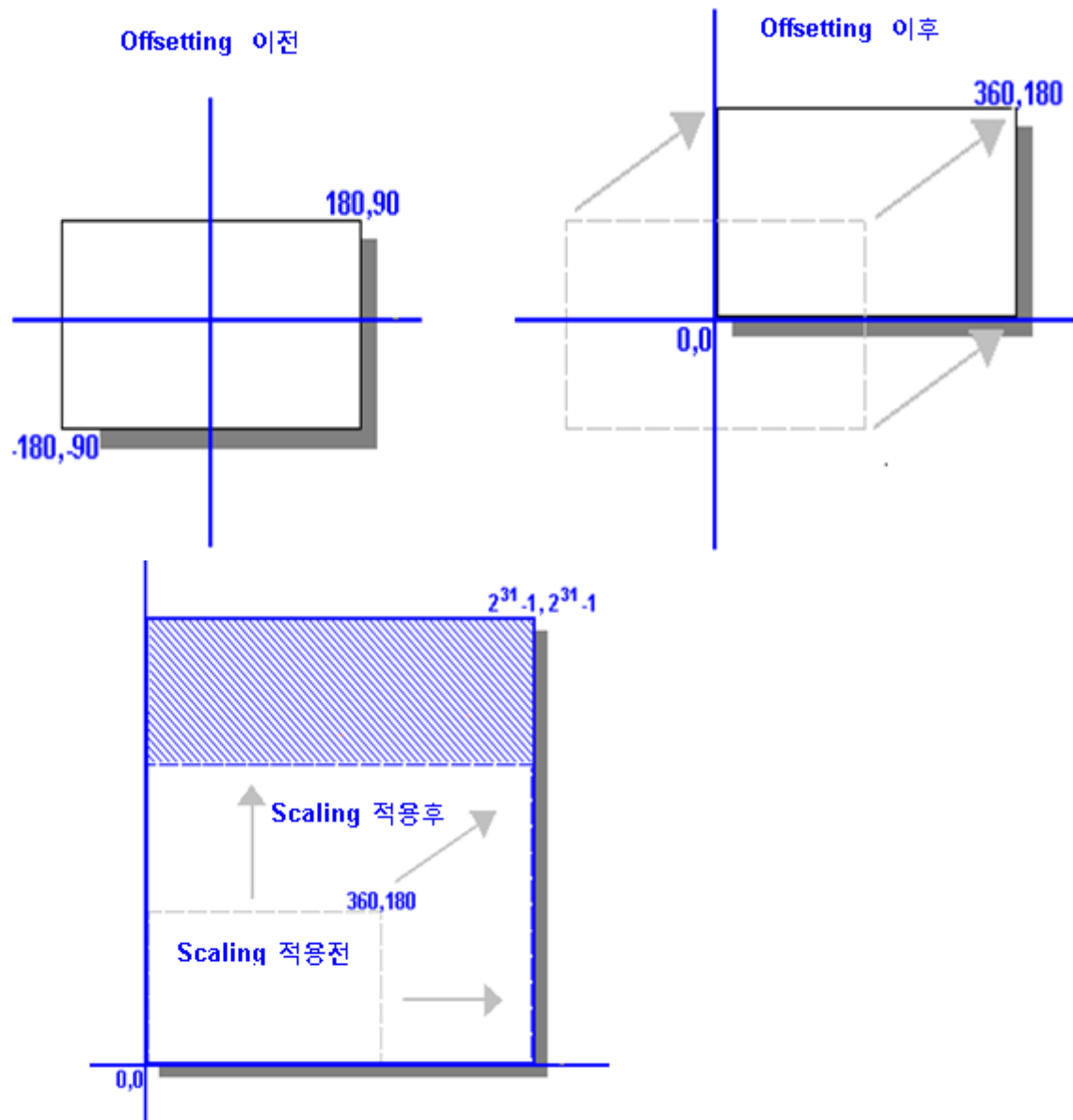
XY offset 과 scale : ArcSDE 는 데이터의 좌표값은 4-byte 의 양정수 값으로 저장한다. 이것은 데이터를 빠르게 처리하기 위함이며 모든 데이터의 좌표값을 양정수로 변화하기 위해 Offset 를 사용하며, 데이터의 정밀도를 유지하기 위해 scale 값을 지정한다.

