



# DB2 소개 및 DB2 vs Oracle. 비교.

IBM Innovation Center.

@business on demand software



## DB2 Agenda

- ✓ DB2 Introduction
- ✓ DB2 vs Oracle
- ✓ Function Comparison
- ✓ DB2 Program Intro.
- ✓ DB2 Command Summary

## Document History

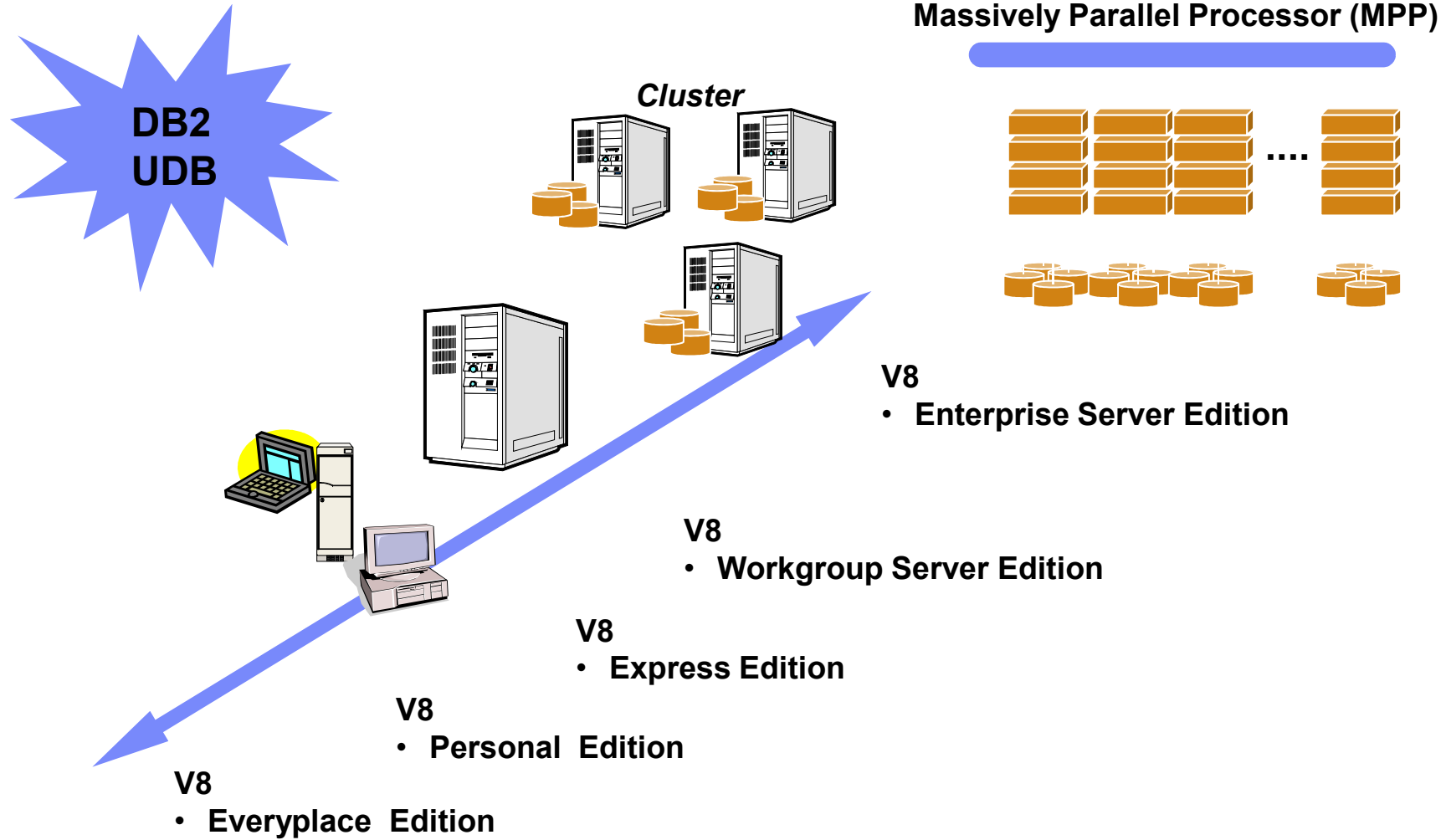
- **Created by Myungjoo Ahn. (2002. )**
- **Merged by Myungjoo ahn. (2004.)**
- **Updated by Myungjoo Ahn. ( 2005.03 ) – nTels.**
- **Updated by Myungjoo Ahn (2005.06.27) – Konan**
- **UPdated by Jungsuk Moon. (2006.07.19)**



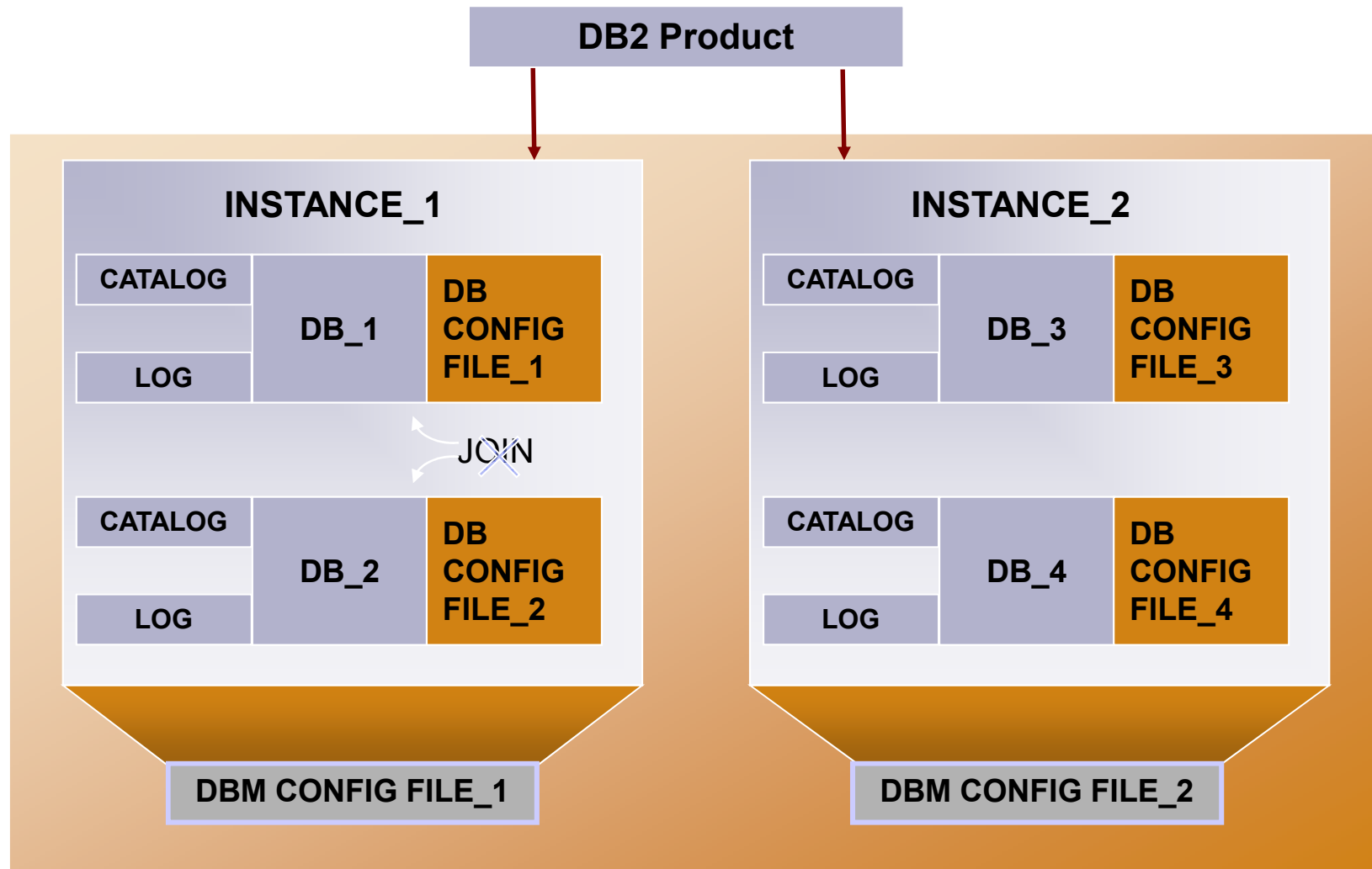
# 2007 DB2 introduction



# DB2 UDB Product



# DB2 인스턴스



# Instance 관리

## db2icrt

- **instance**를 생성하는 명령어.
- **DB2DIR/instance/db2icrt -u <fencedID> <instance\_name>-Unix**
- **db2icrt /u <fencedID> <instance\_name> - Windows**

## db2ilist

- 시스템에 존재하는 **instance**를 나열하는 명령어

## db2idrop

- 시스템에서 **instance**를 제거하는 명령어.

## D B 2 시작/ 중지

- 인스턴스 시작

```
db2start
```

- 인스턴스 중지

```
db2stop
```



# The DB2 UDB Environment

- **DB2 UDB environment**는 **DB**를 제어한다.
- **DB2 environment**를 제어하는 구성 요소
- **Environment variables**
- **DB2 Profile Registry**

# Environment Variables

## •DB2 EnvironmentVariables

•OS	•Location of DB2 Environment Variables
•Windows NT	•[시작]-[설정]-[제어판]-[시스템]에 위치하여 DB2 환경변수를 Setting한다.
•UNIX	•Unix System에 따라 차이가있지만 db2profile이라 하는 Script file (Bourne or Korn shell),or db2cshrc(C shell) •에 시스템 환경변수를 Setting한다.

- 다음은 **DB2 UDB system** 환경변수의 예이다.
  - **DB2INSTANCE** : **DB2 Instance**를 지정한다.
  - **DB2PATH** : **DB2 Path**를 지정한다.

# Setting DB2 Variable Values



## The db2set 명령어

- **Command Line tool**
- **DB2 Profile Registry**를 관리한다.
- **profile variables**값을 **Display, Sets, Resets** 혹은 **Remove** 한다.

**db2set variable=value**

**Options :**

**-g**

**-i instance [node number]**

**-n DAS node [ [-u user id] [-p password] ]**

**-r**

**-l**

**-lr**

**-v**

**-? (or -h)**

**-all**

**-null**

## The db2set 명령어 [ 계속 ]

- **db2set** 명령어 옵션 :
  - **g**    **global profile variables**를 수정한다.
  - **i**    수정하고자 하는 **Registry Profile**의 **instance**를 지정한다.
  - **n**    **remote DB2 Admin Server node name**을 지정한다.
  - **r**    주어진 **instance**에 대한 **profile registry**를 **reset**한다.
  - **l**    **all instance profiles**을 나열한다.
  - **lr**    모든 지원되는 **registry variables**값을 나열한다.
  - **v**    **Verbose mode**
  - **?**    **command help message**를 **Display**한다.
  - **all**    모든 **setting된 Profile Registry variables** 값을 **Display**한다.
  - **-null**    지정된 **registry level**에서 변수값을 **null값으로 Setting**하여 다음 **registry level**에서 그 값을 찾지 못하도록 한다.

# Instance Configuration File.

- **Database Manager Configuration file**

- **Instance**가 생성될때 생성되어진다.
- **instance**당 하나의 **Configuration File**이 존재한다.
- 각 **instance**에 대한 **System resources**에 영향을 준다.

- **DBM CFG**에 대한 명령어.

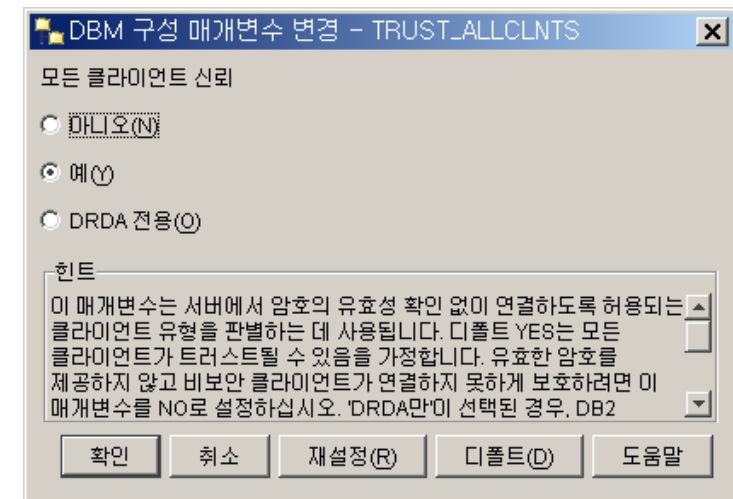
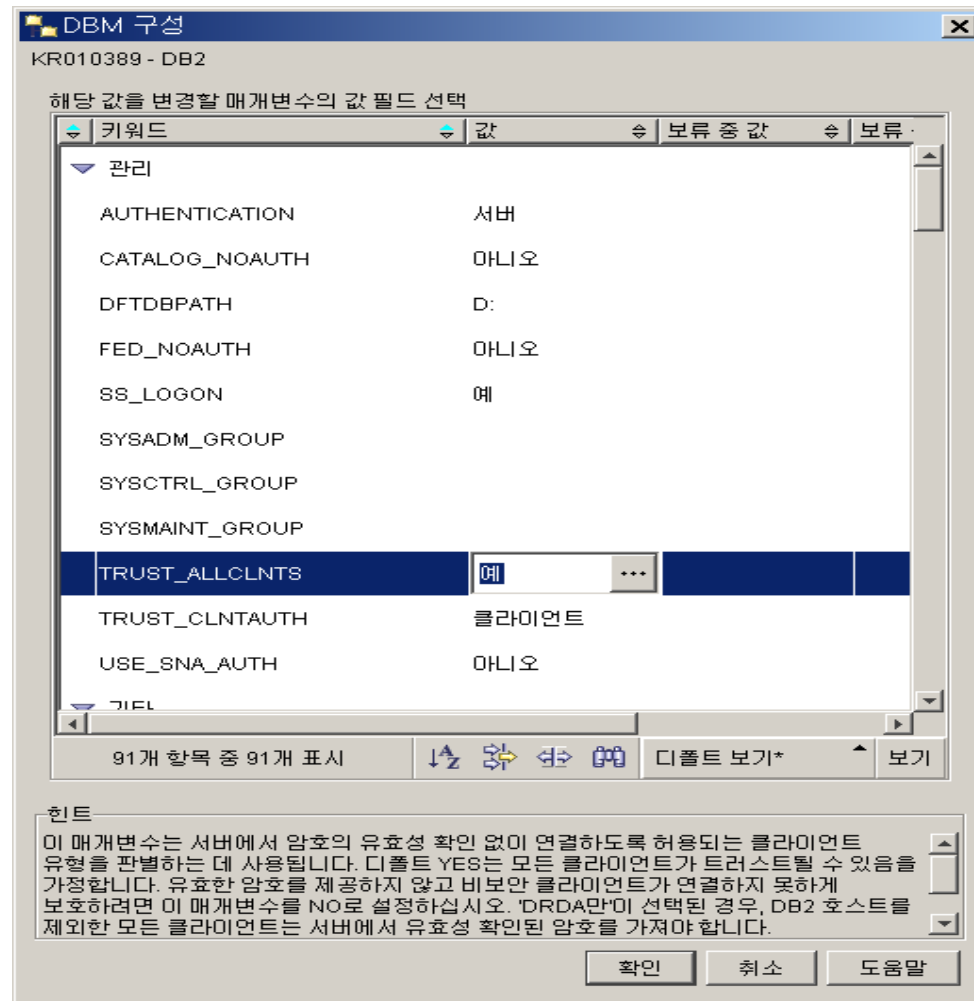
- **db2 get dbm cfg**

- **Configuration File**의 **Parameter**값을 보여주는 명령어.

- **db2 update dbm cfg using <Parameter> <Value>**

- **Configuration File**의 **Parameter**값을 수정하는 명령어.

## Instance Configuration (GUI)



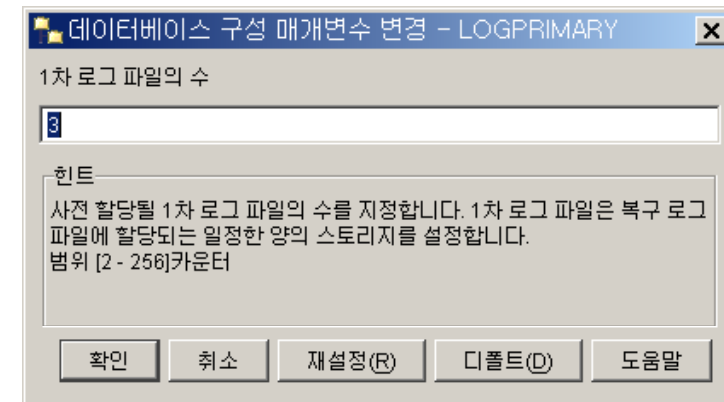
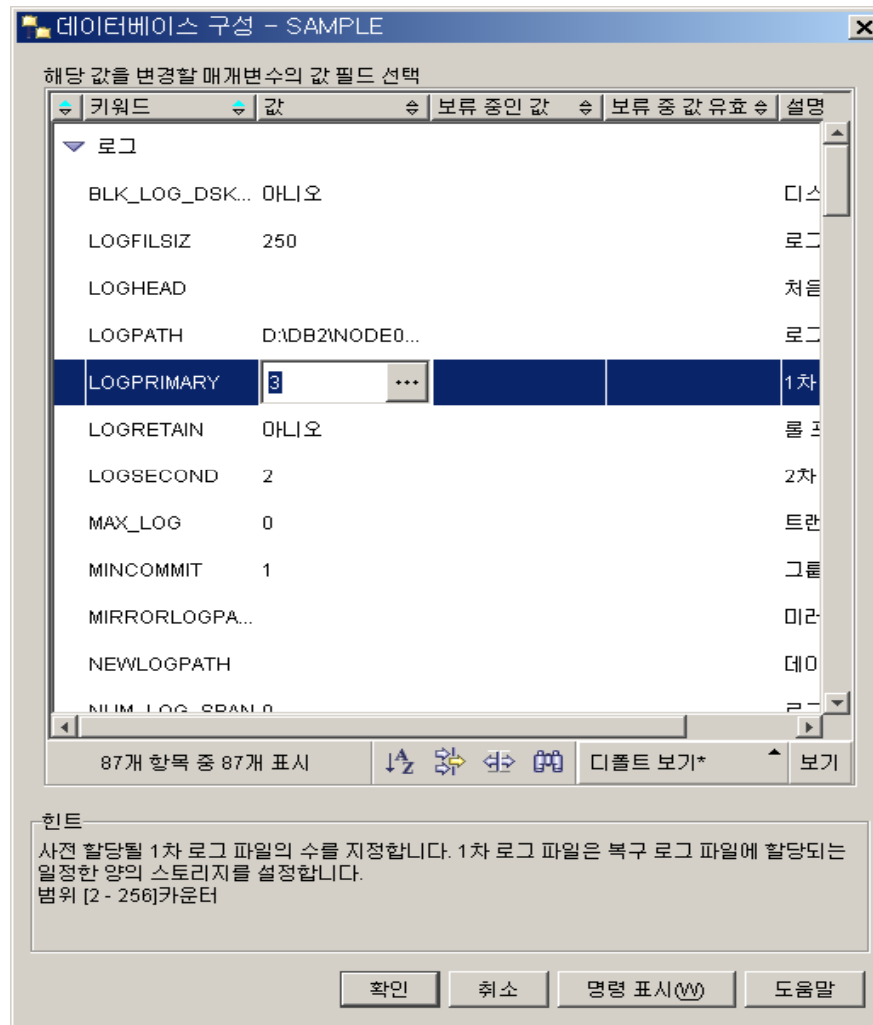
# Database Configuration (GUI)

## • Database Configuration file

- DB가 생성될 때 같이 생성된다.
- DB당 하나의 **Configuration File**이 존재한다.
- **parameter**는 DB에 할당된 **resource**의 양을 지정한다.
- DB CFG에 대한 명령어.
  - **db2 get db cfg for <DBNAME>**
    - **Configuration File의 Parameter값을 보여주는 명령어.**
  - **db2 update db cfg for <DBNAME> using <Parameter> <Value>**

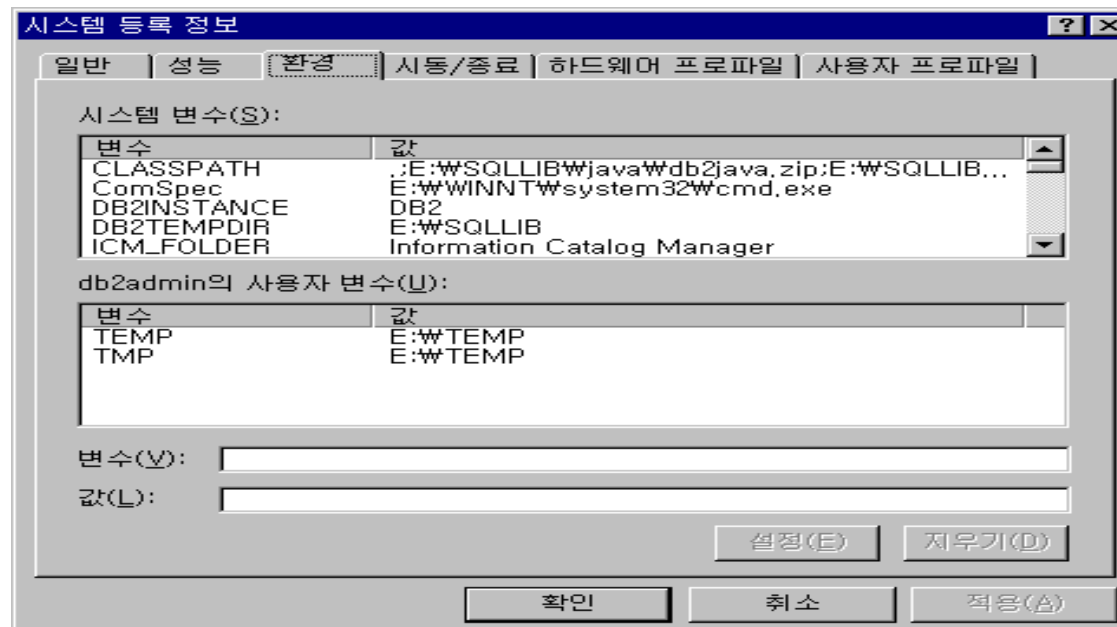


# Database Configuration ( Cont )



## 환경 변수 Setting on NT

- 인스톨이 완료되면 다음 항목이 자동 설정된다.
  - 인스턴스가 자동으로 생성된다.
  - \winnt\system32\drivers\etc\services 파일에 DB2 관련 services 정보가 추가된다.
  - 윈도우 제어판의 환경변수가 자동으로 설정된다.
- [시작]-[설정]-[제어판]-[시스템]을 확인하면 아래 그림과 같이 DB2관련 환경변수가 등록되어 있는 것을 확인할 수 있다.



## 환경 변수 Setting on AIX

- **install**이 완료되면 다음 항목이 자동 설정된다.
  - 설치시 입력했던 **instance**가 생성된다.
  - **instance owner**의 **home directory**에 **sqllib directory**가 추가된다.
  - 인스턴스 작성시 지정한 사용자 (**ex. Db2inst1**)로 로그인하여 **Home**디렉토리의 **.profile**에 아래의 내용이 삽입되었는지 확인한다.

```
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:$HOME/bin:/usr/bin/X11:/sbin:.
```

```
export PATH  
PS1='$PWD/'
```

```
# The following three lines have been added by UDB DB2.  
if [ -f sqllib/db2profile ]; then  
    . sqllib/db2profile  
fi
```

- **/etc/services** file과 데이터베이스 매니저 구성파일을 참조하여 **Port Number**와 **Service Name**을 확인한다.

# Communication Setting

## •TCP/IP Configuration

- **DB2** 은 인스턴스를 생성하면 그 인스턴스에 대한 **TCP/IP** 환경이 자동으로 설정된다. 즉 **winnt\system32\drivers\etc\services** file에 **UDB** 인스턴스에 대한 **Service name**과 **port**가 자동으로 할당된다.
- 기본값 :인스턴스명 = **db2inst1**, **service name=db2cdb2**, 포트번호= **50000**
- 서버 구성 파일은 명령창에서 다음과 같이 입력하면 볼 수 있다. (**svcname** 확인)

```
db2 get dbm cfg
```

- 인스턴스 이름을 변경하거나, **service name**, **port** 등을 변경하고자 할 때는 편집기를 이용 직접 **services** 파일을 변경/수정 하고, 인스턴스 레벨의 변수를 수정하기 위해선 **db2 update dbm cfg using [파라미터] [값]**으로 변경할 수 있다.

service file을 변경할 경우

```
# vi /etc/services
db2cdb2inst2    50002
db2idb2inst2    50003
```

Database Manager 구성 파일을 변경할 경우

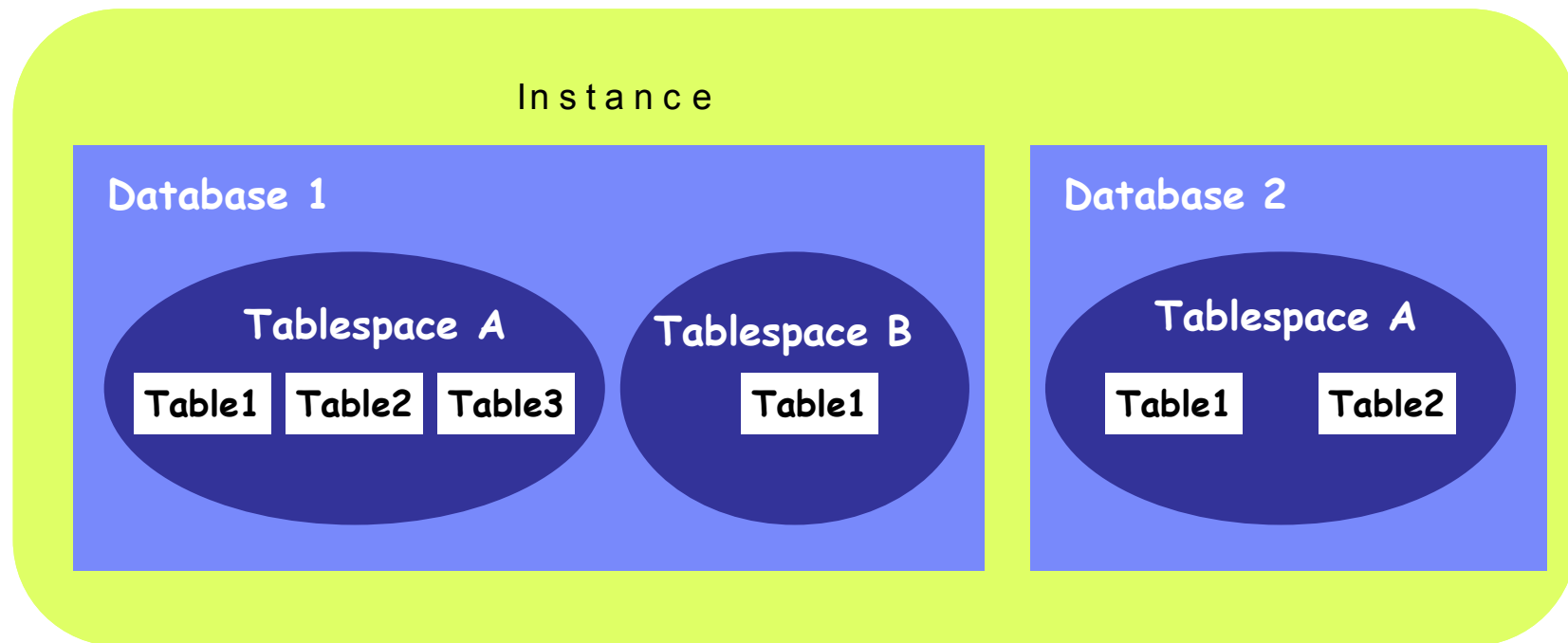
```
# db2 update dbm cfg using svcname db2cdb2inst2
```



## DB / Table Space 생성

@business on demand software

# DataBase / Tablespace / Table



- 테이블 공간은 데이터베이스안에 만들어지는 논리적인 **Layer**이다.
- 테이블은 테이블 공간에 만들어진다.