X,Y offset 과 scale 를 결정하는 과정

Shpinfo 명령을 사용하여 shape file 분석

Layer 로 로드할 모든 데이터의 범위를 확인

수식을 이용한 X,Y offset 의 계산. 데이터의 모든 좌표 값을 양의 정수 부분에 위치 하도록 한다.



coordinate reference

ArcSDE 레이어에 projection 정보를 부여하는 작업은 ArcSDE administration 명령어인 'sdelayer'를 사용

Projection 정보의 부여는 ArcSDE administration 명령어인 'sdelayer'에 대하여 '-G' 옵션을 사용하여 등록 또는 변경할 수 있다.

Administration 명령어 'sdelayer'로는 기존 레이어 데이터를 새로운 projection 으로 변환시킬 수는 없다. 단지 레이어의 메타데이터 정보만을 부여할 수 있으며, projection 정보를 LAYERS 테이블에 있는 coord\_ref\_system 컬럼에 입력하게 된다.

이미 정의된 projection 에 대해서는 \$SDEHOME/include/pedef.h 에 정의된 숫자를 사용하고, 사용자정의 projection 인 경우에는 사용자가 projection file 을 작성하여 사용해야 한다.

ArcSDE 레이어의 기존 projection 데이터를 새로운 projection 으로 변환하기 위해서는 ArcSDE Projection Engine 을 사용해야 한다.

새로운 layer 생성

Shape 파일 정보확인 : shpinfo -o stats -f -d sde

“Shapes 의 개수”

“Points/Shape 의 평균”

“ArcSDE 컬럼 정의 문자열”.

ArcSDE 에 Business 테이블을 생성 : sdetable -o create

Shpinfo 의 결과 내용 중 “ArcSDE column definition string” 항목의 정보를 이용하여 business 테이블 생성 -d <컬럼 정의>

Business 테이블을 spatially enable 하도록 변경 : sdelayer -o add

위의 과정 1 에서 계산된 값을 사용 -x .

layer 에 포함될 모든 feature 유형을 지정 -e <엔티티 마스크> : npsla+

MaxΔ 값의 1/10 은 Spatial index 로 설정 -g <그리드 크기> (후에 재설정 가능)

“Number of Shapes:” 및 “Average Points/Shape:” 을 이용하여 storage parameter 를 임시적으로 계산하여 데이터를 로딩 -f .

새로운 ArcSDE 레이어로 데이터 로드 : shp2sde -o init

모든 Attribute 컬럼을 추가하기 위해 Flag -a {none | all | file=<파일명>}에 all 사용

< 사용 예 >

**shpinfo -o describe -f world30 -d sde**



```
sdetable -o create -t world30 -d "WRLD30_ID double(12,1)" -u <사용자명> -p <패스워드>  
>
```

```
sdelayer -o add -l world30,feature -e psla+ -f 72,121 -g 36 -x -180,-90,5965232 -u <사용  
자명> -p <패스워드>
```

```
shp2sde -o init -l world30,feature -f world30 -a all -u <사용자명> -p <패스워드> -v  
layer 정보
```

layer에 대한 정보는 sdelayer 명령어의 describe 옵션을 이용해서 볼 수 있다. 이 명령어는 현재 서버에서 사용하고 있는 layer의 모든 정보를 보여준다. 그 사용 예는 다음과 같다.

```
$sdelayer -o describe -u <사용자명> -p<패스워드> -i esri_sde
```

이때 -l 옵션을 이용하면 특정 layer의 정보를 선택적으로 볼 수 있다.

```
$sdelayer -o describe -l mcd,shape -u <사용자명> -p <패스워드>
```

```
ArcSDE 9.1 Build 334 Fri Feb 18 07:04:29 PST 2000
```

```
Layer Administration Utility
```

```
-----  
Minor Civil Divisions  
-----
```

```
Table Owner : PADEP
```

```
Table Name : MCD
```

```
Spatial Column : SHAPE
```

```
Layer id : 1
```

```
Entities : na+
```

```
Layer Type : SDE
```

```
I/O Mode : NORMAL
```

```
User Privileges : SELECT, UPDATE, INSERT, DELETE
```

```
Layer Configuration: PADEP_MCD
```