# Create Database 구문

- 권한 부여 : sysadm or sysctrl
- 필수 연결 : Instance.

▶▶ CREATE Database Database-name  
DB Create Database Options

CREATE Database Options :

▶ ON PATH Drive ALIAS Database-alias USING CODESET CODESET

▶ TERRITORY TERRITORY COLLATE USING SYSTEM IDENTITY NUMSEGS numsegs

▶ DFT\_EXTENT\_SZ Dft\_extentsize CATALOG TABLESPACE Tblspace-defn

▶ USER TABLESPACE Tblspace-defn TEMPORARY TABLESPACE Tblspace-defn

# Create Database ( GUI )

데이터베이스 작성 마법사

1. 이름  
2. 사용자 테이블  
3. 카탈로그 테이블  
4. 임시 테이블  
5. 성능  
6. 지역  
7. 요약


사용자의 새 데이터베이스에 대한 이름을 지정하십시오.  
이 마법사는 사용자가 새 데이터베이스를 작성하고 조정하는 데 도움을 줍니다. 기본 데이터베이스를 작성하려면, 새 이름을 입력하고 드라이브를 선택한 다음 '완료'를 누르십시오. 데이터베이스를 사용자의 요구에 맞게 조정하려면 '다음'을 눌러 계속하십시오.

데이터베이스 이름

기본 드라이브

별명

주석



다음 ► 완료 취소

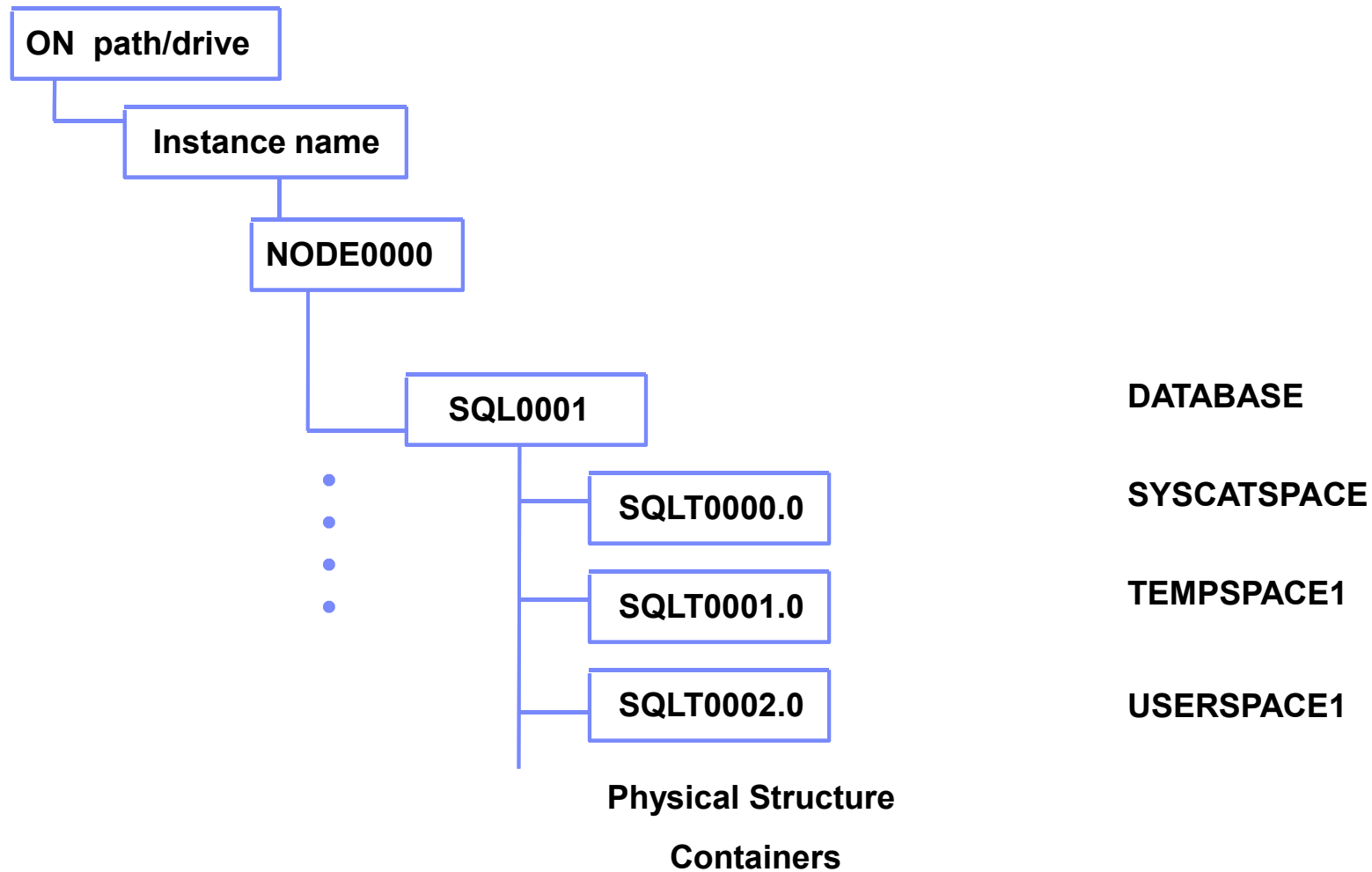


## 데이터베이스 생성중에 수행되는 작업

1. 데이터베이스를 생성한다.
2. **SYSCATSPACE, TEMPSPACE1, USERSPACE1** 테이블 공간을 생성한다.
3. **system catalog tables** 과 **recovery log**를 생성한다.
4. **local database directory** 와 **system database directory**에 데이터베이스를 카탈로그한다.
5. **codeset, territory**, 그리고 **collating sequence**를 지정한다.
6. **SYSCAT, SYSFUN, SYSIBM, SYSSTAT** schemata를 생성한다.
7. 전에 정의된 데이터베이스 **Manager bind file**을 데이터베이스에 바인드한다.
8. 다음과 같은 **privileges**를 **Grant**한다 :
  - **DBADM authority**를 데이터베이스 생성자에게 부여.
  - **System catalog tables** 와 **views**에 대한 **SELECT Privilege**를 **PUBLIC**에게 부여.
  - **BIND**와 **EXECUTE privilege**를 **PUBLIC**에게 부여.
  - **CREATETAB, BINDADD, IMPLICIT\_SCHEMA**와 **CONNECT Privilege**를 **PUBLIC**부여.
  - **USERSPACE1 table space**에 대한 **Use privilege** 를 **PUBLIC** 에게 부여.

## Default Database Configuration

### DB2 Create database TESTDB



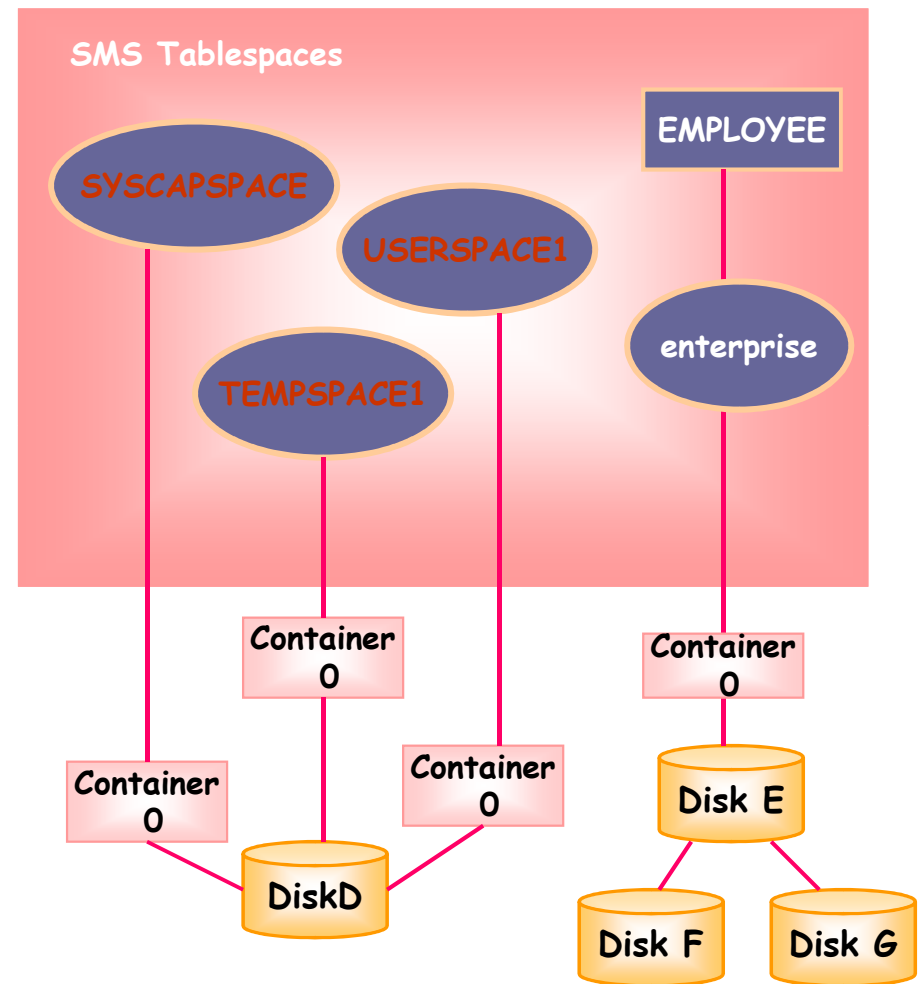
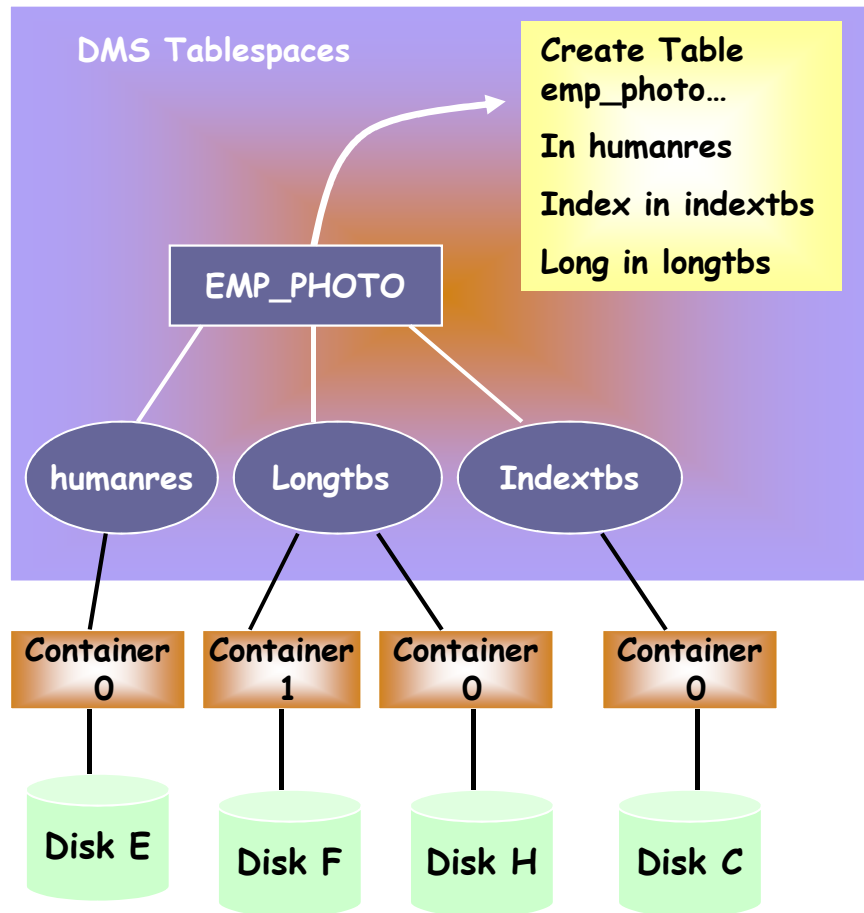
## 테 이 블 공간과 컨테이너

- 테이블 공간
  - 테이블들을 저장하는 장소.
  - 하나 이상의 물리적인 저장 디바이스로 나누어질 수 있다.
  - 테이블 공간의 타입
    - **SMS ( System Managed Space )**
      - **Database Manager**는 **OS**를 이용하여 데이터를 관리한다.
    - **DMS ( Database Managed Space )**
      - **Database Manager**에 의해 직접적으로 관리된다.
- 컨테이너
  - 데이터가 실제로 저장되는 물리적 저장장치이다.
  - 단일 테이블 공간은 여러 컨테이너에 걸쳐 있을 수 있지만, 각 컨테이너는 오직 하나의 테이블 공간에만 속할 수 있다. (**TS**가 여러 컨테이너를 포함한다.)
  - **SMS** 테이블 공간일 경우, 컨테이너는 디렉토리이다. (**OS file system**에 위임)
  - **DMS** 테이블 공간일 경우, 컨테이너는 파일이나 디바이스이다.

# 테이블 공간 타입.

- 테이블 공간 타입
  - **Regular**
    - **Temporary Table**을 제외한 모든 데이터를 저장한다.
  - **long**
    - **long varchar, long vargraphic** 또는 **LOB data**를 저장한다.
    - 반드시 **DMS** 테이블 공간이어야 한다.
    - ★ **SMS** 테이블 공간은 이 타입으로 생성할 수 없다.(**OS file system**에 위임)
  - **temporary**
    - **DB System**에 의해 이용되는 **Temporary** 공간.
- 테이블 **Data** 타입.
  - **Regular** - **long varchar, long vargraphic** 또는 **LOB data**를 제외한 모든 데이터
  - **Long** - **long varchar, long vargraphic** 또는 **LOB data**
  - **Index** – 테이블에 생성되는 **Index**

# Sample 시나리오



# SMS VS DMS

	SMS	DMS
제어	운영체제	Database Manager
컨테이너	디렉토리	파일, 혹은 Row device
공간할당	추가 공간이 요구되어질 때마다 한 <b>Page</b> 씩 할당된다.	공간이 미리 할당되며 하나의 <b>Extent</b> 안에 있는 공간은 물리적으로 연속적이다.
테이블 위치	한 테이블은 한 테이블 공간에 존재해야 한다.	하나의 테이블은 여러 개의 테이블공간으로 나눌 수 있다.
크기를 증가시키는 법.	운영체제 파일 시스템 변경  <b>DB</b>  <b>redirect restore ~ 이용.</b>	<b>DB</b>  <b>Alter tablespace Add container</b>  <b>Alter Tablespace resize</b>  <b>Alter Tablespace extend</b>
장점	요구되어질 때 시스템에 의해 공간이 할당된다.  데이터베이스를 생성할 때 초기 작업이 줄어든다.	컨테이너를 추가함으로써 테이블 공간이 증가된다.  데이터 타입에 따라 테이블을 여러 테이블 공간으로 나눌 수 있다.
적절한 business	작은 <b>personal databases</b> 용으로. <b>Temporary</b> 테이블 공간용으로	크고 점점 커지는 데이터 베이스용으로 .

## S M S P h y s i c a l F i l e s .

다음 파일은 **SMS** 테이블 공간 컨테이너에 존재한다. :

파일 이름	설명
<b>SQLTAG.NAM</b>	데이터베이스가 완전하고 일치하는지 검증하기 위해 사용자가 데이터베이스에 연결할 때 <b>DBM</b> 에 의해 이용된다.
<b>SQLxxxxx.DAT</b>	<b>LONG VARCHAR , LONG VARGRAPHIC ,LOB</b> 데이터를 제외한 모든 데이터가 저장된다.
<b>SQLxxxxx.LF</b>	<b>LONG VARCHAR or LONG VARGRAPHIC</b> 을 포함하는 파일.
<b>SQLxxxxx.LB</b>	<b>BLOB, CLOB, 또는 DBCLOB</b> 데이터를 포함하는 파일
<b>SQLxxxxx.LBA</b>	<b>SQL.LB</b> 파일에 대한 공간 할당 및 여유 공간정보를 포함하는 파일.
<b>SQLxxxxx.INX</b>	해당 테이블에 대한 모든 <b>index</b> 는 이 파일에 저장된다 .

## Page 크기

- 각 페이지는 (페이지 크기에 상관없이) **76** 바이트의 오버헤드를 포함한다.
- 한 행은 여러 개의 페이지로 나누어 저장될 수 없다.
- **random**하게 행을 읽고 쓰는 **OLTP**성 응용 프로그램은 작은 페이지 크기가 더 좋다.
- 한번에 연속된 대량의 행에 접근하는 **DSS (decision support system)** 응용 프로그램은 큰 크기의 페이지 크기가 유리하다.

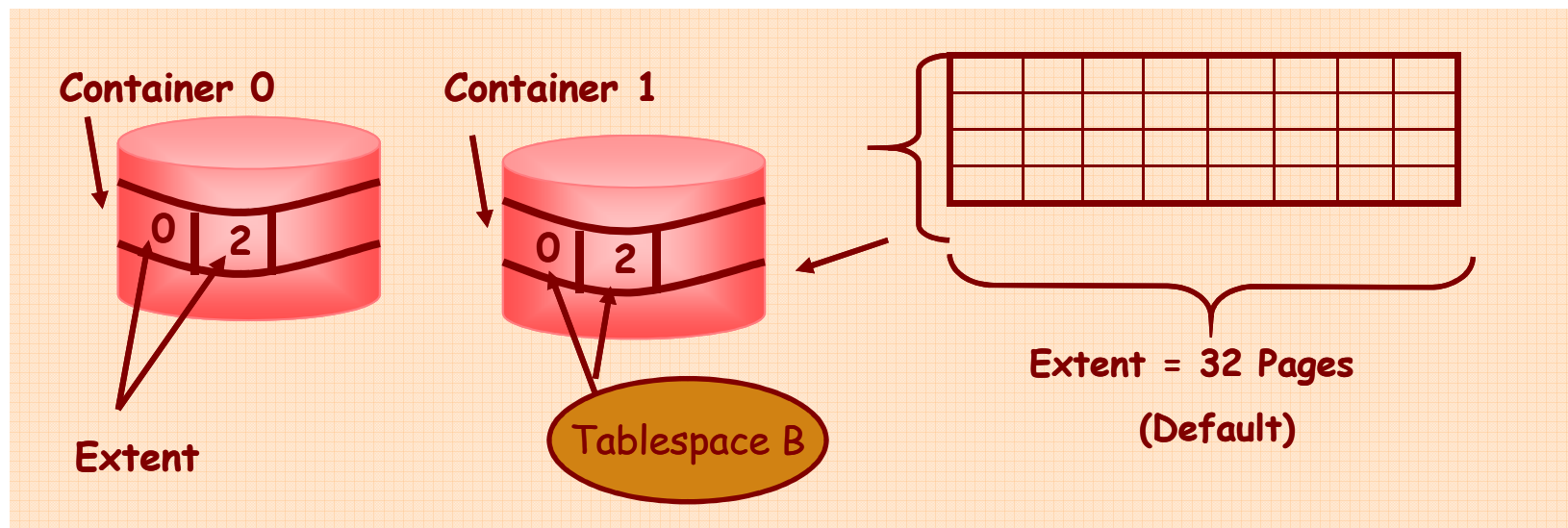
Table Limit

	4K Page	8K Page	16K Page	32K Page
칼럼의 최대수	500개	1012개	1012개	1012개
행의 최대 길이	4005Byte	8101Byte	16293 Byte	32677Byte
테이블의 최대 크기	64GB	128GB	256GB	512GB
행의 최대 수	255 행	255행	255 행	255행



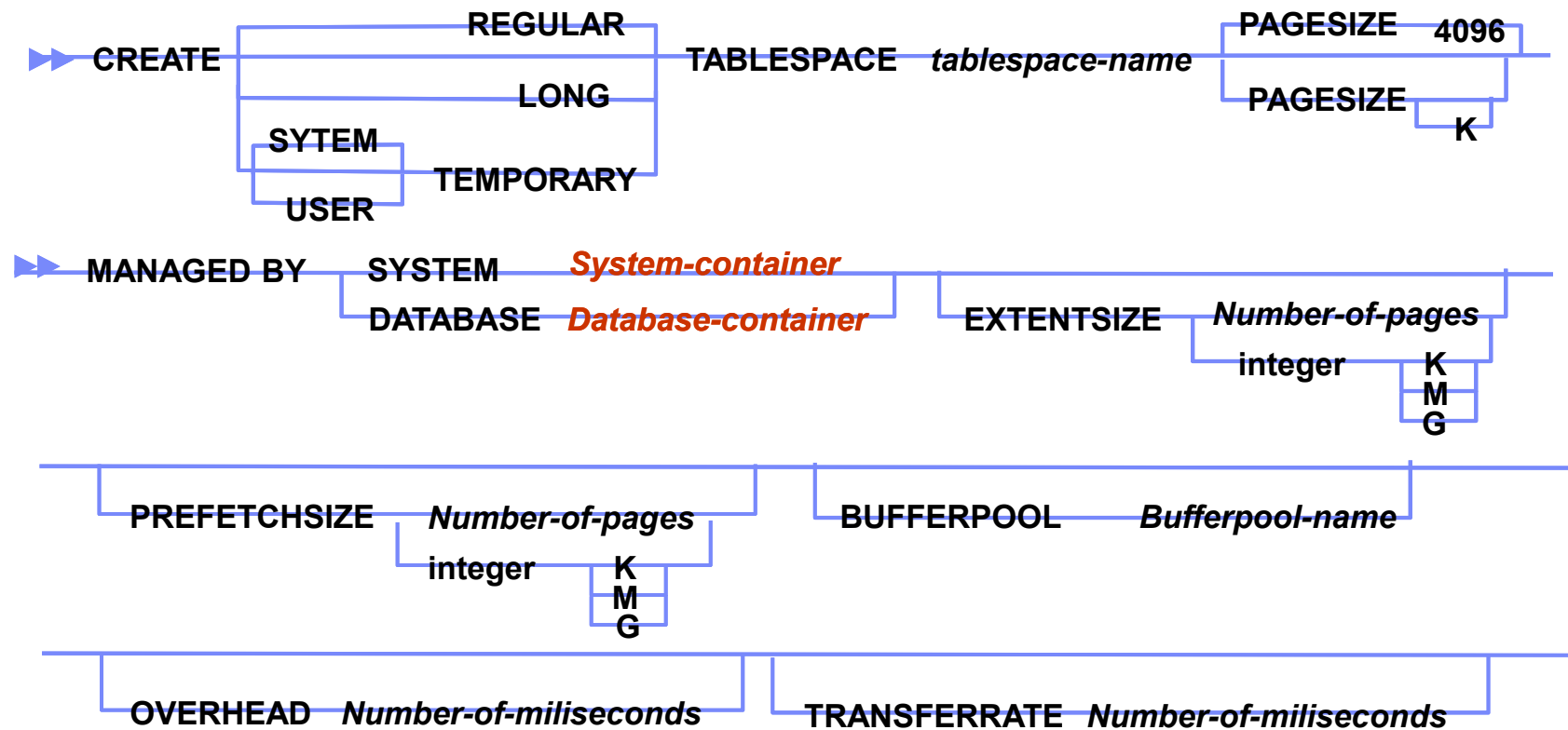
## Extent 크기

- 데이터베이스 관리 프로그램이 다른 컨테이너를 사용하기 전에 하나의 컨테이너에 기록하는 페이지 수
- 데이터는 라운드 로빈 유형으로 테이블 공간 내의 모든 컨테이너에 저장됨으로써 컨테이너간에 데이터의 균형을 이룬다.
- 데이터베이스를 생성할 때 **DFT\_EXTENT\_SZ** 파라미터를 이용하여 **Extent** 크기를 정할 수 있다.
- 테이블공간을 생성할 때 **EXTENTSIZE** 에 값을 지정하여 **Extent** 크기를 정할 수 있다.



# Create TableSpace ~ 구문

• 권한 : *sysadm or sysctrl*



# Create TableSpace ~ 구문 ( 계속 )

## System – containers :



## Database – containers :



## Table 공간 생성 예제 ( GUI )

테이블 공간 생성

KR010389 - DB2 - SAMPLE

테이블 공간 이름 dms01

테이블 공간의 유형

☒ 일반
☐ Long
☐ 시스템 임시
☐ 사용자 임시

공간 관리

☐ 시스템
☒ 데이터베이스

컨테이너 이름

유형

크기(MB)

할당된 총량: 0.00 MB

주석

확인

취소

SQL 표시

고급...

컨테이너 추가

파일 크기

20 MB

☒ 파일
☐ 원시 장치

컨테이너

DMS01

현재 디렉토리

D:\DB2\

파일

DMS01

디렉토리

..

NODE0000

SAMPLE.0

드라이브

D:

1855 MB 사용 ...

확인

취소

적용

도움말

## Table 공간 생성 예제 ( Command )

### ◆ Creating SMS Table Spaces

#### ◆ Unix :

- Create tablespace *SMSTBS* managed by system Using ('/SMS/contain1' , ' /SMS/contain2') prefetchsize 32

#### ◆ intel:

- Create tablespace *SMSTBS* managed by system using ('E:\SMS\contain1','F:\SMS\contain2') prefetchsize 32

### ◆ Creating DMS Table Spaces

#### ◆ Unix :

- Create tablespace *DMSTBS* managed by database using (device '/dev/rdata1' 1024, device '/dev/rdata2' 2048 ) extentsize 32

#### ◆ NT:

- Create tablespace *DMSTBS* managed by database using( file 'E:\DMS\contain1' 2500) extentsize 32

## List tablespaces 명령어

- **db2 list tablespaces**
- **db2 list tablespaces show detail**
- **db2 list tablespace containers for X**
- **db2 list tablespace containers for X show detail**

# List Tablespaces 명령어 예제

## C:\>db2 list tablespaces

현재 데이터베이스에 대한 테이블 공간

테이블 공간 ID	= 0
이름	= SYSCATSPACE
유형	= 시스템 관리 공간
내용	= 임의의 데이터
상태	= 0x0000
테이블 공간 ID	= 1
이름	= TEMPSPACE1
유형	= 시스템 관리 공간
내용	= 시스템 임시 데이터
상태	= 0x0000

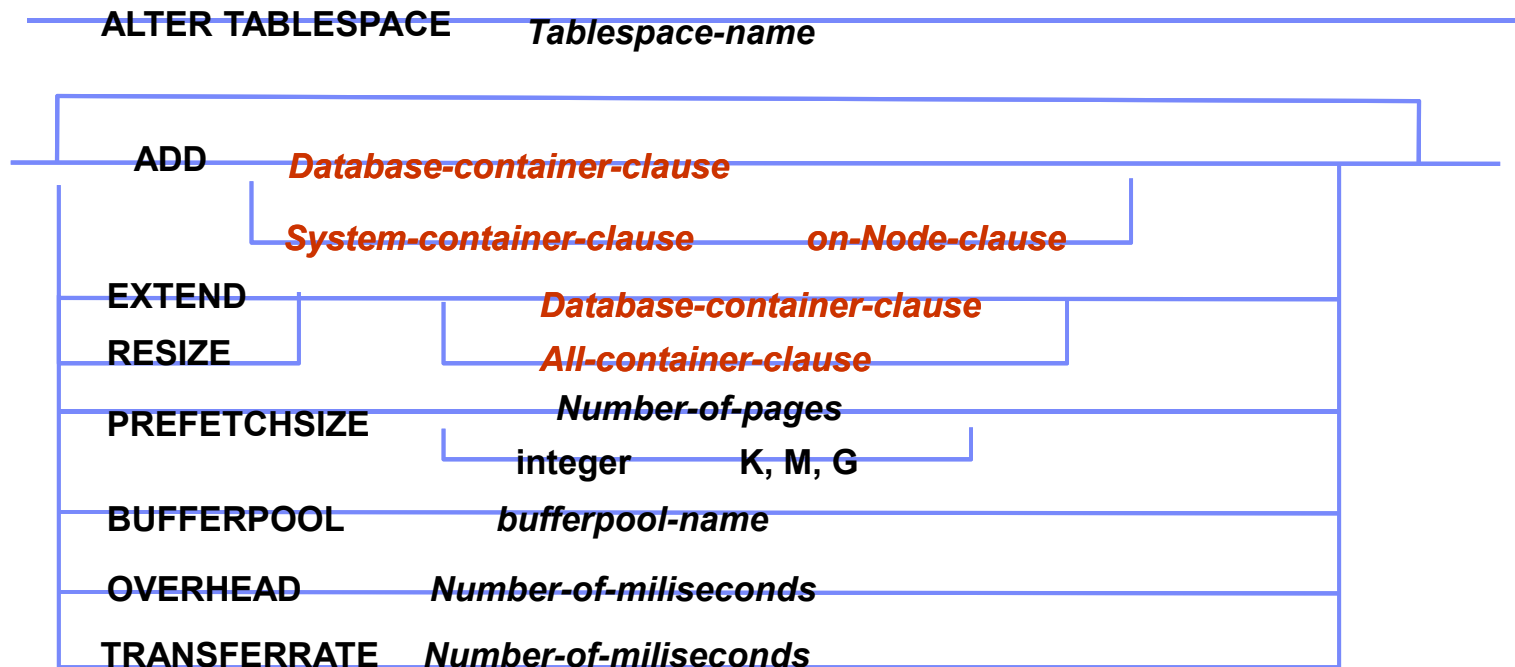
## C:\>db2 list tablespace containers for 1

테이블 공간 1에 대한 테이블 공간 컨테이너

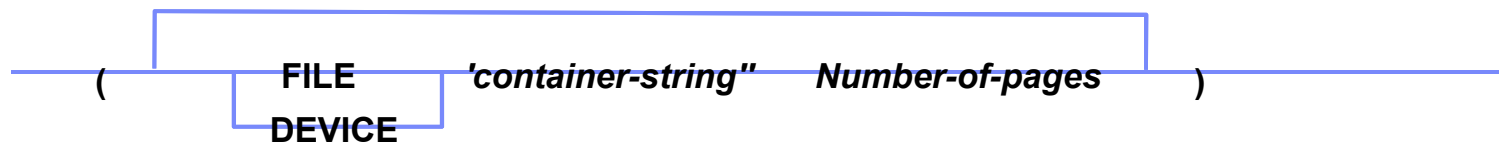
컨테이너 ID	= 0
이름	= C:\DB2\NODE0000\SQL00001\SQLT0001.0
유형	= 경로

# ALTER TABLESPACE ~ 구문

• 권한 : *sysadm or sysctrl*



**Database-container-clause :**





# Alter Table Space ~ 예제

## GUI

## 명령어

```
Alter tablespace dms01 extend (file 'D:\dms\dms01' 5 );  
Alter tablespace dms01 resize (file 'D:\dms\dms01' 25);
```

## Table Space 크기 증가

SMS	DMS
운영 체제의 파일 시스템 변경 <b>redirect restore ~ 이용.</b>	<b>Alter tablespace Add container</b> <b>Alter Tablespace resize</b> <b>Alter Tablespace extend</b>
	<ul style="list-style-type: none"><li>• 테이블 공간은 사용자에게 접근가능.</li><li>• 데이터는 컨테이너사이에서 재 분배됨.</li></ul>

# Placement Considerations

- 인덱스스캔이 요구되면
  - 인덱스와 데이터를 각각 다른 테이블 공간에 저장할 것을 고려.
- 인덱스 테이블 공간을 가장 빠른 **Media**에 놓을 것을 고려.
- 전체 테이블을 자주 **Access**되는 각각 다른 테이블 공간으로 나누어서 놓는다.
- **EXTENTSIZE** 고려.
  - 공간 효율성과 성능사이에 이율 배반적.
- **I/O Prefetch**

## 테이블 공간 관리

- 테이블 공간을 **Drop** 할 경우
  - 테이블 공간에 저장된 모든 데이터가 삭제되고 컨테이너를 **Free**시킨다.
  - 여러 개의 테이블 공간이 지정될 수 있다.; 하나의 테이블이 여러 개의 테이블 공간에 걸쳐 있을 경우 테이블 공간을 **Drop** 할 때 모든 테이블 공간이 지정되어야 한다. 그렇지 않다면 테이블 공간을 **Drop** 할 수 없다.

### SYSCAT.TABLES

TABNAME

TABLEID

TBSPACEID

### SYSCAT.TABLESPACES

S=SMS  
D=DMS

A=All  
L=Long  
T=Temp

TBSPACE

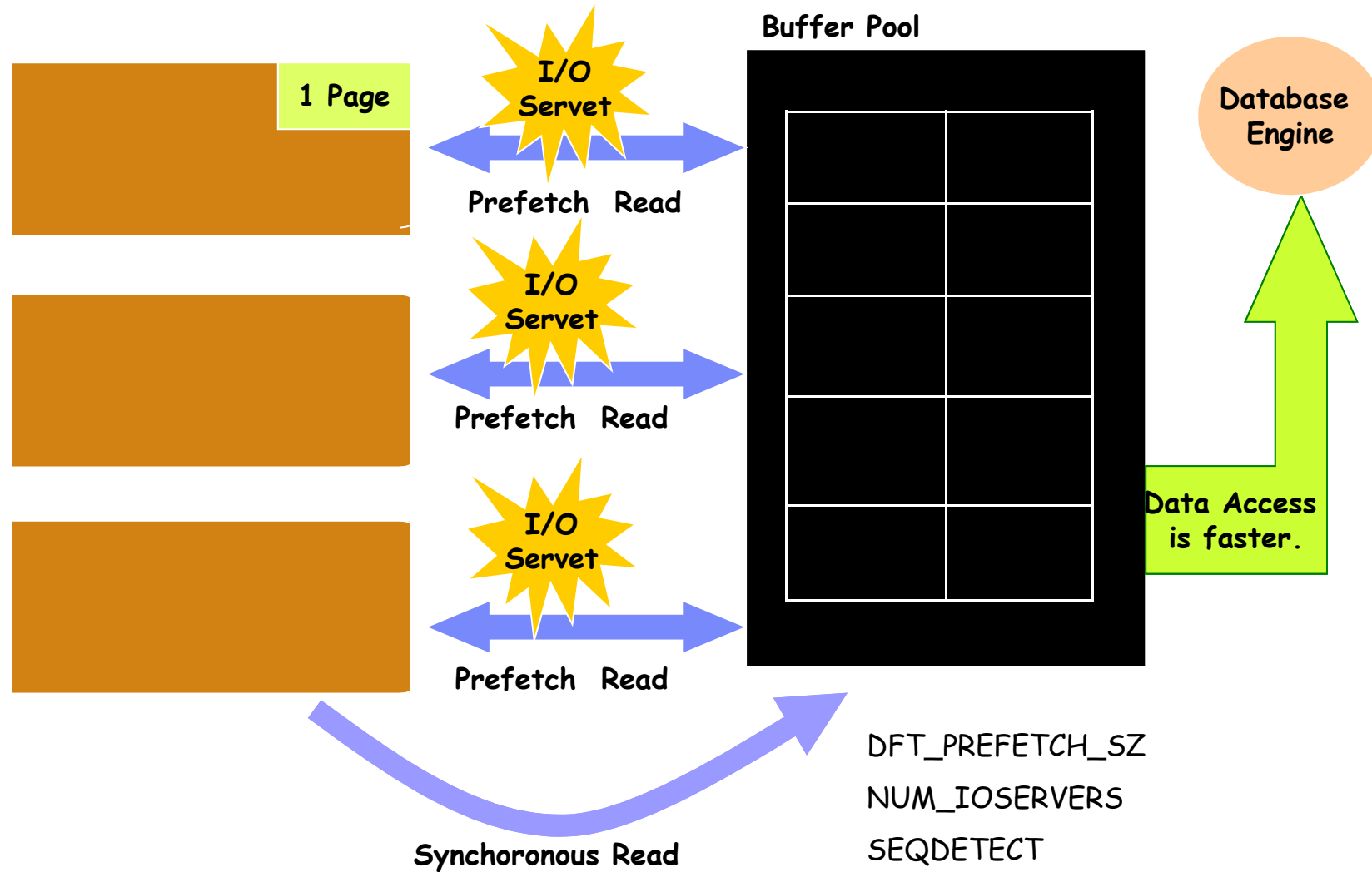
TBSPACEID

TBSPACETYPE

DATATYPE

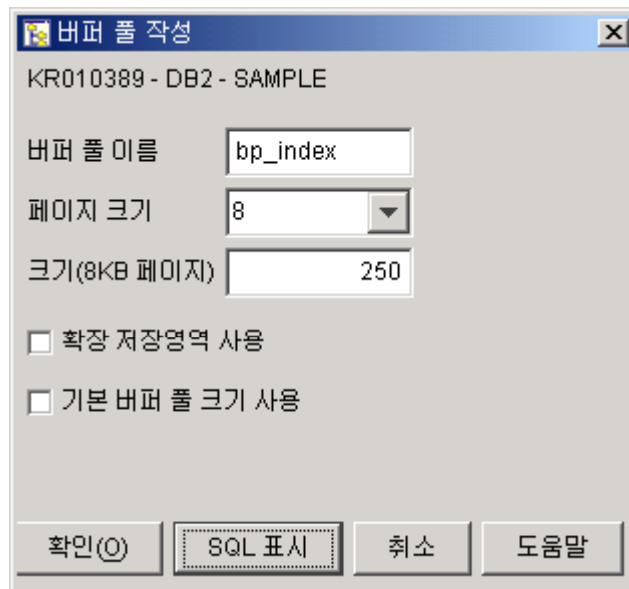
EXTENTSIZE

# I/O Prefetch 예제



# CREATE BUFFERPOOL Example

- 권한 : *sysadm or sysctrl*



테이블 공간은 반드시 테이블 공간의 페이지 크기와 동일한 페이지 크기를 갖는 버퍼 풀이 존재해야 한다.

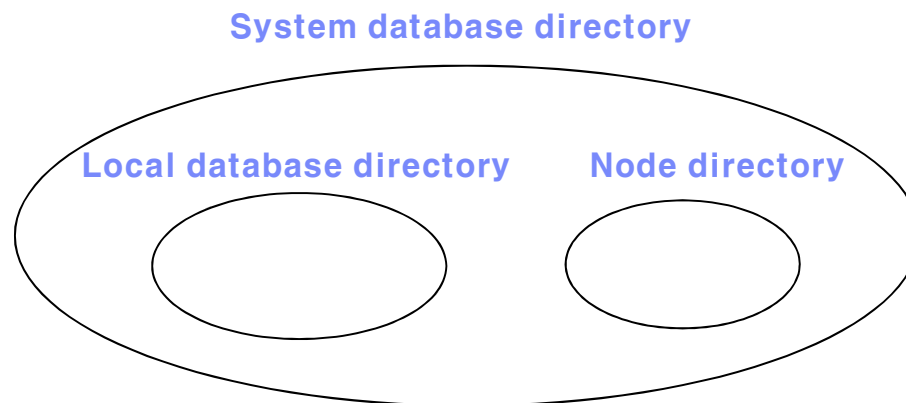
(Buffer Pool의 페이지 크기 = Table Space 페이지 크기)

Create bufferpool bp\_index size 125 pagesize 8K

Alter tablespace indextbs bufferpool bp\_index

## DB2 디렉토리란 ?

- DB2에서 디렉토리는 **Database**와 이에 대한 연결 정보를 가지고 있는 **binary file**이다.
- DB2 디렉토리는 ...
  - 1)System database directory(system db directory)
  - 2)Local database directory(local db directory)
  - 3)Node directory
  - 4)DCS directory( zSeries™:S/390 or iSeries™:AS/400 에서 사용 ) - 무시와 같이 4가지가 있다.



## DB2 Directory Schematic View

## DB2 디렉토리란 ? (Cont)

### 1) System database directory

- **SQLDBDIR** 디렉토리에 존재한다.
- **Local database directory(2)**와 **Node database directory(3)**를 포함한다.
- “*db2 list db directory*” 명령을 실행하면 **system database directory** 목록을 볼 수 있다.
- ‘**Directory Entry Type**(디렉토리 항목 유형)’에 ‘**Indirect**(간접)’이라고 나오면 이는 **local database**이고, ‘**Remote**(원격)’이라고 나오면 이는 **Node database**이다.
- **local database directory**와 **Node directory**를 가리키는 포인터가 들어 있다.

### 2) Local database directory

- **local database**에 대한 연결 정보 항목으로 “*create database ..*” 명령을 통해 자동으로 등록된다.(**Catalog** 명령이 불필요)
- “*db2 list db directory on <path>*” 명령을 통해 목록을 볼 수 있다.

### 3) Node directory

- 원격 **database**에 대한 연결정보를 저장하는데 이를 위해서는 **Node**가 등록되어 있어야 한다.
- **Node**의 등록은 “*db2 catalog tcpip node <node\_name> remote <ip\_addr> server <port\_number>*” 명령을 통해서 할 수 있다.



# DB2 디렉토리란 ? ( Cont )

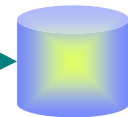


## System DB Directory

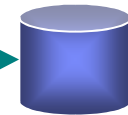
DB Alias	DB Name	TYPE	Node	Pointer
L_DB	DB1	Indirect		
R_DB	DB1	Remote	SPC	

## Local DB Directory

DB Alias	DB Name	TYPE	Location Info
L_DB	DB1	간접	C:\



DB1 (local)



DB1 (remote)

## Node Directory

Node Name	IP Addr	Port Num
SPC	10.x.x.1	50001

# Examining DB2 Directories

## System Database directory

**C:\>db2 list db directory**

시스템 데이터베이스 디렉토리

디렉토리 내 항목의 수 = 3

데이터베이스 1 항목:

데이터베이스 별명	= <b>SAMPLE</b>
데이터베이스 이름	= <b>SAMPLE</b>
데이터베이스 드라이브	= <b>C:\DB2</b>
데이터베이스 릴리스 레벨	= <b>9.00</b>
주석	=
디렉토리 항목 유형	= 간접
카탈로그 노드 번호	= <b>0</b>

## Local Database directory

**C:\>db2 list db directory on C:**

C:상의 지역 데이터베이스 디렉토리

디렉토리 내 항목의 수 = 4

데이터베이스 1 항목:

데이터베이스 별명	= <b>MUSICDB1</b>
데이터베이스 이름	= <b>MUSICDB1</b>
데이터베이스 디렉토리	= <b>SQL00002</b>
데이터베이스 릴리스 레벨	= <b>9.00</b>
주석	=
디렉토리 항목 유형	= <b>홀</b>
카탈로그 노드 번호	= <b>0</b>
노드 번호	= <b>0</b>

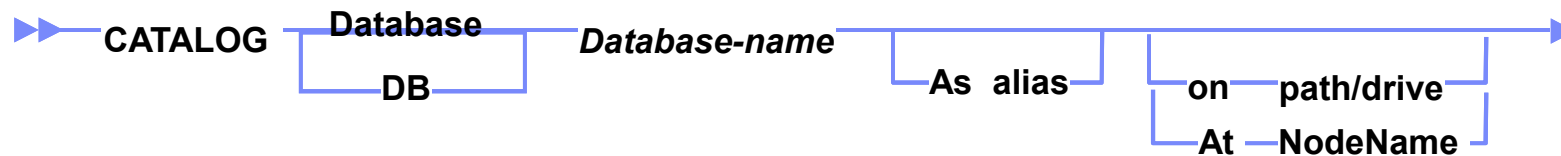
## 카탈로그 정의

- 데이터베이스가 생성될 때, **System Database directory file**에 자동적으로 카탈로그된다.
- **CATALOG DATABASE** 명령어를 이용하여 명시적으로 데이터베이스를 **System Database Directory file**에 카탈로그할 수 있다.
- **UNCATALOG DATABASE** 명령어를 이용하여 명시적으로 데이터베이스를 **System Database Directory file**에서 삭제할 수 있다.
- **Catalog Database** 명령어:
  - 다른 별명으로 데이터베이스를 카탈로그할 때.
  - 시스템 데이터베이스 디렉토리에서 전에 삭제되었던 내용을 다시 카탈로그하려 할 때.
  - 서버시스템에 있는 데이터베이스를 클라이언트 시스템에서 카탈로그하려 할 때.

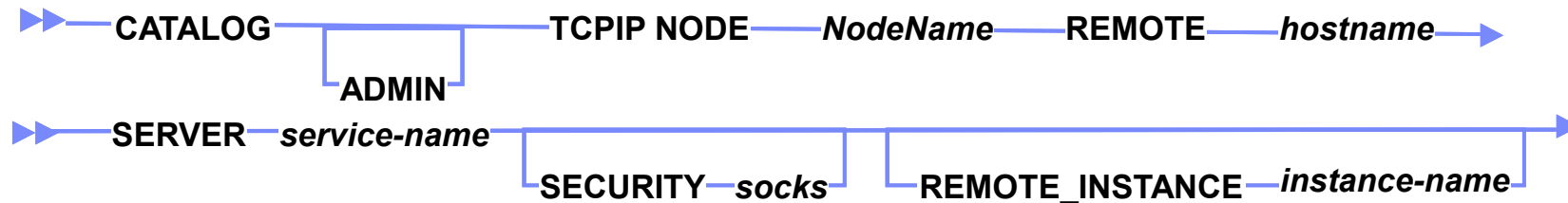
# Catalog/UnCatalog Command 구문

- 권한 부여 : sysadm or sysctrl
- 필수 연결 조건 : None

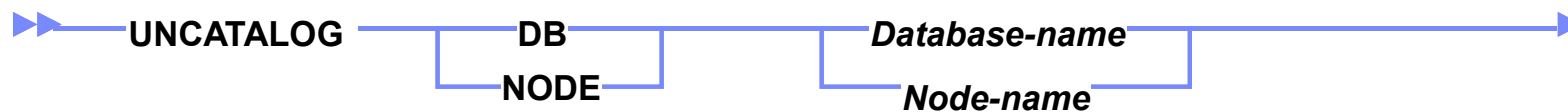
## Catalog Database



## Catalog Node



## UnCatalog DB/Node



## Directory Entries 변경

제어센타 혹은 **CLP** 창에서 **DB2 Directory entry**를 변경할 수 있다.

```
CATALOG DATABASE database-name [AS alias]
                                [ON drive | AT NODE nodename]
```

ex ) catalog db ourdb as testdb on C: (pc) or /home/instnn (unix)

```
UNCATALOG DATABASE <database-alias>
```

ex ) uncatalog db OURDB

### LIST DATABASE DIRECTORY

Database alias	= TESTDB
Database name	= OURDB
Local database directory	= C:\DB2 ( PC) or /home/instnn(UNIX)
Database release level	= 9.00
Comment	= Test Database
Directory entry type	= indirect

## C O N N E C T T O D a t a b a s e

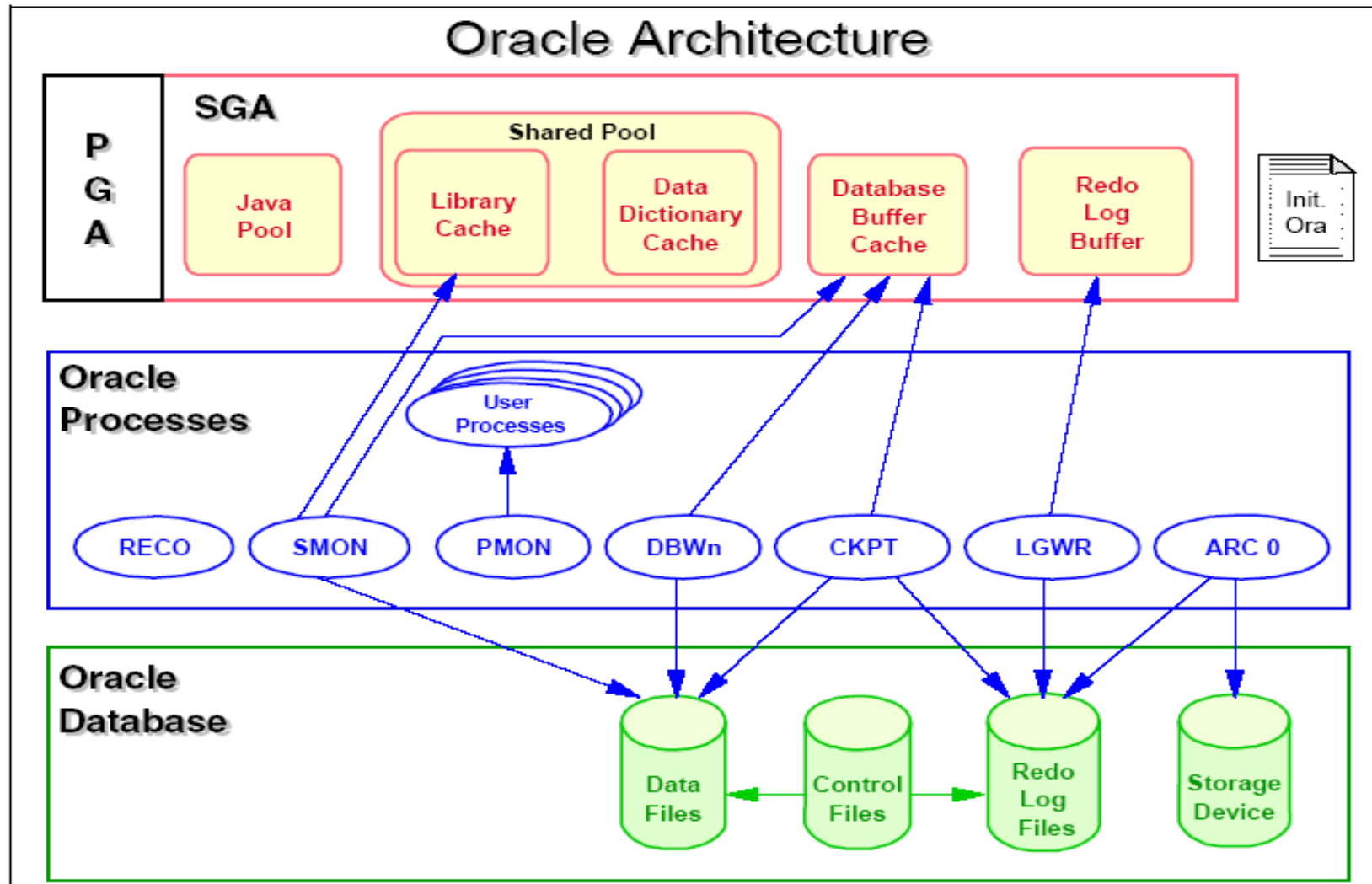
- **CONNECT TO database-alias-name [USER userid USING passwd [ NEW newpass CONFIRM newpass] ]**
- **DB2DBDFT** ( “*db2set -lr*”로 확인 )
  - **Registry variable**
- **CONNECT RESET**
  - **database connection** 해제
- **TERMINATE**(“*db2 terminate*”)
  - **database connection** 과 **CLP background process** 해제.
  - 환경 변수에 대한 변경을 적용할 경우
- 모든 **application** 종료 (“*db2 force application all*”)
  - 일반적으로 **terminate** 후 **application** 종료를 같이 함



# DB2 vs Oracle

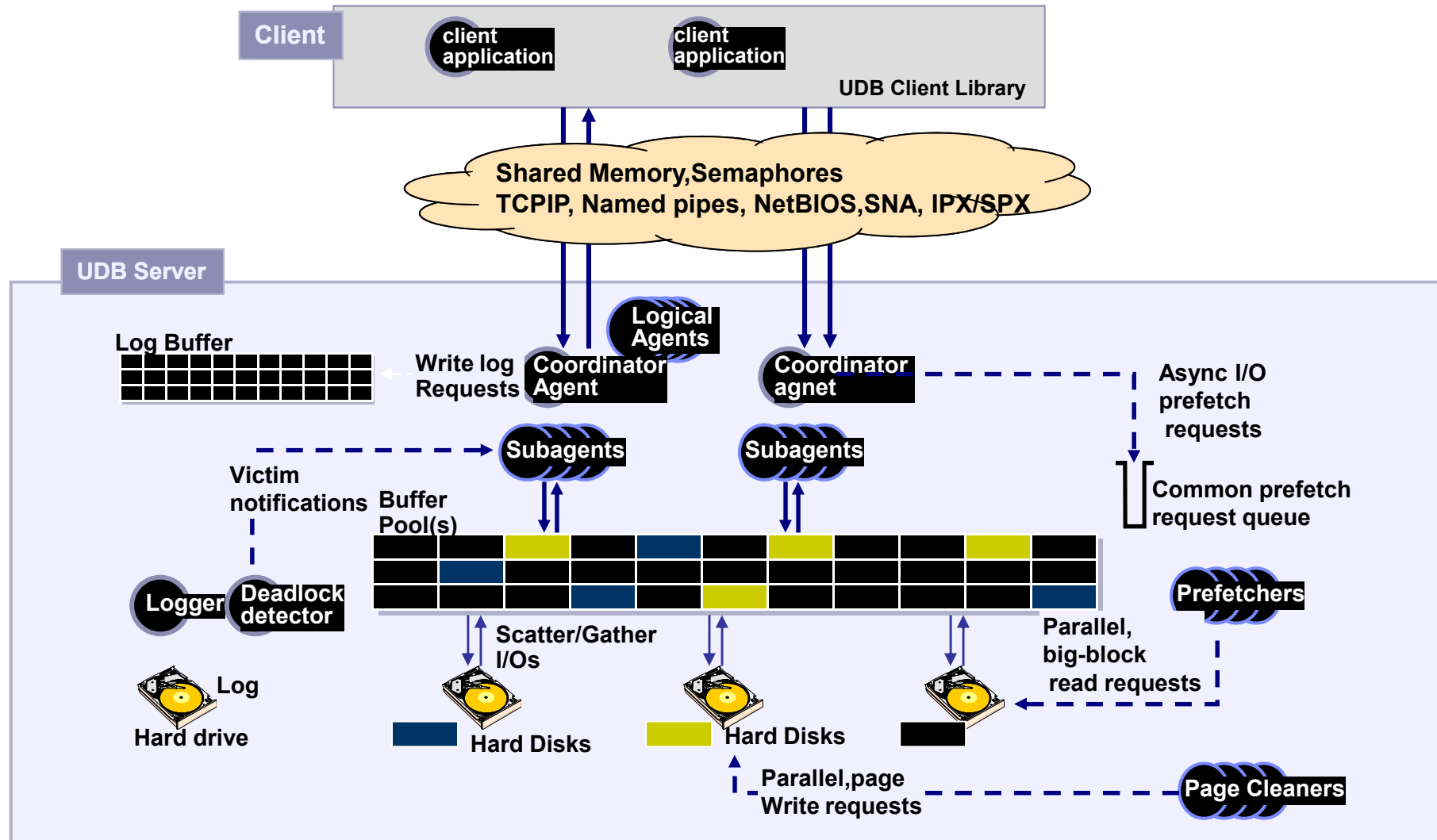
@business on demand software

# Oracle Architecture





# DB2 Architecture



## DB2 Instance / Database



Instance Memory  
monheap, fcm,...