Layer 삭제

ArcSDE layer를 삭제하는 것은 DBMS의 layer 테이블과 LAYERS 테이블에서의 레이어 정의 부분을 drop 시키는 것이다. 만일 사용자에게 의하여 layer가 lock이 걸려 있다면, sdelayer 명령어는 에러 메시지를 보여주며, layer 삭제를 실패 할 것이다. 사용자는 sdemon의 info 옵션을 사용하면 layer의 locking 여부를 알 수 있을 것이다. 물론 sdemon의 kill 옵션을 사용하면 현재의 lock을 해제 할 수 있다.

sdelayer 명령어를 사용하는 예는 다음과 같다.

```
$sdelayer -o delete -l Victoria,parcels -u <사용자명> -p <패스워드> -i esri_sde
```

## Layer 정의의 수정

Sdlayer의 modify 옵션을 사용하여 사용자는 엔티티 마스크, spatial 인덱스 그리드 셀 크기, 좌표 시스템 ID, layer 정보, layer envelope, 구성 keyword 등을 수정 할 수 있다.

엔티티 마스크, 구성 keyword, 좌표 시스템 ID, layer 정보를 수정하는 예.

```
$ sdlayer -o alter -l Victoria,parcels -e al+ -k vict2 -G 4326 -D "Victoria  
Parcels Layer" -i <사용자명> -p <패스워드> -i esri_sde
```

그리드 크기 수정하는 예

사용자는 layer 가 normal\_io 또는 load\_only I/O 모드 일 때에도 그리드 크기를 변환할 수 있다.

```
$sdlayer -o alter -l Victoria,parcels -g 2000,0,0 -u <사용자명> -p <패스워드  
> -i esri_sde -N
```

layer 최외곽경계(envelope)의 수정

sdlayer의 envelope 옵션은 세개의 keyword 를 가지고 있으며, 그 내용에는 'empty', 'calc' , 'xmin,ymin,xmax,ymax' 이다. 그 사용예는 다음과 같다.

```
$sdlayer -o alter -l Victoria,parcels -E calc -u <사용자명> -p <패스워드> -i  
esri_sde
```

Layer I/O 모드의 변환

I/O 모드를 변환은 sdlayer의 load\_only\_io 또는 normal\_io 옵션을 사용하여 변환할 수 있다. 그 사용 예는 다음과 같다.

```
$sdlayer -o load_only_io -l Victoria,parcels -u <사용자명> -p <패스워드> -i esri_sde  
또는
```

```
$sdlayer -o normal_io -l Victoria,parcels -u <사용자명> -p <패스워드> -i esri_sde
```

## Spatial 인덱스의 관리

Spatial 인덱스 적용기법

ArcSDE는 공간 데이터베이스에 대한 spatial 인덱스를 적용함으로써 Spatial 검색 속도를 향상시킨다. 순수 속성테이블을 대상으로 하는 기존의 정보시스템에서는 B-Tree 방식의 1 차원적인 공간 색인 기법을 사용하는 반면에, ArcSDE는 공간데이터의 관리를 위하여 2 차원 또는 다차원적인 그리드 spatial 인덱싱을 사용한다.

ArcSDE에서 채택하고 있는 spatial 인덱스 체계는 해당 레이어에 걸쳐 있는 다차원 그리드 인덱스 방식이며, 등 간격을 갖는 격자형 그리드 셀 크기와 관련하여 3개 레벨의 그리드가 생성된다. 1차 레벨의 그리드는 가장 작은 셀 크기를 가지며, 2차 레벨 및 3차 레벨의 그리드는 좀더 큰 셀 크기를 가진다. 각기 다른 레벨을 둔 까닭은 검색의 대상이 되는 feature가 낮은 레벨의