Processes  
db2sysc, db2log.

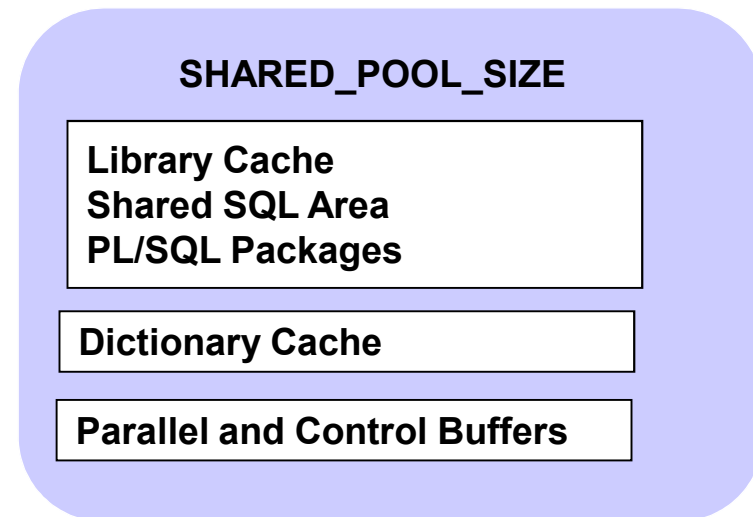
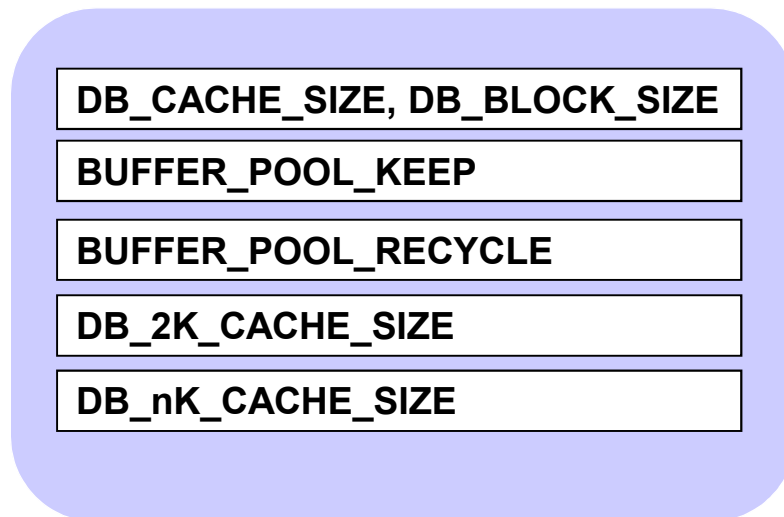
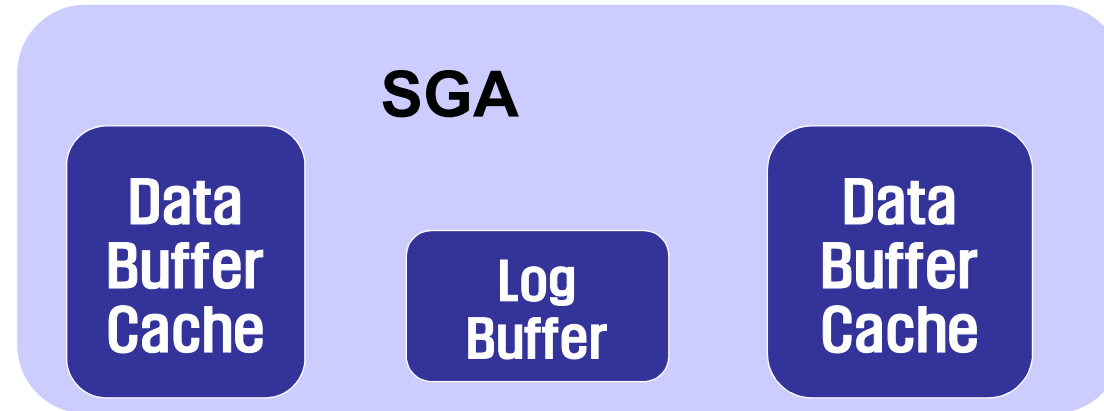
Instance Parms  
dbm cfg

Database Memory  
bufferpool, dbheap, utilheap

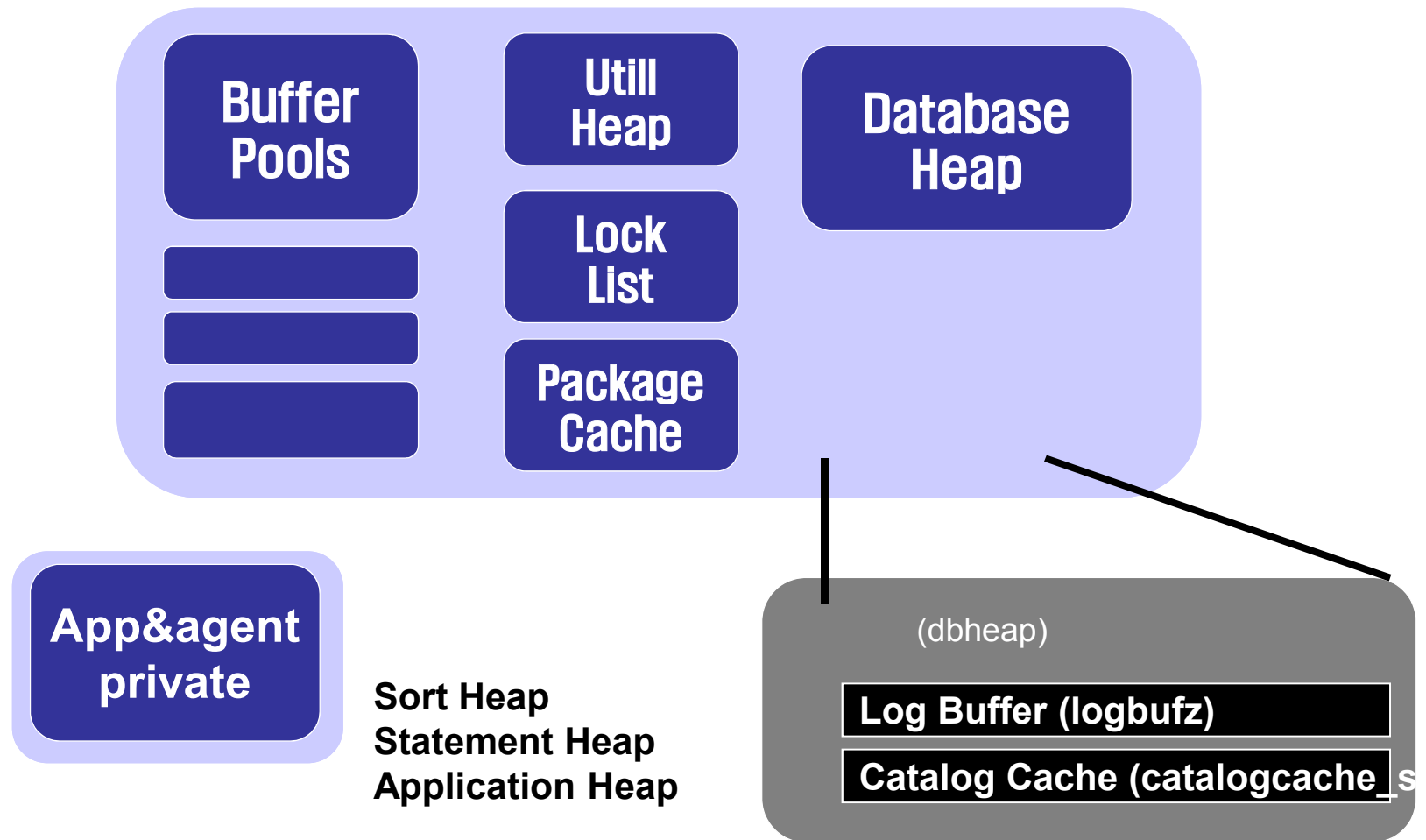
Datafiles, Logs

Database Parms  
db cfg

# Oracle Memory



# DB2 Memory



# Memory Summary

Oracle	DB2
<b>SGA</b>	<b>no equivalent</b>
<b>Buffer Cache</b> - one default, keep, recycle - one for each other page size - defined in init.ora - dynamically resizable	<b>Buffer Pool</b> - as many as you like - defined with CREATE BUFFERPOOL DDL - dynamically resizable
<b>Log Buffer</b> defined in init.ora	<b>Log Buffer</b> defined in database configuration
<b>Shared Pool</b>	<b>No exact equivalent</b>
<b>Library Cache</b>	<b>Package Cache</b>
<b>Dictionary Cache</b>	<b>Catalog Cache</b>
<b>PGA</b>	<b>Application Heap, Sort Heap and others</b>
<b>no equivalent</b>	<b>Utilities Heap</b>
<b>no equivalent</b>	<b>Lock List</b>

# Oracle Process

**PMON**

Process Monitor - performs cleanup if a process dies

**SMON**

Monitor - performs crash recovery

**DBWO**

Data Writers - write dirty pages to disk  
DB\_WRITER\_PROCESS - 1 to 10

**LGWR**

Log Writer - write redo log buffer to redo log file  
Writes on commit, every 3 seconds or when dirty pages are written to disk

**CKPT**

Checkpoint - updates block headers (SCN) when  
checkpoint occurs

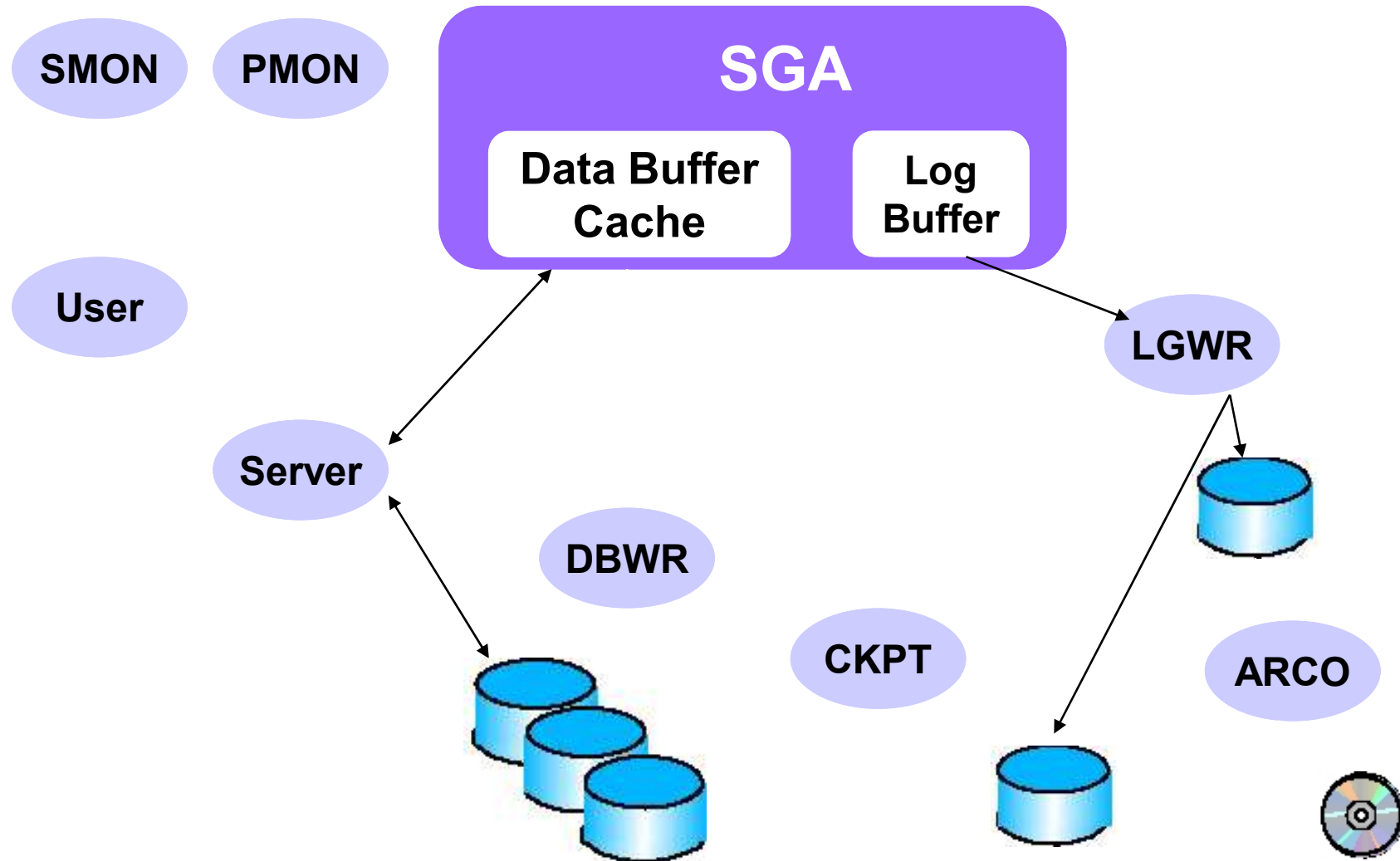
**Server**

Parse, Execute SQL, Read Data Blocks, Return Results

**ARCO**

Archiver - copies online redo log files to archive media  
LOG\_ARCHIVE\_MAX\_PROCESS - 1 to 10

# Oracle Process



## DB2 Process

**db2wdog**

Watchdog - performs cleanup if a process dies

**db2sysc**

System Controller - startup/shutdown of instance

**db2pclnr**

Page Cleaner - write dirty pages to disk

**db2pfchr**

Prefetcher - reads data from disk into bp async

**db2loggr**

Log Writer - writes log buffer to log file

Writes on commit, when log buffer is full, before data pages write to disk

**db2dlock**

Deadlock Detector - finds deadlocks and resolves

**db2agent**

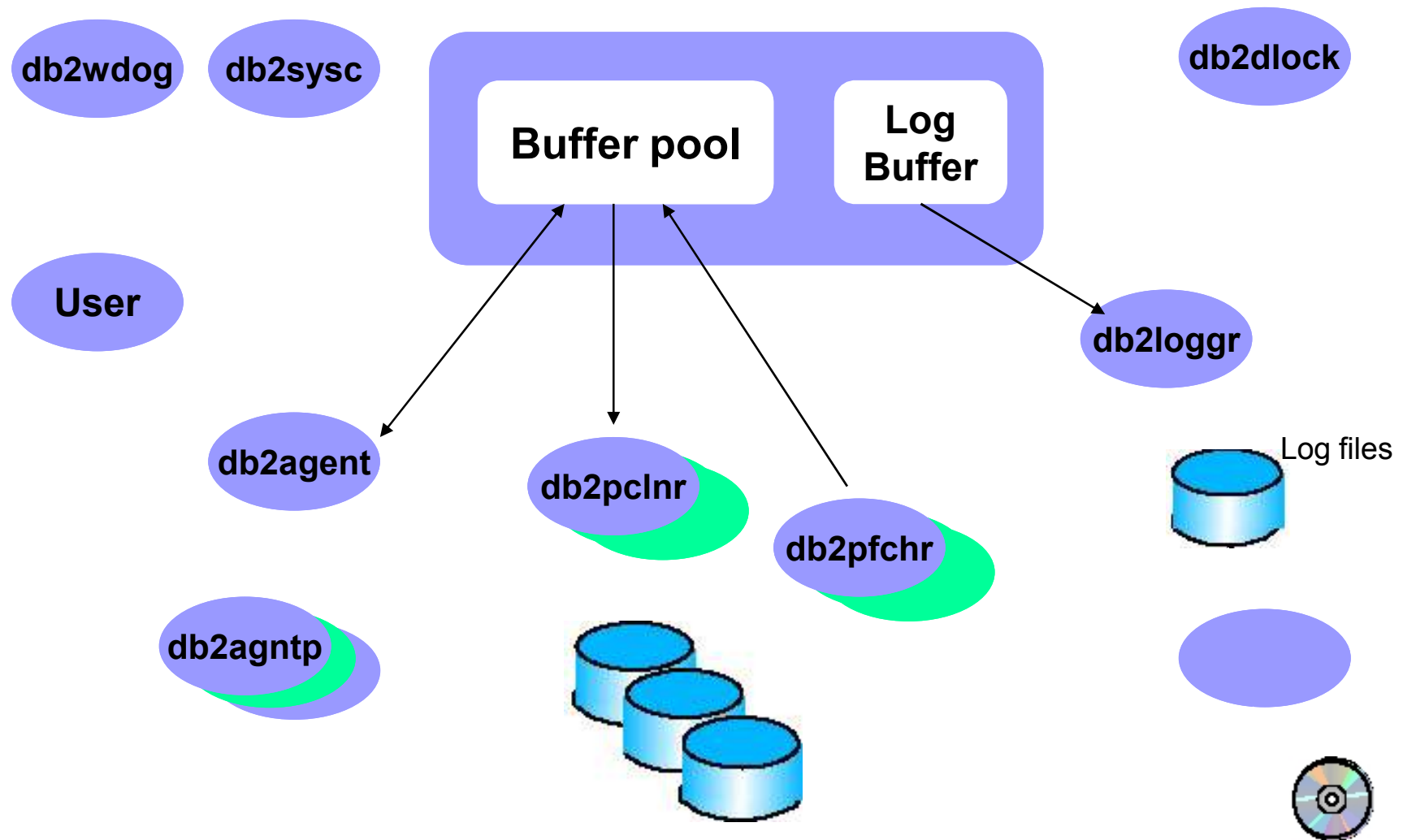
Coordinator Agent - Compile &amp; Execute SQL and return result

**db2agntp**

Subagent - works on a portion of the access plan



# DB2 Process



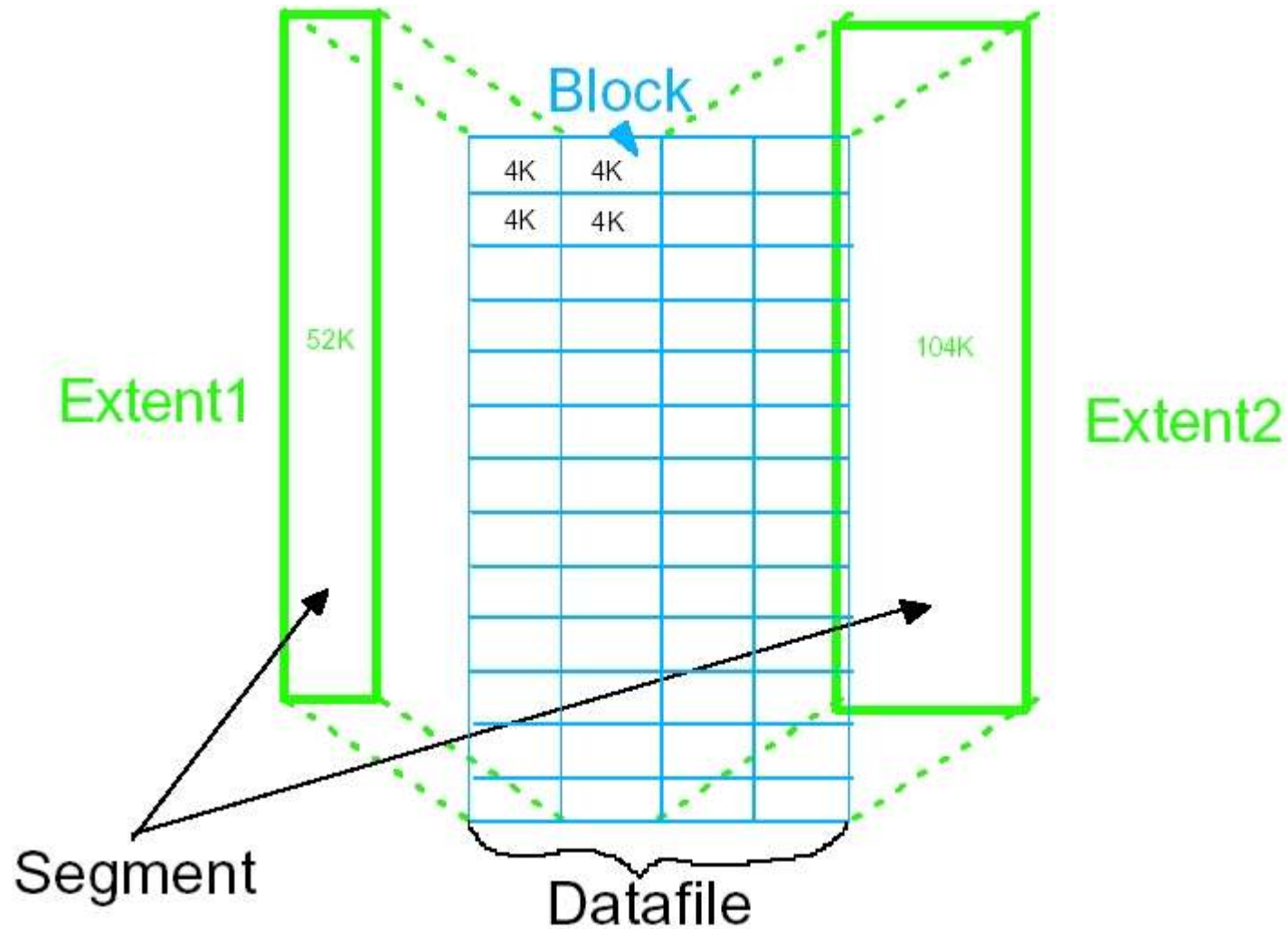
## Process Summary

Oracle	DB2
PMON	db2wdog
SMON	no equivalent
Server	db2agent
DBWx	db2pclnr
CKPT	no equivalent
LGWR	db2loggr
ARCx	no equivalent
no equivalent	db2agntp
no equivalent	db2pfchr

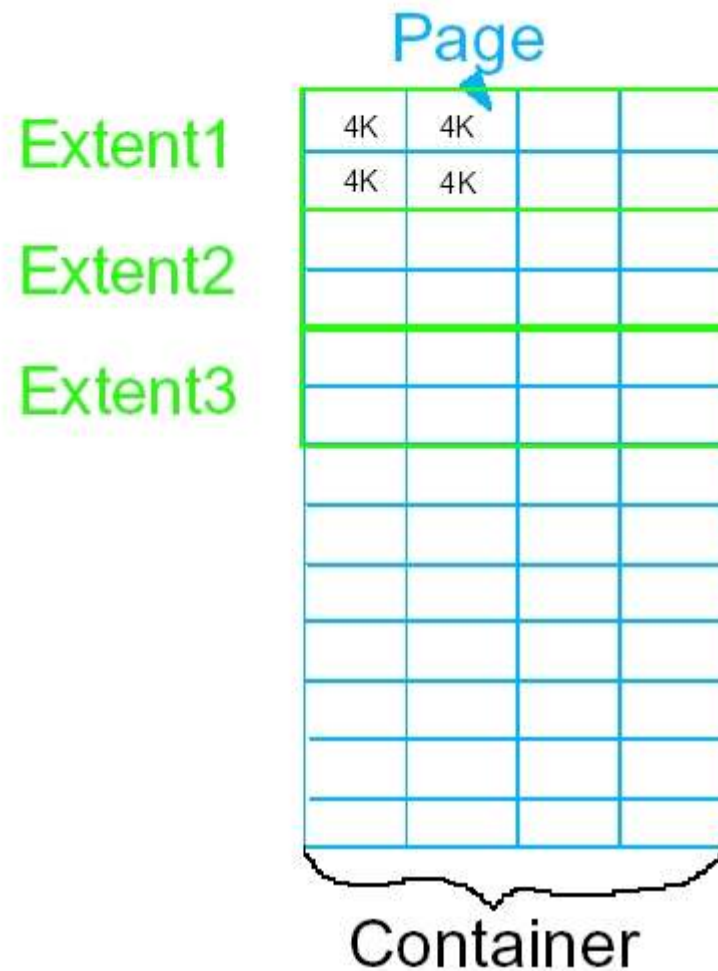
# Tablespace

Oracle	DB2
<p>Datafile의 집합이다.</p> <ul style="list-style-type: none"><li>▪ Locally managed Tablespace<ul style="list-style-type: none"><li>▪ Free space가 테이블공간안에서 관리된다.</li></ul></li><li>▪ Data dictionary managed tablespace<ul style="list-style-type: none"><li>▪ Free space가 data dictionary 에서 관리된다.</li></ul></li></ul>	<p>Container의 집합이다.</p> <ul style="list-style-type: none"><li>▪ SMS<ul style="list-style-type: none"><li>▪ Filesystem의 directory file로 구성되어진다.</li></ul></li><li>▪ DMS<ul style="list-style-type: none"><li>▪ 오라클의 datafile과 유사하며 생성시에 파일사이즈가 지정된다.</li><li>▪ Container로 file과 raw device가 이용되어진다.</li></ul></li></ul>

## 오라클 Data Structure

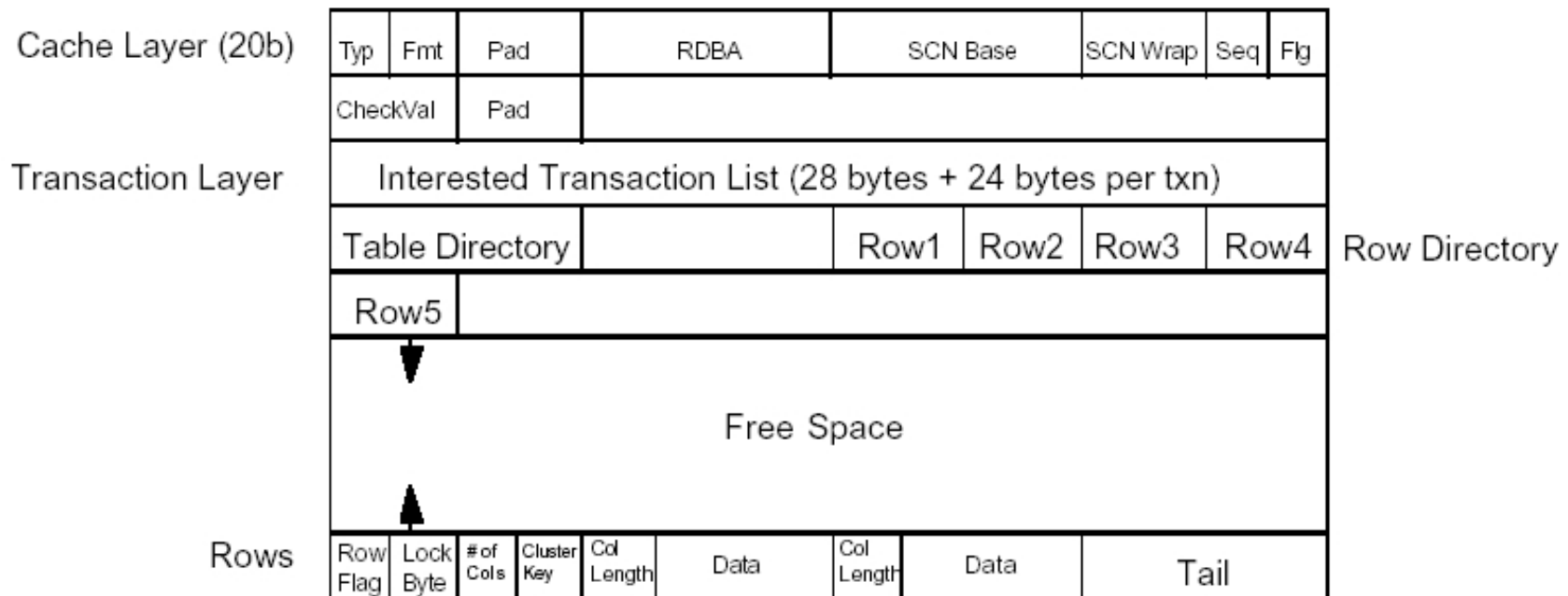


# DB2 Data Structure



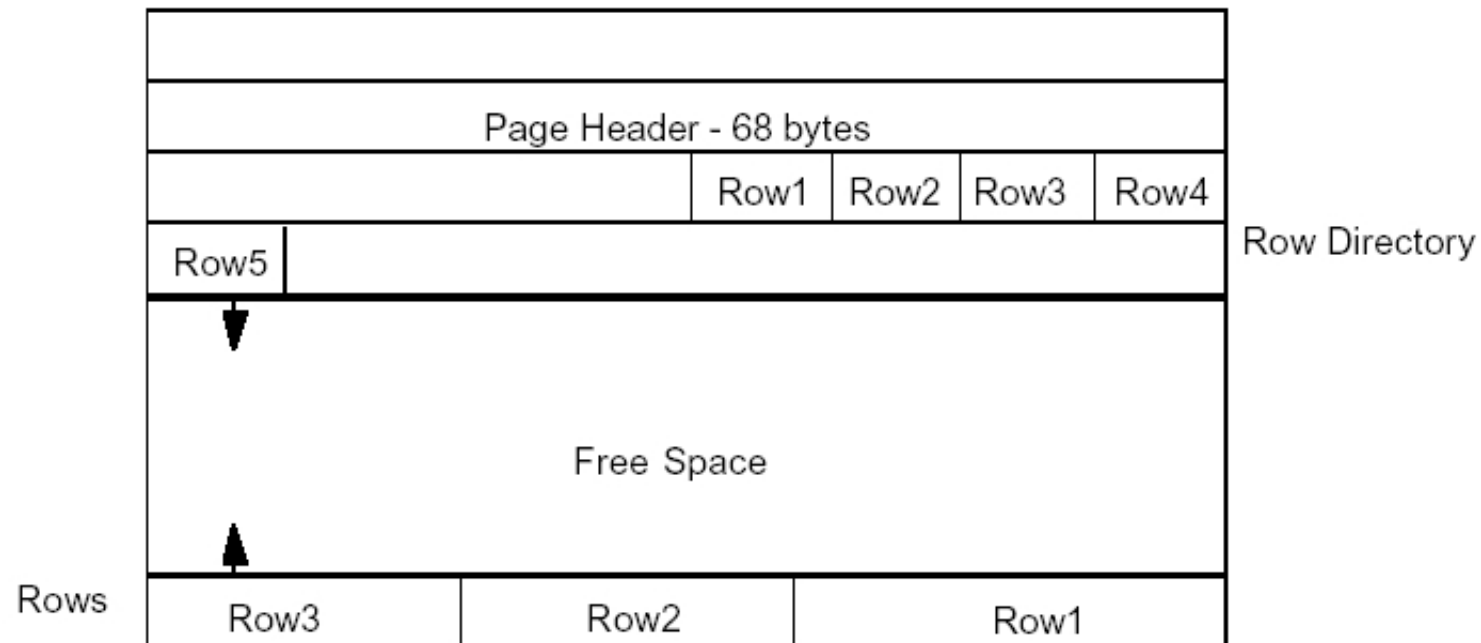
**Object (table/index)  
made up of 1 or more  
extents**

# Oracle Data Block Structure



Oracle과 DB2 page 구조의 가장 큰 차이점은 Lock 정보가 freelist에 저장되어 오라클은 page integrity를 보장하기 위해 처음과 마지막 부분만을 체크한다.

# DB2 Data Page Structure



# Oracle Vs DB2 Tablespace 비교

Oracle	DB2 V
Tablespace - <b>data dictionary managed</b> - <b>locally managed</b>	Tablespace (regular, long) - <b>system managed</b> - <b>database managed</b>
Segment (table, index, rollback)	Object (table, index )
Extent - <b>variable size within tablespace</b> (PCTINCREASE로 정의)	Extent - <b>fixed size within tablespace</b>
Block (2k, 4k, 8k, 16k, 32k on some platforms)	Page (4k, 8k, 16k, 32k)
Datafile	Container
Oracle Flexible Architecture <b>.DBF - data or index file</b> <b>.LOG - redo log file</b> <b>.CTL - control file</b>	DB2 SMS file names <b>.DAT - data file</b> <b>.INX - index file</b> <b>.LB - LOB file (.LBA is the lob metadata)</b> <b>.LF - long field file</b> <b>.TMP - temp table</b>



## Oracle Vs DB2 용어 비교

Oracle	DB2 UDB
Instance	Instance
Database	Database
Initialization File	Database Manager Configuration File
Table spaces	Table spaces
Data Block	pages
Extents	Extents
Data Files	DMS Containers
Redo Log Files	Transaction Log Files
PL/SQL	SQL/PL
Data Buffers	Buffer pool
SGA	Database Manager and Database shared memory
Data Dictionary	Catalog
Library Cache	Package Cache
Large Pool	Utility Heap
Data Dictionary Cache	Catalog cache
SYSTEM tablespace	SYSCATSPACE tablespace

# Command Comparison

DDL or Command	Oracle	DB2
Create Database	builds system and temp tablespace	builds system, temp and user default tablespace
Create Table	max 256 columns then row chaining is used - unrecoverable	500 columns for 4k pages 1012 columns for larger pages. - not logged initially
Alter...	alter table drop column alter table set unused	no equivalent no equivalent
Start instance	startup pfile = ...	db2start
Stop instance	shutdown [abort ...]	db2stop [force]
Collect Statistics	Analyze Table ...	Runstats on table ....
Reorganize data	export/import most common	Reorg table ...
Load...	sqlldr	Load from ... insert into ...
Import/Export	support only Oracle format	positional, delimited or DB2 format (IXF)
Drop Database	leaves files unless you specify including contents. Can't delete files for partitioned tables	Drop tables ... cleans up data files
Backup / Restore	use RMAN utility	DB2 Backup Database dbname to ...

# Data Type Comparison

DB2	Oracle
CHAR(x) x < 256	CHAR(x) x < 2000
VARCHAR(x) x < 4001	VARCHAR2(x) x < 4001
LONG VARCHAR	LONG (character up to 2gig)
FOR BIT DATA	RAW
BLOB(x [K M G])	BLOB
CLOB(x [K M G])	CLOB
DBCLOB(x [K M G])	NCLOB
GRAPHIC(x)	NCHAR(x)
VARGRAPHIC(x)	NVARCHAR2(x) x < 4001
LONG VARGRAPHIC	
DATE	DATE (1/1/4712BC - 12/31/9999)
TIME	no equivalent
TIMESTAMP	TIMESTAMP(s) s = # digits in fractional sec.
SMALLINT (2byte integer)	NUMBER(5)
INTEGER (4byte integer)	NUMBER(10)
BIGINT (8byte integer)	NUMBER(19)
FLOAT (REAL 4byte   DOUBLE 8byte)	FLOAT (precision 38)
DECIMAL(p,s)	NUMBER(p,s) p<39, -85<s<128
no equivalent	ROWID
no equivalent	INTERVAL YEAR or INTERVAL DAY

## C r e a t e P r o c e d u r e

**ORACLE**

```
CREATE OR REPLACE PROCEDURE process_withdrawal ( Account_Id    VARCHAR2
                                                    , Cheque_No    VARCHAR2
                                                    , Amount        NUMBER )
IS
...
```

**DB2**

```
CREATE PROCEDURE process_withdrawal ( IN Account_Id    VARCHAR(10)
                                       , IN Cheque_No    VARCHAR(10)
                                       , IN Amount        DECIMAL(10,2)      )
LANGUAGE SQL
...
```

- DB2 는 REPLACE 를 지원하지 않는다.
- DB2는 Data Type에 Length 를 꼭 지정해야 한다. 그러므로 Column 를 Check 해야 한다.
- Migration Toolkit 이용시 오라클에서 %TYPE 사용시 자동으로 Conversion 한다.

# Example

## ORACLE

```
CREATE TRIGGER emp_history_trg
  AFTER DELETE ON employees
  FOR EACH ROW
BEGIN
  INSERT INTO emp_history(emp_id, first_name, last_name )
  VALUES (:old.emp_id, :old.first_name, old.last_name);
END;
```

## DB2

```
CREATE TRIGGER emp_history_trg
  AFTER DELETE ON EMPLOYEES
  REFERENCING OLD AS d
  FOR EACH ROW
  MODE DB2SQL
BEGIN ATOMIC
  INSERT INTO emp_history ( emp_id, first_name, last_name )
  VALUES ( d.emp_id, d.first_name, d.last_name);
END!
```

## C R E A T E F U N C T I O N

- DB2는 다음과 같은 Function 을 지원한다.
  - SQL scalar, table 또는 row function
  - Sourced 또는 template function
  - OLE DB function
  - EXternal Table function.
  - External Scalar function

```
CREATE FUNCTION function_name ( parameters )  
RETURNS return_type  
LANGUAGE SQL  
READS SQL DATA  
RETURN statement
```

## 변 수 선언

Oracle PL/SQL에서의 변수 선언 영역	DB2 SQLPL
<ul style="list-style-type: none"> <li>Stored Procedure / Function 의 parameter list 에서 선언.</li> <li>Stored procedure / Function / Trigger 의 Body 에서 선언.</li> <li>Package 선언부에서 선언.</li> <li>Package Body 선언부에서 선언.</li> </ul>	<ul style="list-style-type: none"> <li>Stored Procedure / Function 의 Parameter list 에서 선언.</li> <li>Stored Procedure / Function / Trigger의 Body 에서 선언.</li> </ul> <p><b>* DB2는 Stored Procedure / Function 을 그룹화하는 Package 개념이 없고 Schema 로 Stored Procedure/ Function을 그룹화할 수 있다.</b></p>
<pre>I_balance NUMBER(10,2) := 0.0; I_balance := 19.99;</pre>	<pre>DECLARE I_balance NUMERIC(10,2) DEFAULT 0.0; SET I_balance = 19.99;</pre> <p><b>SET 구문은 또한 다음과 같이 local 변수에 Table Column 값을 할당할 수 있다.</b></p> <pre>SET I_balance = (SELECT balance                   FROM account_info                   WHERE account_no = actNo);</pre>

# Condition 구문 및 Flow control

Oracle PL/SQL	DB2 SQLPL
IF - THEN - END IF ; IF - THEN - ELSE - END IF ; IF - THEN - ELSIF - END IF ;	IF - THEN - END IF ; IF - THEN - ELSE - END IF ; IF - THEN - ELSEIF - END IF ;
LOOP statements ; END LOOP;	[L1:] LOOP statements; LEAVE L1; END LOOP [L1];
WHILE condition LOOP statements; END LOOP;	WHILE condition DO statements; END WHILE ;
OPEN cursor_variable FOR select_statements;	FOR variable AS cursor_name CURSOR FOR select_statement DO statements; END FOR ;



## C o n d i t i o n 구문 및 Flow control

Oracle PL/SQL	DB2 SQLPL
<pre>FOR I_count IN lower_bound ..upper_bound LOOP   statements; END LOOP;</pre>	<p>정확히 Mapping 되는 구문은 없으나 MTK 는 다음과 같이 Convert 한다.</p> <pre>SET I_count = lower_bound; WHILE I_count &lt;= upper_bound DO Statements; SET I_count = I_count + 1; END WHILE ;</pre>

## Example

Oracle PL/SQL	DB2 SQLPL
<pre>IF v_TotalStudents = 0 THEN   INSERT INTO temp_table ( char_col )   VALUES ('There are no students registered. '); ELSIF v_TotalStudents &lt; 5 THEN   INSERT INTO temp_table ( char_col )   VALUES ( 'There are only a few students' ); ELSIF v_TotalStudents &lt; 10 THEN   INSERT INTO temp_table (char_col )   VALUES ('There are a little more students'); ELSE   INSERT INTO temp_table ( char_col )   VALUES ('There are many students '); END IF ;</pre>	<pre>IF v_TotalStudents = 0 THEN   INSERT INTO temp_table ( char_col )   VALUES ('There are no students registered. '); ELSEIF v_TotalStudents &lt; 5 THEN   INSERT INTO temp_table ( char_col )   VALUES ( 'There are only a few students' ); ELSEIF v_TotalStudents &lt; 10 THEN   INSERT INTO temp_table (char_col )   VALUES ('There are a little more students'); ELSE   INSERT INTO temp_table ( char_col )   VALUES ('There are many students '); END IF ;</pre>

## E x a m p l e ( C o n t i n u e )

Oracle PL/SQL	DB2 SQLPL
<pre>IF v_TotalStudents = 0 THEN   INSERT INTO temp_table ( char_col )   VALUES ('There are no students registered. '); ELSIF v_TotalStudents &lt; 5 THEN   INSERT INTO temp_table ( char_col )   VALUES ( 'There are only a few students' ); ELSIF v_TotalStudents &lt; 10 THEN   INSERT INTO temp_table ( char_col )   VALUES ('There are a little more students' ); ELSE   INSERT INTO temp_table ( char_col )   VALUES ('There are many students '); END IF ;</pre>	<pre>CASE   WHEN v_TotalStudents = 0 THEN     INSERT INTO temp_table ( char_col )     VALUES ('There are no students registered. ');   WHEN v_TotalStudents &lt; 5 THEN     INSERT INTO temp_table ( char_col )     VALUES ( 'There are only a few students' );   WHEN v_TotalStudents &lt; 10 THEN     INSERT INTO temp_table ( char_col )     VALUES ('There are a little more students' );   OTHER     INSERT INTO temp_table ( char_col )     VALUES ('There are many students '); END ;</pre>

## D y n a m i c S Q L – Dynamic Update 구문

### DB2 SQLPL

```
CREATE PROCEDURE update_emp_office_v1 ( IN      v_emp_id      FLOAT
                                         ,IN      v_office_id    FLOAT
                                         ,OUT     v_num_changes  INTEGER )

LANGUAGE SQL
BEGIN
    DECLARE  v_dyn_sql      VARCHAR(1000);

    SET  v_dyn_sql = 'UPDATE employees' || 'SET office_id = ' || CHAR( v_office_id ) ||
        'WHERE emp_id = ' || CHAR ( v_emp_id );

    EXECUTE IMMEDIATE v_dyn_sql;

    GET  DIAGNOSTICS v_num_changes = row_count;
END!
```

- GET DIAGNOSTICS 구문은 INSERT, UPDATE , DELETE 후에 변경된 row 의 수를 넘겨준다.
- EXECUTE IMMEDIATE 구문은 한번만 수행되어질때 이용되어지며 만약 반복적으로 SQL 문이 수행되어 질 경우에는 PREPARE 와 EXECUTE 구문을 사용해야 한다.
- EXECUTE 구문에는 SELECT 또는 VALUES 구문을 사용할 수 없다.

## Dynamic SQL – Dynamic Update 구문

### DB2 SQLPL – PREPARE 와 EXECUTE 구문 사용시

```
CREATE PROCEDURE update_emp_office_v1 ( IN      v_emp_id      FLOAT
                                         ,IN      v_office_id   FLOAT
                                         ,OUT     v_num_changes INTEGER )

LANGUAGE SQL
BEGIN
    DECLARE  v_dyn_sql      VARCHAR(1000);

    SET  v_dyn_sql = 'UPDATE employees' || 'SET office_id = ' || CHAR( v_office_id ) ||
        ' WHERE emp_id = ' || CHAR ( v_emp_id );

    PREPARE  v_stmt FROM  v_dyn_sql;

    EXECUTE v_stmt USING v_office_id, v_emp_id;

    GET  DIAGNOSTICS v_num_changes = row_count;
END!
```

## 자 바 로 구현된 UDF

## Java User Defined Function with Dynamic SQL - get\_max\_band

```
Import COM.ibm.db2.app.*;
Import java.sql.*;

Public class UDFemp extends UDF
{
    public void maxBand ( int inOfficeID, String outBand ) throws Exception
    {
        try
        {
            Connection con = DriverManager.getConnection ("jdbc:default:connection");
            String query = "SELECT max(band) FROM employees WHERE office_id = ?";

            PreparedStatement stmt = con.prepareStatement(query);
            stmt.setInt(1, inOfficeID);
            ResultSet rs = stmt.executeQuery();

            while(rs.next() ) { outBand = rs.getString(1); }
            set(2, outBand);

            rs.close();
            stmt.close();
            con.close();
        }
    }
}
```

## 자바로 구현된 UDF ( Continue )

### Java User Defined Function with Dynamic SQL - get\_max\_band

```
        catch ( SQLException  sqle )
        {
            setSQLstate("38999");
            setSQLmessage("SQLCODE = " + sqle.getSQLState() );
            return ;
        }
    }
}
```

```
CREATE FUNCTION get_max_band ( INTEGER )
RETURNS CHAR
EXTERNAM NAME 'UDFemp!maxBand'
FENCED
CALLED ON NULL INPUT
VARIANT
READS SQL DATA
PARAMETER STYPE DB2GENERAL
LANGUAGE JAVA
NO EXTERNAL ACTION!
```

## C u r s o r c o n v e r s i o n

Operation	Oracle	DB2 UDB
Declaring a cursor	CURSOR cursor_name [ ( cursor_parameter(s) ) ] IS select_statement	DECLARE cursor_name CURSOR [WITH HOLD] [ WITH RETURN] [T OCALLER   TO CLIENT ] FOR Select_statement
Opening a cursor	OPEN cursor_name [(cursor_parameter(s))]	OPEN cursor_name [USING host- variable]
Fetching from cursor	FETCH cursor_name INTO variable(s)	FETCH [from] cursor_name INTO variable(s)
Update fetched row	UPDATE table_name SET statement(s)... WHERE CURRENT OF Cursor_name	UPDATE table_name SET statements... WHERE CURRENT OF Cursor_name



## C u r s o r c o n v e r s i o n ( C o n t i n u e )

Operation	Oracle	DB2 UDB
Deleted fetched row	DELETE FROM table_name WHERE CURRENT OF Cursor_name;	DELETE FROM table_name WHERE CURRENT OF cursor_name
Closing cursor	CLOSE cursor_name;	CLOSE cursor_name

## Converting Cursor attributes

Oracle	DB2 UDB
%ISOPEN	Cursor를 Open 할 경우 만약 Cursor가 이미 Open 되어져 있다면 DB2는 SQLCODE로 -502 ( or SQLSTATE = 24501) 를 Return Fetch할 경우 Cursor가 Open 되어 있지 않으면 SQLCODE는 -501 ( or SQLSTATE 24502) 이 Return 된다.
%NOTFOUND	If ( SQLCODE == 100 ) or if SQLSTATE = '02000'
%FOUND	If ( SQLCODE == 0 ) or if SQLSTATE = '00000'
%ROWCOUNT	Counter 변수를 이용하여 cursor 로부터 fetch 된 row 수를 retrieve 한다.

## C o n v e r t i n g   C u r s o r   A t t r i b u t e s

	Oracle PL/SQL	DB2 SQLPL
%ISOPEN	<pre>IF c1%ISOPEN THEN   fetch c1 into var1; ELSE   OPEN c1;   fetch c1 into var1; END IF;</pre>	<pre>DECLARE cursor_notopen CONDITION FOR   SQLSTATE 24501; DECLARE CONTINUE HANDLER FOR   cursor_notopen BEGIN   open c1;   FETCH c1 into var1; END;  ..... FETCH c1 into var1;</pre>

## C o n v e r t i n g   C u r s o r   A t t r i b u t e s

	Oracle PL/SQL	DB2 SQLPL
%NOTFOUND	<pre>OPEN cur1; LOOP   FETCH cur1 INTO v_var1;   EXIT WHEN     cur1%NOTFOUND;  ....  END LOOP;</pre>	<pre>DECLARE  SQLCODE into DEFAULT 0; .... OPEN c1; L1: LOOP   FETCH c1 INTO v_var1;   IF SQLCODE = 100 THEN     LEAVE L1;   END IF ; .... END LOOP L1;</pre> <p>* CONDITION HANDLERS 를 이용하여 구현할수도 있다.</p>

## Converting Cursor Attributes

\* %ROWCOUNT 가 처음 10개 Row만 처리하는 Code 일 경우

	Oracle PL/SQL	DB2 SQLPL
%ROWCOUNT	<pre> LOOP   FETCH c1 INTO my_ename,                 my_deptno;   IF c1%ROWCOUNT &gt; 10 THEN     EXIT;   END IF ;   ... END LOOP; </pre>	<pre> DECLARE c1 CURSOR FOR SELECT ename, deptno FROM emp_table   FETCH FIRST 10 ROWS ONLY;  DECLARE CONTINUE HANDLER FOR NOT FOUND BEGIN   SET end-of-fetch = 1; END; L1 : LOOP   FETCH c1 INTO my_ename, my_deptno;   IF end-of-fetch = 1 THEN     LEAVE L1;   END IF;   ..... END LOOP L1; </pre>

## Converting Cursor Attributes

- %ROWCOUNT 가 Cursor로부터 처리되어진 Rows 수를 계산할때

	Oracle PL/SQL	DB2 SQLPL
%ROWCOUNT	<pre> LOOP   FETCH c1 INTO my_ename,                 my_deptno;   IF c1%ROWCOUNT &gt; 10 THEN     ...   END IF ;   ... END LOOP; </pre>	<pre> DECLARE v_CURCOUNT INT DEFAULT 0; ..... L1 : LOOP   FETCH c1 INTO my_ename, my_deptno;   v_CURCOUNT = v_CURCOUNT + 1;   IF v_CURCOUNT &gt; 10 THEN     .....   END IF ;   ..... END LOOP L1; </pre>

## C o n v e r t i n g   C u r s o r   A t t r i b u t e s

\* %ROWCOUNT 가 10개의 row를 삭제할 때 취해야 할 Action이 있을 경우

	Oracle PL/SQL	DB2 SQLPL
%ROWCOUNT	<pre>DELETE FROM emp_table WHERE ....  IF SQL%ROWCOUNT &gt; 10 THEN .... END IF;</pre>	<pre>DECLARE rc INT DEFAULT 0; ....  DELETE FROM emp_table WHERE ... GET DIAGNOSTICS rc = ROW_COUNT;  IF rc &gt; 10 THEN ... END IF;</pre>

## Converting Cursor Attributes

	Oracle PL/SQL	DB2 SQLPL
%FOUND	<pre>DELETE FROM emp_table WHERE empno = my_empno;  IF SQL%FOUND THEN   INSERT INTO emp_table     VALUES ( my_empno,               my_ename); .... END IF;</pre>	<pre>DELETE FROM emp WHERE empno =   my_empno;  IF SQLCODE = 0 THEN   INSERT INTO emp_table VALUES     (my_empno, my_ename);</pre>



## Oracle Data Dictionary vs DB2 Catalog

Oracle	DB2
DBA_TABLES	SYSCAT.TABLES
DBA_TAB_COLUMNS	SYSCAT.COLUMNS
DBA_TABLESPACES	SYSCAT.TABLESPACES
DBA_INDEXES	SYSCAT.INDEXES
DBA_TAB_PRIVS	SYSCAT.TABAUTH
DBA_TRIGGERS	SYSCAT.TRIGGERS
DBA_VIEWS	SYSCAT.VIEWS
DBA_SEQUENCES	SYSCAT.SEQUENCES
DBA_PROCEDURES	SYSCAT.ROUTINES

## Data Type

Oracle	DB2	Notes
CHAR(n)	CHAR(n)	1 <= n <= 254
VARCHAR2(n)	VARCHAR(n)	N <= 32762
LONG	LONG VARCHAR(n)	If n <= 32700 bytes
LONG	CLOB ( 2GB)	If n < 2GB
NUMBER(p)	SMALLINT / INTEGER/ BIGINT	-SMALLINT if 1 <= p <= 4 -INTEGER if 5 <= p <= 9 -BIGINT if 10 <= p <= 18
NUMBER (p, s )	DECIMAL ( p, s )	If s > 0
NUMBER	FLOAT / REAL / DOUBLE	
RAW ( n)	CHAR(n) FOR BIT DATA / VARCHAR(n) FOR BIT DATA BLOB(n)	-CHAR, if n <= 254 -VARCHAR, if 254 < n <= 32672 -BLOB if 32672 < n <= 2GB
LONG RAW	LONG VARCHAR(n) FOR BIT DATA / BLOB(n)	-LONG, if n <= 32700 -BLOB if 32700 < n <= 2GB

## D a t a   T y p e

Oracle	DB2	Notes
BLOB	BLOB(n)	If n <= 2GB
CLOB	CLOB(n)	If n <= 2GB
NCLOB	DBCLOB(n)	If n <= 2GB use DBCLOB(n/2)
DATE	TIMESTAMP	
DATE ( 연월일만 )	DATE (MM/DD/YYYY)	
DATE ( time 만 )	TIME ( HH24:MI:SS )	



# DB2 VS Oracle Function comparison



## CONNECTBY ~ 1

Oracle	DB2
<pre>select level rank, soid, soname, aliasname, uppersoid, sorank from if051 where worktype = 'W' connect by prior soid = uppersoid start with soid = '2900000'</pre>	<pre>with temptab(level,soid,soname,aliasname,uppersoid,sorank,worktype ) AS (     SELECT 1,root.soid, root.soname,            root.aliasname,root.uppersoid,            root.sorank ,root.worktype     FROM IF051 root     where soid = '2900000' UNION ALL     SELECT super.level+1,sub.soid,            sub.soname,            sub.aliasname,sub.uppersoid,            sub.sorank ,sub.worktype     FROM IF051 sub , temptab super     WHERE sub.uppersoid = super.soid ) SELECT * FROM temptab where worktype = 'W' ;</pre>

## C O N N E C T B Y ~ 2

Oracle	DB2
<pre> SELECT soid, soname FROM if051 WHERE disuseyn = 'N' AND worktype = 'W' CONNECT BY prior soid = uppersoid START WITH soid = ( SELECT soid                                 FROM                                 if051                                 WHERE soformtypecd                                 = '1102'                                 AND rownum = 1 CONNECT BY soid = prior uppersoid START WITH soid = '2902103'; </pre>	<pre> create view if051view as with temptab(soid,uppersoid, soformtypecd) AS ( SELECT root.soid, root.uppersoid, root.soformtypecd FROM IF051 root where soid = '2902103' UNION ALL SELECT sub.soid,sub.uppersoid, sub.soformtypecd FROM IF051 sub, temptab super WHERE sub.soid = super.uppersoid ) select soid from ( SELECT row_number() over () AS rn, soid FROM temptab where soformtypecd = '1102' ) AS temp  WHERE rn = 1;  with temptab(soid,soname,disuseyn,worktype ) AS ( SELECT root.soid, root.soname ,root.disuseyn,root.worktype FROM IF051 root where soid = ( SELECT soid from if051view ) UNION ALL SELECT sub.soid, sub.soname,sub.disuseyn,sub.worktype FROM IF051 sub , temptab super WHERE sub.uppersoid = super.soid ) SELECT * FROM temptab where disuseyn='N' and worktype='W'; </pre>