무수히 많은 그리드 셀에 걸쳐 있다면 검색의 효율성이 떨어지기 때문에, 이 경우 셀 크기가 큰 2 차 또는 3 차 레벨의 그리드를 이용하여 신속한 검색을 이루도록 하기 위함이다. 검색의 절차를 보면 먼저 검색기준 feature의 최외곽경계(envelope)로 그리드 격자와 오버레이 한다. 오버레이 결과, 최외곽 경계에 걸치는 그리드 격자만을 남기고 모든 그리드 격자와 그 안에 속한 feature는 검색 대상에서 제외한다. 이제 검색대상이 매우 좁혀진 상태에서 최외곽경계와 데이터 레이어를 직접 오버레이 하고, 그 결과로 최외곽 경계에 걸처지는 feature만을 검색 대상으로 간주한다. 끝으로 검색 대상 feature와 검색기준 feature를 오버레이 하여 검색기준 feature에 걸치는 검색 대상 feature만을 서버의 최종 검색 결과로 인정하고 stream을 이용하여 클라이언트에 전송한다.

### Spatial 인덱스의 튜닝

#### Spatial 인덱스의 튜닝

Spatial index의 크기는 layer를 검색할 때 속도에 영향을 미치기 때문에 최적의 크기로 튜닝이 되어 있어야 한다. 그리고 인덱스의 크기는 다음과 같은 기준에 의해서 설정될 수 있다.

Grid 셀의 크기를 평균적인 질의 윈도우 크기 또는 이것의 4 배 정도로 설정

폴리곤 또는 라인 데이터의 경우 Grid/Feature 비율이 1.25 ~ 3.0 이 되도록 설정

평균적인 Feature/Grid의 값이 75 ~ 150 이 되도록 설정

Feature/Grid의 값이 너무 크지 않도록 설정 (예: 2000 개 이상)

Layer 내의 전체 Feature 수가 적을 경우에는 Grid 셀의 크기를 크게 설정

#### Spatial 인덱스의 통계정보 디스플레이

ArcSDE 명령어로 확인이 가능하며 출력되는 정보는 Index의 현재 상태를 설명하는 여러가지의 항목들을 표현하며, 이는 최적의 Index 크기를 결정할 수 있는 기준을 제시해준다. 인덱스 크기가 하나이상 지정되어 있다면 아래의 결과가 각 레벨마다 다르게 표현된다.

```
$ sdelayer -o si_stats -l countbnd,shape -u padep -p padep
```

ArcSDE 9.1 Build 334 Fri Feb 18 07:04:29 PST 2005

Layer 7 Spatial Index Statistics:

Level 1, Grid Size 50000

Grid Records: 273

Feature Records: 67

Grids/Feature Ratio: 4.07

Avg. Features per Grid: 4.01

Max. Features per Grid: 8

% of Features Wholly Inside 1 Grid: 1.49
Spatial Index Record Count By Group
Grids: <=4 >4 >10 >25 >50 >100 >250 >500
Features: 52 15 0 0 0 0 0 0
% Total: 77% 22% 0% 0% 0% 0% 0% 0%

### 그리드 셀 크기 변경

ArcSDE 명령어로 GRID 인덱스의 셀 크기를 변경할 수 있다. GRID 인덱스의 셀 크기는 레벨 3 까지 지정할 수 있지만 하나의 GRID 셀 크기로 지정하는 것 보다 전체적인 프로세싱 속도가 떨어지게 된다. 만약 레이어 상의 사상간의 크기가 크게 차이가 나지 않는다면 첫번째 레벨만 지정한다.

다음은 ArcSDE 명령어로 GRID 셀 크기를 변경하는 예이다.

```
$ sdelayer -o alter -l countbnd,shape -g 100000 -u <사용자명> -p <패스워드> -i
```

### 3. ArcSDE 관리 명령어

ArcSDE 를 관리자는 일반적으로 DBA(데이터베이스 Administrator)이며, ArcSDE 명령어를 사용하여 관리 한다. 대부분의 명령어는 \$SDEHOME/bin 에 있다.

ArcSDE 명령어는 직관적으로 일수 있게 되어 있다. 예를 들어 sdelayer 는 layer 에 관련된 명령어, sdeimport 는 import, sdeexport 는 export 등과 같다. 구체적인 명령어는 다음과 같다.