# CONNECTBY USING FUNCTION

## DB2

connect to sample user db2admin using db2admin!

```
CREATE FUNCTION get_direct_childs(code varchar(30), parent integer)
RETURNS TABLE(code VARCHAR(30), id INTEGER)
READS SQL DATA
DETERMINISTIC
NO EXTERNAL ACTION
RETURN
    SELECT code || '.' || RTRIM(CHAR(RANK() OVER (ORDER BY child_id))), child_id
    FROM ( SELECT empno FROM emp
           WHERE emp.mgr = get_direct_childs.parent)
    AS T(child_id)!
```

```
create function get_rec_childs(root integer)
RETURNS TABLE (code VARCHAR(30), id INTEGER)
READS SQL DATA
DETERMINISTIC
NO EXTERNAL ACTION
RETURN
WITH rec(code, id)
AS ( VALUES (CAST ('1' AS VARCHAR(30)), root )
UNION ALL
SELECT t.code, t.id
FROM rec, TABLE( GET_DIRECT_CHILDS(rec.code, rec.id ) ) AS T)
SELECT code, id FROM rec !
```

```
CREATE FUNCTION get_level(code VARCHAR(30))
RETURNS INTEGER
DETERMINISTIC
NO EXTERNAL ACTION
RETURN
(length(code) - length(replace(code, '.', '')))!
```

```
SELECT T.code ,T.id,substr( (space(2 * get_level(code)) || emp.ename), 1, 20 ) as ename,emp.mgr
FROM TABLE(get_rec_childs(7782)) AS T,emp where T.id = emp.empno!
```

# TO\_CHAR

Oracle	DB2
<p>Date Type을 Convert 하기 위해서, 오라클은 다양한 Format 을 지원한다.</p> <p>TO_CHAR(sysdate,'YYYY-MM-DD') or TO_CHAR(sysdate,'MM/DD/YYYY')</p>	<p>DB2에서는 다음과 같은 한가지 Format 만 지원한다.</p> <p>TO_CHAR(timestamp_expression,'YYYY-MM-DD HH24:MI:SS');</p> <p>이외에 다음과 같은 형식으로 할때 다양한 결과값을 얻을 수 있으며 필요시 User Define Function으로 생성하여 사용할 수 있다.</p> <p>CHAR(current date, ISO ) =&gt; 2005-03-03            CHAR(current date, USA) =&gt; 03/03/2005            CHAR(current date) =&gt; 2005-03-03            CHAR( current date – 1 days ) =&gt; 2005-03-02            CHAR(current date, JIS) = 2005-03-03            CHAR(current time,USA) = 02:21 PM            CHAR(current time + 2 hours,EUR) = '16.21.23'</p> <p><b>** CHAR 의 첫번째 Argument 가 Timestamp 일 경우는 두번째 Argument 를 지원하지 않는다.</b></p>



# TO\_CHAR

Oracle	DB2
<p>Date Type을 Convert 하기 위해서, 오라클은 다양한 Format 을 지원한다.</p> <p>TO_CHAR(sysdate,'YYYY-MM-DD') or TO_CHAR(sysdate,'MM/DD/YYYY')</p>	<p>DB2에서는 다음과 같은 한가지 Format 만 지원한다.</p> <p>TO_CHAR(timestamp_expression,'YYYY-MM-DD HH24:MI:SS');</p> <p>이외에 다음과 같은 형식으로 할때 다양한 결과값을 얻을 수 있으며 필요시 User Define Function으로 생성하여 사용할 수 있다.</p> <p>CHAR(current date, ISO ) =&gt; 2005-03-03            CHAR(current date, USA) =&gt; 03/03/2005            CHAR(current date) =&gt; 2005-03-03            CHAR( current date – 1 days ) =&gt; 2005-03-02            CHAR(current date, JIS) = 2005-03-03            CHAR(current time,USA) = 02:21 PM            CHAR(current time + 2 hours,EUR) = '16.21.23'</p> <p><b>** CHAR 의 첫번째 Argument 가 Timestamp 일 경우는 두번째 Argument 를 지원하지 않는다.</b></p>

## TO\_CHAR Uder Define Function

### DB2

```
DROP SPECIFIC FUNCTION UDF_TO_CHAR2!
CREATE FUNCTION TO_CHAR2(TS TIMESTAMP, FMT VARCHAR(30))
RETURNS VARCHAR(50)
SPECIFIC UDF_TO_CHAR
RETURN
WITH TMP ( DD, MM, YYYY,HH,MI,SS, NNNNNN ) AS
(
    SELECT
        SUBSTR(DIGITS ( DAY(TS)),9),
        SUBSTR(DIGITS (MONTH(TS)),9),
        RTRIM (CHAR(YEAR(TS))),
        SUBSTR(DIGITS ( HOUR(TS)),9),
        SUBSTR( DIGITS (MINUTE(TS)),9),
        SUBSTR( DIGITS (SECOND(TS)),9),
        RTRIM(CHAR(MICROSECOND(TS)))
    FROM SYSIBM.SYSDUMMY1
)
SELECT
    CASE UCASE(FMT)
        WHEN 'YYYYMMDD'
            THEN YYYY || MM || DD
        WHEN 'MM/DD/YYYY HH24:MI:SS'
            THEN MM || '/' || DD || '/' || YYYY || '-' || HH || ':' || MI || ':' || SS
        WHEN 'YYYY/MM/DD HH24:MI:SS'
            THEN YYYY || '/' || MM || '/' || DD || '-' || HH || ':' || MI || ':' || SS
        ELSE
            'DATE FORMAT NOT RECOGNIZED.'
        END
FROM TMP!
```

## T O \_ D A T E

Oracle	DB2
<p>TO_DATE(char [,fmt[nlsparam]]) 은 Char 를 Date 로 Convert 한다. 만약 fmt 가 지정되지 않으면 Default date format이 사용된다.</p>	<p>TO_DATE(string_expression,'YYYY-MM-DD HH24:MI:SS');</p> <p>이외에 다음과 같은 형식으로 할때 다양한 결과값을 얻을 수 있으며 필요시 User Define Function으로 생성하여 사용할 수 있다.</p> <p>CHAR(current date, ISO ) =&gt; 2005-03-03          CHAR(current date, USA) =&gt; 03/03/2005          CHAR(current date) =&gt; 2005-03-03          CHAR( current date – 1 days ) =&gt; 2005-03-02          CHAR(current date, JIS) = 2005-03-03          CHAR(current time,USA) = 02:21 PM          CHAR(current time + 2 hours, EUR) = '16.21.23'</p> <p>** CHAR 의 첫번째 Argument 가 Timestamp 일 경우는 두번째 Argument 를 지원하지 않는다.</p>

## TO \_ N U M B E R

Oracle	DB2
<p>TO_NUMBER(char [,fmt[nlsparam]])</p>	<div data-bbox="1131 446 2004 670"> <pre> graph LR     subgraph DB2_Types [DB2 Types]         BIGINT         INTEGER         INT     end     subgraph Inputs [Inputs]         direction TB         A["( numeric-expression )"]         B["character-expression"]         C["date-expression"]         D["time-expression"]     end     DB2_Types --- Inputs             </pre> </div> <p>Ex )  SELECT INTEGER(EMPNO) FROM EMP;</p> <p>만약 BirthDate 가 '1964-07-20' 이면  INTEGER(BIRTHDATE) 결과값은 19640720  이다.</p>

## R P A D U s e r D e f i n e F u n c t i o n — Example

## DB2

```
SELECT CHAR(RPAD('ABCDE',12,'*!'),20) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : ABCDE\*!\*!\*!\*

```
SELECT CHAR(RPAD('ABC',12,'*!'),20) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : ABC

```
SELECT char(rpad('ABCDE',20,"") || 'X',50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : ABCDE X

```
SELECT char(rpad('ABCDE',3) || 'X',50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : ABCX

```
SELECT char(rpad(927,12,'*.'),50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : 927\*.\*.\*.\*

```
SELECT char(rpad(927,12,"") || 'X',50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : 927 X

```
SELECT char(rpad(9021,3),20) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : 902

## RPAD User Define Function

### DB2 - fncOrPad.txt file 참조

```
CREATE FUNCTION RPAD (C1 VarChar(4000), N integer, C2 VarChar(4000))
RETURNS VARCHAR(4000)
LANGUAGE SQL
SPECIFIC RPADBase
DETERMINISTIC
CONTAINS SQL
NO EXTERNAL ACTION
RETURN  substr(C1 || repeat(C2,(((sign(N-length(C1))+1)/2)*(N-length(C1)+length(C2))/(length(C2)+1-sign(length(C2))))),1,N) !
```

```
CREATE FUNCTION RPAD (C1 VarChar(4000), N integer)
RETURNS VARCHAR(4000)
LANGUAGE SQL
SPECIFIC RPADVarCharParm2
..
RETURN  RPAD(C1,N,' ') !
```

```
CREATE FUNCTION RPAD (I1 Integer, N integer, C2 Varchar(4000))
RETURNS VARCHAR(4000)
LANGUAGE SQL
SPECIFIC RPADIntParm3
..
RETURN  RPAD(rtrim(char(I1)),N,C2) !
```

```
CREATE FUNCTION RPAD (I1 Integer, N integer)
RETURNS VARCHAR(4000)
LANGUAGE SQL
.. SPECIFIC RPADIntParm2
RETURN  RPAD(rtrim(char(I1)),N,' ') !
```

## L P A D U s e r D e f i n e F u n c t i o n

**DB2 - fncOrPad.txt file 참조**

```
CREATE FUNCTION LPAD (C1 VarChar(4000), N integer, C2 VarChar(4000))
RETURNS VARCHAR(4000)
.... SPECIFIC LPADBase ... RETURN
CASE WHEN N > length(C1) THEN substr(repeat(C2,(N-length(C1)+length(C2))/(length(C2)+1-sign(length(C2)))),1,N-length(C1)) || C1
ELSE substr(C1,1,N)
END !
```

```
CREATE FUNCTION LPAD (C1 VarChar(4000), N integer, C2 VarChar(4000))
RETURNS VARCHAR(4000)
.... SPECIFIC LPADBase
.... RETURN CASE
WHEN N > length(C1) THEN substr(repeat(C2,(N-length(C1)+length(C2))/(length(C2)+1-sign(length(C2)))),1,N-length(C1)) || C1
ELSE substr(C1,1,N)
END!
```

```
CREATE FUNCTION LPAD (C1 VarChar(4000), N integer)
RETURNS VARCHAR(4000)
.... SPECIFIC LPADParm2 .... RETURN LPAD(C1,N,'') !
```

```
CREATE FUNCTION LPAD (I1 Integer, N integer, C2 Varchar(4000))
RETURNS VARCHAR(4000)
... SPECIFIC LPADIntParm3..... RETURN LPAD(rtrim(char(I1)),N,C2) !
```

```
CREATE FUNCTION LPAD (I1 Integer, N integer)
RETURNS VARCHAR(4000)
.... SPECIFIC LPADIntParm2 ..... RETURN
LPAD(rtrim(char(I1)),N,'') !
```

## LPAD User Define Function – Example

**DB2**

```
SELECT char(lpad('ABCDE',15,'*.'),50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : \*.\*.\*.\*.ABCDE

```
SELECT char(lpad('ABCDE',3,'*.'),50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : ABC

```
SELECT char(lpad('ABCDE',15,"") || 'X',50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : ABCDEX

```
SELECT char(lpad('ABCDE',3),20) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : ABC

```
SELECT char(lpad(9021,15,'*.'),50) FROM SYSIBM.SYSDUMMY1;
```

➔ Result : \*.\*.\*.\*.\*9021



# TRUNC User Define Function

## DB2 – fnRoundD.txt 참조

```
CREATE FUNCTION TRUNC(inTS Timestamp, Fmt VarChar(5))
RETURNS Timestamp
.....
BEGIN ATOMIC
DECLARE Jan01, Mon01 DATE;
DECLARE UCASE_Fmt VarChar(5);

SET UCASE_Fmt = UCASE(Fmt);
IF UCASE_Fmt = 'WW' THEN
    SET Jan01 = DATE(SUBSTR(CHAR(inTS),1,4)||'001');
ELSEIF UCASE_Fmt = 'W' THEN
    SET Mon01 = DATE(inTS) - (DAY(inTS)-1) DAYS;
END IF;

RETURN
CASE
WHEN UCASE_Fmt IN ('CC', 'BCC') THEN
    TIMESTAMP(SUBSTR(DIGITS(YEAR(inTS)+100),7,2) || '00-01-01-00.00.00.000000')
WHEN UCASE_Fmt IN ('YYYY', 'YYYY', 'YEAR', 'SYEAR', 'YYY', 'YY', 'Y') THEN
    TIMESTAMP(SUBSTR(CHAR(inTS),1, 4) || '-01-01-00.00.00.000000')
WHEN UCASE_Fmt IN ('MONTH', 'MON', 'MM', 'RM') THEN
    TIMESTAMP(SUBSTR(CHAR(inTS),1, 7) || '-01-00.00.00.000000')
.....
WHEN UCASE_Fmt IN ('HH', 'HH12', 'HH24') THEN
    TIMESTAMP(SUBSTR(CHAR(inTS),1,13) || '.00.00.000000')
WHEN UCASE_Fmt = 'MI' THEN
    TIMESTAMP(SUBSTR(CHAR(inTS),1,16) || '.00.000000')
END; .....
```

## Round User Define Function

### DB2 – fnRoundD.txt 참조

```
CREATE FUNCTION ROUND(inTS Timestamp, Fmt VarChar(5))
RETURNS Timestamp
.....
BEGIN ATOMIC
DECLARE Jan01, Mon01 DATE;
DECLARE UCASE_Fmt VarChar(5);

SET UCASE_Fmt = UCASE(Fmt);

IF UCASE_Fmt = 'WW' THEN
    SET Jan01 = DATE(SUBSTR(CHAR(inTS),1,4)||'001');
ELSEIF UCASE_Fmt = 'W' THEN
    SET Mon01 = DATE(inTS) - (DAY(inTS)-1) DAYS;
END IF;

RETURN
CASE
WHEN UCASE_Fmt IN ('CC', 'BCC') THEN
    TIMESTAMP(SUBSTR(DIGITS(YEAR(inTS)+100),7,2) || '00-01-01-00.00.00.0000000')
WHEN UCASE_Fmt IN ('YYYY', 'YYYY', 'YEAR', 'SYEAR', 'YYY', 'YY', 'Y') THEN
    TIMESTAMP(SUBSTR(CHAR(inTS + 7 MONTHS),1, 4) || '-01-01-00.00.00.0000000')
WHEN UCASE_Fmt IN ('MONTH', 'MON', 'MM', 'RM') THEN
    TIMESTAMP(SUBSTR(CHAR(inTS + 1 MONTH - 15 DAYS),1, 7) || '-01-00.00.00.0000000')
.....
WHEN UCASE_Fmt = 'MI' THEN
    TIMESTAMP(SUBSTR(CHAR(inTS),1,16) || '.00.0000000')
END;
END!
```

# TRIM User Define Function - STRIP

## DB2 – fncStrip.txt 참조

```
CREATE FUNCTION STRIP (STR VarChar(2000), BLT VarChar(8), SC VarChar(1))
RETURNS VARCHAR(2000)
SPECIFIC STRIP3Parm
.... RETURN WITH
  Const_Tbl (BLT_N, SC_N) AS ( VALUES (translate(substr(BLT,1,1)), char(SC,1))) ,Trail (Seq, Trim) AS ( VALUES (length(Str), Str)
UNION ALL
SELECT Seq-1, substr(Trim,1,Seq-1)
  FROM Trail,Const_Tbl
WHERE Seq > 0  AND (BLT_N = 'T' OR BLT_N = 'B')  AND substr(Trim,Seq,1) = SC_N ) ,Int_Result (Trim_L, Trim) AS (
SELECT Seq, Trim
  FROM Trail
WHERE Seq = (Select min(Seq) From Trail)
) ,Lead (Seq, Trim) AS (
SELECT 1, Trim
  FROM Int_Result
UNION ALL
SELECT Seq+1, substr(Pre.Trim,2,Trim_L-Seq)
  FROM Lead    Pre
      ,Int_Result IR
      ,Const_Tbl
WHERE Seq <= 2000
  AND Seq < Trim_L
  AND (BLT_N = 'L' OR BLT_N = 'B')
  AND substr(Pre.Trim,1,1) = SC_N
)
SELECT Trim
  FROM Lead
WHERE Seq = (SELECT max(Seq) FROM Lead)!
```

## TRIM User Define Function - Example

### DB2 – fncStrip.db2 참조

Values char(STRIP('0000345.50','L','0'),20)

➔ Result : 345.50

Values char(STRIP('0000345.50','B','0'),20)

➔ Result : 345.5

Values char(STRIP('0000345.50 ','B','") || 'X',20)

➔ Result : 0000345.50X

Values char(STRIP('0000345.50 ','L','") || 'X',20)

➔ Result : 0000345.50 X

Values char(STRIP('0000345.50 ','T','") || 'X',20)

➔ Result : 0000345.50X

Values char(STRIP('0000345.50 ') || 'X',20)

➔ Result : 0000345.50X

Values char(STRIP(' 0 0345.50 ') || 'X',20)

➔ Result : 0 0345.50X

## Function Comparision.

Oracle	DB2
NVL NVL(TO_CHAR(MANAGER_ID),'No Manager')	COALESCE COALESCE(MANAGER_ID,'No Manager')
MIN (expr) SELECT MIN(DISTINCT(SALARY)) FROM EMP;	MIN ( expr ) SELECT MIN(DISTINCT(SALARY)) FROM EMP;
SUBSTR(string,n[,m])	SUBSTR(string,start[,length])
SUM(expr) SELECT SUMM(DISTINCT(SALARY)) FROM EMP;	SUM(expr) SELECT SUMM(DISTINCT(SALARY)) FROM EMP;

## Decode Statement

Oracle	DB2
DECODE(condition,case1,assign1,case2,assign2.. default)	CASE condition WHEN case1 THEN assign 1 WHEN case2 THEN assign 2 ELSE default END
<pre>SELECT AVG(DECODE( Grade,                     'A',1,                     'B',2,                     'C',3 )) INTO v_Grade FROM Students WHERE DEPARTMENT = p_Department AND Course_ID = p_Course_ID;</pre>	<pre>SELECT AVG(CASE GRADE               WHEN 'A' THEN 1               WHEN 'B' THEN 2               WHEN 'C' THEN 3             END) INTO v_Grade FROM Students WHERE DEPARTMENT = p_Department AND Course_ID = p_Course_ID;</pre>

## RowNum

Oracle	DB2
<pre>SELECT * FROM (   SELECT ROWNUM AS NUM, K.* FROM (     SELECT * FROM T_ALARM_HISTORY     WHERE SeRVICE = ?     ORDER BY ALARM_DATE DESC,              ALARM_TIME DESC ) K   ) N WHERE N.NUM BETWEEN 5 AND 10</pre>	<pre>SELECT * FROM (   SELECT ( ROW_NUMBER() OVER() ) AS NUM, K.* FROM     ( SELECT * FROM T_ALARM_HISTORY       WHERE SeRVICE = ?       ORDER BY ALARM_DATE DESC,                ALARM_TIME DESC ) K   ) N WHERE N.NUM BETWEEN 5 AND 10</pre>

## OuterJoin

Oracle	DB2
SELECT A.last_name, A.id, B.name FROM emp A, Customer B WHERE A.id (+) = B.sales_rep_id;	SELECT A.last_name, A.id , B.name FROM emp A RIGHT OUTER JOIN customer B ON A.id = B.sales_rep_id;
SELECT A.last_name, A.id , B.name FROM emp A, Customer B WHERE A.id = B.sales_rep_id(+)	SELECT A.last_name, A.id , B.name FROM emp A LEFT OUTER JOIN Customer B ON A.id = B.sales_rep_id;
SELECT A.last_name, A.id ,B.name FROM emp A, Customer B WHERE A.id (+) = B.sales_rep_id (+)	SELECT A.last_name, A.id, B.name FROM emp A FULL OUTER JOIN Customer B ON A.id = B.sales_rep_id;



## Sequence Table

Oracle	DB2
<pre>SELECT SEQ_PROCESSID.NEXTVAL AS PSID FROM DUAL;</pre>	<pre>CREATE SEQUENCE Order_SEQ START WITH 1 INCREMENT BY 1 NO MAXVALUE NO CYCLE CHCCE 24; OR CREATE TABLE EMPLOYEES(EMP_ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (START WITH 1 , INCREMENT BY 1 ).....  INSERT INTO order(orderno, cutno) VALUES (NEXTVAL FOR Order_SEQ, 123456);  INSERT INTO line_item(orderno,partno, quantity) VALUES (PREVVAL FOR Order_SEQ, 987654,1);  SELECT NEXTVAL FOR ORDER_SEQ FROM SYSIBM.SYSDUMMY1; OR VALES NEXTVAL FOR ORDER_SEQ</pre>

## D u m m y T a b l e

Oracle	DB2
<pre>SELECT TO_CHAR(SYSDATE,'YYYYMMDDHH24MISS') FROM DUAL;</pre>	<pre>SELECT TO_CHAR( CURRENT TIMESTAMP , 'YYYYMMDDHH24MISS') FROM SYSIBM.SYSDUMMY1;</pre>

## T a b l e / V i e w L i s t

Oracle	DB2
Select table_name from dba_tables; Select table_name from All_tables;	List tables for schema <XXXX> List tables for system => Catalog Table List Select tabname from SYSCAT.TABLES;

## T a b l e 의 구성 보기

Oracle	DB2
Desc scott.tiger	DESCRIBE TABLE DB2INST1.EMP;

# Special Registry

CLP Command	Description
CURRENT DATE (Data type : Date)	Admin server configuration을 Return한다.
CURRENT DEGREE (Data type : Char(5))	Intra-Partition의 Degree 정보를 Return 한다.
CURRENT QUERY OPTIMZATION (Data type : Integer)	Optimization Level 정보를 Return한다.
CURRENT SERVER (Data type : Varchar(18))	현재 접속되어 있는 Database 정보를 Return한다.
CURRENT TIME (Data type : Time)	SQL 문장이 실행되는 시점의 시간을 Return한다.
CURRENT TIMESTAMP (Data type : Timestamp)	SQL 문장이 실행되는 시점의 micro-second을 Return한다
CURRENT TIMEZONE (Data type : Decimal(6,0))	Server의 현재 Time-zone을 Return한다.
USER (Data type : Char(8))	Database에 접속되어 있는 User-ID를 반환한다.
<p>Example )</p> <pre> /home/db2inst&gt; db2 Db2=&gt; values (current user) 1 ----- Db2inst Db2=&gt; values current server 1 ----- SAMPLE </pre>	

## A p p e n d i x

Function Name	Sybase R11.0	Oracle 9.2	DB2 UDB V8.1	File Name
add_months		o	(o)	fnAddMonth.db2
atan2	o	o (ATAN2)	o (ATAN2)	
bitwise_AND [integer]	o (& operator)	o (BITAND)	(o)(BITAND)	fnBitAnd.db2
bit_to_num		o	(o)	fnBitToNum.db2
char (Sybase)	o	o (CHR)	o (CHR)	
charindex	o	o (LOCATE)	o (LOCATE or POSSTR)	
char_length	o	o (LENGTH)	(o)(LENGTH_O for SJIS)	
convert (Sybase)	o		o (CAST Specification)	
convert (char(12),datetime,style)	o		+ (CHAR(DATE(),style))	

## A p p e n d i x

Function Name	Sybase R11.0	Oracle 9.2	DB2 UDB V8.1	File Name
datalength	o	+ (LENGTHB for char)	o (LENGTH)	
dateadd	o		o (datetime arithmetic)	
datediff	o		+ (TIMESTAMPDIFF)	
datename	o	o (TO_CHAR)	o (MONTHNAME, DAYNAME)	
datepart	o	o (EXTRACT)	o (DAY, ... ,WEEK,YEAR)	
getdate	o	o (SYSDATE)	o (CURRENT TIMESTAMP) (o)	
getutcdate	o	o (CURRENT_DATE)	(o)	

## A p p e n d i x

Function Name	Sybase R11.0	Oracle 9.2	DB2 UDB V8.1	File Name
greatest		o	(o)[char, integer; parm count = 2-10]	fnGreatestString.db2 fnGreatestNum.db2
least		o	(o)[char, integer; parm count = 2-10]	fnLeastString.db2 fnLeastNum.db2
hextoint	o		(o)	fnHexToInt.db2
hextoraw		o	(o)[Returns VarChar for bit data]	fnHexToRaw.db2
initcap		o	(o)	fnInitCap.db2
instrb		o	(o) + (LOCATE)	fnInstrb.db2
inttohex	o		+ (HEX)	fnIntToHex.db2
isnumeric	o		(o)	fnIsNumeric.db2



# Appendix

Function Name	Sybase R11.0	Oracle 9.2	DB2 UDB V8.1	File Name
last_day		o	(o)	fnLastDay.db2
length (Oracle)*	o (char_l ength)	o	(o)(LENGTH_O for SJIS**)	fnLengthO.db2
ln	o		o	
log (Sybase)	o		o	
log(m, n) (Oracle)		o	(o)	fnLogMN.db2
log10	o		o	
lpad		o	(o)	fnPad.db2
rpadd		o	(o)	fnPad.db2
mod	o (% operat or)	o	o	

# Appendix

Function Name	Sybase R11.0	Oracle 9.2	DB2 UDB V8.1	File Name
months_between		o	(o)	fnMonthBetween.db2
new_time		o	(o)	fnNewTime.db2
next_day		o	(o)	fnNextDay.db2
nvl	o (isnull)	o	o (COALESCE)	
nvl2		o	o (CASE expression)	
patindex	o		+ (LOCATE)	
pi	o		(o)	fnMath.db2
rawtohex		o	o (HEX)	
replicate	o		o (REPEAT)	
reverse	o		(o)	fnReverse.db2
right	o		o	
round (date fmt)		o	(o)	fnRound.db2
square	o		(o)	fnMath.db2
str	o	o (TO_CHAR)	(o)	fnStr_ms.db2
stuff	o		(o)	fnStuff.db2

## Appendix

Function Name	Sybase R11.0	Oracle 9.2	DB2 UDB V8.1	File Name
substr (Oracle)*		o	(o)(SUBSTR_O for SJIS**)	fnSubstO.db2
substring	o	o (SUBSTRB)	o (SUBSTR)	
sysdate	o (GETDATE)	o	o (CURRENT IMESTAMP) (o)	fnSysdate.db2
to_char (date format)	+ (CONVERT)	o	(o)	fnToChar_Date.db2
to_char (numeric format)	+ (CONVERT)	o	(o)	fnToChar_Num.db2
to_date	+ (CONVERT)	o	(o)	fnToDate.db2
trim		o	(o)(STRIP: different format)	fnStrip.db2
trunc (date fmt)		o	(o)	

## Appendix

- O 은 함수가 존재하는 것을 의미한다.
- O (... ) 은 동작은 같고 이름이 다른 함수가 존재하는 것을 의미한다.
- (O) 은 User Define Function을 제공한다. File을 Build 하여 User Define Function을 만들어 준다.
- (+) 은 제한된 Spec 으로 User Define Function을 제공한다.

## Appendix – Document

- Manual :
  - Command Reference ; db2 system command 부분과 Import / export 등에 대해 설명한다.
  - DB2 Application Server Programming – Stored Procedure 와 Function 에 대해 설명한다.
  - DB2 Application Client Programming - Client Program ( C/ C++/ JAVA ) 에 대해 설명한다.
  - CLP Quick Reference : DB2 Command 와 SQL Command 에 대해 일괄적으로 설명한다.
- URL : IBM Developwork Site 및 DB2 자료를 찾을 수 있는 홈페이지를 모아놓은 곳.
- Sample Source : User Define Function 및 Sample Source 을 모아 놓은 곳.
- Porting Guide : DB2로 Porting 시에 타 DBMS 와 비교한 문서 및 DB2 포팅시 참고할 수 있는 참고 Redbook을 모아놓은 곳이다.



# DB2 Programming



## H o s t S t r u c t u r e S u p p o r t i n C / C + +

```
struct tag {  
    short id;  
    struct { short length ;  
            char data[10]; } name;  
    struct { short years;  
            double salary; } info;  
} staff_record;
```

```
EXEC SQL SELECT id,name,years,salary INTO :staff_record FROM staff  
WHERE id = 10 ;
```

OR