Sdeexport - ArcSDE export 파일 생성

Sdeimport - ArcSDE export 파일로부터 import

Sdelayer - Map layer 관리

Sdelog - 로그 파일 관리

Sdemon - ArcSDE 서버 관리 및 모니터링

Sdetable - business 테이블 관리

Sde2shp - ArcSDE 데이터를 shape 파일로 변환

Shp2sde - shape 파일을 ArcSDE 데이터로 변환

Cov2sde – Coverage 데이터를 sde 데이터로 변환  
Sde2cov – sde 데이터를 coverage 데이터로 변환  
Shpinfo – ESRI shape 파일의 정보  
Sde2tbl – ArcSDE 테이블을 DBMS 테이블로 변환  
Tbl2sde – DBMS 테이블을 ArcSDE 테이블로 변환  
Sdegrouop – Single part shape 를 Multi part shape 로 결합  
Sderealese – ArcSDE 서버의 Release 의 리스트 및 업그레이드  
Sdexinfo – ArcSDE export 파일의 정보  
Sdeversion – ArcSDE 의 version 정보

#### 4. ArcSDE 에러 메시지별 대응절차

##### 라이선스 관련 에러 메시지별 대응절차

FLEXlm error : invalid returned data from license server (12)

##### **Program not run**

해결 방법 : 이 에러는 사용자가 이전 버전의 FLEXlm 을 사용하려 했을 때 발생하는 에러 메시지이다. SDE 3.0.2.2 버전에서는 FLEXlm v6.0 보다 이전 버전의 FLEXlm 을 사용하면 안된다. 이 에러는 SDE 3.0.2.2 설치시 제공되는 FLEXlm 을 사용하면 해결된다.

9/21 18:00 (ESRI) BAD CODE for ArcSDE

\$SDEHOME/sysgen/lmutil lmstat 명령어를 이용하면 현재의 라이선스 구동 상태를 확인할 수 있으며 이때 다음과 같은 오류 내용을 확인 할 수 있다.

해결 방법 : ArcSDE FEATURE 의 라이선스 내용이 잘못된 코드를 가지고 있다는 오류 메시지이며, 이때에는 다시 한번 제공된 라이선스 파일을 점검한다. 점검 내용은 revision number, time-out date, 라이선스의 개수, keycode 순이다.

FLEXlm Error : encryption code in license file is inconsistent (8)

##### **Program not run.**

해결 방법 : keycode, 즉, 라이선스 파일내의 FEATURE 를 포함하는 라인의 암호화된 코드 부분이 부 정확하다

FLEXlm error : cannot connect to license server (15)

##### **Program not run.**

해결 방법 : 라이선스 관리자가 구동 되지 않았다.

\$SDEHOME/sysgen/lmgrd -c ./license.dat 또는 .cshrc file 에서 정의한 lmstart 명령어를 사용하여 라이선스 관리자를 구동한다.

참고로 .cshrc file 에서 정의 되어있는 lmstart 내용은 다음과 같다.

```
Unix_prompt> more .cshrc
```

~

```
alias lmstart '$SDEHOME/sysgen/lmgrd -c $SDEHOME/sysgen/license.dat &'
```

```
alias lmstat '$SDEHOME/sysgen/lmutil lmstat -a -c  
$SDEHOME/sysgen/license.dat'
```

```
alias lmdown '$SDEHOME/sysgen/lmutil lmdown -c  
$SDEHOME/sysgen/license.dat'
```

~

9/21 17:38 (lmgrd) license manager : Not a valid server host, exiting.

해결 방법 : 사용자가 라이선스 관리자를 실행하려는 중에 \$SDEHOME/sysgen/license.dat 에 있는 호스트 이름과 실제 시스템의 호스트 이름이 다르다. 시스템의 호스트이름과 라이선스 관련 파일(\$SDEHOME/sysgen/license.dat)에 있는 호스트 이름과 정확하게 일치시킨다.

9/21 17:45 (ESRI) Wrong hosted, exiting

해결 방법 : 사용자가 라이선스 관리자를 실행하려는 중에 \$SDEHOME/sysgen/license.dat 에 있는 호스트 ID 와 실제 시스템의 호스트 ID 가 다르다. 시스템의 호스트 ID 와 라이선스 관련 파일(\$SDEHOME/sysgen/license.dat)에 있는 호스트 ID 와 정확하게 맞추어 준다.

### ArcSDE 사용중 에러 메시지별 대응 방법

UX:sh: ERROR: sdemon: Not found

해결방법 : PATH 변수에 \$SDEHOME/bin 디렉토리가 설정되어 있지 않다. .cshrc 파일에 path 를 설정한다.

ArcSDE Already Running

해결방법 : 이미 I/O 관리자가 구동 되어 있다. 만일 I/O 관리자가 비 정상적으로 종료 되었다면 I/O 프로세서를 kill 명령어로 강제 종료 시킨다.

Old IPC Objects Need Be Removed

해결방법 : Shared 메모리와 세마포어가 종료되지 않았다. 이것은 UNIX 명령어인 'ipcs', 'ipcrm'을 사용하여 Shared 메모리와 세마포어를 정리한다. 참고로 ipcs, ipcrm 명령어의 사용법은 다음과 같다.

```
$ipcs -sm
```

```
IPC status from kernel as of Fri Jul 28
```

```
T ID KEY MODE OWNER GROUP
```

Shared Memory:

m 1 0x0f0b0010 --rw-rw-rw- sde sde

m 226 0x1992048f --rw----- sde sde

Semaphores:

s 0 0x00000069 --ra-ra-ra- root root

s 2 0x0f0b0010 --ra-ra-ra- sde sde

s 23 0x1992048f --ra----- sde sde

만일 사용자가 세마포어 23 번과 shared 메모리 226 번을 삭제하고자 한다면 다음과 같이 한다.

**\$ ipcrm -m 226 -s 23**

삭제한 후 다시 한번 ipcs -sm 을 하면 다음과 같이 보여진다.

IPC status from kernel as of Fri Jul 28

T ID KEY MODE OWNER GROUP

Shared Memory:

m 1 0x0f0b0010 --rw-rw-rw- sde sde

Semaphores:

s 0 0x00000069 --ra-ra-ra- root root

s 2 0x0f0b0010 --ra-ra-ra- sde sde

\$ sdemon -o start -p <패스워드>

**This instance's service name esri\_sde not found in system services file.**

해결 방법 : \$SDEHOME/etc/services.sde 에 있는 esri\_sde 5151/tcp 과 시스템의 /etc/services 에 등록된 tcp port 가 일치 하여야 한다. /etc/services 파일에 esri\_sde 5151/tcp 를 추가해준다.

\$ sdemon -o start -p <패스워드>

ArcSDE 9.1 for Oracle8i Build 334 Fri Feb 18 07:04:29 PST 2006

-----  
init\_DB DB\_connect error: -51

DBMS error code: 1034

ORACLE error: code is 1034, func is not used

ORA-01034: ORACLE not available

Could not start ArcSDE -- Check Network, \$SDEHOME disk and DBMS.

해결 방법 : ORACLE 이 구동 되어 있지 않다. ORACLE 을 기동시킨다.

USER 라는 속성이 이미 속성 테이블에 존재한다. 이 경우 다음과 같은 에러 메시지가 발생한다.

ArcSDE 9.1 for Oracle9i Build 334 Fri Feb 18 07:04:29 PST 2006

-----  
USER is an SQL keyword, and hence not a legal column name.

SDE Code (-53) : Invalid column data type

Not able to create business table buildings

Starlifter[SDE]:

해결 방법 : 다른 속성이름을 가진 shape 파일을 재 생성 한다.

저장 파라미터 값이 정확하지 않다. 이 경우 다음과 같은 에러 메시지가 발생한다.

ArcSDE 9.1 for Oracle9i Build 334 Fri Feb 18 07:04:29 PST 2006

-----  
SDE code (-51) = Underlying DBMS error

Extended DBMS error code : 1631

ORACLE error : code is 1631, func is OEXEC,OEXN

ORA-01631 : max # extents (505) reached in table CLASS.R70

Error in stream loading...

SDE Code (-1) – Operation Failed

Exiting on account of SDE error.

SDE Code (-37) : DBMS table not mode.

31941 features converted

30000 features stored.

해결 방법 : 이미 생성되어 있던 layer와 테이블을 삭제하고 저장 파라미터를 재 조정 후 재 로딩한다.