```
EXEC SQL SELECT id,name,years,salary  
INTO :staff_record.id, :staff_record.name, :staff_record.info.years, :staff_rec  
ord.info.salary FROM staff WHERE id = 10;
```

## DB2 Application Code – Structure Type

```
struct tag {  
    short id;  
    struct { short length ;  
            char data[10]; } name;  
    struct { short years;  
            double salary; } info;  
} staff_record;
```

```
EXEC SQL SELECT id,name,years,salary INTO :staff_record FROM staff  
WHERE id = 10 ;
```

OR

```
EXEC SQL SELECT id,name,years,salary  
INTO :staff_record.id, :staff_record.name, :staff_record.info.years, :staff_rec  
ord.info.salary FROM staff WHERE id = 10;
```



## D B 2 C o m p i l e

DB2 Connect to < Database >

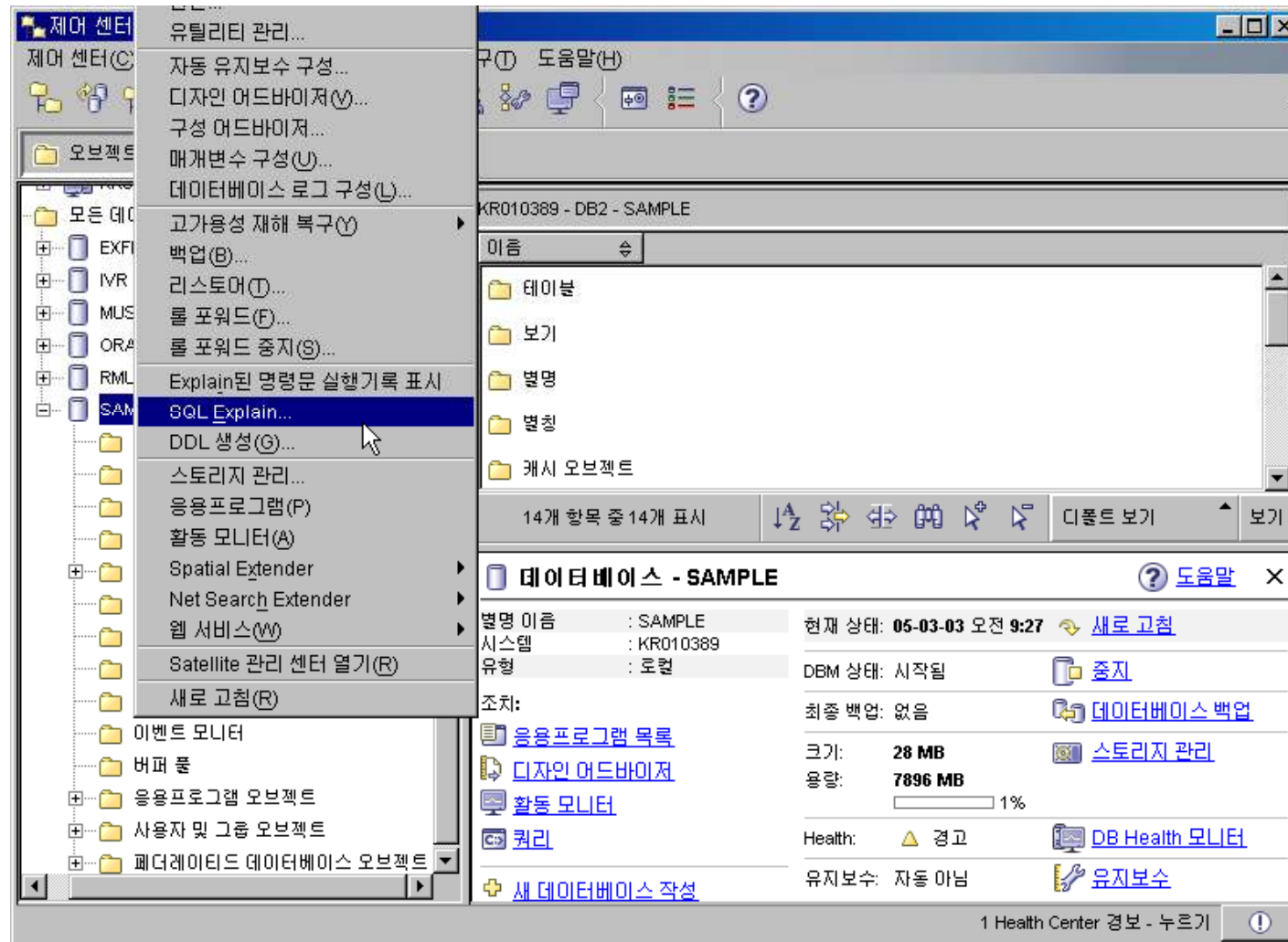
DB2 PREP filename.sqc BINDFILE

DB2 BIND filename.bnd blocking all

- Precompile 후 네가지 유형의 출력 생성.
  - 수정된 소스 파일 - SQL문을 DB2 Runtime API 호출로 변환시킨다.
  - 패키지 - PACKAGE 옵션을 사용하거나 BINDFILE 을 지정하지 않으면 자동으로 Package 가 생성된다.
  - 바인드 파일 - BINDFILE 옵션을 사용하면, DB2 SQL 문이 포함된 바인드 파일이 작성된다.
  - 메시지 파일- MESSAGES 옵션을 사용하여 파일을 지정하면 Precompile 중에 발생하는 오류 및 경고 메시지를 파일에 쓴다.
- Bind 란 ?
  - 응용프로그램이 실행될 때 데이터베이스에 액세스하기 위해 데이터베이스 관리 프로그램이 필요로 하는 패키지를 작성하는 과정을 말한다.
- 패키지
  - 패키지는 Precompile에서 PACKAGE 옵션이 지정되지 않으면 bind 과정에서 filename.bnd 파일의 이름과 동일하며 파일의 이름이 8자 이상일 경우 8자로 절단된다.
  - 새로 작성된 패키지 이름이 데이터베이스에 있는 패키지과 같은 경우 새 패키지는 이전의 패키지를 대체한다.



# SQL Analysis – SQL Explain



## SQL Analysis – SQL Explain

SQL문 Explain - SAMPLE

KR010389 - DB2 - SAMPLE

SQL 텍스트

SELECT COUNT(\*) FROM SALES

가져오기(G)

저장(S)

쿼리 번호 1

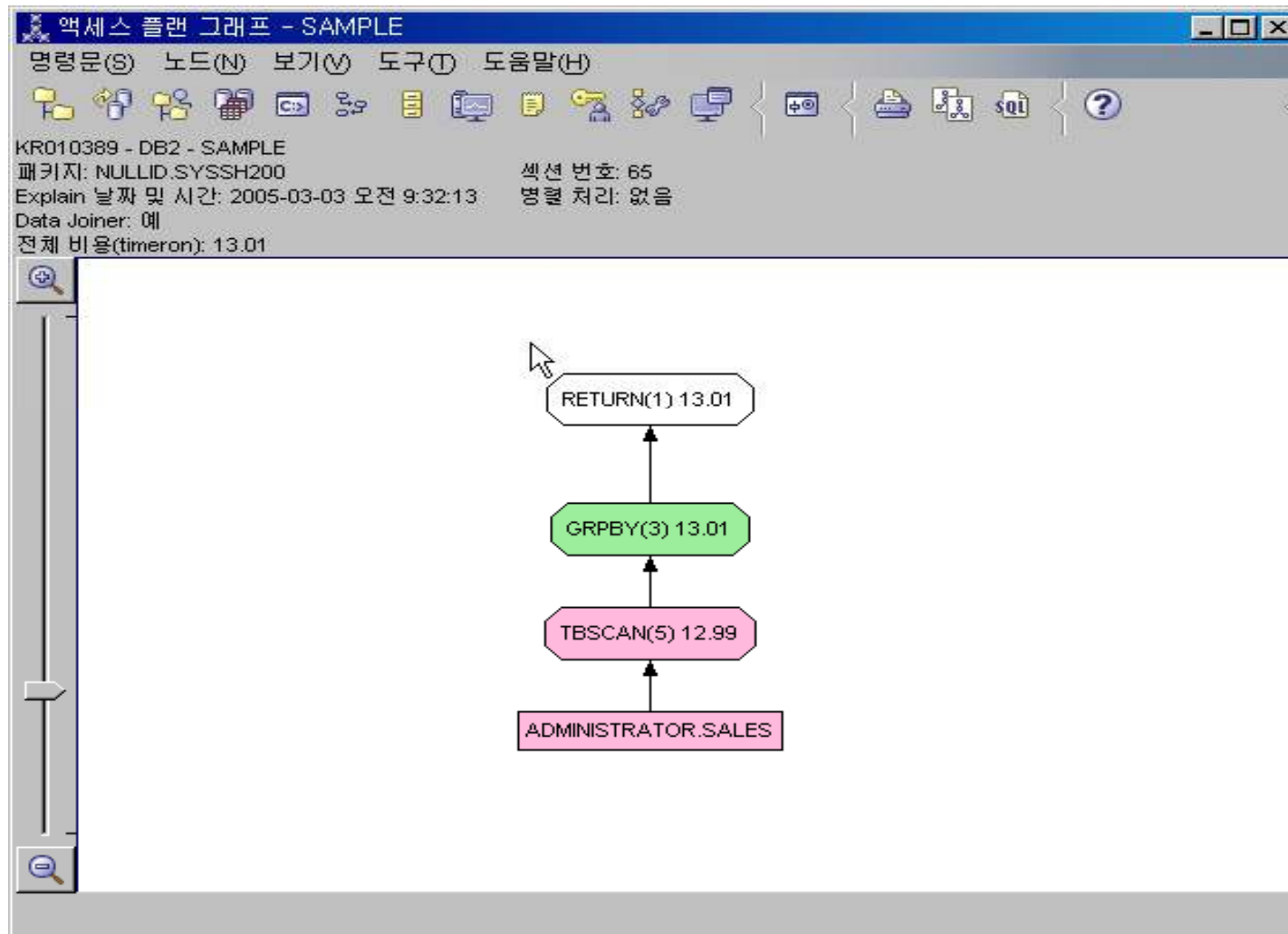
쿼리 태그

최적화 클래스 5

☒ Explain 테이블의 모든 컬럼 처리

확인 취소 도움말

## SQL Analysis – SQL Explain



## SQL Tuning

- 한번에 수행할 수 있는 Multiple Statement 를 피하라

.Bad Case	Good
<pre>INSERT INTO T1 VALUES (A1,B1,C1); INSERT INTO T1 VALUES (A2,B2,C2); INSERT INTO T1 VALUES (A3,B3,C3);</pre>	<pre>INSERT INTO T1 VALUES (A1,B1,C1),(A2,B2,C2), (A3,B3,C3);</pre>

- Multiple SQL 문장을 하나의 SQL 문장으로 변경한다.

.Bad Case	Good
<pre>IF ( PRICE &lt;= MaxPrice ) THEN   INSERT INTO T1( C1,C2 ) Values ( Oid,Price ) ; ELSE   INSERT INTO T1(C1,C2 ) Values (Oid , MaxPrice ) ; END IF;</pre>	<pre>INSERT INTO T1 ( C1,C2 ) VALUES (Oid, CASE WHEN (Price &lt;= MaxPrice ) THEN Price, ELSE MaxPrice END );</pre>
<pre>IF ( Name IS NOT NULL ) THEN   SET ProdName = Name; ELSE IF (NameStr IS NOT NULL ) THEN   SET ProdName = NameStr; ELSE   SET ProdName = DefaultName; END IF;</pre>	<pre>SET ProdName = ( CASE   WHEN ( NAME IS NOT NULL ) THEN Name   WHEN ( NameStr IS NOT NULL ) THEN NameStr   ELSE DefaultName END ); Or SET ProdName = COALESCE(Name,NameStr,DefaultName );</pre>

## SQL Statement tuning

- SELECT ... FOR UPDATE 문을 이용하면 Select 되는 모든 row에 U Lock 을 걸어 향후 Select 된 Row를 Update할때 deadlock을 방지한다.
- SELECT ... FOR READ ONLY 문은 result Table이 read-only 라는 것을 의미하며 이것은 FETCH operation 을 수행할때 DB2가 Blocking ( multiple row를 Client 에게 return ) 을 수행하도록 함으로써 성능을 향상시킬수 있다.
- 반복되는 문장이면 literal 를 사용하는 대신 parameter maker를 사용하도록 한다.

Example >

Bad Case	Good
INSERT INTO T1 VALUES (A1,B1,C1); INSERT INTO T1 VALUES (A2,B2,C2);	INSERT INTO T1 VALUES ( ?, ?, ? );



# DB2 명령어 Summary.





## Instance 관련 명령어

CLP Command	Description
db2start	Database Instance Start.
db2stop	Database Instance Stop.
get admin cfg	Admin Server의 현재 값을 Return한다.
get dbm cfg	DBM CFG의 현재(변경요청)값을 Return한다.
get dbm cfg show detail	DBM CFG의 현재 값과 변경 될 값을 Return한다.(V8)
update dbm cfg using <p> <v>	DBM Parameter <p>를 Value <v>로 변경한다.
get instance	환경변수 DB2INSTANCE의 값을 Return한다.
list active databases	Active databases 와 connection 수를 나열한다.
list application [show detail]	현재 접속되어 있는 Application의 정보를 Return한다.
force application (h1 [,h2,...,hn])	Handle Number에 의해 해당 Application을 Disconnect한다.
force application all	Database에 모든 Application을 Disconnect한다.
attach to <node> [ user <userid> using <pwd> ]	노드 <Node>에 사용자 <userid>와 비밀번호<pwd>로 Attach한다.

## 데이터베이스 관련 명령어

CLP Command	Description
<b>create database &lt;dbname&gt;</b>	Database <dbname>을 생성한다.
<b>activate database &lt;dbname&gt;</b>	Database <dbname>을 활성화(Start) 한다.
<b>deactivate database &lt;dbname&gt;</b>	Database <dbname>을 비활성화(Stop) 한다.
<b>connect to &lt;dbname&gt; [ [user &lt;userid&gt;] using &lt;pwd&gt; ]</b>	Database <dbname>에 사용자 <userid>와 비밀번호<pwd>로 Connect한다.
<b>get connection state</b>	Database 접속 상태를 확인한다.
<b>connect reset</b>	Database 접속을 해지한다.
<b>connect terminate</b>	Database 접속을 해지하고, <b>Back End Process</b> 를 종료하며, <b>Registry Variable</b> 에 변경된 환경변수를 값을 적용한다.
<b>get db cfg show detail</b>	DB CFG의 현재 값과 변경 될 값을 Return한다. (V8)
<b>get db cfg for &lt;dbname&gt;</b>	Database <dbname>의 구성 값을 Return한다.
<b>update db cfg for &lt;dbname&gt; using &lt;p&gt; &lt;v&gt;</b>	Database <dbname>의 매개변수 <p>를 Value <V>로 변경한다.
<b>describe table &lt;tablename&gt; [show detail]</b>	Table/View의 Column 정보를 표시한다.
<b>describe indexes for table &lt;tablename&gt; [show detail]</b>	Table의 Index 정보를 표시한다.
<b>list tablespaces [show detail]</b>	Table space ID, name, type, contents와 state를 표시한다.
<b>list tablespace containers for &lt;id&gt; [show detail]</b>	Table space <id>의 Container 정보를 표시한다.
<b>quiesce tablespaces for table &lt;tablename&gt; &lt; SHARE   INTENT TO UPDATE   EXCLUSIVE   RESET &gt;</b>	Table space 상태를 변경한다.
<b>LIST TABLES FOR {USER   ALL   SYSTEM   SCHEMA schema-name}} [SHOW DETAIL]</b>	Table List를 표시한다.

# Connectivity

CLP Command	Description
<b>catalog [admin] &lt;protocol&gt; node ...</b>	<b>&lt;protocol&gt;</b> 에 해당하는 항목을 <b>Node Directory</b> 에 생성한다.
<b>list [admin] node directory</b>	<b>Node directory</b> 의 항목을 표시한다.
<b>catalog database &lt;dbname&gt;...</b>	<b>Database &lt;dbname&gt;</b> 항목을 <b>Database Directory</b> 에 생성한다.
<b>list database directory [on &lt;path&gt;]</b>	<b>Database Directory</b> 내의 <b>Instance</b> 에 해당하는 항목을 표시한다.
 <b>catalog tcpip node mynode remote 127.0.0.1 server 50000</b>  <b>catalog db sample as mydb at node mynode</b>  <b>uncatalog db mydb</b>  <b>uncatalog node mynode</b>	

# Performance

CLP Command	Description
<b>get monitor switches</b>	실행 중인 <b>Session</b> 의 <b>Monitor Switch</b> 를 나열한다.
<b>update dbm cfg using dft_mon_bufpool [global] [on off]</b> <b>update dbm cfg using dft_mon_lock [global] [on off]</b> <b>update dbm cfg using dft_mon_sort [global] [on off]</b> <b>update dbm cfg using dft_mon_stmt [global] [on off]</b> <b>update dbm cfg using dft_mon_table [global] [on off]</b> <b>update dbm cfg using dft_mon_timestamp [global] [on off]</b> <b>update dbm cfg using dft_mon_uow [global] [on off]</b> <b>update dbm cfg using health_mon [global] [on off]</b>	<b>Database Manager</b> 의 <b>Monitor Switch</b> 를 변경한다.
<b>update monitor switches using bufferpool [global] [on off]</b> <b>update monitor switches using lock [global] [on off]</b> <b>update monitor switches using sort [global] [on off]</b> <b>update monitor switches using statement [global] [on off]</b> <b>update monitor switches using table [global] [on off]</b> <b>update monitor switches using timestamp [global] [on off]</b> <b>update monitor switches using uow [global] [on off]</b>	<b>Session</b> 의 <b>Monitor Switch</b> 를 변경한다.
<b>reset monitor all</b>	<b>Monitor Switch</b> 의 값을 초기화 한다.
<b>reorgchk [current update] statistics on table user</b> <b>reorgchk [current update] statistics on table system</b> <b>reorgchk [current update] statistics on table all</b> <b>reorgchk [current update] statistics on table</b> <b>&lt;schema&gt;.&lt;tablename&gt;</b> <b>reorgchk [current update] statistics on all</b> <b>reorgchk [current update] statistics on schema &lt;schema&gt;</b>	<b>Table</b> 의 <b>Reorganization</b> 필요 여부를 점검한다.

# Performance

CLP Command	Description
<p>get snapshot for dbm</p> <p>get snapshot for all           on &lt;dbname&gt;</p> <p>get snapshot for db           on &lt;dbname&gt;</p> <p>get snapshot for tables       on &lt;dbname&gt;</p> <p>get snapshot for tablespaces on &lt;dbname&gt;</p> <p>get snapshot for locks       on &lt;dbname&gt;</p> <p>get snapshot for bufferpools on &lt;dbname&gt;</p> <p>get snapshot for applications on &lt;dbname&gt;</p> <p>get snapshot for dynamic sql on &lt;dbname&gt;</p> <p>get snapshot for application agentid 4</p> <p>get snapshot for application applid *LOCAL.DB2.00..</p> <p>get snapshot for locks for application agentid 4</p> <p>get snapshot for locks for application applid *LOCAL.DB2.00..</p> <p>db2 get health snapshot for dbm                           [show detail]</p> <p>db2 get health snapshot for all databases               [show detail]</p> <p>db2 get health snapshot for all on &lt;dbname&gt;           [show detail]</p> <p>db2 get health snapshot for database   on &lt;dbname&gt; [show detail]</p> <p>db2 get health snapshot for tablespaces on &lt;dbname&gt; [show detail]</p>	<p>Instance / Database / Table space / Table / Bufferpool / Lock Application / SQL / Agent 및 Health 등 다양한 Level의 Performance 정보를 나열한다.</p>

# Performance

CLP Command	Description
<b>runstats on table db2admin.employee ;</b> <b>runstats on table db2admin.employee with distribution on columns (empid, empname) ;</b> <b>runstats on table db2admin.employee with distribution default num_freqvalues 50 ;</b> <b>runstats on table db2admin.employee for indexes db2user.empl1, db2user.empl2 ;</b> <b>runstats on table db2admin.employee for indexes all ;</b> <b>runstats on table db2admin.employee and sampled detailed indexes all ;</b>  <b>runstats on table db2admin.employee</b> <b>with distribution on columns (empno, firstnme, workdept num_freqvalues 50</b> <b>num_quantiles 100) default num_freqvalues 5 num_quantiles 10 and indexes all;</b>  <b>runstats on table db2user.t1</b> <b>with distribution on columns (c1, c2, c3 num_freqvalues 20 num_quantiles 40,</b> <b>c4, c5, c6, c7, c8) default num_freqvalues 0, num_quantiles 0 and indexes all;</b>  <b>runstats on table db2admin.employee on key columns and indexes all ;</b>  <b>runstats on table db2admin.employee on columns (empno, (firstnme, firstnme),</b> <b>(firstnme, lastname), workdept);</b>  <b>runstats on table db2admin.employee on columns (empno like statistics, workdept);</b>	<b>Database</b> <b>Directory</b> 내의 <b>Instance</b> 에 해당하는 항목을 표시한다.

# Administration

CLP Command	Description
export export to emp.del of del select * from employee	Database data를 Flat 파일로 추출한다.
import import from emp.del of del messages emp.msg insert into employee	Import Utility를 사용하여 Database에 Data를 저장한다.
load load from employee of del modified by dumpfile=emp.dump identityoverride warningcount 100 messages emp.msg insert into employee ;	Load Utility를 사용하여 Database에 Data를 저장한다.
load query table <tbname> [to local-message-file] [nosummary   summaryonly] [showdelta]	Load Utility의 진행상태를 표시한다.
backup database <dbname> [to <path>]	Database를 Backup 한다.
(Sun) db2 backup db sample to /tsm (Mon) db2 backup db sample online incremental delta to /tsm (Tue ) db2 backup db sample online incremental delta to /tsm (Wed) db2 backup db sample online incremental to /tsm (Thu ) db2 backup db sample online incremental delta to /tsm (Fri ) db2 backup db sample online incremental delta to /tsm (Sat ) db2 backup db sample online incremental to /tsm	Offline Full backup / Incremental / Delta Backup을 한다.
restore database <dbname> [from <path>]	Database를 Restore 한다.

### Admin Server

CLP Command	Description
get admin server	Admin server configuration을 Return한다.
update admin cfg using <p> <v>	Admin CFG 구성변수 <p>를 <v> 로 변경한다.

### Application Development

CLP Command	Description
get routine into <filename> from [specific] procedure <routine-name> [hide body]	SQL Stored Procedure를 Binary 파일로 추출한다.
put routine from <filename> [owner <newowner> [use registers]]	Binary 파일로 부터 SQL Stored Procedure를 생성한다.

### Transaction

CLP Command	Description
LIST INDOUBT TRANSACTIONS [WITH PROMPTING]	INDOUBT Transaction을 나열한다.

### History

CLP Command	Description
LIST HISTORY {BACKUP   ROLLFORWARD   REORG   CREATE TABLESPACE   ALTER TABLESPACE   DROPPED TABLE   LOAD   RENAME TABLESPACE   ARCHIVE LOG} {ALL   SINCE timestamp  CONTAINING {schema.object_name   object_name}} FOR [DATABASE] database-alias db2 list history create tablespace all for sample	BACKUP   ROLLFORWARD   REORG   CREATE TABLESPACE   ALTER TABLESPACE   DROPPED TABLE   LOAD   RENAME TABLESPACE   ARCHIVE LOG 등의 History 정보를 표시한다.



## Special Register

CLP Command	Description
CURRENT DATE (Data type : Date)	Admin server configuration을 Return한다.
CURRENT DEGREE (Data type : Char(5))	Intra-Partition의 Degree 정보를 Return 한다.
CURRENT QUERY OPTIMIZATION (Data type : Integer)	Optimization Level 정보를 Return한다.
CURRENT SERVER (Data type : Varchar(18))	현재 접속되어 있는 Database 정보를 Return한다.
CURRENT TIME (Data type : Time)	SQL 문장이 실행되는 시점의 시간을 Return한다.
CURRENT TIMESTAMP (Data type : Timestamp)	SQL 문장이 실행되는 시점의 micro-second을 Return한다
CURRENT TIMEZONE (Data type : Decimal(6,0))	Server의 현재 Time-zone을 Return한다.
USER (Data type : Char(8))	Database에 접속되어 있는 User-ID를 반환한다.

# Instance Example

## 1.1 Instance 사용자 계정 생성

root 사용자 계정으로 db2 관련 Group 및 계정을 생성한다.

ex ) on AIX

```
mkgroup id=500 db2adm8
```

```
mkuser id=501 pgrp=db2adm8 groups=db2adm8 home='/DB2/db2inst' db2inst <-- Instance 사용자 계정
```

```
mkuser id=502 pgrp=db2adm8 groups=db2adm8 home='/DB2/db2fnce' db2fnce <-- fenced 사용자 계정
```

```
mkuser id=503 pgrp=db2adm8 groups=db2adm8 home='/DB2/db2as' db2as <-- administrator 사용자 계  
정
```

## 1.2 DB2 Instance를 생성한다.

DB2 Version 7/8에 따라 Instance 및 Admin Server를 생성한다.

ex) /usr/opt/db2\_08\_01/instance/db2icrt -u db2fnce db2inst

<-- Instance 생성

```
/usr/opt/db2_08_01/instance/dascrt db2as
```

<-- Admin server 생성

```
/usr/opt/db2_08_01/instance/db2ilist
```

<-- Instance list

```
/usr/opt/db2_08_01/instance/daslist
```

<-- Admin Server list

```
/usr/opt/db2_08_01/instance/db2idrop db2inst
```

<-- Instnace Drop

```
/usr/ppt/db2_08_01/instance/dasdrop db2as
```

<-- Admin Server Drop

ex) /usr/lpp/db2\_07\_01/instance/db2icrt -u db2fnce db2inst

<-- Instance 생성

```
/usr/lpp/db2_07_01/instance/dasicrt db2as
```

<-- Admin server 생성

```
/usr/lpp/db2_07_01/instance/db2ilist
```

<-- Instance list

```
/usr/lpp/db2_07_01/instance/daslist
```

<-- Admin Server List

```
/usr/lpp/db2_07_01/instance/db2idrop db2inst
```

<-- Instnace

```
/usr/lpp/db2_07_01/instance/dasidrop db2as
```

<-- Admin Server Drop

# Instance Example

## 1.3 Instance Start

Instance User계정으로 Login 한다.

환경변수 DB2INSTANCE를 변경한다.

export DB2INSTANCE=<instance user name>

ex) **db2start**

## 1.4 Admin Server Start

admin server계정으로 Login 한다.

ex) db2admin start

## 1.5 Admin CFG 및 Instance 상태 확인

Instance User 계정으로 Login 한다.

ex) **db2 get admin cfg**

db2 get instance

## 1.6 Node 및 DB catalog

Remote Node 및 DB를 Catalog한다.

ex) db2 uncatalog node REMOTENODE

db2 uncatalog dcs db REMOTEDB

db2 uncatalog db REMOTEDB

db2 catalog tcpip node REMOTENODE remote 127.0.0.1 server 50000

db2 catalog dcs db REMOTEDB

db2 catalog db REMOTEDB at node REMOTENODE authentication dcs

## 1.7 환경변수 및 Instance 구성 매개변수 확인

Instance User의 db2 Repository 변수를 확인한다.

ex) **db2set -all**

**db2 get dbm cfg**

# Database Example

## 2.1 Database 생성

Sample Database 를 생성한다.

ex) **db2sample**

사용자 구성에 맞는 Database를 생성한다.

```
ex) create db sample on '/home/db2sample1'  
    alias 'sample'  
    using codeset 1363 territory kr collate using identity  
    collate using identity  
    catalog    tablespace managed by system using ( '/home/db2sample2' )  
    user       tablespace managed by system using ( '/home/db2sample3' )  
    temporary tablespace managed by system using ( '/home/db2sample4' )  
    dft_extent_sz 32  
    extentsize 16 prefetchsize 64;
```

## 2.2 Node / DCS / Database list 확인

```
ex) list node directory  
    list dcs directory  
    list db directory
```

## 2.3 Database 접속

connect to sample user <instance name> using <password>

## 2.4 Table list 확인

```
list tables
```

# Database Example

## 2.5 현재 활성화 되어 있는 Database 목록 확인

`list active databases`

## 2.6 사용자의 Database 접속 확인

`get connection state`

## 2.7 사용자가 접속한 Database 확인

`values ( current server )`

## 2.8 Database의 구성 매개변수 확인

`get db cfg for sample` // "sample" : database name

## 2.9 Database의 접속 해 있는 Application의 목록을 확인

`list applications for sample`

## 2.10 Database의 접속 해 있는 Application의 강제 종료

`force applications (1)` // "1" : Application Agent id

`force applications all` // 모든 Applications 종료

## 2.11 Database의 접속 종료

`terminate or reset`

## 2.12 Database의 구성 매개변수 변경

ex) Database에 접속 할 수 있는 동시 사용자 수 변경

`update db cfg for sample using MAXAPPLS 50`

## 2.13 Database의 구성 매개 변수 변경사항 반영

`force applications all`

`terminate`

`activate database`

## 2.14 Database의 구성 매개 변수 변경사항 반영

`autoconfigure using mem_percent 60`

`workload_type simple`

`tpm 60`

`is_populated no`

`num_local_apps 10`

`num_remote_apps 10`

`apply db and dbm > autoconfig.txt`

# BufferPool

## 3.1 MYBP8K라는 1000크기의 Page size 8k의 Buffer pool 생성

```
create bufferpool MYBP8K size 1000 pagesize 8K
```

## 3.2 MYBP4K라는 1000크기의 Page size 4k의 Buffer pool 생성

```
create bufferpool MYBP4K size 1000 pagesize 4K
```

## 3.3 Buffer pool의 크기 변경

```
alter bufferpool ibmdefaultbp size 5000
```

```
alter bufferpool mybp8k size 5000
```

## 3.4 Buffer pool의 목록을 확인

```
select * from syscat.bufferpools
```

## 3.5 MYBP8K Buffer pool을 사용하는 Table space MYTS8K를 생성

```
create tablespace MYTS8K pagesize 8K
```

```
managed by database using ( FILE 'c:\myts8k.0' 1000, FILE 'c:\myts8k.1' 1000)
```

```
extentsize 8 prefetchsize 16
```

```
bufferpool MYBP8K
```

# Tablespace

## 4.1 Table space의 목록을 확인한다.

`list tablespaces [show detail]`

`list tablespace containers for <tablespace-id> [show detail]`

## 4.2 DMS 방식의 Table space를 생성

`create regular tablespace ts01 managed by database using (file '/mytbs/ts01.dat' 1000) prefetchsize 64`

`create regular tablespace ts02 managed by database using (file '/mytbs/ts02.dat' 200)`

## 4.3 DMS 방식의 Long Table space를 생성

`create long tablespace ts03 managed by database using ( file '/mytbs/ts03.dat' 1000)`

## 4.4 SMS 방식의 Table space를 생성

`create regular tablespace ts04 managed by system using ('/mytbs/ts04')`

## 4.5 SMS 방식의 System Temporary Table space를 생성

`create system temporary tablespace ts05 managed by system using ( '/mytbs/ts05')`

## 4.6 Pagesize 8k, MYBP8K를 사용하는 SMS 방식의 regular tablespace ts06을 생성하시오.

`create tablespace ts06 pagesize 8K managed by system using ('/mytbs/ts06') bufferpool mybp8k`

## 4.7 Pagesize 8k, MYBP8K 을 사용하는 SMS 방식의 system temporary용 tablespace ts07을 생성하시오.

`create system temporary tablespace ts07 pagesize 8K managed by system using ('/mytbs/ts07') bufferpool mybp8k`

## 4.8 Tablespace ts01의 bufferpool을 mybp4k로 바꾸시오.

`alter tablespace ts01 bufferpool mybp4k`

## 4.9 Tablespace ts01에 container를 추가하시오.

`alter tablespace ts01 add (file '/mytbs/ts0112.dat' 1000)`

# Tablespace

- 4.10 Tablespace ts01의 container 크기를 10 page씩 증가시키시오.  
`alter tablespace ts01 extend (all 10)`
- 4.11 Tablespace ts02의 container 크기를 300 page가 되도록 바꾸시오.  
`alter tablespace ts02 resize (all 300)`
- 4.12 Tablespace ts02의 container ts02.dat크기를 200 page 줄이시오.  
`alter tablespace tb02 reduce ( file 'B.DBF' 200 )`
- 4.13 접속해 있는 Database의 tablespace의 목록을 다시 확인하시오.  
`list tablespaces`
- 4.14 Tablespace의 ID를 통해 Container를 확인하시오.  
`list tablespace containers for <id> show detail`
- 4.15 Tablespace ts05, ts06, ts07, userspace1을 삭제하시오.  
`drop tablespace ts05, ts06, ts07, userspace1`
- 4.16 생성된 Tablespace의 정보를 system catalog에서 확인하시오.  
`select * from syscat.tablespaces`



# Schema, Table, View, Alias, Trigger

5.1 MY\_SCHM라는 schema를 생성한 후 user01에게 그 사용 권한을 부여하시오.

```
create schema MY_SCHM authorization user01
```

5.2 Schema 의 정의를 확인하시오.

```
select * from syscat.schemata where tabschema= 'MY_SCHM'
```

5.3 Table MY\_SCHM.DEPT를 생성하시오.

```
create table MY_SCHM (  id          smallint          not null,
                        name        varchar(20)         not null,
                        man          smallint,
                        budget       int
                        ) in ts04;
```

5.4 Table MY\_SCHM.EMPL을 생성하시오.

```
create table MY_SCHM.empl (  id          smallint          not null,
                             name        varchar(30)         not null,
                             sex         char(1),
                             mydept     smallint,
                             hiredate   date,
                             salary     smallint,
                             email       varchar(30)         not null,
                             resume      clob(1K)            not logged
                             ) in ts01 index in ts02 long in ts03;
```

5.5 MY\_SCHM.dept table에 대한 index를 생성하시오.

```
create unique index MY_SCHM.dept_ix on MY_SCHM.dept (id) include (name) cluster
```

5.6 MY\_SCHM.empl table에 대한 index를 생성하시오.

```
create unique index MY_SCHM.empl_ix on MY_SCHM.empl (id) cluster
create index MY_SCHM.empl_ix2 on MY_SCHM.empl (mydept)
```

# Schema, Table, View, Alias, Trigger

5.7 MY\_SCHM.DEPT table에 대해 아래의 **constraint**를 추가하시오.

```
alter table MY_SCHM.dept add constraint dept_pk primary (id)
```

5.8 MY\_SCHM.EMPL table에 대해 아래의 **constraint**를 추가하시오.

```
alter table MY_SCHM.empl add constraint empl_pk primary (id)
```

```
alter table MY_SCHM.empl add constraint empl_uk unique (email)
```

```
alter table MY_SCHM.empl add constraint empl_fk foreign key (mydept) references dept on delete restrict
```

```
alter table MY_SCHM.empl add constraint empl_cc check (sex in ('M', 'F'))
```

5.9 Table empl에 관한 정보를 확인하시오.

```
describe table MY_SCHM.empl
```

```
select * from syscat.tables where tabschema= 'MY_SCHM' and tabname = 'EMPL'
```

```
select * from syscat.references where tabschema= 'MY_SCHM' and tabname = 'EMPL'
```

```
select * from syscat.colchecks where tabschema= 'MY_SCHM' and tabname = 'EMPL'
```

5.10 Table MY\_SCHM.empl의 **index**에 관한 정보를 확인하시오.

```
describe indexes for table MY_SCHM.empl show detail
```

```
select * from syscat.indexes where tabschema= 'MY_SCHM' and tabname = 'EMPL'
```

5.11 MY\_SCHM.empl table에 대해 아래의 **view**를 생성하시오.

```
create view empl_v1 as select * from MY_SCHM.empl where salary > 200 with check option
```

```
create view empl_v2 as select a.name, b.name as dname
```

```
from MY_SCHM.empl a, MY_SCHM.dept b
```

```
where a.mydept = b.id
```

5.12 view empl\_v1, empl\_v2의 정의를 확인하시오.

```
select * from syscat.tables where tabname like 'EMPL_V%'
```

5.13 아래와 같이 **alias**를 생성하시오.

```
create alias empl for MY_SCHM.empl
```

```
create alias empl_a1 for empl
```

# Schema, Table, View, Alias, Trigger

5.14 alias empl의 정의를 확인하시오.

```
select * from syscat.tables where type = 'A' and tablename like 'EMPL'
```

5.15 Table의 목록을 확인하시오.

```
list tables  
list tables for all  
list tables for system  
list tables for user  
list tables for schema MY_SCHM
```

5.16 Table uplist를 생성하시오.

```
create table MY_SCHM.empl (  id          smallint          not null,  
                             namevarchar(30)      not null,  
                             salary      smallint,  
                             nsalary     smallint,  
                             ts          timestamp  
                             ) in ts04;
```

5.17 Trigger empl\_t1을 생성하시오.

```
create trigger empl_t1 after update of salary on empl referencing old as o new as n  
for each row mode db2sql  
when ( n.salary > o.salary * 1.2 )  
insert into uplist values ( o.id, o.name, o.salary, n.salary, current timestamp )
```

5.18 Trigger empl\_t1의 정의를 확인하시오.

```
select * from syscat.triggers where trigrname = 'EMPL_T1'
```

# Export

## 6.1 DEL 형식의 file x.del, y.del을 생성하시오.

```
export to x.del of del modified by coldel; select * from empl  
export to y.del of del select * from empl
```

## 6.2 WSF 형식의 file x.wsf를 생성하시오.

```
export to x.wsf of wsf messages x.msgs select * from empl
```

## 6.3 IXF 형식의 file x.ixf, y.ixf를 생성하시오.

```
export to x.ixf of ixf select * from empl where id < 100  
export to y.ixf of ixf method N (c1,c2,c3,c4,c5,c6,c7) select * from empl where id < 100
```

## 6.4 Join을 이용하여 IXF 형식의 file w.ixf를 생성하시오.

```
export to w.ixf of ixf select e.id, e.name, d.name from empl e, dept d where e.mydept = d.name
```

## 6.5 Database에서 IXF 형식의 file z.ixf과 DEL 형식의 file z.del을 생성하시오.

```
export to z.ixf of ixf select * from emp_photo  
export to z.del of del lobs to c:\mylob\ modified by lobsinfile select * from emp_photo  
export to myfile.del of del lobs to ./mylobs/ lobfile emp_lobs modified by lobsinfile select * from emp_photo;  
export to emp.del of del lobs to ./emp/ lobfile emp_bobs modified by lobsinfile select * from emp_photo;
```

# Import

7.1 IXF 형식의 file을 이용하여 empl2, empl3 table을 생성하시오.

```
import from x.ixf of ixf create into empl2
```

```
import from y.ixf of ixf method N (c1,c2,c3,c6,c5,c4,c7) create into empl3
```

7.2 DEL 형식의 file을 이용하여 empl2, empl3 table에 data를 추가, 대체하시오.

```
import from x.del of del modified by coldel; commitcount 3 replace into empl2
```

```
import from y.del of del method P (1,2,3,6,5,4,7) messages y.msg insert into empl3
```

```
import from y.del of del restartcount 1 replace into empl2
```

7.3 z.ixf, z.del을 이용하여 emp\_phote2 table을 생성하고 data를 추가하시오.

```
import from z.ixf of ixf create into emp_photo2
```

```
import from z.del of del lobs from c:\mylob\ modified by lobsinfile insert into emp_photo2
```

7.4 ASC 형식의 file을 이용하여 empl2 table의 기존 data를 입력 file의 data로 대체하시오.

```
91조용필 M 12001-04-07100 phil@mymail.com
```

```
92심수봉 F 22001-01-02200 bong@mymail.com
```

```
import from x.dat of asc method L (1 2,3 10,11 11,13 13,14 23,24 27,28 46) replace into empl2
```

7.5 WSF 형식의 file을 이용하여 empl2 table에 data를 추가하시오.

```
import from x.wsf of wsf insert into empl2
```



# Load

9.1 empl table을 위한 exception table인 emplexcp를 생성하시오.

```
create table emplexcp like empl
alter table emplexcp add column ts timestamp add column msg clob(32k)
```

9.2 data file y.dat를 생성하시오.

```
11,"이문세","M", 1, "2002-03-07", 100, "lms@mymail.com"
11,"김경호","M", 2, "2001-04-25", 200, "kkh@mymail.com"
13,"이기찬","M", 1, "2002-02-19", 300, "lkc@mymail.com"
14,"김현정","F", 3, "2002-07-17", 400, "lkc@mymail.com"
15,"김건모","m", 2, "2001-08-02", 500, "kkm@mymail.com"
16,"제이","F", 1, "2000-05-08", 120, "j@mymail.com"
17,"양희은","F", 2, "2002-10-20", "130", "yhe@mymail.com"
18,"M", 2, "2001-11-29", 140, "god@mymail.com"
19,"신화","M", 4, "2001-04-07", 150, "sh@mymail.com"
20,"엄정화","F", 1, "2001-04-28", 160, "ejw@mymail.com"
```

9.3 REPLACE mode를 이용한다면 backup을 받아두시오.

```
backup db sample
backup db sample tablespace (ts01,ts02,ts03)
```

9.4 y.dat를 이용하여 empl table에 load 하시오.

```
load from y.dat of del savecount 5 messages y.msg insert into empl for exception emplexcp
```

9.5 만약 Backup Pending에 걸렸다면 empl이 속한 tablespace를 backup 하시오.

```
backup db sample
backup db sample tablespace (ts01,ts02,ts03)
```

9.6 만약 Check Pending에 걸렸다면 empl table에 대한 constraint를 check 하시오.

```
select const_checked from syscat.tables where tabname = 'EMPL'
set integrity for empl immediate checked for exception in empl use emplexcp
```

# Load

9.7 emplexcp, empl table을 확인하시오.

```
select * from emplexcp  
select * from empl
```

9.8. 만약 load 작업을 실패했다면 다음과 같이 재시작하시오.

```
load from y.dat of del savecount 5 messages y.msg restart into empl for exception emplexcp
```

9.9. 만약 load 작업을 실패한 경우 재시작을 원하지 않는다면 다음과 같이 종료하시오.

```
load from y.dat of del savecount 5 messages y.msg terminate into empl for exception emplexcp
```

9.10. db2move를 이용하여 load 하시오.

```
db2move sample load
```



# Restore Recovery using Offline Backup

10.1 empl table의 data 건수를 확인하시오.

```
select count(*) from empl
```

10.2 Database sample을 Offline mode로 backup 하시오. (BACKUP-A)

```
connect reset
```

```
backup db sample
```

10.3 'BACKUP-A' 에 대한 정보를 확인하시오.

```
list backup all for sample
```

10.4 empl table의 자료를 아래와 같이 삭제하시오.

```
delete from empl where id < 13
```

10.5 empl table의 data 건수를 확인하시오.

10.6 'BACKUP-A' image를 이용하여 database를 복원하시오.

```
connect reset
```

```
restore db sample
```

10.7 empl table의 data 건수를 확인하시오.

10.8 'BACKUP-A' image를 이용하여 새로운 database sample2를 생성하시오.

```
restore db sample into sample2 redirect
```

```
set tablespace containers for 3 using (file 'c:\mycont\db02\ts01\file01.dat' 1000)
```

```
set tablespace containers for 4 using (file 'c:\mycont\db02\ts02\file01.dat' 1000)
```

```
set tablespace containers for 5 using (file 'c:\mycont\db02\ts03\file01.dat' 1000)
```

```
set tablespace containers for 6 using (path 'c:\mycont\db02\ts04')
```

```
restore db sample continue
```

10.9 sample2에서 tablespace와 table에 관한 정보를 확인하시오.

# Rollforward Recovery using Offline Backup

11.1 sample의 logging 정책을 archival logging으로 바꾸시오.

update db cfg for sample using ROGRETAIn ON

update db cfg for sample using TRACKMOD ON

11.2 Backup Pending을 풀기 위해 sample을 backup 하고. (BACKUP-B)

backup db sample

11.3 sample에 접속한 후, empl table의 현재 data 건수를 기록하고, 그 시각을 'TS-0' 라 정하시오.

select count(\*), current timestamp - current timezone from empl

11.4 자료를 삭제한 후, data 건수를 기록하고, 그 시각을 'TS-A' 라 정하시오.

delete from empl where id <13

select count(\*), current timestamp - current timezone from empl

11.5 다시 자료를 삭제한 후, data 건수를 기록하고, 그 시각을 'TS-B' 라 정하시오.

delete from empl where id <15

select count(\*), current timestamp - current timezone from empl

11.6 다시 자료를 삭제한 후, data 건수를 기록하고, 그 시각을 'TS-C' 라 정하시오.

delete from empl where id <17

select count(\*), current timestamp - current timezone from empl

11.7 'BACKUP-B' image를 이용하여 sample database를 복원하시오.

connect reset

restore db sample taken at 20020501160000

connect to sample

11.8 rollforward Pending에 걸렸다면 최근까지의 log file을 적용하시오.

rollforward db sample to end of logs and stop

connect to sample

# Rollforward Recovery using Offline Backup

11.9 아직도 rollforward Pending에 걸렸다면 log file의 적용을 중지하십시오.

```
rollforward db sample stop
connect to sample
```

11.10 empl table의 data 건수를 확인하십시오.

```
select count(*) from empl
```

11.11 다시 'BACKUP-B' image를 이용하여 database를 복원 후 'TS-B' 까지 log file 을 적용하십시오.

```
connect reset
restore db mydb1 taken at 20020501160000
rollforward db sample to 2002-05-01-07.05.00 and stop
select count(*) from empl
```

11.12 다시 'BACKUP-B' image를 이용하여 database를 복원 후 log file을 전혀 적용하지 마시오.

```
connect reset
restore db sample taken at 20020501160000
rollforward db sample stop
connect to sample
```

11.13 특정 시간까지 Roll forward를 위한 Restore Image를 조사한다.

```
db2ckrst -d sample -t 20030420200844 -r database
```

Suggested restore order of images using timestamp 20030420200844 for database sample.

```
=====
restore db sample incremental taken at 20030420200844
restore db sample incremental taken at 20030417141610
restore db sample incremental taken at 20030420200844
=====
```

# Rollforward Recovery using Online Backup

12.1 session A에서 sample에 접속하시오.

connect to sample

12.2 session B에서 sample에 접속하시오.

connect to sample

12.3 session B에서 empl table의 자료를 삭제하는 명령을 준비만 했다가 4번이 실행되는 도중에 실행하시오.

delete from empl where id < 11

12.4 session A에서 sample를 backup 하고. backup image를 'BACKUP-C' 라 정하시오.

backup db sample online

12.5 session B에서 자료를 삭제한 후 data 건수를 기록하고, 그 시각을 'TS-D' 라 정하시오.

delete from empl where id <15

select count(\*), current timestamp - current timezone from empl

12.6 다시 session B에서 자료를 삭제한 후 data 건수를 기록하고, 그 시각을 'TS-E' 라 정하시오.

delete from empl where id <17

select count(\*), current timestamp - current timezone from empl

12.7 다시 session B에서 자료를 삭제한 후 data 건수를 기록하고, 그 시각을 'TS-F' 라 정하시오.

delete from empl where id <19

select count(\*), current timestamp - current timezone from empl

12.8 session B에서 'BACKUP-C' image 를 이용하여 database를 복원 후 최근까지의 log file을 적용하시오.

connect reset

restore db sample taken at 20020501170000

rollforward db sample to end of logs and stop

connect to sample

select count(\*) from empl

# Rollforward Recovery using Online Backup

12.9 session B에서 'BACKUP-C' image를 이용하여 database를 복원 후 'TS-E' 까지 log file을 적용하시오.

```
connect reset
restore db sample taken at 20020501170000
rollforward db sample to 2002-05-01-08.06.10 and stop
connect to sample
select count(*) from empl
```

12.10 session B에서 'BACKUP-C' image를 이용하여 database를 복원 후 log file을 전혀 적용하지 않은 경우 오류가 발생하는 것을 확인하시오.

```
connect reset
restore db sample taken at 20020501170000
rollforward db sample stop
```

12.11 session B에서 'BACKUP-C' image를 이용하여 database를 복원 후 최소한의 log file을 적용하시오.

```
list backup all for sample
rollforward db sample to 2002-05-01-08.00.03 and stop
connect to sample
select count(*) from empl
```

12.12 session B에서 sample에 대한 rollforward 작업 내역을 확인 하시오.

```
list history rollforward all for sample
```

12.13 session B에서 sample에 대한 recovery history file 작업 내역 중 20020501 이전의 기록을 삭제하시오.

```
prune history 20020501000000
```

# Snapshot

13.1 session A에서 **monitor switches**의 현재 상태를 확인하시오.

get monitor switches

13.2 session A에서 **sample**에 접속하시오.

connect to sample

13.3 session B에서 **sample**에 접속하시오.

connect to sample

13.4 session A에서 **sample**에 접속 중인 **application**의 목록을 확인하시오.

list applications for db sample

13.5 session A에서 **sample** 내에서 발생한 **lock**에 관한 정보를 확인하시오.

get snapshot for locks on sample

13.6 session B에서 **update** 문을 실행하시오.

db2 +c update salary = salary + 1 where id = 13

13.7 session A에서 **sample** 내에서 발생한 **lock**에 관한 정보를 확인하시오.

get snapshot for locks on sample

13.8 session B에서 **commit** 문을 실행하시오.

commit

13.9 session A에서 **sample** 내에서 발생한 **lock**에 관한 정보를 확인하시오.

get snapshot for locks on sample

# Snapshot

13.10 session A에서 lock과 buffer pool에 관한 monitor switch를 on 시키시오.  
update monitor switches using lock on bufferpool on

13.11 session A에서 monitor switches의 현재 상태를 확인하시오.  
get monitor switches

13.12. session A에서 sample에 관한 system monitor data 값을 초기화 시키시오.  
reset monitor for db sample

13.13 session B에서 update 문을 실행하시오.  
db2 +c update salary = salary + 1 where id = 13

13.14 session A에서 sample 내에서 발생한 모든 monitor 정보를 확인하시오.  
get snapshot for all on sample

13.15 session B에서 commit 문을 실행하시오.  
commit

13.16 session A에서 sample 내에서 발생한 모든 monitor 정보를 확인하시오.  
get snapshot for all on sample

# Explain

14.1 Package에 대한 정보를 나열하시오.

```
list packages
```

14.2 MYAPP라는 package에 대한 access plan을 확인하시오.

```
db2expln -d sample -c nullid -p db2sampl -s 0 -t
```

14.3 아래 select 문에 대한 access plan을 확인하시오.

```
dynexpln -d sample -q "select * from employee where empno > 'A' order by firstnme" -t
```

14.4 다음 File의 access plan을 확인하시오.

```
exp.db2
```

```
select * from employee where empno > 'A' order by firstnme ;
```

```
select * from org;
```

```
dynexpln -d sample -f exp.db2 -t -z “,”
```



# Authority, Privilege, Authentication

**15.1 Instance의 SYSADM, SYSCTRL, SYSMANT 권한을 가진 group을 확인하시오.**

```
get dbm cfg | find "GROUP"
```

**15.2 현재 사용자가 가진 authority를 확인하시오.**

```
get authorizations
```

**15.3 inst01에서 SYSADM group을 admgrp로 변경하시오.**

```
update dbm cfg using SYSADM_GROUP admgrp
```

**15.4 sample을 deactivate한 후 다시 접속하시오.**

```
deactivate db sample  
connect to sample
```

**15.5 현재 사용자가 가진 authority를 확인하시오.**

```
get authorizations
```

**15.6 database에 관한 authority 및 privilege를 확인하시오.**

```
select * from syscat.dbauth
```

**15.7 PUBLIC 으로부터 CREATETAB 권한을 제거하시오.**

```
revoke createtab on database from public
```

**15.8 DBADM authority를 user01에게 부여한 후 확인하시오.**

```
grant dbadm on database to user user01  
select * from syscat.dbauth where grantee = 'USER01'
```

**15.9 tablespace ts04에 대한 use privilege를 user02에게 부여하시오.**

```
grant use of tablespace ts04 to user user02
```

**15.10 CREATETAB privilege를 user02에게 부여하시오.**

```
grant createtab on database to user user02
```

# Authority, Privilege, Authentication

**15.11 MY\_SCHM.empl table에 대한 select, salary column에 대한 update privilege를 user02에게 부여하시오.**  
grant select, update (salary) on table my\_schm.empl to user user02

**15.12 tablespace 와 table 에 관한 privilege 중 user02 에게 부여된 것은 무엇인지 확인하시오.**  
select \* from syscat.tbpaceauth where grantee = 'USER02'  
select \* from syscat.tabauth where grantee = 'USER02'

**15.13 sample에 user02 사용자를 이용하여 접속한 후, empl table과 동일한 empl4를 생성하시오.**  
connect to sample user user02 using user02  
create table db2admin.empl4 like my\_schm.empl in ts04

**15.14 table empl4에 관한 권한 부여 상황을 확인하시오.**  
select \* from syscat.tabauth where tabname = 'EMPL4'

**15.15 sample에 user01 사용자를 이용하여 접속하시오.**  
connect to sample user user01 using user01

**15.16 my\_schm.empl table에 대한 select privilege를 user02로부터 제거하시오.**  
revoke select on table my\_schm.empl from user user02

**15.17 sample에 user02 사용자를 이용하여 접속하시오.**  
connect to sample user user02 using user02

**15.18 user01에게서 DBADM authority를 제거한 후, 그 정보를 확인하시오.**  
revoke dbadm on database from user user01  
select \* from syscat.dbauth where grantee = 'USER01'

**15.19 sample 에서 index, schema, package, tablespace 등에 관한 권한 부여 상황을 확인하시오.**  
select \* from syscat.colauth where tabschema = 'MY\_SCHM' and tabname= 'EMPL'  
select \* from syscat.indexauth where indschema = 'MY\_SCHM' and indname= 'DEPT\_X'  
select \* from syscat.schemaauth where schemaname= 'MY\_SCHM'  
select \* from syscat.packageauth where pkgname = 'MYAPP'  
select \* from syscat.tbpaceauth where tbpace = 'TS01'

# Reorgchk, reorg, runstats

## 16. Reorgchk, Reorg, Runstats

### 16.1 table에 대한 재구성 필요 여부를 확인하시오.

```
reorgchk on table all  
reorgchk on table user  
reorgchk on table system  
reorgchk on table my_schm.empl  
reorgchk on table sysibm.sysindexes
```

### 16.2 my\_schm.empl table을 재구성하시오.

```
reorg table my_schm.empl index my_schm.empl_A use temp space 1
```

### 16.3 통계 정보를 갱신하시오.

```
runstats on table my_schm.empl  
runstats on table my_schm.empl for index my_schm.empl_A  
runstats on table my_schm.empl for indexes all  
runstats on table my_schm.empl and index my_schm.empl_A  
runstats on table my_schm.empl and indexes all
```

### 16.4 package myapp에 대한 access plan을 갱신하시오.

```
rebind myapp
```

### 16.5 sample에 있는 모든 package에 대한 access plan을 갱신하시오.

```
db2rbind sample /I rebind.log all
```